Shih, JL. et al. (Eds.) (2023). Proceedings of the 31st International Conference on Computers in Education. Asia-Pacific Society for Computers in Education

Design and Development of a Personalized Recommender System of Student Question-Generation Exercises for Programming Courses

Chih-Hung LAI[a], Pham-Duc THO[b*]
[a]*Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan, Republic of China*
[b]*International School, Vietnam National University, Hanoi, Vietnam*
*mrthoedu@gmail.com

Abstract: Computer programming courses often face the challenge of high dropout rates, prompting educators to seek effective solutions to improve student engagement and retention. Student Question-Generation (SQG) have been explored as potential remedy to this issue, enabling advancements in computer science education. However, the majority of existing SQG support systems lack personalized features to enhance student learning. In response to this gap, the study introduces a personal recommender system of SQG exercise tailored for programming courses. The system leverages individual student preferences and SQG exercise complexities to personalize learning experiences within SQG activities. This paper presents the design of the system and its user interface and delves into motivations and technical bases underlying its development.

Keywords: programming course, Student Generated Question, Recommender System, collaborative filtering

1. Introduction

Computer programming is a foundational part of computer science education, and many educational institutions are updating their curricula to address the challenges in programming courses. High student failure and dropout rates in traditional classrooms are attributed to factors like limited class time, resource constraints, and more. Scholars propose remedies, including collaborative learning, problem-solving techniques (Wang et al., 2017), visualization tools (Minjie, 2015), psychological analysis, visual programming environments (Minjie, 2015), and student-generated questions (Andrew et al., 2011; Hsu & Wang, 2018; Lai et al., 2017).

Among these solutions, SQG have emerged as a critical pedagogical strategy with long-standing recognition (Barak & Rafaeli, 2004). SQG has been noted to enhance comprehension of acquired knowledge, and cultivate skills in algorithmic thinking (Hsu & Wang, 2018), motivation, engagement and learning performance (Crogman & Crogman, 2018; Yeong et al., 2019). Through SQG, students are empowered to apply learned concepts to novel challenges and bridge newly acquired knowledge/skills with prior understanding (Lai et al., 2017). This approach encourages students to adopt an alternative cognitive mode and enhance their repertoire of learning strategies (Hsu & Wang, 2018; Yu & Chen, 2014).

Despite the positive outcomes observed, there were some noteworthy concerns surrounding the implementation of this approach. Specifically, it came to light that certain students expressed a lack of motivation when engaging with assignments involving SQG, as proposed by the instructor throughout the course. This lack of motivation was evident in cases where high-achieving students were presented with relatively straightforward SQG exercises, or conversely, where less experienced students were confronted with overly complex ones.

To address this challenge and fully capitalize on the advantages of SQG, this research introduces an innovative system known as PERS (Personalized Exercise Recommender System). PERS builds upon the SQG concept by incorporating a Recommender System (RS) that suggests appropriate SQG exercises or problem statements to individual students. Operating on the principles of collaborative filtering, PERS intelligently recommends SQG exercises based on students' interests and their assessed proficiency level.

2. Related work

2.1 Question generation and programming learning

Question Generation, also referred to as Question Posing, involves the active process wherein students formulate examination questions based on the content they have read. Even in the face of unfamiliar scenarios or exceptional circumstances, individuals, including children, naturally pose questions. Constructivist educational theories emphasize the significance of this innate curiosity as a driving force for learning and advocate for instructors to facilitate the emergence of such spontaneous inquiry. This facilitative role of the instructor is seen as pivotal in nurturing the inherent questioning process.

Question types span diverse formats, encompassing coding exercises, multiple-choice, matching, short-answer, true-false, fill-in-the-blank, and word puzzles (Barak & Rafaeli, 2004; Lai et al., 2017). When generating questions, students are required to engage deeply with the pertinent information in their texts and subsequently formulate correct answers and distractors (Lai et al., 2017). The ability of students to elucidate the reasoning behind their crafted answer options serves as an indicator of their grasp of the reading material (Lan & Lin, 2011;). Furthermore, question generation allows instructors to identify students' comprehension challenges, thereby enabling targeted instructional interventions (Lan & Lin, 2011; Yeh & Lai, 2012).

Numerous web-based learning systems have been developed to foster student question generation, including QPPA (Yu et al., 2002), Question-Authoring and Reasoning Knowledge System (QuARKS) (Yu, 2009), CodeWrite (Denny et al., 2011), StudySieve (Andrew et al., 2011), and PIPLS (Lai & Tho, 2016). While many of these systems are domain-agnostic, accommodating various question types and multimedia content, and supporting anonymous interactions, only a limited few are specifically tailored for programming courses (Denny et al., 2011; Lai & Tho, 2016; Reilly, 2012) and none of them incorporate automated methods to recommend SQG exercises to students.

2.2 Recommender systems and learning environments

A RS is a software tool designed to discern and propose content that a specific user would find valuable (Ricci et al., 2011). It falls within the realm of information filtering systems, which exploit user data to predict ratings or preferences a user might assign to particular items. Consequently, the primary advantage of an RS is its ability to identify the most suitable array of items for a given user, optimizing rating predictions.

Ricci et al. (2011) delineated five types of RSs: content-based, knowledge-based, demographic, community-based, collaborative, and hybrid. Collaborative Filtering RSs (CFRSs) have seen extensive utilization (Elahi et al., 2016). CFRSs operate on the premise that users may favor items endorsed by other users with similar preferences in the past. This principle underpins the RS employed in this study. CFRSs consist of two main approaches: user-user and item-item (Elahi et al., 2016), both commonly employing the Nearest Neighbors algorithm (Nikolakopoulos et al., 2021).

RSs have found broad application in e-learning environments within the scope of Technology-Enhanced Learning (TEL) to enhance students' self-directed learning (Manouselis et al., 2011). In e-commerce, RSs suggest products; in e-learning, they recommend educational resources (such as papers, books, or courses) to participants like students and teachers (Liu et al., 2022).

RSs have been analyzed for their application in e-learning systems (Santos & Boticario, 2011), presenting three technological prerequisites for developing semantic education RSs.

Enhancing RSs in collaborative learning environments, Anaya et al. (2013) introduced an influence diagram employing machine learning to assess user collaboration, leading to the development of an automatic RS with a pedagogical decision tree. A cloud-based architecture for recommending learning elements based on the learner's emotional state was also proposed (Leony Arreaga et al., 2013).

An extensive survey on RS evaluation in the context of TEL highlighted the need for better evaluation strategies (Erdt et al., 2015). In conclusion, the review underscores the popularity of RS in education, especially technology-driven approaches for enhancing learning. Furthermore, it notes the absence of RSs supporting programming learning with SQG, which forms the rationale for the proposed system.

3. PERS: An integrated approach

In the traditional SQG approach, instructors utilize classroom lectures and student-created SQGs, which are added to a shared Question Bank. Challenges involve overseeing student interaction and addressing dissatisfaction, especially among advanced students who may find exercises too simple. Another approach is more lab practice with instructor oversight, offering better control over SQG exercise interaction. Yet, it risks overemphasizing formal learning, conflicting with broader goals. The challenge is enhancing the current method while reducing direct instructor involvement, maintaining its advantages.

In response to these considerations, this study introduces PERS, a web application that integrates SQG as a teaching tool along with a RS for peer exercises. It is important to note that this proposal not only facilitates interactions similar to the existing approach but also personalizes students' learning journeys through SQG exercises. This empowers the instructor to guide students' learning paths by creating exercises integrated into the system. Moreover, students are afforded the opportunity to evaluate and indicate their preferences for SQG exercises, enabling the system to gain insights into their individual preferences. Using this feedback, the system generates suggestions for new SQG exercises tailored to each student, capitalizing on the notion that students with shared preferences and perceptions of exercise complexity are prime sources for recommendations.

From an educational perspective, this proposition emerges as a novel intermediary between instructors and students. In the ensuing section, the proposal is presented in meticulous detail, outlining the developed Web Application's principal modules and features. Subsequently, the technical elements of the implemented RS are elucidated.

The primary interaction workflow between students and the proposed system is illustrated in Figure 1. Upon logging in, the system assesses whether the student has previously encountered SQG exercises. If they have, the system employs a collaborative filtering approach to recommend 10 SQG exercises. If not, the system selects the most approachable and engaging SQG exercises from the available pool. Additionally, the system assembles a list of 10 SQG exercises intended for presentation to students.

The subsequent stages can be readily comprehended by referring to the diagram. However, it's noteworthy that following the submission of evaluations for the SQG exercises, the system employs collaborative filtering to furnish the student with a fresh set of recommended SQG exercises. As a result, students may engage with varied SQG exercises based on their individual interactions.

Consequently, the student profile is developed based on their evaluations (Favorites and Complexity) of the completed SQG exercises. The current version of PERS was crafted using Python and employs the PostgreSQL Server as its underlying database system.

Among its noteworthy features, PERS encompasses:

- User Authentication Area: This area permits the entry of previously registered students and teachers.
- Student Interface: Students benefit from an informative panel (as depicted in the right panel of Figure 2), which empowers them to manage their profiles,

access a summary of their completed SQG exercises, and review the evaluations of those exercises.

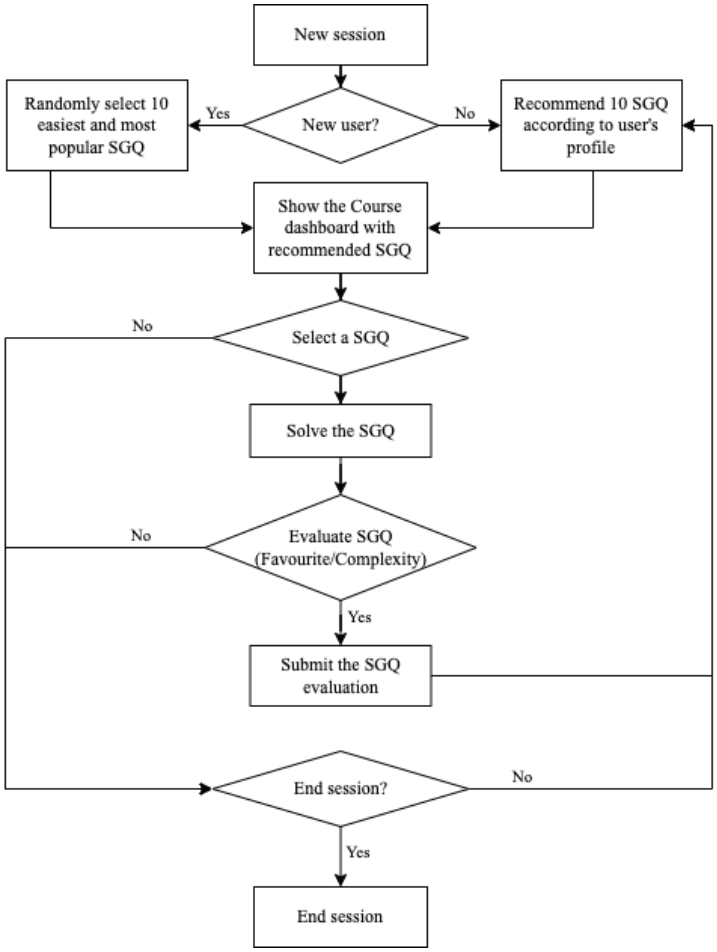

Figure 1. The main workflow of the interaction between the student and the proposed system

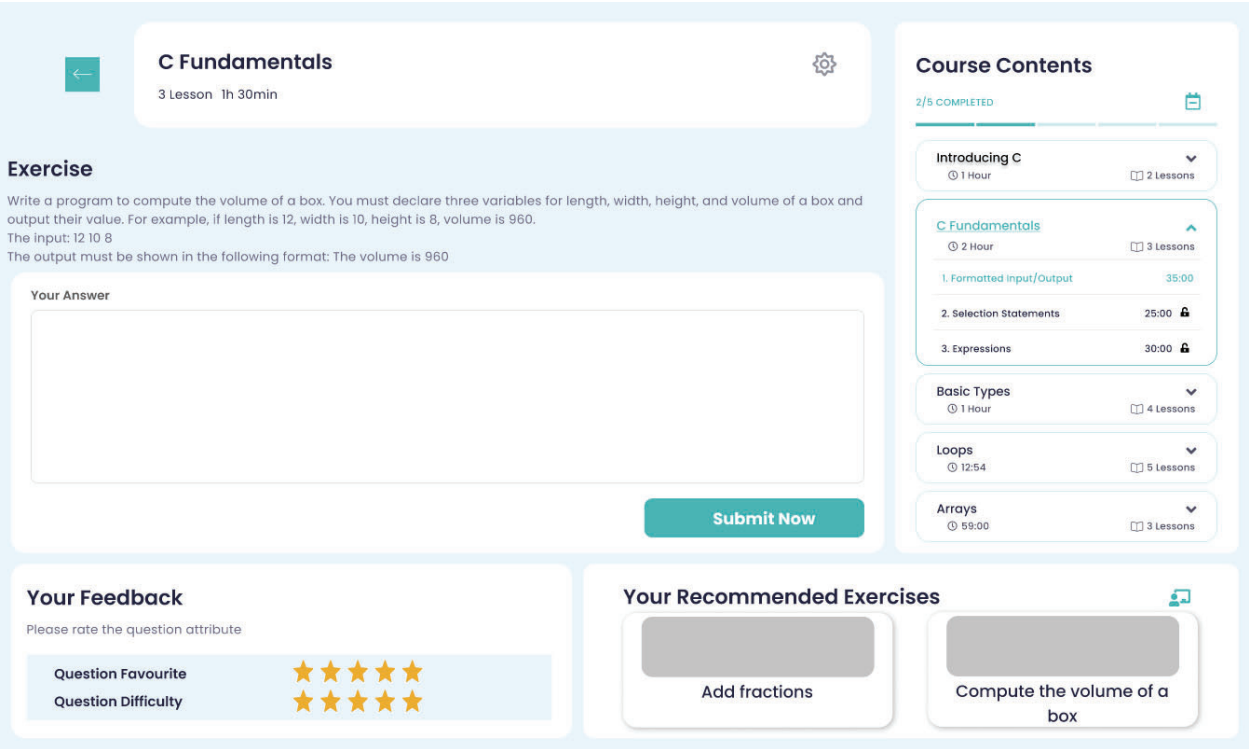

Figure 2: PERS front-end

- Central Display Area: The central region (as shown in Figure 2) is tailored to exhibit the description of the chosen SQG exercise along with its evaluation criteria.
- Bottom-Right Area: The section located at the bottom-right (depicted in Figure 2) provides either a roster of recommended SQG exercises or an exhaustive list of all SQG exercises.
- Teacher Profile: In the teacher's profile, educators can seamlessly integrate new SQG exercises and gain access to statistical insights regarding both the SQG exercises and student interactions.

Importantly, the inclusion of the RS within PERS was realized through the utilization of Surprise. This Python Scikit, abbreviated as SciPy Toolkits, facilitates the creation and evaluation of recommender systems, underscoring PERS's technological foundation.

4. Conclusion and future work

In this study, PERS, a user-friendly Web App is introduced seamlessly combining SQG with a robust recommender system to support students in learning Programming more effectively compared to other systems. In the forthcoming stages, this study will improve the functionalities of the existing systems and assessing the impact of the system on students' academic performance. The insights gleaned from this study will stimulate heightened interest among researchers in this domain.

Acknowledgements

This research was funded by National Science and Technology Council, Taiwan, R.O.C. (Contract NO: MOST 111-2410-H-259-009-)

References

Anaya, A. R., Luque, M., & García-Saiz, T. (2013). Recommender system in collaborative learning environment using an influence diagram. *Expert Systems with Applications, 40*(18), 7193-7202.

Andrew, L. R., Denny, P., Plimmer, B., & Bertinshaw, D. (2011). Supporting student-generated free-response questions. 16th annual joint conference on Innovation and technology in computer science education,

Barak, M., & Rafaeli, S. (2004, 7//). On-line question-posing and peer-assessment as means for web-based knowledge sharing in learning. *International Journal of Human-Computer Studies, 61*(1), 84-103. https://doi.org/10.1016/j.ijhcs.2003.12.005

Berry, J. W., & Chew, S. L. (2008). Improving Learning Through Interventions of Student-Generated Questions and Concept Maps. *Teaching of Psychology, 35*(4), 305-312, Article Pii 905075619. https://doi.org/10.1080/00986280802373841

Chang, S. B., Huang, H. M., Tung, K. J., & Chan, T. W. (2005). AGQ: a model of student question generation supported by one-on-one educational computing. 2005 conference on Computer support for collaborative learning: learning 2005: the next 10 years!,

Crogman, H., & Crogman, M. T. (2018, May). Modified generated question learning, and its classroom implementation and assessment [Article]. *Cogent Education, 5*(1), 1-41, Article 1459340.

Denny, P., Luxton Reilly, A., Tempero, E., & Hendrickx, J. (2011). CodeWrite: supporting student-driven practice of java. 42nd ACM technical symposium on Computer science education,

Elahi, M., Ricci, F., & Rubens, N. (2016, 2016/05/01/). A survey of active learning in collaborative filtering recommender systems. *Computer Science Review, 20*, 29-50. https://doi.org/https://doi.org/10.1016/j.cosrev.2016.05.002

Erdt, M., Fernandez, A., & Rensing, C. (2015). Evaluating recommender systems for technology enhanced learning: a quantitative survey. *Ieee Transactions on Learning Technologies, 8*(4), 326-344.

Hsu, C. C., & Wang, T. I. (2018, 2018/06/01/). Applying game mechanics and student-generated questions to an online puzzle-based game learning system to promote algorithmic thinking skills. *Computers & Education, 121*, 73-88. https://doi.org/10.1016/j.compedu.2018.02.002

Lai, C.-H., Tho, P.-D., & Liang, J.-S. (2017). Design and evaluation of question-generated programming learning system. 2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI),

Lai, C. H., & Tho, P.-D. (2016). Development of a Programming Learning System Based on a Question Generated strategy. 24th International Conference on Computers in Education, India.
Lan, Y. F., & Lin, P. C. (2011). Evaluation and improvement of student's question-posing ability in a web-based learning environment [Article]. *Australasian Journal of Educational Technology, 27*(4), 581-599.
Leony Arreaga, D. A., Parada Gélvez, H. A., Muñoz Merino, P. J., Pardo Sánchez, A., & Delgado Kloos, C. (2013). A generic architecture for emotion-based recommender systems in cloud learning environments.
Leovy, E., Ruth, C., Liliana, M., & Ivan, C. (2017). Using collaborative learning scenarios to teach programming to non-CS majors. *Computer Applications in Engineering Education, 25*(5), 719-731. https://doi.org/10.1002/cae.21832
Linxiao, M. (2007). *Investigating and Improving Novice Programmers' Mental Models of Programming Concepts* [Dissertation, University of Strathclyde]. United Kingdom.
Liu, T., Wu, Q., Chang, L., & Gu, T. (2022, 2022/12/01). A review of deep learning-based recommender system in e-learning environments. *Artificial Intelligence Review, 55*(8), 5953-5980. https://doi.org/10.1007/s10462-022-10135-2
Manouselis, N., Drachsler, H., Vuorikari, R., Hummel, H., & Koper, R. (2011). Recommender systems in technology enhanced learning. *Recommender systems handbook*, 387-415.
Minjie, H. (2015). *Teaching Novices Programming: A Programming Process Using Goals & Plans with a Visual Programming Environment* University of Otago]. Dunedin, New Zealand.
Nikolakopoulos, A. N., Ning, X., Desrosiers, C., & Karypis, G. (2021). Trust your neighbors: a comprehensive survey of neighborhood-based methods for recommender systems. *Recommender systems handbook*, 39-89.
Piteira, M., Costa, C. J., & Aparicio, M. (2018). Computer programming learning: how to apply gamification on online courses? *Journal of Information Systems Engineering & Management, 3*(2), 11-20.
Reilly, A. L. (2012). *The Design and Evaluation of StudySieve: A Tool That Supports Student-Generated Free-Response Questions, Answers and Evaluations.* [Dissertation, The University of Auckland]. NewZealand.
Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to Recommender Systems Handbook. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender systems handbook* (pp. 1-35). Springer US. https://doi.org/10.1007/978-0-387-85820-3_1
Rosenshine, B., Meister, C., & Chapman, S. (1996, Sum). Teaching students to generate questions: A review of the intervention studies [Review]. *Review of Educational Research, 66*(2), 181-221.
Santos, O. C., & Boticario, J. G. (2011). Requirements for semantic educational recommender systems in formal e-learning scenarios. *Algorithms, 4*(3), 131-154.
Vladimiras, D., Tatjana, J., & Valentina, D. (2018). From Android games to coding in C—An approach to motivate novice engineering students to learn programming: A case study. *Computer Applications in Engineering Education, 26*(1), 75-90. https://doi.org/10.1002/cae.21862
Wan, X., Jamaliding, Q., & Okamoto, T. (2011). Analyzing Learners' Relationship to Improve the Quality of Recommender System for Group Learning Support. *J. Comput., 6*(2), 254-262.
Wan, X., & Okamoto, T. (2011). Utilizing learning process to improve recommender system for group learning support. *Neural Computing and Applications, 20*, 611-621.
Wilson, E. V. (2004). ExamNet asynchronous learning network: augmenting face-to-face courses with student-developed exam questions. *Computers & Education, 42*(1), 87-107.
Yeh, H. C., & Lai, P. Y. (2012). Implementing online question generation to foster reading comprehension [Article]. *Australasian Journal of Educational Technology, 28*(7), 1152-1175. <Go to ISI>://WOS:000314400200006
Yeong, F. M., Chin, C. F., & Tan, A. L. (2019). Use of a competency framework to explore the benefits of student-generated multiple-choice questions (MCQs) on student engagement. *Pedagogies: An International Journal*, 1-23. https://doi.org/10.1080/1554480X.2019.1684924
Yu, F. Y., & Chen, Y. J. (2014, Mar). Effects of student-generated questions as the source of online drill-and-practice activities on learning. *British Journal of Educational Technology, 45*(2), 316-329.
Yu, F. Y., Wu, W. S., & Huang, H. C. (2018, Oct). Promoting Middle School Students' Learning Motivation and Academic Emotions via Student-Created Feedback for Online Student-Created Multiple-Choice Questions [Article]. *Asia-Pacific Education Researcher, 27*(5), 395-408. https://doi.org/10.1007/s40299-018-0398-x