International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 9 Issue 7 ǁ 2021 ǁ PP. 14-22

Recommending Best Teaching Practices Based on Instructor's Interests

Khulud Alshahrani and Jawad Berri
College of Computer and Information Sciences, King Saud University
Riyadh, Saudi Arabia

ABSTRACT
Improving student learning outcomes is a primary goal for elearning. While most efforts in instructional design has been spent on developing ingenious learning content, the teaching process needs more attention and efforts to be enhanced. This includes applying the available technologies to help instructors elicit, share and reuse their knowledge and teaching expertise. This can be done through Best Teaching Practices (BTP) which are successful daily teaching experiences and practices developed and used by instructors. BTPs represent the main building blocks of the teaching process; they need to be expressed and organized in order to be shared by instructor's communities. In this research, we propose a system that allows sharing and reusing BTPs by automatically recommending the most suitable BTPs to instructors' interests. To promote and enhance BTP reuse, we propose an interactive personalized BTP recommendation system based on matching the teaching context of the instructor with the context of the recommended BTP.

KEYWORDS
Recommender system, best teaching practice, machine learning, content-based recommendation

---

Date of Submission: 06-07-2021 Date of acceptance: 20-07-2021

I. INTRODUCTION

Expertise of instructors has a significant effect in improving student learning outcomes in classical and e-learning [1, 2]. Beyond the borders of the preparation of the learning content, instructors should also pay attention to applying the suitable teaching strategies, using the appropriate instructional methods and deciding on the right tools and technologies [3]. As stated by many researches [4, 5], there is a strong relationship between teaching experiences of an instructor and the learning content understanding. To fill out this gap, researchers should concentrate on developing methods and systems that are able to gather and exchange in a large scale these experiences among instructors. Best Teaching Practices (BTPs) can be the facilitators for this effort, BTPs can be defined as the successful teaching experiences and instructional practices used by instructors which have proved good effect on the teaching process and learning content understanding. BTPs need to be effectively organized and shared among communities of instructors in order to be gathered and reused. This represents the motivation of this research which proposes a system that manages gathering, classification and retrieval of BTPs among communities of instructors. For this purpose, the system classifies and stores BTPs shared by instructors in the computing field by using a standard classification system; the ACM (Association of Computing Machinery) computing classification system [6]. In addition, it provides users with the advantage of ranking the search results based on the similarity to user's profile.

In this research we propose an interactive personalized recommendation system which proposes the most relevant BTPs for instructors based on their needs and preferences. We are specifically interested in the BTPs used in teaching the computing courses. To promote and enhance the BTP's reusing process, we apply a personalized BTP recommendation system based on matching the teaching context with the context of the recommended BTPs. The system uses instructor's feedback to improve the recommendation accuracy and solve the cold-start and overspecialization problems. The recommendation process starts by recommending a list of BTPs based on the instructor's interests in his profile, and then, the system gets the instructor's feedback on the recommended BTPs to update the recommendation list according to BTPs he likes or dislikes.

The proposed system provides instructors with the ability to browse and reuse BTPs based on the context by exploring a list of computing subject fields where each of them contains the BTPs that could be applied in that subject. It proposed also a personalized BTP recommendation system that matches the teaching context with the context of the recommended BTPs. The system gets the instructors feedback on the recommended BTPs to improve the recommendation accuracy and solve the overspecialization problem.

II. BACKGROUND

2.1. Best Teaching Practice

A BTP is any method, technique, strategy or practice used by instructors in teaching and have been proven to be effective in delivering the learning content and managing the teaching process [7]. It includes direct and indirect teaching practices. Indirect teaching practices are strategies that are used to improve the delivery of information and knowledge to students. On the other hand, the direct teaching practices deal with the classroom management and lesssons' presentation.

2.2. Content-based Recommendation

Content-based recommendation (CBR) and collaborative filtering are the most commonly used recommendation approaches for developing recommendation systems [8]. Collaborative filtering is based on using the user's behavior and his past decisions to predict the relevant recommendation items. On the other hand, content based recommendation solves the cold start problem by ignoring users' past behavior. In CBR, the items are described by attributes that will be used to match the user's attributes in order to find the relevant items. The items' characteristics can better explain the reasons for recommending a certain item so that users can get the most relevant items. The concept of CBR can be described with the following steps: first, a vector is created to represent the user's interests. Afterwards, a text vector is generated by performing some processing including: segmentation, indexing, weighing of word frequency statistics. Finally, the similarity between the user vector and text vector is calculated and the items with high similarities are recommended for the user.

2.3. Recommendation Systems Challenges

There are many problems facing the development of recommendation systems, the most prominent are the cold start and over specification problems.

- Cold start problem: this problem occurs when collaborative filtering based recommendation systems try to recommend items for new users [9]. These systems are based on exploiting the user interactions with other users and items and his past behavior. In this case, the system does not have any information that leads the recommendation process for the new users. On the other hand, some systems are based on recommending items using attributes such as item popularity or item ratings. The content based recommendation techniques, which use the users and items information, are considered as a solution for this problem.
- Overspecialization: this problem is considered as a disadvantage of the content based recommender systems. It happens when the user is recommended with items that are very similar to his profile contents, resulting in a very narrow range of recommended items [10]. This prevents the recommendation diversity, which is a desirable characteristic of all recommendation systems.

III. RELATED WORK

There are three main categories of recommendation systems: Content based, Collaborative Filtering and Hybrid which integrates both content and collaborative recommendation techniques [11, 12].

Cheng et al. presented in [13] a solution for the implicit-feedback-based recommendation problem through the principle of item similarity. The researchers introduce a new mixed similarity model including a recommendation algorithm that is based on the mixed similarity and pairwise preference assumption. The proposed model: (i) use a symmetric similarity to identify the useritem interactions, (ii) apply the asymmetric similarity to model the global correlations between items, and (iii) use the pairwise preference assumption to remove the uncertain implicit feedback. For the evaluation of the model, four public datasets obtained via converted implicit feedback are used. A comparison of the test results shows increased accuracy recommendation than the traditional recommendation strategies.

Xie and Wang proposed in [14] a novel hybrid web page recommendation framework that uses a novel twofold clustering algorithm for web page organization. The proposed clustering algorithm is based on density-based clustering and the k-means clustering. The hybrid recommendation introduced in this paper (i) organize the web pages according to its content to identify the relations between the web page clusters and users, and (ii) use the web page clusters as items to apply the SVD method on and users for mining the collaborative relations and include the freshness and popularity factors in the model. The empirical studies prove improved recommendation performance when compared to the several recently published algorithms including Goo, ClickB, 15 Bilinear, Bandit, SCENE and GU. The authors use F1 score to evaluate the model accuracy and the variance of F1 score to evaluate stability.

In [15], Sundermann et al. introduced a solution for the lack of automatic methods to incorporate the contextual information into the recommendation procedure. The researchers propose an unsupervised method to collect the contextual information from the web page text instead of the user's context. The proposed method uses the privileged information including domain terms and named items found in the web page text to create topic hierarchies. These topics are used by the context-aware recommendation systems as contextual information. An evaluation of the proposed approach is conducted using two access log bases: the Embrapa Agency of Technology Information and the Informal Dictionary to build 10 topic hierarchies for each data set. The evaluation results are compared against a recommendation system that does not consider contextual information on the same datasets and show an optimized recommendation quality using a context-aware recommendation system.

Miao and Lang introduced in [16] a recommendation system based on text mining methods applied to the Chinese language. The developed system assigns a score to the user's sentiment extracted from his reviews by a classifier. To train the classifier, an improved logistic regression in sentiment analysis is used. Moreover, to increase the recommendation accuracy, the system builds an item feature matrix to compute the feature similarity of the items. A website data including user ID, merchant ID, user score, and user reviews are used as the datasets to evaluate the system performance. The researchers choose the Mean absolute (MAE) as the evaluation criteria. The results give a clear vision on the improved quality of the proposed system.

Hsieh et al. in [17] presented a solution to address the cold-start issue by proposing two modules to establish a keyword-aware recommendation system. The first module is responsible for estimating the initial user ratings by extracting keywords from external domains and external information related to the users and items. To increase the dimensionality of the keywords space, WEB scrappers are used to acquire the external information. On the other hand, the second module performs the basic recommendation services. To test the proposed system, the descriptions of the users and movie-related items are considered as the external information and used to estimate the initial user ratings. The empirical results indicate lower prediction errors of item recommendation 16 than the other recommendation systems by calculating their RMSE values, and higher recommendation accuracy. A shortcoming of the proposed architecture is the need for a big data environment to estimate the user ratings.

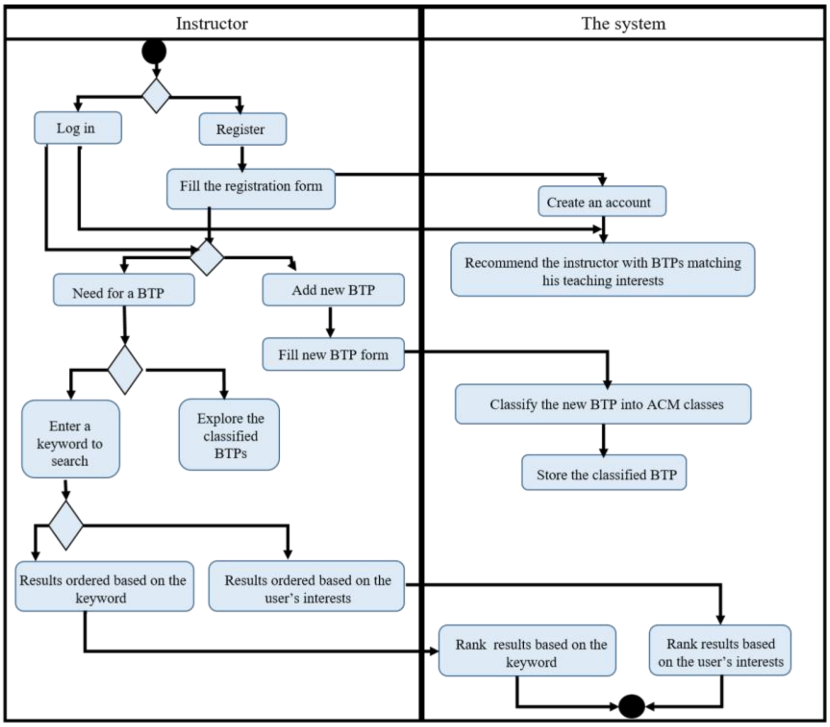

Figure 1. The system flow.

IV. SYSTEM OVERVIEW

The system is designed to share and reuse BTPs among instructors. To achieve this goal, instructors can explore the stored BTPs by the ACM classification through a dropdown list or search for a certain BTP by specifying a keyword in the search box without having to register in the website. However, having an account will add an advantage to instructors' experience with the system. The information in the instructor's account is used to enhance the searching process by ranking the search results according to the instructor interests. Also, this information is used to recommend the BTPs that best match the instructor's teaching interests. On the other hand, for instructors who have some BTPs to share, they must have an account. Instructors can interact with the system through the following steps. The first step includes registering the instructor into the system. In this phase, a registration form is filled by the instructor and his account is created. Once the account is created, the BTPs that match the instructor's interests mentioned in the registration form are recommended to him. After the registration phase, the instructor can either add new BTPs to the system or explore the stored BTPs by their classes. Adding new BTPs involve filling a new BTP form (examples of BTPs are presented in Fig. 3 and Fig. 4). The form requires the following information: BTP title, Keywords, BTP description, level of students for whom the BTP can be applied, courses in which the BTP can be applied, attachments files and the copy / publish rights. Once the form is filled and the add new BTP button is clicked, the BTP description entered by the instructor is used to start the classification process. At the end of the classification process, the new BTP will be stored in the system database along with its corresponding ACM classes predicted by the classifier. Fig.1 indicates the activity diagram of the system and shows the flow of information.

V. THE PROPOSED SYSTEM

According to [18], to design a good recommendation system, the user's context and preferences should be exploited to recommend the most relevant content, which results in a highly personalized recommendation system. Content-based recommendation is based on exploiting the user information and ignoring contributions from other users and the user behavior data. This advantage offers a solution for the cold start problem. On the other hand, recommendation based on constant characteristics may results in the problem of overspecialization. To overcome this problem, an interactive assigned-weight method is proposed in [7], that assigns weights to a list of recommendation candidates according to real-time feedback. The candidates list is obtained through a hybrid recommendation algorithm. The research in [19] used the same user interaction principle where the candidates list is created based on tag information. In this research, we used a similar principle that is based on two-phase recommendation process as shown in Fig 2. The first phase involves creating an initial list of the recommended items using content-based recommendation technique. To avoid the overspecialization problem, the second phase involves user interaction process to assess the initial list of recommended items created in the first phase.

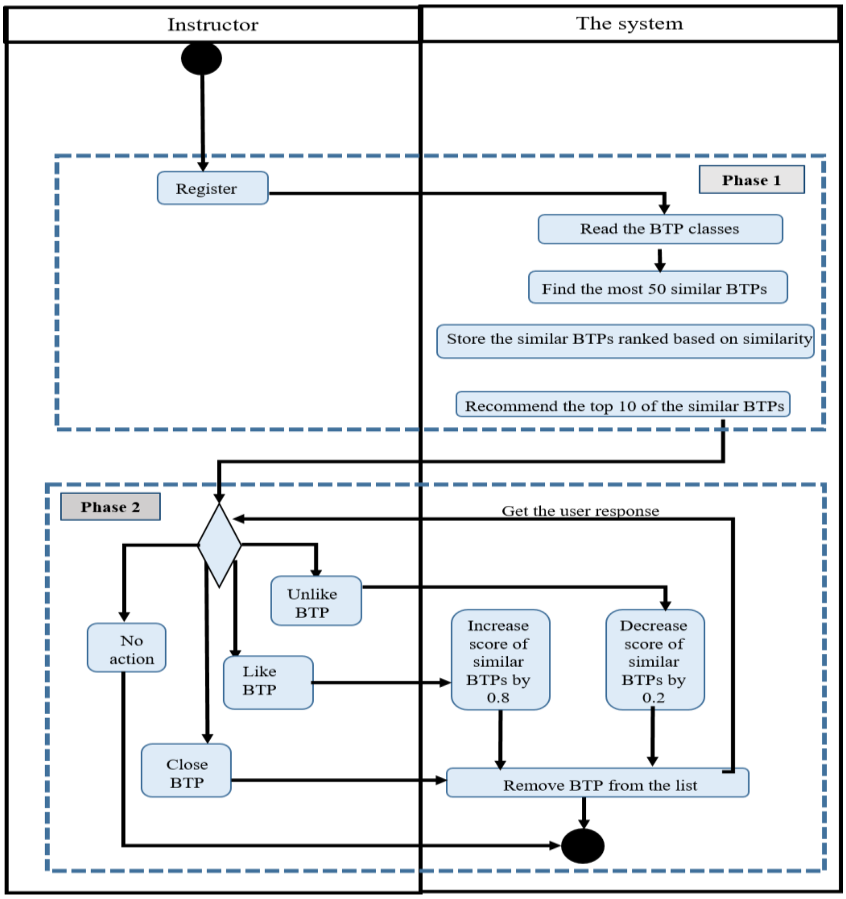

Figure 2. The flow of the recommendation system

5.1. The first phase

This phase starts after the user registration process. The system uses a content-based recommendation algorithm that utilizes the user's interests to find the most similar 50 BTPs as an initial list. After that, these BTPs are ranked based on the degree of similarity with user interests and stored in the userbtps table in the database. Once the user log in the system, the top 10 BTPs are recommended to him.

In order to find the similarity between user's interests and the stored BTPs, the function cosine_similarity() is used. This function, imported from sklearn.metrics.pairwise library, takes two parameters x and y to create a similarity matrix where the first parameter is the users' interests matrix and the second parameter is the BTPs classes matrix. The intersection of each row and column represent the value of cosine similarity between that row and column.

Since the values of cosine similarity function represent the ratio of similarity, we had to arrange these values to find the most similar BTPs ids. We need to get the top 50 cosine similarities between all the users and all the stored BTPs.

To store the similar BTPs in the user profile, every similar BTP is stored in user_btps table in the database along with the corresponding user id. The primary key in this table is a composite primary key consists of two columns: (userid, BTPid). In this way, we can guarantee no repeated BTPs for a user and vice versa. The recommended BTPs are displayed with a close button to remove BTPs and a read button that allow user to open the selected BTP in another page to view its details.

5.2. The second phase

In this phase, the system gets the user response about the 10 items recommended at the first phase. Each recommended BTP is displayed with its title, three lines of its description, close and read more buttons. Reading the BTP allows the user to like or dislike the BTP. Like and dislike the recommended BTP is the interaction that will be used as a feedback to update the recommendation list for each user according to what he likes and what he does not. The update process is performed by increasing the scores of the BTPs similar to the liked BTP and decreasing the score of the BTPs similar to the disliked BTP. The user is expected to respond to the recommended BTPs into one of the following four interactions:

1. The user opens the BTP and like it. In this case, the score of all BTPs similar to the liked BTP are increased by 0.8 (resulted from 1*0.8) Then, the liked BTP is removed from the recommendation list in the user profile and substituted by the next BTP. Therefore, the next recommendation process involves new recommended BTPs due to the updated scores. Clicking like link of a recommended BTP calls the likeRecommendedBTP() function which execute the following steps: 1. Gets the user id. 2. Calculate the cosine similarity between all the BTPs in the system database. 3. Find the 10 most similar BTPs. 4. Get the scores of the BTPs from the user_btps table. _5. Increase the score of each similar BTP by 0.8 6. Update the user_btps table by the new socres 7. Delete the liked BTP from user_btps table so that it will not be recommended again.
2. The user opens the BTP and dislike it. In this case, the score of all BTPs similar to the disliked BTP are decreased by 0.2 (resulted from 1*0.2) Then, the disliked BTP is removed from the recommendation list in the user profile and substituted by the next BTP. Therefore, the next recommendation process involves new recommended BTPs due to the updated scores.
3. The user closes the BTP. In this case, the closed BTP is removed from the recommendation list in the user profile and substituted by the next BTP. Clicking dislike link of a recommended BTP calls the dislikeREcommendedBTP() function which execute the following steps: 1. Gets the user id. 2. Calculate the cosine similarity between all the BTPs in the system database. 3. Find the 10 most similar BTPs. 4. Get the scores of the BTPs from the user_btps table. 5. Decrease the score of each similar BTP by 0.2 6. Update the userbtps table by the new scores. 7. Delete the disliked BTP from user_btps table so that it will not be recommended again.
4. The user doesn't open, close, like or dislike any recommended BTP. Thus, no change is made to the recommendation list waiting for next response.

5.3. Weighted Linear Combination method

Using the Weighted Linear Combination method, we make the like and dislike processes add a weight to the score. This method is a multi-attribute decision making analytical method that is used in cases of having more than one weighting criteria [20]. It is based on the following steps:

1. Define the weighting criteria: two criteria are used to find the value of each BTP as in Table 1.

Table 1: BTP scoring criteria

Criteria	**BTP value**
Matching to a liked BTP	Increased by one
Matching to a disliked BTP	Decreased by one

2. Assign weights for each criterion based on its importance: since the recommendation is based on similarity to the liked BTP, the criteria of matching to a liked BTP is the most important see table 2.

Table 2: Scoring criteria and its weights

Criteria	Importance	Weight
Matching to a liked BTP	80%	0.8
Matching to a disliked BTP	20%	0.2

3. Calculating the weight of each BTP by multiplying the weight assigned to each criteria by the value of the BTP for that criteria, and then all the products over all criteria are summed up to get the final weight. This is expressed in Eq. (1).

$$W = \sum w_i v_i \quad (1)$$

where W is the final BTP weight, w_i is weight of criteria i, and v_i is the BTP value for that criteria i. The weighting equation to calculate the weight of a BTP b is Eq. (2).

$$W_b = (0.8 \times matching\ to\ a\ liked\ BTP\ value_b) + (0.2 \times matching\ to\ a\ disiked\ BTP\ value_b) \quad (2)$$

VI. RESULTS AND DISCUSSION

6.1. Dataset

The data used in this experiment was collected manually from different websites on the internet. Examples of the internet websites are www.tes.com, www.teachthought.com, www.teachervision.com, www.educationworld.com, http://csteachingtips.org. The BTPs dataset collection demands time-consuming manual procedure of organizing the instructors' expertise. The collected BTPs were stored as text data as a list of records. Each record represents a BTP along with its title, keywords, description, student's level, courses and attachments. Two types of BTPs were used: specialized BTPs related to teaching computing courses and are classified according to ACM CCS. And general BTPs related to general instructional and pedagogical aspects that can be applied in teaching any course and labeled as general pedagogical BTP. Fig .3 shows a specialized BTP that could be applied in the course: Introduction to CS. Fig .4 presents a general pedagogical BTP as an example.

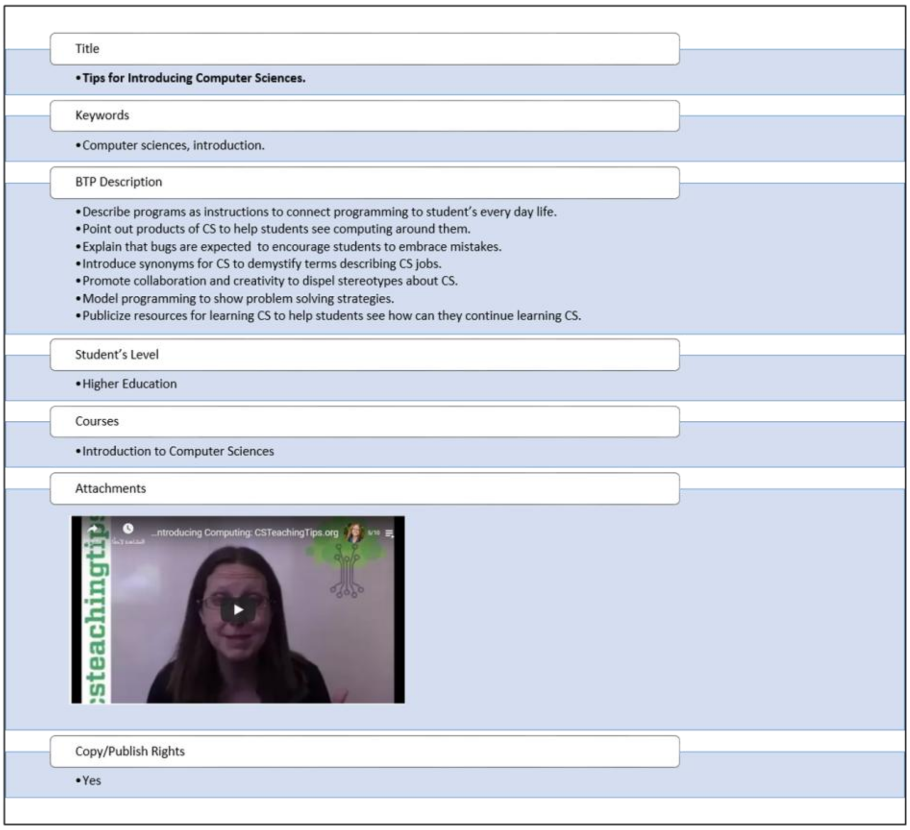

Figure 3. Example of a specialized BTP

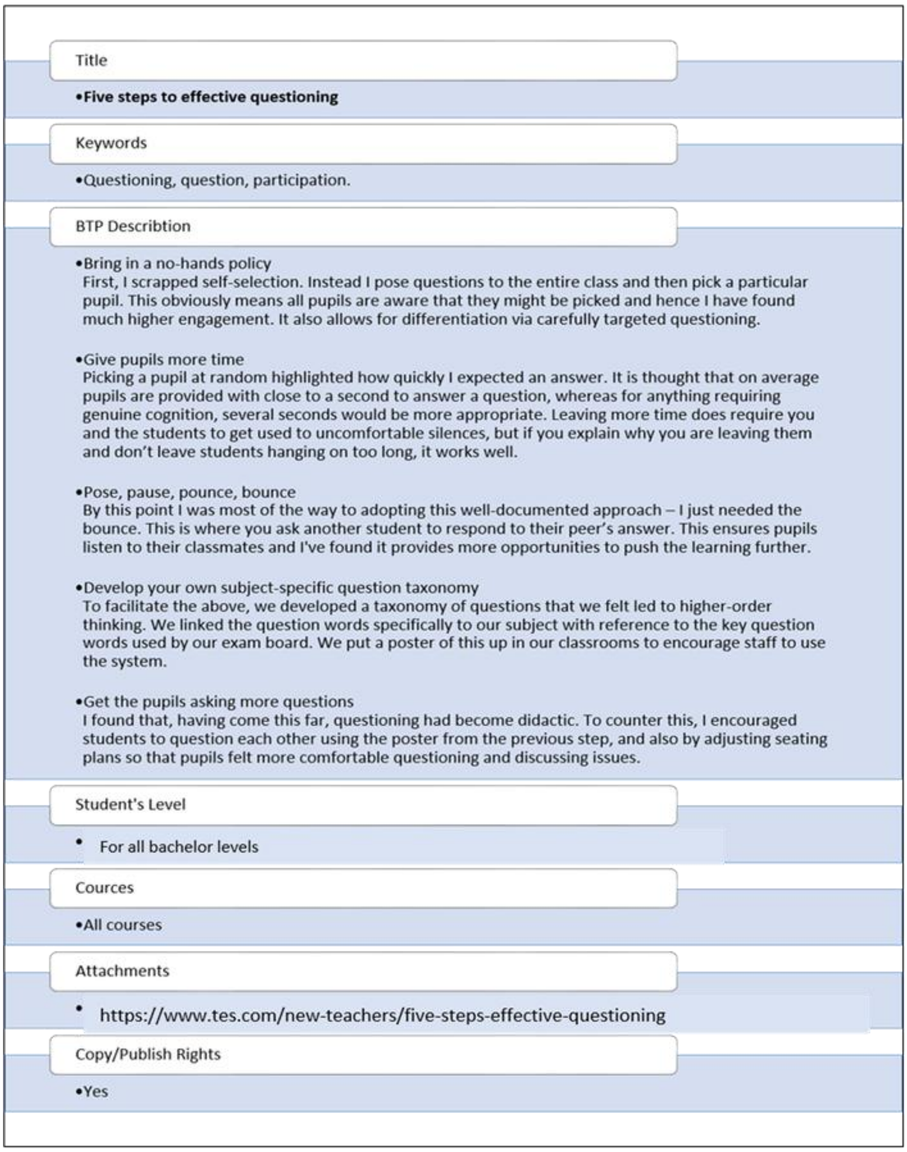

Figure 4. Example of general BTP

6.2. Evaluation

To evaluate the recommendation process, we used the top 10 recommended BTPs for three users. The first user has two interests, the second user have three interests and the third user have four interests. The evaluation is performed on the recommendation list before the users make any interaction and after making interaction by liking or disliking a BTP.

The evaluation of the recommendation process is performed using two evaluation metrics: precision and recall. Precision and recall metrics are considered as the most commonly used metrics to measure the accuracy of recommendation systems [21].

• Precision: Precision represents the proportion of the recommended relevant BTPs in the top N recommended BTPs list and is calculated using Eq. (3). To calculate precision value, we classify all the recommended BTPs according to its relevance to the user as in Table 3.

Table 3: Classifying the recommended BTPs

	Recommended	**Not Recommended**
Relative to user interests	True-Positive (TP)	False-Negative (FN)
Not relative to user interests	False-Positive (FP)	True-Negative (TN)

$$Precision = \frac{Number\ of\ TP}{Number\ of\ TP + Number\ of\ FP} \quad (3)$$

To evaluate the precision at different number of recommendations, we had to find precision @k. Where k represents the number of recommendations on which the precision is measured. Fig. 5 shows the precision levels for the top 5, 8 and 10 recommended BTPs respectively for the three users. We observe that all the three users have a precision value of 1 at k=5, which means that all the top five recommendations are relevant to the users. Beyond that point, the curves start falling down as the k value increase. For the first and third users, the precision level reduced gradually to meet at k=8 with a precision score of 0.875. At k=10, both user1 and user3 reach their lowest precision with a value of 0.8 for user1 and 0.7 for user3. The precision level of the second user also falls down as the k value increase, starting from 1 at k=5 and ends by 0.7 at k=10. These results could be concluded to the precision of the BTPs recommender decrease as we go down in the recommendation list, and the number of user's interests does not affect the precision of the recommendation process.

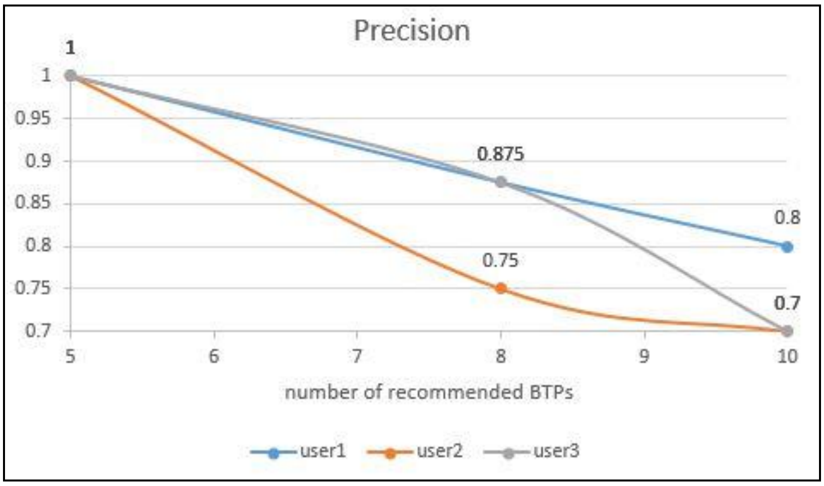

Figure 1. Precision of the recommendation process

• Recall: Recall represents the proportion relevant BTPs that have been recommended in the top-N [22] is calculated using Eq.(4). Fig. 6 shows the recall levels for the top 5, 8 and 10 recommended BTPs respectively for the three users. We can observe that the recall levels increased across the three values of k for the three users. Users 1 and 2 moved gradually from their lowest recall score of 0 at k=5 to their highest recall values at k=10. On the other hand, user 2 had a higher starting at k=5 by a recall value of 0.29 and raised at k=10 to reach 0.37.

$$Recall = \frac{Number\ of\ FP}{Number\ of\ FP + Number\ of\ TN} \tag{4}$$

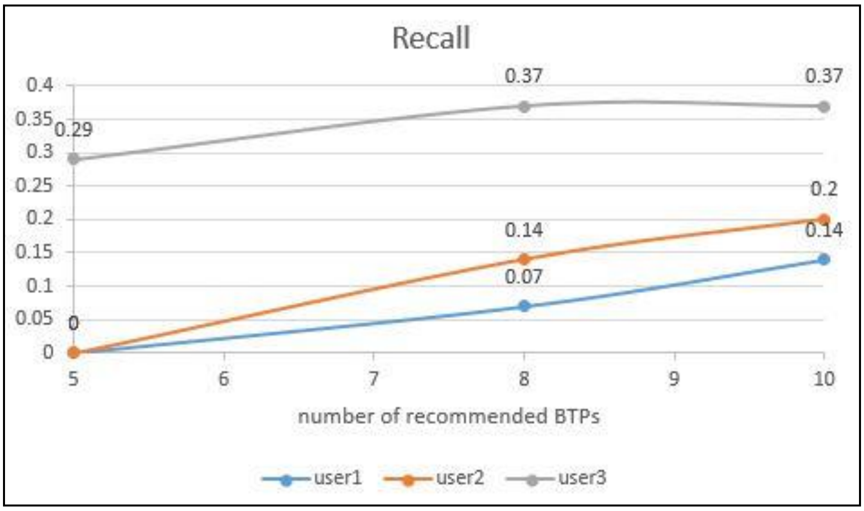

Figure 2. Recall of the recommendation process

VII. CONCLUSION AND FUTURE WORK

To reach good learning outcomes, the teaching process and teachers practices need to get more enhancement and improvements. Nowadays, the technology provides solutions to improve both. One of these solutions, proposed in this research, is to develop a system that organize the instructor's expertise to make it effectively reused and shared among communities of instructors. Best teaching practices (BTPs), which represent the successful teaching experiences that have been used by instructors and proved to have good impact on the teaching results. In this research, we proposed a system to manage the expertise of instructors through sharing and reusing their best teaching practices in teaching computing courses. By creating a repository for gathering, organizing and storing BTPs and providing an interactive personalized recommendation subsystem to propose the most relevant BTPs for instructors based on their interests. If the instructor likes any of the recommended BTPs, he will be recommended with more similar BTPs. And reversely, if he dislikes any of the recommended BTPs, the similar BTPs will move to the end of the recommendation list. The proposed system was evaluated the recommendation process using the most commonly used information retrieval metrics: precision and recall. The results prove that the number of user's interests does not affect the accuracy of the recommendation process. Future work will focus on encouraging instructors to share their BTPs. This includes the design and implementation of a scoring algorithm to increase the scores of instructor and give them extra features.

REFERENCES

[1]. M. M. Keller, K. Neumann, and H. E. Fischer, "The Impact of Physics Teachers' Pedagogical Content Knowledge and Motivation on Students' Achievement and Interest," *J. Res. Sci. Teach.*, vol. 54, no. 5, pp. 586–614, 2017.

[2]. C. E. Wolff, N. Van Den Bogert, and H. Jarodzka, "Keeping an Eye on Learning: Differences between Expert and Novice Teachers' Representations of Classroom Management Events," *J. Teach. Educ.*, no. January 2015.

[3]. S.-C. Kong, M. Lai, and D. Sun, "Teacher development in computational thinking: Design and learning outcomes of programming concepts, practices and pedagogy," *Computers & Education*, vol. 151, p. 103872, 2020.

[4]. H. Ladd and L. Sorensen, "Returns to Teacher Experience: Student Achievement and Motivation in Middle School", *Education Finance and Policy*, vol. 12, no. 2, pp. 241-279, 2017. Available: 10.1162/edfp_a_00194.

[5]. A. G. Wilhelm, I. Chen, T. M. Smith, and K. A. Frank, "Selecting Expertise in Context: Middle School Mathematics Teachers' Selection of New Sources of Instructional Advice," *Am. Educ. Res. J.*, vol. 53, no. 3, pp. 456–491, 2016.

[6]. ACM Computing Classification System, https://dl.acm.org/ccs, accessed on 28/12/2020.

[7]. A. Al-Rasheed and J. Berri, "Effective reuse and sharing of best teaching practices," *Computer Applications in Engineering Education*, vol. 25, no. 2, pp. 163–178, 2017.

[8]. Y. Guo, M. Wang, and X. Li, "An interactive personalized recommendation system using the hybrid algorithm model," *Symmetry (Basel)*. vol. 9, no. 10, 2017

[9]. Y. Zhou, Z. Tang, L. Qi, X. Zhang, W. Dou, and S. Wan, "Intelligent Service Recommendation for Cold-Start Problems in Edge Computing," *IEEE Access*, vol. 7, pp. 46637– 46645, 2019.

[10]. J. Son and S. B. Kim, "Content-based filtering for recommendation systems using multiattribute networks," *Expert Systems with Applications*, vol. 89, pp. 404–412, 2017.

[11]. Pathan, Apashabi, A., 2015. Recommender System Challenges and Methodologies in Social Network: Survey, *International Journal of Science and Research (IJSR)*, 4(11), pp.286-289.

[12]. Areej Bin Suhaim, Jawad Berri, Context-Aware Recommender Systems for Social Networks: Review, Challenges and Opportunities, *IEEE Access*, vol. 9, pp. 57440-57463, 2021.

[13]. Y. Cheng, P. Hsu and Y. Liu, "Identifying and recommending user-interested attributes with values", *Industrial Management & Data Systems*, vol. 118, no. 4, pp. 765-781, 2018. Available: 10.1108/imds-04-2017-0164.

[14]. X. Xie and B. Wang, "Web page recommendation via twofold clustering: considering user behavior and topic relation," *Neural Computing and Applications*, vol. 29, no. 1, pp. 235–243, 2018.

[15]. C. Sundermann, M. Domingues, M. Conrado and S. Rezende, "Privileged contextual information for context-aware recommender systems", *Expert Systems with Applications*, vol. 57, no. 0957-4174, pp. 139-158, 2016. Available: 10.1016/j.eswa.2016.03.036.

[16]. D. Miao and F. Lang, "A Recommendation System Based on Text Mining," 2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), pp. 318–321, 2017.

[17]. M. Y. Hsieh, T. H. Weng, and K. C. Li, "A keyword-aware recommender system using implicit feedback on Hadoop," *Journal of Parallel and Distributed Computing*, vol. 116, pp. 63– 73, 2018.

[18]. A. Klašnja-Milićević, M. Ivanović, and A. Nanopoulos, "Recommender systems in elearning environments: a survey of the state-of-the-art and possible extensions," *Artif. Intell. Rev.*, vol. 44, no. 4, pp. 571–604, 2015.

[19]. Q. Xie, F. Xiong, T. Han, Y. Liu, L. Li and Z. Bao, "Interactive resource recommendation algorithm based on tag information", *Springer Science+Business Media*, vol. 21, no. 6, pp. 1655- 1673, 2018. Available: 10.1007/s11280-018-0532-y

[20]. D. Mezzogori, G. Romagnoli, and F. Zammori, "Deep learning and WLC: how to set realistic delivery dates in high variety manufacturing systems", *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 2092–2097, 2019.

[21]. A. Kla, M. Ivanovi, B. Vesin, and Z. Budimac, "Enhancing e-learning systems with personalized recommendation based on collaborative tagging techniques", *Appl. Intell.*, vol. 48, pp. 1519–1535, 2018.

[22]. T. Silveira, M. Zhang, X. Lin, Y. Liu, and S. Ma, "How good your recommender system is? A survey on evaluations in recommendation," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 5, pp. 813–831, 2017