Interaction-based Collaborative Recommendation: A Personalized Learning Environment (PLE) Perspective

Syed Mubarak Ali[1], Imran Ghani[1], Muhammad Shafie Abd Latiff[2]
[1,2]Faculty of Computing,
Universiti Teknologi Malaysia, Skudai, Johor, 81310, Malaysia
[e-mail: mubarakali.00@gmail.com, imransaieen@gmail.com, shafie@utm.my]
*Corresponding author: Imran Ghani

Received October 12, 2014; revised December 1, 2014; accepted December 18, 2014; published January 31, 2015

Abstract

In this modern era of technology and information, e-learning approach has become an integral part of teaching and learning using modern technologies. There are different variations or classification of e-learning approaches. One of notable approaches is Personal Learning Environment (PLE). In a PLE system, the contents are presented to the user in a personalized manner (according to the user's needs and wants). The problem arises when a new user enters the system, and due to the lack of information about the new user's needs and wants, the system fails to recommend him/her the personalized e-learning contents accurately. This phenomenon is known as cold-start problem. In order to address this issue, existing researches propose different approaches for recommendation such as preference profile, user ratings and tagging recommendations. In this research paper, the implementation of a novel interaction-based approach is presented. The interaction-based approach improves the recommendation accuracy for the new-user cold-start problem by integrating preferences profile and tagging recommendation and utilizing the interaction among users and system. This research work takes leverage of the interaction of a new user with the PLE system and generates recommendation for the new user, both implicitly and explicitly, thus solving new-user cold-start problem. The result shows the improvement of 31.57% in Precision, 18.29% in Recall and 8.8% in F1-measure.

***Keywords*:** Recommender system, Collaborative filtering, e-Learning, Personalized Learning

A preliminary version of this paper was presented at APIC-IST 2014 and was selected as an outstanding paper.

http://dx.doi.org/10.3837/tiis.2015.01.028 **ISSN : 1976-7277**

1. Introduction

In recent years, e-learning has become a well-known teaching and learning approach. The e-learning approach uses up-to-date and latest educational technologies to create a learning environment where information technology is integrated in curriculum. This approach helps in creating more effective learning environment than traditional learning systems [1] [2]. In comparison with traditional 'face-to-face' learning, e-learning has obtained much attention. Personal Learning Environment (PLE), on the other hand, is becoming quite popular in the line of Technology Enhanced Learning (TEL). PLE refers to "a set of learning tools, services, and artifacts gathered from various contexts to be used by the learners" [3]. Furthermore, according to Van Harmelen [4] PLE helps the learners to enjoy ICT-based environments for learning activities so that a learner can connect to different networks in order to collaborate on shared outcomes and acquire necessary professional competences through its usage.

As there is so much information available on the Internet-based e-learning system, so it becomes quite difficult for a learner to find the most appropriate contents for learning. This problem arises significantly when a user is new to the learning system and he/she has very little personal experience in using the system. Recommendations can be very useful in different aspects of PLE as well. For example, for finding relevant tools, get recommendations for learner to interaction in specific situations [5].

Due to this feature recommender systems have become popular during the past few years. A recommender system is defined as "system that produces individualized recommendations as output or have the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options" [6] [7] [38] [39] [40]. Recommendation systems are used in e-commerce and social networking sites commonly and now these systems have gained a lot of attention in the e-Learning community as well. In regards with classification, there are different types of recommendation techniques. The most popular techniques are 'Collaborative Filtering', 'Content-based Filtering' and 'Hybrid Filtering' [6] [8] [9].

In this paper, we present a detailed implementation and evaluation of our previously proposed interaction-based collaborative filtering approach [41]. A number of new experiments have been conducted which depict an increased recommendation accuracy for the new-user cold start problem specifically in the personal learning environment. The approach is discussed in detail in section 3. In the next section, an extensive literature review on PLE, recommender systems and the current problems regarding recommendation are presented.

2. Literature Review

2.1 e-Learning Approach and its Limitations

The academic and learning environment of the 21^{st} century cannot thrive or do much without an important component, the e-learning platform. The e-learning system effectively offers tailored information and guides its users considering factors such as the needs of the individual students, their learning ability, and adaptation to knowledge. However, the existing e-learning systems lack in interaction hence become static content-centric repositories [10]. One of the reasons of this deficiency is that information overload in traditional e-learning systems is too much and seems unavoidable. Hence, the need for a personalization function in the e-Learning environment seems essential so that learners can have the appropriate information at the right time.

2.2 Personal Learning Environment (PLE) and its Limitations

A Personal Learning Environment (PLE) is defined as a set of tools, services and communities that establish an independent learning platform for learners' use for the sole purpose of accomplishing their educational objectives via self-paced learning. It is important to highlight that Learning Management Systems (LMS) are sometimes compared with PLE. They, however, are dissimilar. As LMS is mainly course-centric while PLE thrives on a learner-centric model (**Fig. 1**). Therefore, it is important to realize that the PLE is more like an ideology about how learners achieve their task of learning rather than a particular software application.

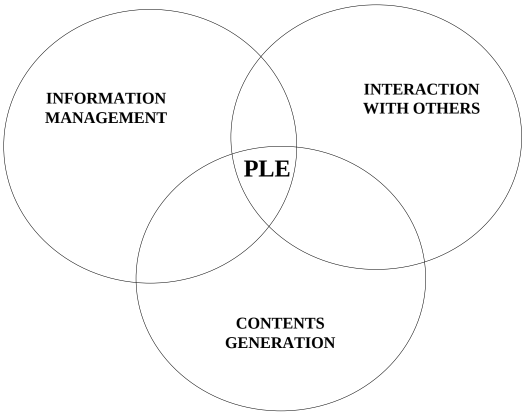

Fig. 1. Personal Learning Environment (PLE)

As shown in **Fig. 1**, three important parts of the environment must be taken in consideration. First is the information management part. The main idea behind this concept is that there is more than enough information available in a typical e-learning system but not all the information is required by every learner. So, there must a mechanism which should look after the management of that information and personalized the information for a particular learner. Second part is content generation, which is very simple. When a learner wants to publish or share some information on the e-learning system, there must be mechanism which should help the user in generating that content. For example, a leaner posts comments on a blog, starts some discussion thread or share a published article. Third part is interaction with others phase. In this part, a very important task is described, which is, to interact with other learners either for sharing information or asking for help on some issue. Thus, interaction is an integral part in a learning environment.

As it is obvious that Internet is a rich source of information, so much that students are confused that which content is more relevant to them. Ideally, they are only interested in contents that are most relevant in fulfilling their learning objectives. To tackle this challenge, recommendations come to the aid of the learner by filtering the available information, customize and recommend the relevant information according to their needs. 'Good' recommendations are vital, if a PLE is to be considered successful. Otherwise its failure leads to loss of learner's trust in the PLE [11].

2.3 Interaction Concept

Interaction can be defined as a form of communication involving two or more entities. According to Ha and James [12] "Interactivity should be defined in terms of the extent to which the communicator and the audience respond to or are willing to facilitate each other's communication needs." In the perspective of web systems, it can be defined as how a user communicates with a system [13] [14]. When a user communicates with the system, it goes far beyond explaining the interaction alone but also characterizes the peculiar level and type of interaction.

2.3.1 Levels of Interaction

The levels of satisfaction and communication are the essential part in defining the levels of interaction. According to Steuer [15], there are two levels of interaction starting from low interaction to high interaction. These interactions are categorized based on the capacity of communication in real time scenarios' [14]. For instance, an online chat program is considered as high level of interaction whereas, a newspaper falls in the category of low level of interaction (**Fig. 2**). A high level of interaction provides with a real time feedback, the user has more control of the environment. The user can become easily involved in this level of interaction. For example, online chat programs like skype or msn messenger are considered as real high levels of interaction. A low level of interaction provides the user less control over the environment and there is not a real-time feedback such as a book, a movie, and newspaper. In all of these, there is only one way of communication flow, the user feedback is not real-time.

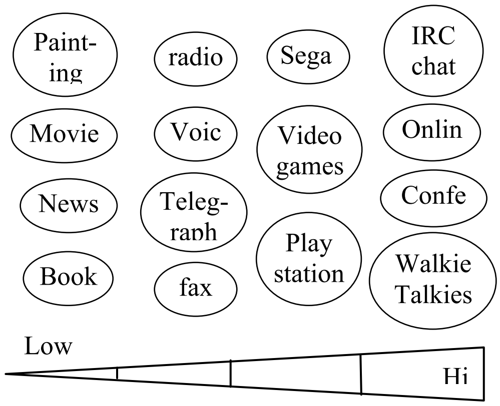

Fig. 2. Level of interactions [15]

2.3.2 Types of Interaction

Although interaction can be divided in many sub-categories depending upon the domain of research but according to the scope of this research work, interaction is categorized as Implicit Interaction and Explicit Interaction [13]. Implicit interaction can be defined as "the interaction which is not intentional". An action which is performed without the intention of interaction but regardless, the environment understands it as in input, is considered as implicit interaction. Explicit interaction can be defined as "the interaction which is intentional". In this kind of interaction, a user intentionally interacts with a system and gives input. Furthermore, implicit interaction falls in the category of low level of interaction, whereas explicit interaction falls in the category of high level of interaction.

2.4 Recommender System

This is a form of information filtering engine that gives suggestion to users about products, goods and services that they may be interested in. It assists users in the decision making process when presented with multiple choices on selected items [16]. As the name of the

engine reads, the recommender system only suggests options to users and in no way does it make decision on behalf of its subjects or users. Given its wide acceptance, it thrived successfully in the e-commerce environment where e-commerce sites and applications are powered by this system behind the scenes hence its huge popularity. In the e-commerce realm, such recommendations aid customers in locating what they may be looking for, thus increasing sales and profitability. Furthermore, users' spending behavior and pattern can be monitored with the recommender system to provide tailored and specific goods or services that suite them best.
There are two generic entities that are dealt with in any recommender system namely: the user (customer) and the item of interest. The user spectrum is diverse depending on the kind of system powered by the recommender system. They could be customers in an e-commerce platform or a passionate book reader looking for suitable books or a particular subject or title of interest. Users' ratings and comments on items bought or borrowed are accumulated in repositories the processed by algorithms to provide suggestions or recommendations to future users based on their requests hence reducing the burden of decision making.

2.5 Issues in Recommender System

Some of the major problems in recommender systems are discussed as follows.

a) Cold Start: The cold start problem typically affects new user or item when they are newly registered in a system. Usually, there is insufficient information and ratings for causing the recommendation algorithm not to accurately predict or recommend to the users. All recommendation techniques use ratings or item history logs for effective and efficient recommendation. However, it's hard for an algorithm to process the ratings given the scarcity of information leading to one of the major challenges of recommender systems. Cold-start problem is divided into two broad categories namely: new-user cold-start problem and new-item cold-start problem [17] [19].

b) Data Sparsity: In a system, the users may not rate some of the items. Therefore, the user-item matrix may have many missing ratings and be very sparse. Therefore, finding correlations between users and items becomes quite difficult and can lead to weak recommendations. Many users do not rate every item they like, so it is not necessary that if a user had not rated an item necessarily means that the user did not like the item. This is a major issue in recommender systems [18] [20] [21].

2.6 Types of Recommender Systems

There are different types of recommender systems like collaborative filtering recommender systems, content-based filtering recommender systems, hybrid filtering recommender systems, knowledge-based filtering recommender systems, demographic-based filtering recommender systems and utility-based filtering recommender systems [6] [8] [9] [22]. Here we are going to discuss collaborative filtering and content-based recommender system.

2.6.1 Collaborative Filtering (CF)

The CF approach [9] [17] [20] [23] [26] provides user with recommendations based on what a community of users with likely or similar interests or preferences might have liked previously on an e-commerce site. The CF approach takes advantage of existing semblances in user interest patterns to make recommendation via correlation of their preferences stored in the system's repositories over time. The CF-based systems are often classified as "memory-based CF" and "model-based CF".

The main advantages of collaborative filtering recommender systems are that they are more effective when it comes to customer satisfaction as they recommend the most appropriate items to users which are personalized at the same time [19]. The collaborative filtering algorithms are designed such that the accuracy of their prediction increases tremendously over item as more user preferences are added to the database, irrespective of the size of the database [21].

The main drawback of CF is that the system would not be very effective if a user changes his/her preferences unexpectedly, as the system still focuses on past interests of the user. It is also known as gray sheep problem [21]. For example, if a user has brought books on astrophysics for a long period of time but if user suddenly gains interest in other areas like psychology, he would still be recommended with the books on astrophysics, which might not be very useful to the user. It also suffers from the problems of sparsity of ratings [18] [20] [21] or single votes for items in the database along with the problem where a new user or new item is added to the database [17] [19].

2.6.2 Content-based Recommender System

Content-based recommendation systems or content-based filtering approaches [6] [8] [16] [33] are based on textual information such as documents. These items are typically described with keywords and weights. Using nearest neighbor functions or clustering methods can allow this recommendation system to analyze these keywords and document content and use them as a basis to recommend a suitable item. The type of information for the user's profile derived by the content-based recommender systems is largely depended on the learning method used, like decision tree, neural networks and etc. [8].

The main advantage of this method is that it does not depend on the user ratings of items in the database and hence, even if the database does not contain user preferences, the prediction accuracy is not affected. Even if the user preferences change, it has the capacity to adjust its recommendations in a short span of time. The main drawback of this approach is the need to know all the details of an item really well, even where the features of the item are stored in the database in a way where it cannot be inferred directly.

In this research, we have presented an approach to improve the accuracy of collaborative recommendation for new-user cold-start problem. In the next section we present our approach and an in depth discussion about the approach.

3. Proposed Approach

The proposed approach attempts to address the cold-start effect faced by new users in getting accurate recommendations in the PLE domain. This approach attempts to exploit the interaction between disparate users in making concise suggestions to the new users, a set of existing users that has huge similarity with the new users thus reducing the cold-start effect. The components of our approach can be seen in **Fig. 3**.

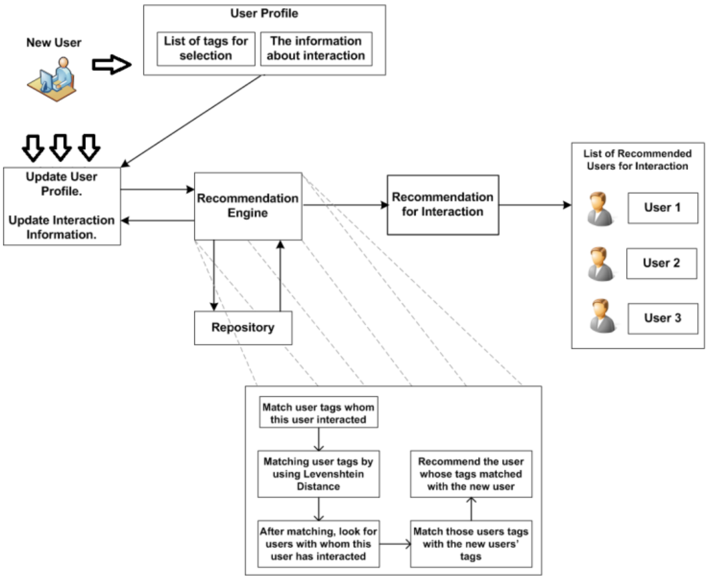

Fig. 3. Proposed collaborative filtering approach

The main focus of this approach is on the new users when they log on to the system. During the initial step, after the user creates a profile, a set of tags are presented to the new user for selection as an attempt to identify the user's interest. Having selected few of this tags as many as preferred, the profile is updated and recommendations are generated for this new user

As shown in **Fig. 3**, after a complete set-up of the user profile, it is sent to the recommendation engine for suggestions. The strength of the proposed approach lies in user interaction hence the recommendation tends to become more useful and efficient as the users continue to interact with each other in the system. The profile of interacting users are opened and matched with that of new users. The recommender system looks for a user with whom this other user has interacted. The profile of that user then opens up and that profile is then matched with the new user. If the user profile matches, this user is then recommended to the new user otherwise the engine will look for a user with whom this third user has interacted and then matches its profile with new users' profile as well. This fetch and match cycle continues until a profile that matches that of the new user is discovered. This interaction is labeled as implicit interaction which will be described in detail in Section 4.4.

The core of this approach depends immensely on the 'Levenshtein Distance' algorithm [34] for matching tags among user profiles. In this algorithm, the variation between two strings is calculated by the algorithm to identify similarities and differences between the users tags saved in the user profiles. So, when the tags are matched with the users' selected tags, the user is then recommended to the new user otherwise the system looks for another user with whom this user has interacted. Finally, a user is recommended to the new user whose choices and interests are similar to those of the new user. The pictorial presentation of problem can be seen in **Fig. 4**.

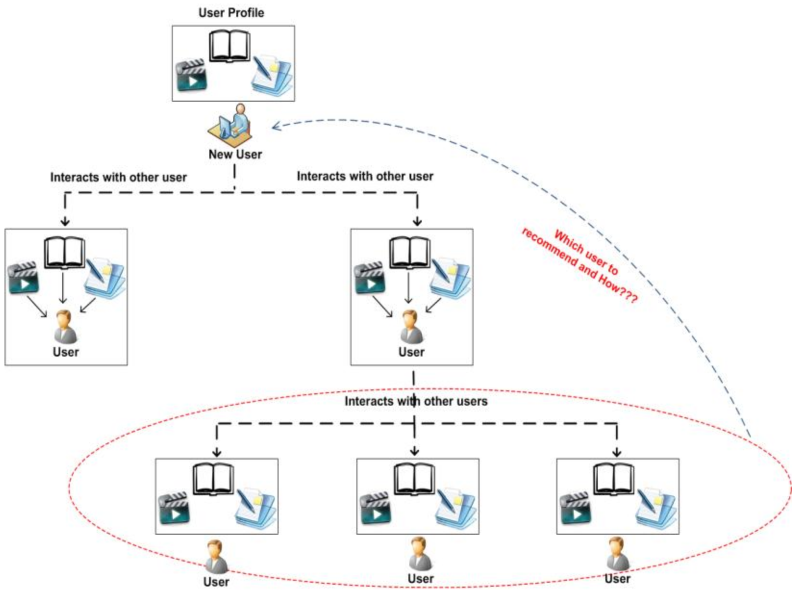

Fig. 4. Recommendation problem

Fig. 4 depicts the recommendation problem occurs when a new user is registered in the PLE. The problem manifests when the new user logs on to a PLE system. There is little or no information about the likes and interests of that user hence the question arises, *"how should the recommender system suggest existing users with similar preferences to the new user?"* The proposed approach intends to answer this by leveraging on the user interactions within the system. For instance, if a new user, while interacting with the system interacts with items such as document, book and etc. on other existing users' profile, this information is saved in the new user's profile in the form of tags. This interaction information is then used to match the profile of both users for similarities with the algorithm. Therefore, if the profile matches, that user is recommended. A complete description of how this approach works is discussed in the next section.

3.1 Interaction in Proposed Approach

In fact, the interaction is the communication between two entities. In a web based system, interaction is defined as the way in which a user communicates with the system. Furthermore, there are two types of interaction; explicit interaction and implicit interaction. Explicit interaction is defined as the intentional interaction. For example, when a user interacts with the system and provides input. Implicit interaction is defined as the interaction which is not intentional. The levels of interaction are between high and low depending upon the user control, satisfaction and the interaction frequency.

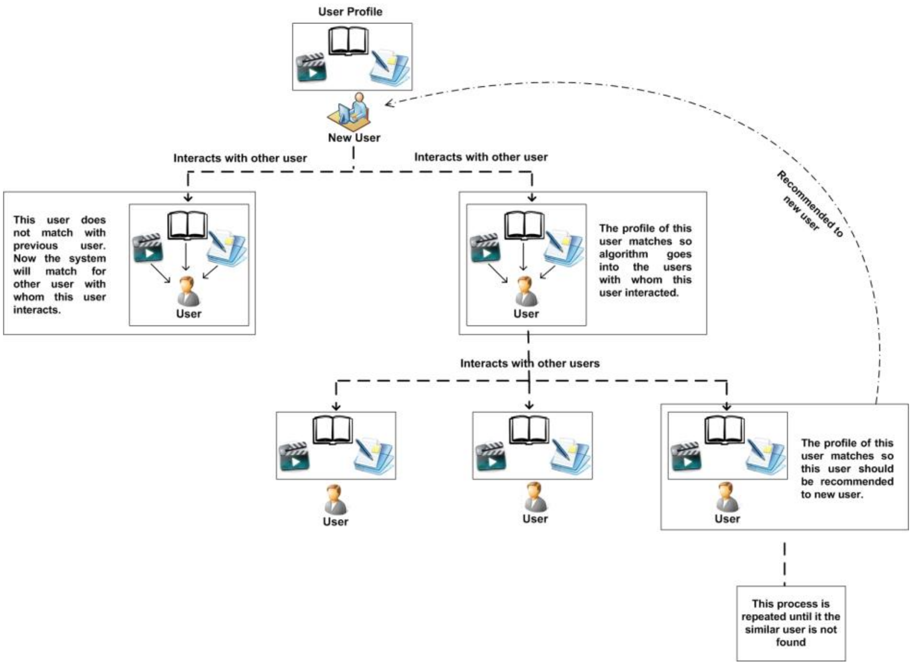

Fig. 5. Interaction in proposed approach

After the fundamental introduction of interaction, the application of interaction in the proposed approach is discussed. As shown in **Fig. 5**, when a new user enters in the system, a user profile with all the tags that the user selects based on his interests, is created for him. Next, when this new user interacts with an existing user, the system opens up the profile of this new user and looks up to whom the existing user has interacted earlier. The system then opens up the profile of the user that this current user has interacted with. This method explores the chain of interactions among existing users until the last user is reached.

The tags of this new user are then matched against the tags of the existing user. If the profile matches, this new user is recommended to the existing user; otherwise the system keeps on repeating the interaction-exploring process. The system keeps doing this recursive process until a user is found whose interests are same the new users' interest. Once, the profile matches that user is recommended to the new user.

The profile perspective of interaction explaining the process of matching of user profile of new user with the existing profiles of other users is shown in **Fig. 6**. The **Fig. 6** explains in detail about the use of user profile in matching and recommendation process. The tags selected by users are saved in users' profile and the information about the interaction is also saved in it.

In order to explain the activity involved in this whole process, an activity diagram is presented here for the activity wise understanding of the proposed approach.

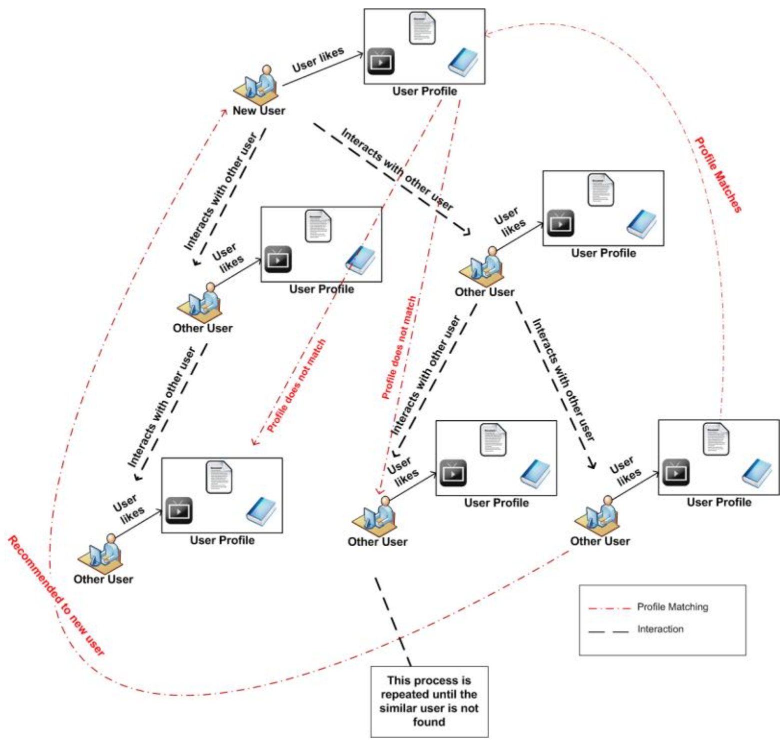

Fig. 6. Profile perspective of interaction

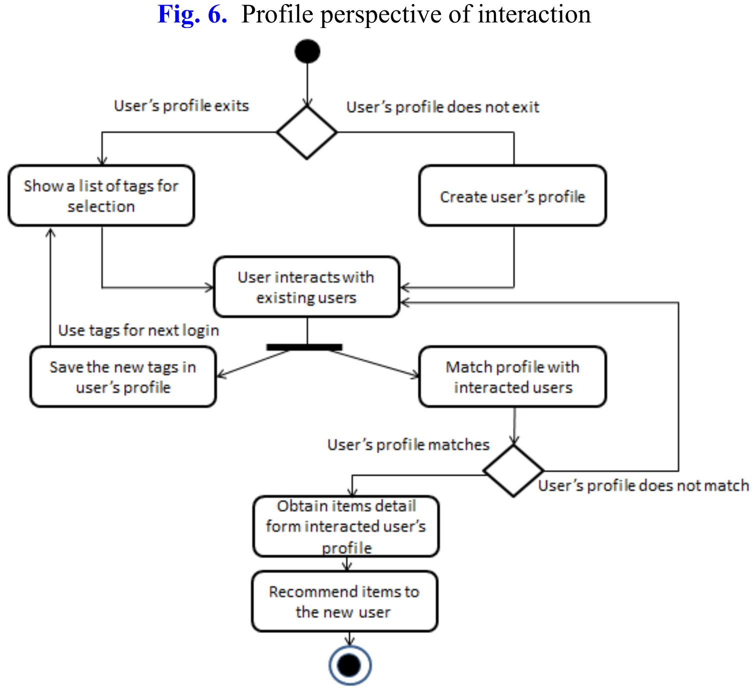

Fig. 7. Activity Diagram

As presented in **Fig. 7**, the very first operation the system performs is, when the user enters in the system it checks for the user's profile if it has been created or not. If it is not created then the system gives the user a list of tags for selection and saves these tags in this user's profile. The next activity is to capture the interaction of the new user with another user. Once the new user interacts with another user the system extracts both the users' profile and matches them. If they do not match, the system waits for another interaction of new user. But if the profile matched, the system reads this user's profile and looks for another user with whom this user has interacted. This process goes on until a suitable user is recommended to the new user.

The levenshtein distance algorithm is described below. The algorithm of the proposed approach is presented below along with the complete explanation which is given in **Table 1**. The algorithm is presented in a precise way with detailed explanation.

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j)=0 \\ \min \begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+[a_i \neq b_j] \end{cases} & \end{cases}$$

The use and the implementation of above equation is described in detail in the algorithm presented in **Table 1**.

Table 1. Proposed algorithm

The annotation of symbols
$U(u_1,u_2,u_3,...,u_n)$ //set of all users;
$T(t_1,t_2,t_3,...,t_m)$ //set of all tags which belongs to every user;
$I(i_1,i_2,i_3,...,i_k)$ //set of interactions which belongs to every user;
$k(1,2,3,...,n)$ //the number
$\forall u_k \ \exists T$ //for each user u_k there is a set of tags;
$\forall i \ \exists \ u_k, u_{k+1}$ //for each interaction *i* there are two users u_k and u_{k+1};
First Phase of user profile matching
//for each interaction *i*, if there is a set of tags which belongs to user u_k which is similar to the set of tags which belongs to user u_{k+1}.
$\forall i$ if $\exists \mathrm{T} \in u_k \equiv T \in u_{k+1}$
//if the above condition satisfied, go to the second phase.

Second Phase of user profile matching

//for each interaction *i*, if there is a set of tags which belongs to user u_{k+1} which is similar to the set of tags which belongs to user u_{k+2}

$\forall i$ if $\exists\, \mathrm{T} \in u_{k+1} \equiv T \in u_{k+2}$

//if the above condition satisfied, go the final phase.

Final phase of matching and recommendation

//if there is a set of tags which belongs to user u_{k+2} which is similar to the set of tags which belongs to user u_k, then user u_{k+2} is the selected user.

if $\exists\, \mathrm{T} \in u_{k+2} \equiv T \in u_k$

u_{k+2} is selected.

Here, important point to notice is that both explicit and implicit interactions are happening in this process but the main focus of this approach is on the implicit interaction. Since, the user is new and the user has very limited knowledge about the system and contents, explicit interaction might not be so helpful. Thus implicit interaction based recommendation helps in recommending the most appropriate existing users to the new user. As it can be seen clearly that in order for this approach to efficiently work, the new user only needs to interact with one user explicitly, from there the approach starts doing the process implicitly which is very effective since the user has no prior knowledge about the system.

4. Experimental Evaluation

4.1 Introduction to Dataset

The dataset used in the evaluation of the proposed approach is Movie lens dataset (**Table 2**) which contains the cold start users (users with less than 20 votes). This dataset contains 10000054 ratings and 95580 tags applied to 10681 movies by 71567 users. The minimum rating is 0.5 and maximum rating is 5 with the step size of 0.5. The mean, median and standard deviation of ratings is 3.61, 3.72 and 1.06 respectively.

For our experiments, we looked for users and tags information. Since cold-start users (users with less than 20 votes) are not accounted in Movielens dataset, so in order to perform experiments related to cold-start users, we have removed votes from the dataset. The users who have rated between 2 and 20 items and have few tags selected are referred to as cold-start users. We have slightly modified the data and added more data relevant to PLE.

Table 2. Main parameters of the Movie lens dataset

Number of Users	71567
Number of Movies	10681
Number of Tags	95580
Number of Ratings	10000054
Minimum Ratings	0.5
Maximum Ratings	5

We have extracted the data from Movielens dataset according to the feasibility for the experiments. We created three tables from the dataset. The first table constains the structure of the table of tag id and the tags from these tags a user will select. The second table contains the tag id and the tags, in the string format. The third table contains the user id and user name. The user names are added to the dataset which helped in presenting the generated list of users for recommendation.

4.2 Experimental Setup

In order to validate the improvement gained using this new approach, an experiment has been carried out with a new user account in the PLE domain given the nature of the problem. In this scenario, we showed that how this interaction-oriented approach solves the problem of the cold-start effect.
When a new user enters in the PLE system, a list of different keywords is shown to user for selection of a few tags. These tags are saved in the user profile and the profile is then used to generate first list of recommendation for the user. Since this approach is related to interaction, we assume that the user has updated the profile by visiting other links, other users' profile and some other articles. While the new user browses the PLE system, his profile is updated with the form of tags in user profiles he visited. This updated profile serves as input parameters for the algorithm to select the best user profile matches for the new user thereby making improved recommendation. A step by step description is described below.

Step 1: **Updating New User's Profile with Tags**– A new-user is registered in the system. This is the input for this step. A list of few tags is displayed to user for selection in order to describe his learning interest. The new user's profile is now updated with the selected tags which is the output of this step. This user's profile is continuously updated whenever the user interacts with the system, clicks on some article or visit some other users' profile. The input is a new user, the user selects few tags from a list of tags which is data and the output is the profile which is saved for every user.

Step 2: Tags Matching among Users – In this step the selected user's tags are retrieved from the updated profile of the user in the database when the user logs on to the system. The Levenshtein Distance [34] algorithm seamlessly matches the retrieved tags against other users' tags with whom this new user has interacted. This algorithm measures the difference between two strings. The algorithm cross-references the user profiles while searching and finally presents the best recommendation to the new user. A detailed explanation of Levenshtein Distance algorithm is given in **Table 3**.

In this step, the user profile serves as the input data while the output is the list of recommended users.

Step 3: Recommending A User to Another – In this step, we perform an in-depth exploration of other users' profile in order to discover the network of users that have interacted with these users. The fetching and matching process of each user's tags that this user has interacted with continues until all suitable user profile matches are generated. In simple terms, we recommend a user to another user if the tags between them match; else no recommendation is made and we continue the third step in a loop until all matches are found and presented to the new user. The results are discussed and shown in Section 4.3. The input for this step is the user profile. From this profile, again all tags information is retrieved and once again computed with other users profile in order to find the closest match for the new user. This is the data for this step. A list of 5 users is then recommended to the new user, which are the most appropriate candidate for the interaction. This information is updated in the profile as output.

Table 3. Levenshtein distance algorithm

- Set n to be the length of x, and m to be the length of y
- Create a matrix with m rows and n columns and initialize the first row and column to $0 \ldots m$ and $0 \ldots n$ respectively.
- Examine each of the characters of x and y to 1 to n and 1 to m.
- If $x[i] = y[i]$, the characters are equal and the transformation cost is 0. If $x[i] \mathrel{!}= y[i]$, the characters are not equal and the transformation cost is 1.
- The value of cell $d[i, j]$ is set to the minimum of {*d[i-1, j] + 1 (the cell above + 1)*}, {*d[i, j-1] + 1 (the cell to the left + 1)*}, or {*d[i-1, j] + cost (the cell diagonally above and to the left)*}.
- Step 3-5 is repeated until the distance score is found in cell $d[n, m]$.

After the execution of the algorithm, a matrix will be created in Matlab indicating the possible user for recommendation. The levenshtein algorithm is presented in Tables 3 and the output after the execution is shown in **Table 4**.

Table 4. The output matrix after matching

	U_1	U_2	U_3	I
U_1	t_3	t_1	t_2	✓
U_2	t_1	t_3	t_2	✓
U_3	t_2	t_2	t_3	✗

In **Table 4**, it can be seen that first step is to determine which is appropriate to look in. It can be decided by matching the tags for the new-user to the second user that this user has interacted with. If the tags matched among them, then all the users, with whom this second user has interacted, are matched one by one with the new-user. If a user with similar interests is found, it is recommended to the new-user.

After completing the experiment with all results documented, the latter was validated using Precision, Recall and F1 measures. A Comparison was done using the latest approach with the new experimental result showing noticeable and remarkable improvement. The details are given below in next section.

4.3 Results and Discussion

In this section, the results of the evaluation of our proposed approach are discussed. The experimentation process is discussed in detail in previous Sections. Precision and Recall are used to evaluate the performance of the proposed approach. It is apparent from the results that the proposed approach corroborates good results for new-user cold start issue in Personal Learning Environment.

Precision and Recall measure are the most commonly used measures to validate the results [16] [35] [36]. Precision is the measure of the retrieved items which are actually relevant while Recall is the measure of the overall retrieved items which maybe contain irrelevant items as well [35]. Both of these measures are reciprocal of each other. F1 score is the measure of the accuracy of a test. It is the weighted average of Precision and Recall and shows the overall performance of both of them [16] [36]. The formula for each of these measures is given below.

$$\text{Precision} = \frac{\left|(\text{retrieved items})\right|}{\left|(\text{relevant items}) + (\text{retrieved items})\right|}$$

$$\text{Recall} = \frac{\left|(\text{relevant items})\right|}{\left|(\text{relevant items}) + (\text{retrieved items})\right|}$$

$$F_1 = 2\left(\frac{\text{precision.recall}}{\text{precision} + \text{recall}}\right)$$

After the experimentation and the acquiring of results, these results are validated by Precision, Recall and F1 measures. Comparison is done with latest approach and the result shows significant improvement. The details are given below.

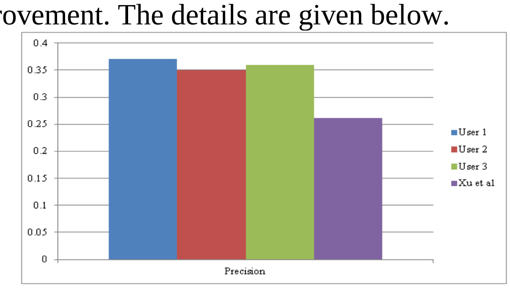

Fig. 8. Precision comparison

Fig. 8 shows the comparison of precisions of our approach with [37]. The experiments are performed with three different user profiles and the result shows improvement of 31.57%.

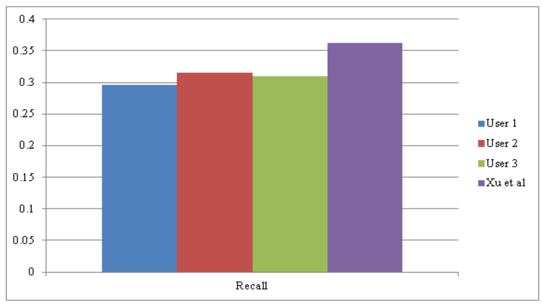

Fig. 9. Recall comparison

The recall comparison of our approach can be seen in **Fig. 9**. As it can be clearly seen, the recall value is dropped by average of 18.29% when compared with the recall of [37]. The recall is also compared with three different user profiles.

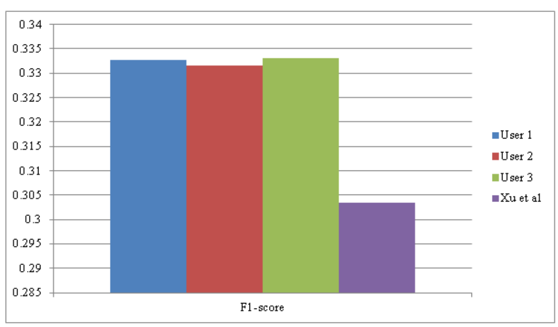

Fig. 10. F1-score comparison

F1-score are compared and it shows improvement too which is shown in **Fig. 10**. As it can be seen, it shows improvement of 8.8% when compared with [37]. To calculate F1-score we also use three different user profiles.

5. Conclusion

E-learning has become an integral part of online learning process for the learners. As the technology advances, so does the problem. A very common problem in this regard is the problem of new-user cold start. It is the problem in which a new user enters a PLE system. The system has to recommend personalized contents to the learner. However, as the learner is new and there is not much information about the learner, the system fails to recommend most appropriate learning content to the learning. In this research paper, we have presented the implementation of interaction-based collaborative filtering approach to improve the recommendation accuracy for new–user (new learner) cold start problem. We have discussed about our approach and the experimental evaluation in detail in sections 3 and 4 respectively. The result shows the improvement in the accuracy of recommendation for the new-user cold start problem.

Acknowledgement

This research was supported by a research grant from Ministry of Higher Education (MOHE) Malaysia under Vot No. 4F315. In addition, we are thankful to Universiti Teknologi Malaysia (UTM) for providing research facilities.

References

[1] S. Duo and L. X. Song, "An E-Learning System based on Affective Computing," in *Proc. of International Conference on Applied Physics and Industrial Engineering*, 2012, pp. 1893-1898. Article (CrossRef Link)

[2] B. Wu, *et al.*, "Experience Effect in E-Learning Research," *Physics Procedia*, vol. 24, pp. 2067-2074, 2012. Article (CrossRef Link)

[3] F. Henri, *et al.*, "Understanding PLE as an Essential Component of the Learning Process," in *Proc. of World Conference on Educational Multimedia, Hypermedia and Telecommunications*, 2008, pp. 3766-3770. Article (CrossRef Link)

[4] M. V. Harmelen, "Design Trajectories: Four Experiments in PLE Implementation," *Interactive Learning Environment*, vol. 16, pp. 35-46, 2008. Article (CrossRef Link)

[5] F. Modritscher, "Towards a Recommender Strategy for Personal Learning Environments," in *Proc. of 1st Workshop on Recommender Systems for Technology Enhanced Learning*, 2010, pp. 2775-2782. Article (CrossRef Link)

[6] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," *User Modeling and User-Adapted Interaction*, vol. 12, pp. 331-370, 2002. Article (CrossRef Link)

[7] M.W. Chghtai, *et al.*, "Short systematic review on e-learning recommender systems," *Journal of Theoretical and Applied Information Technology*, vol. 57, pp. 139-148, 2013. Article (CrossRef Link)

[8] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the Start-of-the-Art and Possible Extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, 2005.Article (CrossRef Link)

[9] X. Zhou, *et al.*, "The State-of-the-Art In Personalized Recommender Systems for Social Networking," *Artificial Intelligence Review*, vol. 37, pp. 119-132, 2012. Article (CrossRef Link)

[10] S. Duo and L. X. Song, "An E-learning System based on Affective Computing," *Physics Procedia*, vol. 24, pp. 1893-1898, 2012. Article (CrossRef Link)

[11] U. Kirschenmann, *et al.*, "Demands of modern PLEs and the ROLE approach," *Sustaining TEL: From Innovation to Learning and Practice*, pp. 167-182, 2010. Article (CrossRef Link)

[12] L. Ha and E. L. James, "Interactivity reexamined: A baseline analysis of early business web sites," *Journal of Broadcasting & Electronic Media*, vol. 42, pp. 457-474, 1998. Article (CrossRef Link)

[13] R. Sun, *et al.*, "The interaction of the explicit and the implicit in skill learning: a dual-process approach," *Psychological review*, vol. 112, p. 159, 2005. Article (CrossRef Link)

[14] J.-n. Sun and Y.-c. Hsu, "An experimental study of learner perceptions of the interactivity of web-based instruction," *Interacting with Computers*, vol. 24, pp. 35-48, 2012.

Article (CrossRef Link)
[15] J. Steuer, "Defining virtual reality: Dimensions determining telepresence," *Journal of communication*, vol. 42, pp. 73-93, 1992. Article (CrossRef Link)
[16] A. Gunawardana and G. Shani, "A Survey of Accuracy Evaluation Metrics of Recommendation Tasks," *Journal of Machine Learning Research*, vol. 10, pp. 2935-2962, 2009. Article (CrossRef Link)
[17] J. Bobadilla, *et al.*, "A collaborative filtering approach to mitigate the new user cold start problem," *Knowledge-Based Systems*, vol. 26, pp. 225-238, 2012. Article (CrossRef Link)
[18] C. Birtolo, *et al.*, "Personalized Suggestions by means of Collaborative Filtering: A Comparison of two different Model-based Techniques," in *Proc. of 3rd World Congress on Nature and Biologically Inspired Computing NaBIC '11*, 2011, pp. 444-450. Article (CrossRef Link)
[19] N. Koenigstein, *et al.*, "Yahoo! Music Recommendations: Modeling Music Ratings with Temporal Dynamics and Item Taxonomy," in *Proc.of the fifth ACM Conference on Recommender systems RecSys '11*, 2011, pp. 165-172. Article (CrossRef Link)
[20] J. L. Herlocker, *et al.*, "Evaluating Collaborative Filtering Recommender Systems," *ACM Transactions on Information Systems (TOIS)*, vol. 22, pp. 5-53, 2004. Article (CrossRef Link)
[21] M. López-Nores, *et al.*, "Property-based collaborative filtering for health-aware recommender systems," *Expert Systems with Applications*, 2012. Article (CrossRef Link)
[22] Y. Jiao and G. Cao, "A Collaborative Tagging System for Personalized Recommendation in B2C Electronic Commerce," in *Proc. of the International Conference on Wireless Communications, Networking and Mobile Computing, 2007*. Article (CrossRef Link)
[23] J. Bobadilla, *et al.*, "A collaborative filtering similarity measure based on singularities," *Information Processing & Management*, vol. 48, pp. 204-217, 2012. Article (CrossRef Link)
[24] J. B. Schafer, *et al.*, "Collaborative Filtering Recommender Systems," *Lecture Notes in Computer Science*, vol. 4321, pp. 291-324, 2007. Article (CrossRef Link)
[25] N. Barbieri and G. Manco, "An Analysis of Probabilistic Methods for Top-N Recommendation in Collaborative Filtering," in *Proc. of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases ECML PKDD '11*, 2011, pp. 172-187. Article (CrossRef Link)
[26] H. N. Kim, *et al.*, "Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation," *Electronic Commerce Research and Applications*, vol. 9, pp. 73-83, 2010. Article (CrossRef Link)
[27] H. Lanseth and T. D. Nielsen, "A Latent Model for Collaborative Filtering," *International Journal of Approximate Reasoning*, vol. 53, pp. 447-466, 2012. Article (CrossRef Link)
[28] J. S. Breese, *et al.*, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," in *Proc. of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, 1998. Article (CrossRef Link)
[29] T. Hofmann, "Collaborative Filtering via Gaussian Probabilistic Latent Semantic Analysis," in *Proc. of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pp. 259-266, 2003. Article (CrossRef Link)
[30] D. M. Blei, *et al.*, "Latent Dirichlet Allocation," *The Journal of Machine Learning Research*, vol. 3, pp. 993-1022, 2003. Article (CrossRef Link)
[31] M. Grcar, *et al.*, "kNN Versus SVM in the Collaborative Filtering Framework," *Data Science and Classification*, pp. 251-260, 2005. Article (CrossRef Link)

[32] A. Paterek, "Improving Regularized Singular Value Decomposition for Collaborative Filtering," in *Proc. of 13th ACM International Conference on Knowledge Discovery and Data Mining*, 2007, pp. 39-42. Article (CrossRef Link)
[33] S. Shishehchi, *et al.*, "Review of Personalized Recommendation Techniques for Learners in E-learning Systems," in *Proc. of the International Conference on Semantic Technology and Information Retrieval (STAIR)*, 2011. Article (CrossRef Link)
[34] L. Yujian and L. Bo, "A normalized Levenshtein distance metric," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, pp. 1091-1095, 2007. Article (CrossRef Link)
[35] S. M. Ali and I. Ghani, "A Review on Recommender Techniques, Systems and Evaluation Metrics," *Science International-Lahore*, vol. 4, pp. 503-511, 2012. Article (CrossRef Link)
[36] C.-F. Tsai and C. Hung, "Cluster ensembles in collaborative filtering recommendation," *Applied Soft Computing*, vol. 12, pp. 1417-1425, 2012. Article (CrossRef Link)
[37] D. Xu, *et al.*, "A Collaborative Tag Recommendation Based on User Profile," in *4th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2012*, pp. 331-334.Article (CrossRef Link)
[38] M. W. Chughtai, *et al.*, "E-learning recommender systems based on goal-based hybrid filtering," *International Journal of Distributed Sensor Networks, Article ID 912130*, pp 1-18, 2014. Article (CrossRef Link)
[39] M. W. Chughtai, *et al.*, "Goal-based Framework for cold-start problem using multi-user personalized similarities in e-Learning scenarios," in *Proc. of 2nd International Conference on E-Learning and E-Technologies in Education, ICEEE 2013*, pp 334-338, 2013. Article (CrossRef Link)
[40] I. Ghani, *et al.*, "Semantics-oriented approach for information interoperability and governance: Towards user-centric enterprise architecture management," *Journal of Zhejiang University: Science C*, vol. 11, pp 227-240, 2010. Article (CrossRef Link)
[41] S. M. Ali and I. Ghani, "An Interaction-based Collaborative Filtering Approach for Personal Learning Environment", *Science International-Lahore*, Vol 25(3), pp 477-482, 2013. Article (CrossRef Link)

Syed Mubarak Ali received the B.S degree in Information Technology from PIMSAT (Karachi) and M.Sc. degree in software engineering from the Universiti Teknologi Malaysia (UTM), Malaysia. Since then, he has been associated with Wavelet Solutions, Malaysia, in Research and Development for ERP Systems. He is currently working on web services and the improvement of ERP systems to support offline transactions. His area of interests includes artificial intelligence, software engineering, agile methodologies, object oriented programming, databases and project management.

Imran Ghani is a Senior Lecturer at Faculty of Computing, Universiti Teknologi Malaysia (UTM), Johor Campus. He received his Master of Information Technology Degree from UAAR (Pakistan), M.Sc. Computer Science from UTM (Malaysia) and Ph.D. from Kookmin University (South Korea). His research focus includes agile software development methods and practices, semantics techniques, secure software development life cycle, web services, software testing, enterprise architecture and software architecture.

Muhammad Shafie Abd Latiff is an Associate Professor at Faculty of Computing, Universiti Teknologi Malaysia (UTM), Johor Campus. He received his BSc in Computer Science and MSc in Computer Science from Universiti Teknologi Malaysia (UTM) and PhD in AI method from Bradford University, United Kingdom. His research focus in AI methods, includes mobility algorithm for improving network coverage in wireless Nano-sensor networks, and Virtual Environments.