information

Article

Path-Based Recommender System for Learning Activities Using Knowledge Graphs

Christos Troussas * and **Akrivi Krouska**

Department of Informatics and Computer Engineering, University of West Attica, 12243 Egaleo, Greece
* Correspondence: ctrouss@uniwa.gr

Abstract: Recommender systems can offer a fertile ground in e-learning software, since they can assist users by presenting them with learning material in which they can be more interested, based on their preferences. To this end, in this paper, we present a new method for a knowledge-graph-based, path-based recommender system for learning activities. The suggested approach makes better learning activity recommendations by using connections between people and/or products. By pre-defining meta-paths or automatically mining connective patterns, our method uses the student-learning activity graph to find path-level commonalities for learning activities. The path-based approach can provide an explanation for the result as well. Our methodology is used in an intelligent tutoring system with Java programming as the domain being taught. The system keeps track of user behavior and can recommend learning activities to students using a knowledge-graph-based recommender system. Numerous metadata, such as kind, complexity, and number of questions, are used to describe each activity. The system has been evaluated with promising results that highlight the effectiveness of the path-based recommendations for learning activities, while preserving the pedagogical affordance.

Keywords: knowledge graphs; knowledge-graph-based recommender system; path-based reasoning; recommender system; intelligent tutoring system; learning activity recommendations

check for updates

Citation: Troussas, C.; Krouska, A. Path-Based Recommender System for Learning Activities Using Knowledge Graphs. *Information* **2023**, *14*, 9. https://doi.org/10.3390/info14010009

Academic Editors: Pierpaolo Basile and Annalina Caputo

Received: 22 November 2022
Revised: 20 December 2022
Accepted: 21 December 2022
Published: 23 December 2022

1. Introduction

The amount of available data has multiplied tremendously with the quick development of the internet. As such, users may face an unfriendly environment when they use a system, since they struggle to select what may interest them from a plethora of options due to the overabundance of information. Toward assisting users in the process of proceeding to the most adequate selection, recommender systems have been used to provide suggestions for items that are most relevant to a certain user [1]. In the literature, recommender systems have been used in different fields of application, e.g., e-shops, online movie recommendations, and songs recommendations.

Recommender systems have been used to support e-learning software [2]. In e-learning recommender systems, the recommendation mostly depends on the item content and user behaviors observed when the user is interacting with the online tutoring system. For instance, recommender systems in e-learning software can provide recommendations concerning students' learning styles, learning resources, learning activities, courses, or even learning pathways [3–5]. Learning activities' content and types may be ambiguous and uncertain information for students. In many e-learning systems, students are provided with specific learning activities to conduct. However, recommender systems can help in this direction by proposing specific activities to students in a personalized way, while preserving the educational quality of the learning process.

For the development of recommender systems, the approaches that have been mainly used by researchers are collaborative filtering, content-based filtering, or hybrid approaches [6]. The premise behind collaborative filtering is that people who have previously agreed will do so again and that they will continue to enjoy the same kinds of things. Recommendations are generated by the algorithm solely based on data from rating profiles for various persons or objects. It produces recommendations by using this location to find peer users or objects with rating histories similar to the one of the current user or item. However, a description of the item and a profile of the user's preferences serve as the foundation for content-based filtering techniques. These techniques work best when information about the item (name, location, description, etc.), but not the user, is known. Content-based recommenders approach recommendations as a user-specific classification issue and learn a classifier for a user's preferences based on the characteristics of an item. In this approach, the things are described using keywords, and a user profile is created to show the kinds of items this user prefers. In other words, these algorithms strive to suggest products that are comparable to those that a consumer has previously enjoyed or is now looking at. Finally, the combination of the aforementioned techniques can form hybrid recommender systems.

Recently, researchers have become interested in adding a knowledge graph (KG) as side information to recommender systems [7]. A KG is a heterogeneous network in which the nodes serve as entities and the relations signify the connections among such entities. To comprehend the relationships between items, it is possible to map items and their properties into the KG. Furthermore, the incorporation of users and user-side data into the KG allows for a more precise recording of user relationships with items as well as user preferences. The explainability of recommendation results is another advantage of KG-based recommender systems.

Knowledge-graph-based recommender systems are classified according to how they use the KG data, as follows: embedding-based methods, path-based methods, and unified methods [6]. Embedding-based methods directly augment the representation of items or users with data from the KG. Knowledge graph embedding (KGE) techniques must be used to encode the KG into low-rank embeddings before the KG information can be used. Path-based techniques create a user–item graph and use the entity's connectivity patterns in the network to provide recommendations. Path-based approaches have been referred to in research studies as recommendations in the heterogeneous information network. The connection similarity of users and/or objects is typically exploited by these models to improve recommendations. Unified techniques that incorporate both the semantic representation of entities and relations and the connectivity information have been proposed in order to fully use the data in the KG for improved suggestions. Embedding propagation theory serves as the foundation for the unified method. These techniques improve entity representation under the direction of the KG's connective structure.

In this paper, we present a novel approach for a path-based recommender system for learning activities using knowledge graphs. The presented model uses the connection similarity of users and/or items in order to improve the recommendation of learning activities. Our approach uses the user (student)–item (learning activity) graph to uncover path-level similarities for items (learning activities), either by pre-defining meta-paths or by automatically mining connective patterns. The path-based method can also deliver an explanation for the outcome. Our model is incorporated in an intelligent tutoring system, and the domain to be taught is the programming language Java. The system monitors the actions of users and through the knowledge-graph-based recommender system can suggest learning activities to students. Each activity is characterized by several metadata, such as type, difficulty, number of questions, and revised Bloom's taxonomy (RBT) level. The novel recommender system can preserve the pedagogical affordance through the sophisticated way of suggesting learning activities. In view of this, the contribution of this paper lies in the presentation of a novel path-based recommender system using knowledge graphs in the field of education and specifically for suggesting adequate learning activities. Furthermore, it includes the use of specific learning activity metadata as well as the monitoring of learners' actions.

The main research questions that this paper answers are (1) how effective are the recommendations of learning activities to users and (2) does the recommender system have a positive impact on learning?

The rest of this paper is organized as follows. In Section 2, a review of the related research works is presented. In Section 3, a description of the path-based recommender system is provided, as its incorporation in the e-learning software is shown and an example of operation is provided. In Section 4, the evaluation results are presented and a discussion on them is provided. Finally, in Section 5, conclusions are drawn and future research steps are described.

2. Related Literature

In this section, the related literature on recommender systems in general and knowledge-graph-based recommender systems is analyzed.

2.1. Recommender Systems

Recommender systems have been investigated for various fields in the related scientific literature. Their application in e-learning environments [2,8–10], entertainment websites [11–14], social settings [15–18], and tourist systems [19–22] are a few examples. Collaborative filtering, content-based filtering, machine learning, and hybrid approaches are the algorithmic strategies that have been used most frequently in the research papers mentioned before. According to a study of 2019 [23], a single-criteria rating serves as the main source for the recommendation process for the majority of recommender systems (overall rating). Another study of 2020 [24] reports that collaborative filtering, a technique that has been widely used in the literature to construct recommender systems, has a variety of drawbacks, including the cold-start problem, which means that the system cannot make inferences about people or things for which insufficient data have been collected.

As mentioned earlier, the research field of recommender systems in e-learning environments is active. In such systems, there are two crucial elements: (1) the learners to whom the system offers recommendations have specific characteristics that may be difficult to be defined, such as knowledge level, and (2) the suggestions that are to be delivered to the learners may have a strong impact on knowledge acquisition, such as learning activities, learning material, and assessment units.

In light of these findings, it is evident that additional research is needed in the area of recommender system development.

2.2. Knowledge-Graph-Based Recommender Systems

Knowledge graphs have been introduced in recommender systems as auxiliary data that can track changes in user interests and add insight to recommendations. Indeed, there has been an active research area that explores knowledge-graph-based recommender systems. They have been used in various domains, such as travel websites [25–27], online museums [28–30], biomedical systems [31–33], e-commerce services [34–36], and movie websites [37–39]. In the aforementioned studies, and as clearly stated in two recent review papers [6,7], the methods of recommender systems with knowledge graphs that have been used in the literature are embedding-based methods, connection-based methods, and propagation-based methods.

More specifically, concerning path-based recommender systems, in [40], the authors developed the SemRec, which considers the interaction of the user's favorite and disliked prior items. To integrate attribute values in the link, this architecture uses a weighted meta-path. More accurate item relations and user similarities can be depicted using these channels to disseminate the actual user preference by modeling both positive and negative preference patterns. In [41], the authors introduced MCRec, which generates explicit meta-path representations to reflect the interaction context of user–item pairs. In [42], the authors introduced a recurrent knowledge graph embedding technique that automatically mines the route link between user and item without requiring the user to provide meta-paths. In [43]. The authors designed a technique for sequential recommendation that leverages users' fluctuating interests. Finally, in [44], the authors' strategy seeks to offer an ordered path of important academic papers, which are extremely useful in assisting researchers in understanding the evolution of a certain issue. They use both content and network structure to learn the representation of a document during the process. Following that, the representation is used to assess the similarity of papers.

Analyzing the related literature of the area of knowledge-graph-based recommender systems in the field of e-learning (needless to say, for the recommendation of learning activities to students), it needs to be emphasized that there is significant room for improvement in this direction.

In view of this, the motivation for this research emerged from the fact that the aforementioned technological advancements create a fertile ground for the development of knowledge-graph-based systems that embrace a high degree of sophistication in their recommendation mechanism tailored to the field of e-learning. The research effort in this field is still in its infancy, and many aspects remain unexplored.

3. Description of the Knowledge-Graph-Based Recommender System

3.1. Path-Based Method for Recommendations

In our approach, the incorporated path-based method creates a user–item graph and uses the entity's connectivity patterns in the network to provide recommendations. The connection similarity of users and/or objects is exploited to improve the recommendation of learning activities. In the generated user–item graph, the nodes concern users or items, while edges concern the interactions between users and items.

3.1.1. Representation of the Network

A graph G = (E, R) is used to represent our network, where E = {e_1, e_2,..., e_n} denotes n entities and R is the set of relations in G. Every node e and every link r are connected with their respective mapping functions, y(e): $E \rightarrow K_E$ and z(e): $R \rightarrow K_R$. The sets of pre-defined objects and relation types are denoted by K_E and K_R. To learn the latent representation of learning activities, a mapping function Γ: $E \rightarrow \mathbb{R}^h$ ($h \leq |E|$) is used.

3.1.2. Representation of the Learning Activity

A path in the form of $e_{i,1} \rightarrow e_{i,2} \rightarrow \ldots \rightarrow e_{i,k}$ that represents a chain from new learning activities to previous learning activities is referred to as a learning activity path. There is a relationship between the path's nearby nodes $e_{i,k-1}$ and $e_{i,k}$. It should be noted that learning activities are characterized by metadata, as mentioned before.

3.1.3. Representation of the Path Recommendation Probability

The probability of recommending a learning activity path is defined as the similarity between the representation of the provided learning activity and that of the suggested path, given a path P = {$e_{i,1}$ $e_{i,2}$... $e_{i,k}$}, which signifies a path of recommended learning activities.

3.1.4. Representation of the Multi-Relational Graph

In its most basic form, the network is a multi-relational heterogeneous graph, whose relations denote numerous connections and contain various degrees of semantic relatedness. In our model, we consider the learning activity–learning activity network and user–learning activity–RBT level–difficulty network.

Learning activity–RBT level: The graph's homogeneous network is created by the relationship between learning activities. We use a random walk-based sampling technique to learn the latent representation of the nodes in order to assess the latent relationships between them. We use the two parameters p and q to bias our random walks toward the local area or to tend to move further away when we generate them. We define a window size of k after producing a random walk, and Ns(u) stands for the neighborhood for node u in the sliding window. Assume that γ_1: $E \rightarrow \mathbb{R}^h$ is the function that maps nodes to feature representations. The similarity of two nodes e_i and e_j in learning activity–learning activity space are:

$$\mathrm{f_c}(e_i, e_j) = \cos(\gamma_1(e_i),\ \gamma_1(e_j)) = \frac{\gamma_1(e_i)\ \gamma_1(e_j)}{||\gamma_1(e_i)||\ ||\ \gamma_1(e_j)||}$$

Student–Learning Activity–RBT level–Difficulty–Number of Units: Without a defined relationship, the learning activity–learning activity relationship cannot provide relatedness information between learning activities. We also consider additional ties to obtain a better relatedness rating. We use the "student–learning activity–student" and "student–learning activity–RBT level–learning activity–student" meta-path schemes, which are the two most popular and useful meta-path schemes. We learn efficient node representations by including various types of nodes into skip-grams using a meta-path-based random walk technique. We additionally sue negative sampling for network learning to achieve effective optimization. The similarity of the two nodes e_i and e_j in the "student–learning activity–RBT level–difficulty–number of units" space can be defined as follows after learning the network representation:

$$\mathrm{f_a}(e_i, e_j) = \cos(\gamma_2(e_i),\ \gamma_2(e_j)) = \frac{\gamma_2(e_i)\ \gamma_2(e_j)}{||\gamma_2(e_i)||\ ||\ \gamma_2(e_j)||}$$

3.1.5. Explanation of Notations

G: The knowledge graph G consists of a set of nodes E and a set of relationships R.
E: The set of nodes E includes a collection of entities, such as students.
R: The set of relationships R includes a collection of edges that connect nodes in the graph and represent various types of relationships between entities. For example, R can include edges representing "complete," "level," etc.
e: A specific edge in the graph is referred to as e.
K_E: K_E is a set of attributes that describe each entity in the graph. For example, K_E can include attributes such as "complexity" for learning activity.
K_R: K_R is a set of attributes that describe each relationship in the graph.
γ: γ is a function that maps each entity node to a feature vector representation.
h: h is the dimensionality of the feature vector representations produced by γ.
An example of the described knowledge graph can be as follows:
Matching activity:
Entity: "Matching activity"
Properties:
Name: "Matching Java concepts with explanation"
Type: "Matching activity"
Difficulty level: "Easy"
RBT-level: "RBT-L1"
Score: "90%"
Number of units: "5"
Memory activity:
Entity: "Memory activity"
Properties:
Name: "Remember the operators in Java"
Type: "Memory activity"
Difficulty level: "Moderate"
RBT-level: "RBT-L1"
Score: "75%"
Number of units: 6
Multiple-choice activity:
Entity: "Multiple-choice activity"
Properties:
Name: "Multiple-choice Java quiz"

Type: "Multiple-choice activity"
Difficulty level: "Challenging"
RBT-level: "RBT-L1"
Score: "65%"
Number of units: "8"

The knowledge graph could then include relationships between the different activities and the student, such as the student completing the activities and receiving a certain score on them. This information could be used to track the student's progress and to identify areas where they may need additional support. Here is an example of a knowledge graph that could be used to represent a student's activities:

G: The set of all entities in the knowledge graph, including students and activities.
G = {student, matching activity, memory activity, multiple-choice activity}
E: The set of all edges in the knowledge graph, representing relationships between entities.
E = {(student, completes, matching activity), (student, completes, memory activity), (student, completes, multiple-choice activity), (matching activity, level, RBT-L1), (memory activity, level, RBT-L1), (multiple-choice activity, difficulty, moderate)}
R: The set of all entity types in the knowledge graph, such as "student" and "learning activity."
R = {student, activity}
e: A function that maps an edge to its corresponding entities.
e(student, completes, matching activity) = (student, matching activity)
K_E: The set of all edge types in the knowledge graph, e.g., "completes."
K_R: The set of all possible attribute–value pairs for each entity type.
K_R(student) = {name, age, grade level}
KR(activity) = {name, type, difficulty level, RBT-level, score, number of units}
γ: A function that maps an entity to its corresponding entity type.
γ(student) = student
γ(matching activity) = activity
h: A function that maps an entity to its corresponding attribute–value pairs.
h(student) = {name: "Akrivi Krouska", age: "*20*", grade level: "undergraduate"}
h(matching activity) = {name: "Matching Java concepts with explanation", type: "Matching activity", difficulty level: "Easy", RBT-level: "RBT-L1", score: "90%", Number of units: "5"}

In our example, the function $f_c(e_i,e_j)$ is used to measure the similarity of two students based on the learning activities they completed. The function considers these data and calculates a similarity score based on the degree of overlap between the two sets. The higher the overlap, the higher the similarity score.

In general, the function $f_c(e_i,e_j)$ is designed to measure any type of similarity between two nodes in the presented knowledge graph.

3.2. Overview of the Recommended Items of the E-Learning Software

The domain knowledge of the e-learning software comprises concepts of the programming language Java in an undergraduate level of the computer engineering curriculum. The domain knowledge consists of 13 chapters, ranging in complexity from basic ideas to more complex ones.

The domain knowledge includes various learning activities of different types based on the revised Bloom's taxonomy (RBT) [45]. This taxonomy offers a framework for categorizing learning outcomes into six categories, from the most fundamental to the most complicated, in accordance with students' cognitive abilities. The RBT levels are as follows: remembering (RBT-L1), understanding (RBT-L2), applying (RBT-L3), analyzing (RBT-L4), evaluating (RBT-L5), and creating (RBT-L6).

In addition to this, the learning activities can range in difficulty, meaning that each activity can be either easier or more difficult. The levels of difficulty are three: easy, moderate, and challenging. Information about the learning activities is presented in Table 1.

Table 1. RBT levels.

RBT Level	Types of Learning Activities	Difficulty Level
RBT-L1	True/false activity, book marking, flash cards, reading material, memory activities, watching presentations and videos, matching activity	Easy, moderate, challenging (applied to all RBT levels)
RBT-L2	Create an analogy, group discussions, taking notes, storytelling, diagrams, flowcharts	
RBT-L3	Concept maps, problem-solving examples, learning through short answers, demonstrations, group work, practice and calculate	
RBT-L4	Fishbowls, debating, run a test, case studies, compare and contrast (with charts, tables), group investigation, questionnaires	
RBT-L5	Survey, review papers, blogging, lists with advantages/ disadvantages	
RBT-L6	Create a new model, programming or debugging activities, research projects, develop and describe new solutions or plans, brainstorming	

Finally, metadata of learning activities include the number of units, existing in each learning activity, e.g., a true/false activity may include 10 questions to be answered.

It needs to be noted that the number of recommended activities can be changed by the instructor.

3.3. Example of Operation

Figure 1 illustrates an example of operation of a knowledge-graph-based recommendation provided by the system. Our system recommends a matching activity, a memory activity, and a multiple-choice activity to Student 1. The presented knowledge graph is a snapshot of the developed recommender system, and it contains students, learning activities, RBT levels, difficulty level, and number of units as entities, while complete level, difficulty, belong, number, and classmates are the relationships between entities. The knowledge graph connects learning activities and students with many latent relationships, which helps to increase the accuracy of recommendations. The results of recommendations can be better illustrated and explained, which is another advantage of the knowledge-graph-based recommender system. Following the relationship sequences in the student–learning activity graph in the same scenario will reveal the justifications for suggesting these aforementioned learning activities to Student 1. For instance, one reason for recommending the matching activity is that it belongs to the same RBT level as flash cards, which has been previously completed by Student 1 successfully.

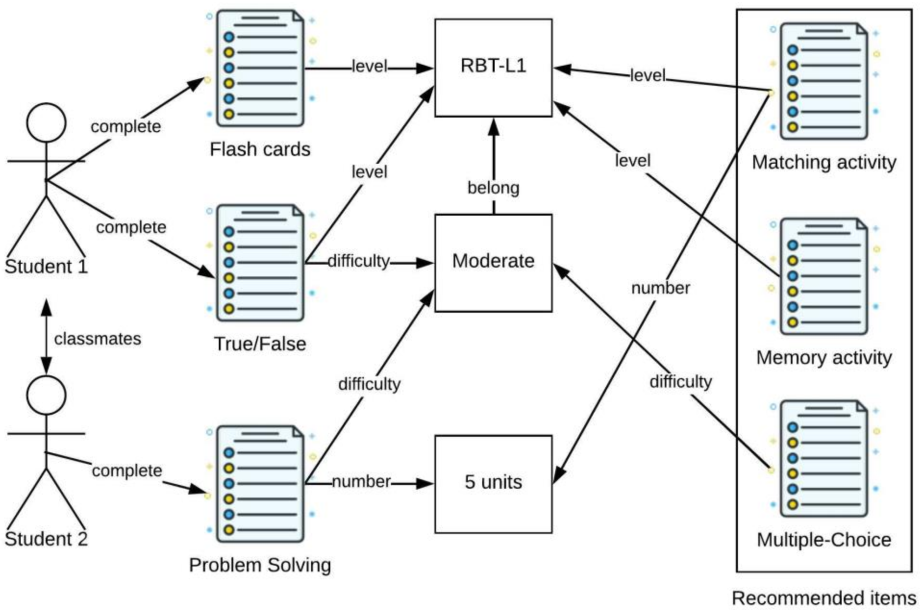

Figure 1. Example of operation.

4. System Evaluation

The phase of software evaluation is significant in order to assess its level of efficacy and acceptance by its users. In our case, the duration of the evaluation phase was an academic semester. The students used the e-learning software in the context of their compulsory course on object-oriented programming with Java language. Though the evaluation, we answered the two research questions (of Section 1). Particularly, both research questions are answered in the first part of the evaluation, while the second part of the evaluation answers additionally answers the second research question.

4.1. Population

The students who participated in the experiment are in their undergraduate studies, and their field of study is informatics and computer engineering in a public university in the capital city of the country (Table 2). In total, 100 students took part in the experiment and were divided into two equal groups by the evaluators and their university professors. Students in group 1 used the presented e-learning software incorporating the knowledge-graph-based recommender system. Students in group 2 used another version of the system, which had the same domain to be taught as well as the same interface. The difference was that the recommendations of the learning activities were given solely based on the RBT level to which they belonged.

Table 2. Characteristics of the population.

Characteristics	Group 1	Group 2
Average age	19.4	19.3
Gender	24 female, 26 male	25 female, 25 male
Demographics	Same number of students of urban and rural origins	
Computer expertise	Advanced computer skills	
Prior knowledge level in computer programming	All participants are students in the same year of studies and have successfully passed the previous programming courses.	
Motivation	All students attended the “Programming with Java” course and wanted to achieve a high grade.	

It needs to be noted that the presented system was used as a supplement to formal education. The same applies to the system’s leaning activities, which supplement the in-class activities. The instructors provided help to students while using the system, but they did not interfere with the recommendations delivered to learners by the system. Finally, the online learning activities were graded, but this grade did not affect the final grade of learners in the course; the students participated in the final examination, and only this grade was considered in order to pass the course successfully.

4.2. Results and Discussion

First, the system evaluation considers three aspects: user experience, effectiveness of the recommender system, and impact on learning [46,47]. As a result, a questionnaire with a 10-point Likert scale, with two questions for the evaluation of the first aspect as well as three questions for the evaluation of the second and third aspects, respectively (Table 3), was delivered to students. Toward evaluating the reliability of the questionnaire, we used Cronbach’s alpha, which was run on the sample. The alpha coefficient was 0.96, showing that our scale and the particular sample had a high level of internal consistency.

Table 3. Questionnaire.

Aspect	No.	Questions
User experience (UE)	Q1	Rate the user interface of the e-learning software.
	Q2	Rate the learning experience after your last interaction with the software.

Table 3. *Cont.*

Aspect	No.	Questions
Effectiveness of the recommender system (ER)	Q3	Did the learning activities correspond to your cognitive level?
	Q4	Rate the adequacy of difficulty level of the learning activities that have been recommended to you.
	Q5	Rate the adequacy of the degree of complexity of the learning activities that have been recommended to you.
Impact on learning (IL)	Q6	Did you find the e-learning software help you advance your knowledge in Java programming?
	Q7	Would you like to use this platform in other courses as well?
	Q8	Would you suggest the software to your friends to use it?

All the students answered the questionnaire when delivered at the end of the academic semester.

The students' answers were aggregated based on the aspect to which the corresponding questions belong. Furthermore, the 10-point Likert scale answers were converted into three categories, namely:

- Low: ranging from 1 to 3;
- Average: ranging from 4 to 7;
- High: ranging from 8 to 10.

In Figure 2, the results concerning the answers of students in groups 1 and 2 are presented.

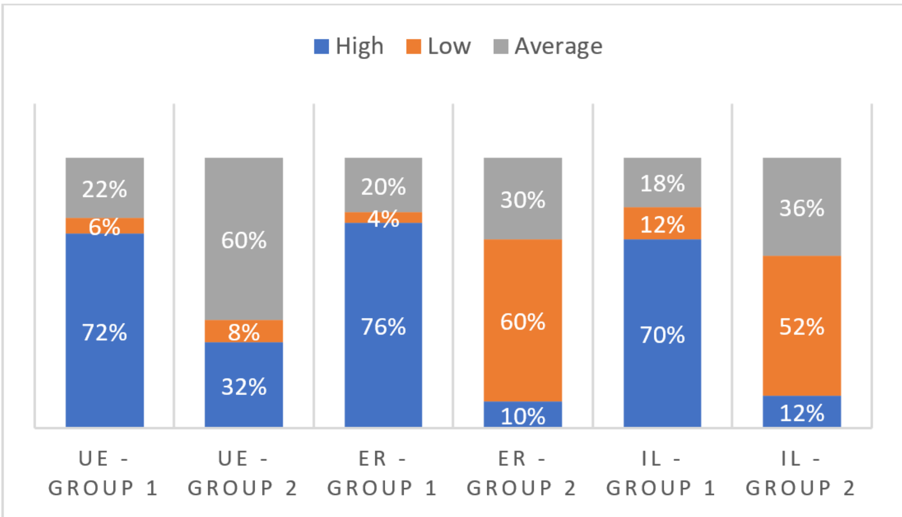

Figure 2. Questionnaire results.

Concerning the aspect of user experience, the responses of students in group 1 showed a high rating (72%), while the responses of students in group 2 showed a 32% rating. This difference reveals that the experience of students in group 1 was more positive, probably because their responses were affected by the adequate and personalized recommended learning activities (even though they were not explicitly asked about it in the questions of the first aspect). Moreover, 76% of students in group 1 appraised the effectiveness of the recommender system, while only 10% of students in group 2 appraised the effectiveness of the recommender system. This difference was expected since the presented recommender system incorporates the path-based method of a recommender system with a knowledge graph. The presented method suggests learning activities to students based on sophisticated reasoning, whereas the RBT-based method of recommendations of the system used by group 2 explores solely the RBT levels of learning activities. Finally, the results of the responses in the questions of the aspect impact of learning were 70% for group 1 and 10%

for group 2. This fact accentuates the importance of the presented knowledge-graph-based recommender system, highlighting its positive impact on learning.

Toward further exploring the effect of the recommender system on students, the statistical hypothesis test (*t*-test) was used, comparing the presented system (used by group 1) to its conventional version (used by group 2). The *t*-test was applied in questions Q3–Q5. Table 4 presents the *t*-test findings.

Table 4. The *t*-test findings.

	Q3		Q4		Q5	
	Group 1	**Group 2**	**Group 1**	**Group 2**	**Group 1**	**Group 2**
Mean	8.48	3.64	8	3.54	8.38	4
Variance	2.703673	4.888163	3.142857	3.559592	4.117959	6
Observations	50	50	50	50	50	50
Pooled variance	3.795918		3.351224		5.05898	
Hypothesized mean difference	0		0		0	
df	98		98		98	
t-Stat	12.42101		12.18157		9.736719	
$P(T \leq t)$ one-tailed	3.85×10^{-22}		1.24×10^{-21}		2.26×10^{-16}	
$P(T \leq t)$ two-tailed	7.71×10^{-22}		2.47×10^{-21}		4.51×10^{-16}	

Based on these results, it can be inferred that that there is a statistically significant difference between the means of the two trials concerning Q3, Q4, and Q5. More specifically, it was observed that the software used by group 1 performed significantly well in recommending learning activities to students that corresponded to their cognitive level in comparison to the conventional version used by group 1 (Q3: *t*-stat ≈ 16.42, $p < 0.05$). Additionally, there was a significant difference in the adequacy of the difficulty level of the learning activities that were recommended to students (Q4) in group 1 (mean = 8, variance ≈ 3.14) and group 2 (mean = 3.54, variance ≈ 3.55), where *t*-stat ≈ 12.18 and $p \approx 2.47 \times 10^{-21}$. The same applies for the adequacy of the degree of complexity of the learning activities that were recommended to students (Q5) in group 1 (mean = 8.38, variance ≈ 4.11) and group 2 (mean = 4, variance = 6), where *t*-stat ≈ 9.73 and $p \approx 4.51 \times 10^{-16}$.

These findings imply that the suggested approach performs better than its conventional version in terms of the appropriateness of learning activities that correspond to students' cognitive level, the adequacy of the difficulty level, and the degree of complexity of the learning activities that were recommended to students. These results were expected since the software, which uses a path-based method for recommender systems using a knowledge graph, recommends adequate learning activities to students. Such activities can create a personalized learning path for students and improve further their learning outcomes. This path-based reasoning for selecting learning activities may have important pedagogical implications since the recommendation of appropriate learning activities could further enhance the learning experience.

As mentioned in Section 2, similar knowledge-graph-based recommender systems have not yet been applied sufficiently in e-learning settings. Comparing the presented approach to others that have been used in recommender systems in e-learning software (e.g., content-based filtering, collaborative filtering, hybrid methods, machine learning), it should be highlighted that the user–item graph's inclusion of the potential information between users and items, which carries the entire information, particularly semantic information, into the graph, is one of major advantages of the path-based method.

5. Conclusions and Future Work

In this paper, we explored the path-based method for recommender systems using knowledge graphs in an e-learning environment. The system monitors the actions of users and through the knowledge-graph-based recommender system can suggest learning activities to them. Each activity is characterized by several metadata, such as type, difficulty, and number of questions. The novel recommender system has pedagogical affordance through the sophisticated way of suggesting learning activities. Our investigation shows that this method has positive results and can create a fertile ground for future research to this direction.

Limitations of this research include the exploration of only one method of recommender systems with knowledge graphs, namely the path-based method. In the related literature, there are other methods as well, such as embedding-based methods or even unified methods that have not been explored as well.

Future work includes the incorporation of machine learning into the presented technique to explore whether it could be further improved. Moreover, future research plans include the enrichment of recommendations to students with peers to collaborate. Finally, part of our future plans is to explore the number of the learning activities that is required to incorporate in the recommender system to render it robust and offer adequate personalization.

Author Contributions: Conceptualization, C.T. and A.K.; methodology, C.T. and A.K.; software, C.T. and A.K.; validation, C.T. and A.K.; formal analysis, C.T. and A.K.; investigation, C.T. and A.K.; resources, C.T. and A.K.; data curation, C.T. and A.K.; writing—original draft preparation, C.T. and A.K.; writing—review and editing, C.T. and A.K.; visualization, C.T. and A.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are available on request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ko, H.; Lee, S.; Park, Y.; Choi, A. A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields. *Electronics* **2022**, *11*, 141. [CrossRef]
2. Marques, G.A.; Rigo, S.J.; Alves, I.M.D.R. Graduation Mentoring Recommender—Hybrid Recommendation System for Customizing the Undergraduate Student's Formative Path. In Proceedings of the 2021 XVI IEEE Latin American Conference on Learning Technologies (LACLO), Arequipa, Peru, 19–21 October 2021; pp. 342–349. [CrossRef]
3. Papakostas, C.; Troussas, C.; Krouska, A.; Sgouropoulou, C. Personalization of the Learning Path within an Augmented Reality Spatial Ability Training Application Based on Fuzzy Weights. *Sensors* **2022**, *22*, 7059. [CrossRef] [PubMed]
4. Agarwal, A.; Mishra, D.S.; Kolekar, S.V. Knowledge-based recommendation system using semantic web rules based on Learning styles for MOOCs. *Cogent Eng.* **2022**, *9*, 2022568. [CrossRef]
5. Li, J.; Ye, Z. Course Recommendations in Online Education Based on Collaborative Filtering Recommendation Algorithm. *Complexity* **2020**, *2020*, 6619249. [CrossRef]
6. Guo, Q.; Zhuang, F.; Qin, C.; Zhu, H.; Xie, X.; Xiong, H.; He, Q. A Survey on Knowledge Graph-Based Recommender Systems. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 3549–3568. [CrossRef]
7. Liu, J.; Duan, L. A Survey on Knowledge Graph-Based Recommender Systems. In Proceedings of the 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 12–14 March 2021; pp. 2450–2453. [CrossRef]
8. Amara, S.; Subramanian, R.R. Collaborating personalized recommender system and content-based recommender system using TextCorpus. In Proceedings of the 2020 6th IEEE International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6–7 March 2020; pp. 105–109. [CrossRef]
9. Fernández-García, A.J.; Rodríguez-Echeverría, R.; Preciado, J.C.; Manzano, J.M.C.; Sánchez-Figueroa, F. Creating a Recommender System to Support Higher Education Students in the Subject Enrollment Decision. *IEEE Access* **2020**, *8*, 189069–189088. [CrossRef]
10. Chrysafiadi, K.; Troussas, C.; Virvou, M. A Framework for Creating Automated Online Adaptive Tests Using Multiple-Criteria Decision Analysis. In Proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Miyazaki, Japan, 7–10 October 2018; pp. 226–231. [CrossRef]

11. Kannikaklang, N.; Wongthanavasu, S.; Thamviset, W. A Hybrid Recommender System for Improving Rating Prediction of Movie Recommendation. In Proceedings of the 2022 19th IEEE International Joint Conference on Computer Science and Software Engineering (JCSSE), Bangkok, Thailand, 22–25 June 2022; pp. 1–6. [CrossRef]
12. Gupta, M.; Thakkar, A.; Gupta, V.; Rathore, D.P.S. Movie recommender system using collaborative filtering. In Proceedings of the 2020 IEEE International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2–4 July 2020; pp. 415–420.
13. Singla, R.; Gupta, S.; Gupta, A.; Vishwakarma, D.K. FLEX: A content-based movie recommender. In Proceedings of the 2020 IEEE International Conference for Emerging Technology (INCET), Belgaum, India, 5–7 June 2020; pp. 1–4.
14. Troussas, C.; Krouska, A.; Virvou, M. Multi-algorithmic techniques and a hybrid model for increasing the efficiency of recommender systems. In Proceedings of the 2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI), Volos, Greece, 5–7 November 2018; pp. 184–188.
15. Yang, X.; Dong, M.; Chen, X.; Ota, K. Recommender system-based diffusion inferring for open social networks. *IEEE Trans. Comput. Soc. Syst.* **2020**, *7*, 24–34. [CrossRef]
16. Wongkhamchan, T.; Namvong, A.; Surawanitkun, C. Personalized recommender system using a social network based collaborative filtering technique. In Proceedings of the 2019 16th IEEE International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Pattaya, Thailand, 10–13 July 2019; pp. 846–849.
17. Mughaid, A.; Obeidat, I.; Hawashin, B.; AlZu'bi, S.; Aqel, D. A smart geo-location job recommender system based on social media posts. In Proceedings of the 2019 Sixth IEEE International Conference on Social Networks Analysis, Management and Security (SNAMS), Granada, Spain, 22–25 October 2019; pp. 505–510.
18. Liang, B.; Xu, B.; Wu, X.; Wu, D.; Yang, D.; Xiao, Y.; Wang, W. A community-based collaborative filtering method for social recommender systems. In Proceedings of the 2019 IEEE International Conference on Web Services (ICWS), Milan, Italy, 8–13 July 2019; pp. 159–162.
19. Baker, O.; Yuan, Q. Machine learning: Factorization Machines and Normalized Discounted Cumulative Gain for Tourism Recommender System Optimisation. In Proceedings of the 2021 IEEE International Conference on Computing (ICOCO), Kuala Lumpur, Malaysia, 17–19 November 2021; pp. 31–36. [CrossRef]
20. Srisawatsakul, C.; Boontarig, W. Tourism Recommender System using Machine Learning Based on User's Public Instagram Photos. In Proceedings of the 5th IEEE International Conference on Information Technology (InCIT), Chonburi, Thailand, 21–22 October 2020; pp. 276–281. [CrossRef]
21. Chen, C.; Wu, X.; Chen, J.; Pardalos, P.M.; Ding, S. Dynamic grouping of heterogeneous agents for exploration and strike missions. *Front. Inform. Technol. Electron. Eng.* **2022**, *23*, 86–100. [CrossRef]
22. Kbaier, M.E.B.H.; Masri, H.; Krichen, S. A Personalized Hybrid Tourism Recommender System. In Proceedings of the 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA), Hammamet, Tunisia, 30 October–3 November 2017; pp. 244–250. [CrossRef]
23. Al-Ghuribi, S.M.; Mohd Noah, S.A. Multi-Criteria Review-Based Recommender System–The State of the Art. *IEEE Access* **2019**, *7*, 169446–169468. [CrossRef]
24. Choi, S.-M.; Jang, K.; Lee, T.-D.; Khreishah, A.; Noh, W. Alleviating item-side cold-start problems in recommender systems using weak supervision. *IEEE Access* **2020**, *8*, 167747–167756. [CrossRef]
25. Lan, J.; Shi, R.; Cao, Y.; Lv, J. Knowledge Graph-based Conversational Recommender System in Travel. In Proceedings of the 2022 IEEE International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 18–23 July 2022; pp. 1–8. [CrossRef]
26. Dai, G.R.; Xu, J.Y.; Zhou, H.; Zhou, L.; Cao, Z.H.; Zhu, M.D. A Travelling Recommendation System based on Knowledge Graph. In Proceedings of the 2021 3rd International Academic Exchange Conference on Science and Technology Innovation (IAECST), Guangzhou, China, 10–12 December 2021; pp. 407–412. [CrossRef]
27. Dadoun, A.; Troncy, R.; Defoin-Platel, M.; Solano, G.A. Predicting your next trip: A knowledge graph-based multi-task learning approach for travel destination recommendation. In Proceedings of the 2021 Workshop on Recommenders in Tourism, RecTour 2021, Amsterdam, The Netherlands, 26 September 2021; Volume 2974.
28. Deladiennee, L.; Naudet, Y. A graph-based semantic recommender system for a reflective and personalised museum visit: Extended abstract. In Proceedings of the 2017 IEEE 12th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP), Bratislava, Slovakia, 9–10 July 2017; pp. 88–89. [CrossRef]
29. Minkov, E.; Kahanov, K.; Kuflik, T. Graph-based recommendation integrating rating history and domain knowledge: Application to on-site guidance of museum visitors. *J. Assoc. Inf. Sci. Technol.* **2017**, *68*, 1911–1924. [CrossRef]
30. Angelis, S.; Kotis, K.; Spiliotopoulos, D. Semantic Trajectory Analytics and Recommender Systems in Cultural Spaces. *Big Data Cogn. Comput.* **2021**, *5*, 80. [CrossRef]
31. Sousa, D.; Couto, F.M. Biomedical Relation Extraction With Knowledge Graph-Based Recommendations. *IEEE J. Biomed. Health Inform.* **2022**, *26*, 4207–4217. [CrossRef] [PubMed]
32. Gogleva, A.; Polychronopoulos, D.; Pfeifer, M.; Poroshin, V.; Ughetto, M.; Martin, M.J.; Thorpe, H.; Bornot, A.; Smith, P.D.; Sidders, B.; et al. Knowledge graph-based recommendation framework identifies drivers of resistance in EGFR mutant non-small cell lung cancer. *Nat. Commun.* **2022**, *13*, 1667. [CrossRef] [PubMed]

33. Fecho, K.; Bizon, C.; Miller, F.; Schurman, S.; Schmitt, C.; Xue, W.; Morton, K.; Wang, P.; Tropsha, A. A Biomedical Knowledge Graph System to Propose Mechanistic Hypotheses for Real-World Environmental Health Observations: Cohort Study and Informatics Application. *JMIR Med. Inform.* **2021**, *9*, e26714. [CrossRef] [PubMed]
34. Vijayakumar, S.; Jetcheva, J.G. SMARTREC: A Conversational Recommendation System Using Semantic Knowledge Graphs. In Proceedings of the 2022 IEEE Eighth International Conference on Big Data Computing Service and Applications (BigDataService), Newark, CA, USA, 15–18 August 2022; pp. 53–60. [CrossRef]
35. Lully, V.; Laublet, P.; Stankovic, M.; Radulovic, F. Enhancing explanations in recommender systems with knowledge graphs. *Procedia Comput. Sci.* **2018**, *137*, 211–222. [CrossRef]
36. Kartheek, M.; Sajeev, G.P. Building Semantic Based Recommender System Using Knowledge Graph Embedding. In Proceedings of the 2021 Sixth International Conference on Image Information Processing (ICIIP), Shimla, India, 26–28 November 2021; pp. 25–29. [CrossRef]
37. Han, W.; Wang, Q. Movie recommendation algorithm based on knowledge graph. In Proceedings of the 2019 2nd International Conference on Safety Produce Informatization (IICSPI), Chongqing, China, 28–30 November 2019; pp. 409–412. [CrossRef]
38. Sarkar, R.; Goswami, K.; Arcan, M.; McCrae, J.P. Suggest me a movie for tonight: Leveraging Knowledge Graphs for Conversational Recommendation. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 4179–4189.
39. Lee, C.; Han, D.; Han, K.; Yi, M. Improving Graph-Based Movie Recommender System Using Cinematic Experience. *Appl. Sci.* **2022**, *12*, 1493. [CrossRef]
40. Shi, C.; Zhang, Z.; Luo, P.; Yu, P.S.; Yue, Y.; Wu, B. Semantic path based personalized recommendation on weighted heterogeneous information networks. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, Melbourne, Australia, 18–23 October 2015; pp. 453–462.
41. Hu, B.; Shi, C.; Zhao, W.X.; Yu, P.S. Leveraging metapath based context for top-n recommendation with a neural co-attention model. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 1531–1540.
42. Sun, Z.; Yang, J.; Zhang, J.; Bozzon, A.; Huang, L.-K.; Xu, C. Recurrent knowledge graph embedding for effective recommendation. In Proceedings of the 12th ACM Conference on Recommender Systems, Vancouver, BC, Canada, 2 October 2018; pp. 297–305.
43. Huang, X.; Fang, Q.; Qian, S.; Sang, J.; Li, Y.; Xu, C. Explainable interaction-driven user modeling over knowledge graph for sequential recommendation. In Proceedings of the 27th ACM International Conference on Multimedia, Nice, France, 21–25 October 2019; pp. 548–556.
44. Hua, S.; Chen, W.; Li, Z.; Zhao, P.; Zhao, L. Path-Based Academic Paper Recommendation. In *Web Information Systems Engineering—WISE 2020. WISE 2020*; Huang, Z., Beek, W., Wang, H., Zhou, R., Zhang, Y., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2020; Volume 12343. [CrossRef]
45. Krathwohl, D.R. A revision of Bloom's taxonomy: An overview. *Theory Pract.* **2002**, *41*, 212–218. [CrossRef]
46. Alepis, E.; Troussas, C. M-learning programming platform: Evaluation in elementary schools. *Informatica* **2017**, *41*, 471–478.
47. Krouska, A.; Troussas, C.; Sgouropoulou, C. A Cognitive Diagnostic Module Based on the Repair Theory for a Personalized User Experience in E-Learning Software. *Computers* **2021**, *10*, 140. [CrossRef]