information

MDPI

Article

Enhancing Personalized Educational Content Recommendation through Cosine Similarity-Based Knowledge Graphs and Contextual Signals

Christos Troussas [1,*], Akrivi Krouska [1], Panagiota Tselenti [1], Dimitrios K. Kardaras [2] and Stavroula Barbounaki [3]

[1] Department of Informatics and Computer Engineering, University of West Attica, 122 43 Egaleo, Greece; nota@uniwa.gr (P.T.)
[2] Business Informatics Lab, Department of Business Administration, School of Business, Athens University of Economics and Business, 104 34 Athens, Greece
[3] Department of Midwifery, University of West Attica, 122 43 Egaleo, Greece
* Correspondence: ctrouss@uniwa.gr

Abstract: The extensive pool of content within educational software platforms can often overwhelm learners, leaving them uncertain about what materials to engage with. In this context, recommender systems offer significant support by customizing the content delivered to learners, alleviating the confusion and enhancing the learning experience. To this end, this paper presents a novel approach for recommending adequate educational content to learners via the use of knowledge graphs. In our approach, the knowledge graph encompasses learners, educational entities, and relationships among them, creating an interconnected framework that drives personalized e-learning content recommendations. Moreover, the presented knowledge graph has been enriched with contextual signals referring to various learners' characteristics, such as prior knowledge level, learning style, and current learning goals. To refine the recommendation process, the cosine similarity technique was employed to quantify the likeness between a learner's preferences and the attributes of educational entities within the knowledge graph. The above methodology was incorporated in an intelligent tutoring system for learning the programming language Java to recommend content to learners. The software was evaluated with highly promising results.

Keywords: knowledge graph; recommender system; intelligent tutoring system; educational software; learning content; learning style; learning goals; knowledge level

check for updates

Citation: Troussas, C.; Krouska, A.; Tselenti, P.; Kardaras, D.K.; Barbounaki, S. Enhancing Personalized Educational Content Recommendation through Cosine Similarity-Based Knowledge Graphs and Contextual Signals. *Information* **2023**, *14*, 505. https://doi.org/10.3390/info14090505

Academic Editors: Pierpaolo Basile and Annalina Caputo

Received: 22 August 2023
Revised: 5 September 2023
Accepted: 12 September 2023
Published: 14 September 2023

1. Introduction

The rapid proliferation of data has been greatly facilitated by the swift advancement of the Internet. Consequently, users often find themselves in a daunting environment where the abundance of choices makes selecting relevant information challenging. This dilemma arises from the overabundance of available data. To mitigate this issue and aid users in making optimal selections, recommender systems have emerged as a solution, offering tailored suggestions that align with specific user preferences [1]. The field of recommender systems extends across diverse domains, including e-commerce platforms, e-learning systems, online movie platforms, and music streaming service providers.

In the context of e-learning, recommender systems have been employed to support learners during the educational process [2]. In these systems, recommendations largely hinge on both the content of the learning material and user behaviors observed during interactions with online learning platforms. For instance, e-learning recommender systems can provide personalized recommendations related to students' learning styles, resources, activities, courses, and learning pathways [3–5]. Often, content and types of learning activities may be an obstacle for learners when selecting them. Thus, many e-learning platforms assign specific learning activities to students. However, recommender systems step in by suggesting tailored activities, thus ensuring a personalized approach that maintains the educational rigor of the learning process.

Researchers predominantly rely on collaborative filtering, content-based filtering, or hybrid approaches for developing recommender systems [6]. Collaborative filtering operates on the assumption that users who have previously agreed on preferences are likely to continue doing so. Recommendations are formulated based on data obtained from rating profiles of various users or items. This method identifies similar users or objects based on historical ratings, thus generating suggestions. Content-based filtering, on the other hand, builds recommendations by focusing on item descriptions and user profiles. This approach is most effective when item information, rather than user information, is accessible. These algorithms view recommendations as user-specific classification problems, learning a user's preferences based on item attributes. The combination of these techniques yields hybrid recommender systems that leverage the strengths of both approaches and minimizes their limitations.

Recent interest has turned towards integrating knowledge graphs (KGs) as supplementary information in recommender systems [7]. KGs represent intricate networks where nodes symbolize entities and relationships denote connections between these entities. KGs offer a deeper comprehension of item relationships and user preferences, enhancing recommendation quality and interpretability. Knowledge graph-based recommender systems are categorized as embedding-based methods, path-based methods, or unified methods [6]. Embedding-based methods directly enhance item or user representations with KG data. Before utilizing KG data, knowledge graph embedding techniques encode the KG into compact embeddings. Path-based methods create a user–item graph, leveraging entity connectivity patterns to provide recommendations. These methods exploit connection similarity among users and objects to improve suggestions. Unified methods incorporate both semantic entity–relation representations and connectivity data, enhancing recommendation quality. The unified approach is rooted in embedding propagation theory, enhancing entity representation based on KG connectivity structure.

In view of the above, this paper presents a novel approach for enhancing personalized educational content recommendation by using a knowledge graph. The knowledge graph includes learners, educational entities, and relationships, resulting in an interconnected architecture that generates personalized e-learning material recommendations. Furthermore, it has been enhanced with contextual signals which correspond to various features and information that provide context and insights about the characteristics of learners, such as prior knowledge level, learning style, and current learning goal. To refine the recommendation process, the integration of cosine similarity is explored. Cosine similarity measures the alignment between learners' attributes and the features of educational entities, leading to refined recommendations. To recommend appropriate content to learners, the aforementioned approach was implemented in an intelligent tutoring system for learning the programming language Java. The novelty of the presented methodology arises from the fact that the knowledge graph involves multiple learner characteristics that collectively define the learner's profile and learning preferences. This multi-dimensional view enhances the granularity of recommendations. Also, the inclusion of attributes such as learning style preference and current learning goal adds a layer of contextual information, making the recommendations more relevant.

2. Related Work

This section presents the review of the literature on recommender systems as well as the utilization of knowledge graphs for generating recommendations.

2.1. Recommender Systems

Recommender systems have been explored across various domains in the relevant scientific literature. Their utilization spans e-learning platforms [2,8–10], entertainment websites [11–14], social contexts [15–18], and tourism applications [19–22], among others. Collaborative filtering, content-based filtering, machine learning, and hybrid methodologies represent the primary algorithmic techniques, commonly employed in the aforementioned research papers. According to a study in 2019 [23], the primary data source for most recommender systems (overall rating) is single-criterion ratings. Another study from 2020 [24] underscores the limitations of collaborative filtering, a widely used approach in constructing recommender systems, including challenges such as the cold-start problem, where insufficient data impedes making inferences about individuals or items.

The area of recommender systems within e-learning environments remains highly active. In such systems, two pivotal factors come into play: (1) the learners, who are the recipients of recommendations, possess nuanced attributes that are sometimes complex to define, such as their knowledge level; and (2) the recommendations made to learners can profoundly influence knowledge acquisition, encompassing elements such as learning activities, educational resources, and assessment components.

Given the above observations, it is evident that further investigation and advancements are required in the field of recommender systems development.

2.2. Knowledge Graphs for Recommendations

Knowledge graphs have been integrated into recommender systems as supplementary means to monitor shifts in user preferences and contribute depth to recommendations. Indeed, there exists an active research domain dedicated to exploring recommender systems grounded in knowledge graphs. Their application spans diverse sectors, including travel websites [25–27], virtual museums [28–30], biomedical platforms [31–33], e-commerce platforms [34–36], and film portals [37–39]. Across these studies, as explicitly highlighted in recent comprehensive reviews [40,41], the prevalent methods for knowledge graph-infused recommender systems in the literature encompass embedding-based strategies, connection-oriented approaches, and propagation-derived methodologies.

Delving into knowledge graph-based recommender systems, the authors of [42] introduced SemRec, a model that accounts for users' favored and disliked historical items. This architecture employs a weighted meta-path to incorporate attribute values into links, facilitating more precise depiction of item relationships and user similarities. This framework enables the portrayal of positive and negative preference patterns, thereby disseminating the true user preference. MCRec, introduced in [43], generates explicit meta-path representations to capture the interaction context of user-item pairs. Alternatively, a recurrent knowledge graph embedding technique was proposed in [44] to automatically uncover the route link between users and items without relying on user-provided meta-paths. The authors in [45] designed a method for sequential recommendation that capitalizes on users' evolving interests. In [46], a strategy was devised to arrange a sequence of pivotal academic papers, immensely valuable for aiding researchers in comprehending the evolution of specific topics. This approach employs both content and network structure to acquire document representations, subsequently employed to evaluate paper similarity. Finally, in [47], the authors present a path-based approach for recommender systems using knowledge graphs in an e-learning environment. This method emphasizes the sequence and progression of learning activities while the presented approach in this paper considers a broader set of interconnected elements and attributes to personalize recommendations on educational content.

When the pertinent literature in the domain of knowledge graph-based recommender systems tailored to the field of e-learning is analyzed, it is evident that substantial opportunities exist for advancements in this direction. This research is motivated by the potential of knowledge graphs to provide a fertile ground for sophisticated recommendations mechanisms customized for e-learning.

3. Knowledge Graph Presentation

The knowledge graph, described in this paper, represents a structured and interconnected framework that captures various aspects of e-learning, leading to personalized content recommendation. This graph serves as a visual representation of relationships between different entities, attributes, and learners within an educational context. Each element in the graph contributes significantly to improving the recommendation process and adapting it to the needs of individual learners. The explanation of the key components of the knowledge graph (Figure 1) are provided as follows:

A. Learners (Nodes):

Learners are distinct entities within the knowledge graph, each with their own set of attributes and preferences. These learners are represented as nodes in the graph. Learner characteristics, referred to as contextual signals, provide crucial insights into their learning context:

- Prior Knowledge Level (PKL): This attribute reflects the learner′s proficiency in the programming language Java. It could be labeled as "Beginner", "Intermediate", or "Advanced", depending on their familiarity with the content. This classification aligns with established educational norms and ensures a balanced distribution of learners across distinct proficiency levels [48].
- Learning Style Preference (LSP): This attribute describes how a learner prefers to gain knowledge. It is based on the VARK model and could be "Visual", "Auditory", "Reading"-"Writing", or "Kinesthetic" [49]. The VARK model was chosen because its categories can be expressed as knowledge graph properties, allowing for integration with other contextual attributes and relationships.
- Current Learning Goal (CLG): This attribute specifies the specific goal a learner aims to achieve. In our case, goals may include "Master Java Programming", "Improve Problem-Solving Skills", etc. and are established for the course under the guidance of its university professors.

In our methodology, we collected these signals via initial surveys and actual platform usage. Rigorous validation was conducted by comparing the signals against observed user behavior, bolstering their reliability. To accommodate changes in preferences and evolving needs over time, our system incorporates a feedback mechanism through which learners can declare any change in their preferences.

B. Educational Entities (Nodes):

Educational entities include various elements that are important to the learning process. Each of these entities is represented in the graph as a node:

- Courses: Courses represent broader subject areas and are nodes that encapsulate a collection of linked topics and learning resources.
- Topics: Within a course, topics are finer-grained concepts. They represent individual units of study and are linked with courses.
- Learning Resources: The learning resources are nodes, containing different materials such as articles, videos, interactive simulations, and assessments. They are linked to specific topics, demonstrating their relevance.

An example of a course in Java may be "Java Data Structures and Algorithms". The topics that it can include may be: Introduction to Data Structures and Algorithms, Arrays and Linked Lists in Java, Stacks and Queues Implementation, Hashing and Hash Tables, Trees and Binary Search Trees, Graphs and Graph Algorithms, Sorting Algorithms (e.g., Bubble Sort, Merge Sort), Searching Algorithms (e.g., Binary Search), Dynamic Programming and Greedy Algorithms, Advanced Data Structures (e.g., Heaps, Hash Maps) and Time and Space Complexity Analysis. The learning resources may be: Comprehensive Video Lectures on Data Structures, Interactive Code Challenges for Practicing Algorithms, Java Code Examples for Different Data Structures, Algorithm Visualizations and Demonstrations, Books and Online Resources on Algorithms in Java, Online Coding Competitions for Algorithmic Problem Solving, Case Studies on Real-World Algorithm Implementations, Discussion

Forums for Algorithmic Questions and Answers, Coding Templates and Best Practices for Java Algorithms, and Online Courses on Advanced Data Structures and Algorithms.

C. Relationships (Edges):

The relationships (edges) connecting the nodes define how the entities are linked and influence the content recommendation process:

- *Learner–Course Relationship*: This edge denotes the enrollment of a learner in a specific course. It expresses the learner's intention to study a specific subject.
- *Course–Topic Relationship*: This edge connects courses to their corresponding topics. It provides information on the topics that are covered in each course.
- *Topic–Learning Resource Relationship*: This edge links topics to corresponding learning resources. It determines the applicability of resources to specific subjects.
- *Learner–Learning Resource Relationship*: This edge represents the interaction of a learner with learning resources. It displays the resources with which a learner has engaged.
- *Learner–PKL Relationship*: This edge connects the learner with his/her prior knowledge level. It expresses the learner's proficiency in a particular subject.
- *Learner–LSP Relationship*: This edge defines the connection between a learner and his/her learning style.
- *Learner–CLG Relationship*: This edge represents the link between a learner and his/her current learning goal. It defines the exact goal that the learner aspires to achieve through the educational process.
- *Learning Resource–CLG Relationship*: The edge provides a link between a learning resource and a learner's current learning goal. This link guarantees that the recommended resources are relevant to the learner's unique goal.

D. Contextual Attributes:

The attributes associated with learners play a pivotal role in shaping the content recommendation process:

- *Leveraging Prior Knowledge Level*: The complexity of recommended information is influenced by the learner's previous knowledge level. For example, an "Intermediate" learner, for example, may be given more advanced resources.
- *Adapting to Learning Style Preference*: Recommendations can be adjusted to the learner's learning style. For instance, visual learners may be given more video content, and auditory learners may be given podcasts.
- *Aligning with Current Learning Goal*: The system customizes recommendations to assist learners in achieving their goals. For example, a learner with an objective of "Develop Java Programming Skills" would obtain resources that support that goal.

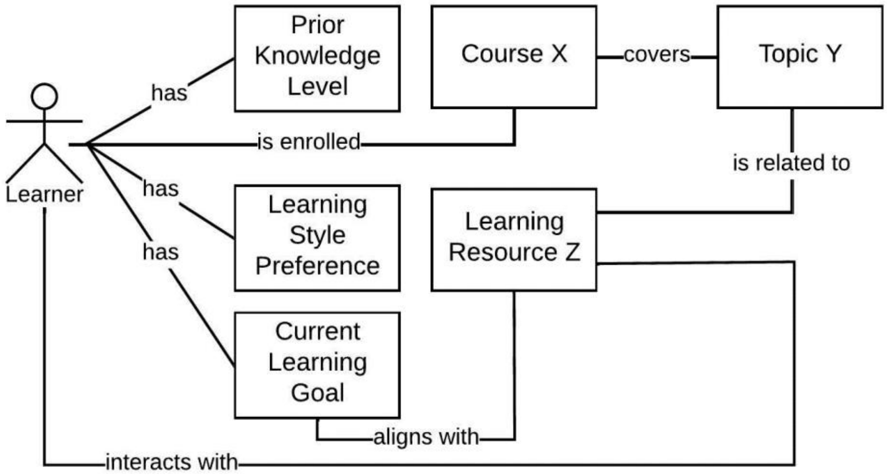

Figure 1. Knowledge graph.

In this system, learners provide information about their knowledge level, learning style preferences, and current learning goals. These attributes are used to assign values to their vectors. Additionally, attributes of educational resources are assigned values based on their relevance to learners' needs. This way, the vectors capture learners' preferences and help in providing personalized recommendations.

4. Personalized Educational Recommendation

The recommendation process begins with the use of the nodes and edges of the knowledge graph, which capture the dense network of connections between various educational aspects. The recommendation method uses the structure of the knowledge graph to identify educational materials that are closely related to the interests and aims of the learners. The system can intelligently forecast and recommend information that enables a continuous progression of knowledge by recognizing nodes. For instance, consider "Learner A", an Intermediate learner with a Visual Learner preference. The graph can identify "Topic Y", which matches Learner A's proficiency and learning goal, a critical step enabled by relationships between learners and educational entities.

The intricate relationships embedded inside the graph act as personalization facilitators. The interaction between topics and learning resources, as well as the relationship between learners and their courses, all contribute to the smooth orchestration of recommendations. Throughout the interaction of Learner A with the system, the graph's insights are constantly refined. Based on Learner A's interactions, the system can adapt the recommendations in a dynamic way.

To enhance the recommendation precision, we employ cosine similarity in the knowledge graph-based recommender system with the following steps:

1. Representation of learners and educational entities (courses, topics, learning resources) as vectors based on their attributes. The dimension of each vector matches with a contextual attribute (e.g., PKL, LSP, CLG).
2. Calculation of the cosine similarity between a learner′s vector (numerical representation of the learner′s attributes, such as PKL, LSP, and CLG) and an educational entity′s vector (numerical representation of the educational entity′s characteristics and content) to determine how similar a learner′s vector is to an educational entity′s vector, use the following formula:

$$\text{Cosine similarity} = \frac{\text{learner vector} \cdot \text{entity vector}}{||\text{learner vector}|| \cdot ||\text{entity vector}||}$$

3. Ranking of the educational entities based on their cosine similarity scores in descending order.
4. Presentation of the top-ranked educational entities as recommendations to the learner.

Using cosine similarity within the context of the knowledge graph enhances the sophistication of the recommendation system by considering the multidimensional relationships and preferences of learners and educational entities. It allows for more nuanced and accurate recommendations that go beyond simplistic comparisons.

Although cosine similarity is a significant technique, its effectiveness can be influenced by the choice of attributes and features used in the calculation. While cosine similarity captures similarities in vector spaces, it may not fully account for nuanced relationships between learners and educational entities. To this end, our approach combines cosine similarity-based knowledge graphs with contextual signals to ensure a multi-dimensional assessment of learners' characteristics and preferences. As such, the contextual signals add a layer of personalization beyond the scope of cosine similarity alone.

5. Example of Operation

In this example, we will consider the knowledge graph for the Java programming course. The graph will illustrate how topics, resources, and learning styles can be interconnected to provide personalized recommendations for learners.

A. Nodes and Relationships:

1. Nodes (Topics): Variables, Data Types, Control Structures, Arrays, Functions, Object-Oriented Programming (OOP), Inheritance, Polymorphism, Exception Handling, File I/O, GUI Development, Multithreading, Networking, Java Libraries.
2. Nodes (Resources): Video Tutorials, Textbooks, Online Code Editors, Coding Challenges, Quiz Assessments, Interactive Java Code Examples, Online Forums.
3. Nodes (Learning Styles): Visual Learner, Auditory Learner, Reading/Writing, Kinesthetic Learner.
4. Relationships: Each topic node is connected to related resources such as tutorials, textbooks, and code examples. Learning styles are associated with specific topics and resources.

B. Example Scenario:

Emily is a postgraduate student enrolled in the Java programming course. Emily provided the input for her vectors within the system, reflecting her prior knowledge level, learning style preferences, and current learning goal. She is identified as a visual learner and has an intermediate proficiency level in Java (Intermediate). As follows, we will observe how the knowledge graph, along with cosine similarities, can provide personalized recommendations for her:

1. Contextual Attributes and Vector Representation:
   a. Emily's learner vector:
      i. Prior Knowledge Level: [1.0, 0.0, 0.0] (Beginner)
      ii. Learning Style Preference: [0.8, 0.0, 0.2] (Visual, Auditory, Reading/Writing Learner, Kinesthetic Learner)
      iii. Current Learning Goal: [0.0, 1.0, 0.0] (Master GUI Development) (for the example, we assume three CLG)
   b. Topic "Inheritance" vector: [0.8, 0.1, 0.1] (Video Tutorials, Textbooks, Coding Challenges)
   c. Topic "GUI Development" vector: [0.5, 0.3, 0.2] (Video Tutorials, Online Code Editors, Interactive Java Code Examples)
2. Cosine Similarity Calculation
   a. Cosine similarity (Emily, Inheritance) = ([1.0, 0.8, 0.0] × [0.8, 0.1, 0.1])/(sqrt($1.0^2 + 0.8^2 + 0.0^2$) × sqrt($0.8^2 + 0.1^2 + 0.1^2$)) = 0.922
   b. Cosine similarity (Emily, GUI Development) = ([1.0, 0.8, 0.0] × [0.5, 0.3, 0.2])/(sqrt($1.0^2 + 0.8^2 + 0.0^2$) × sqrt($0.5^2 + 0.3^2 + 0.2^2$)) = 0.775
3. Personalized Recommendations
   a. Emily's high cosine similarity with "Inheritance" indicates strong alignment with her visual learning style and relevant resources. The graph suggests she explores resources related to inheritance concepts.
   b. For "GUI Development", Emily's cosine similarity suggests a moderate match. The graph recommends video tutorials and interactive code examples to cater to her visual learning preference.
4. Adaptive Updates:
   a. As Emily progresses, the graph adapts by considering her interactions with the recommended resources. Her feedback and performance on coding challenges and quizzes influence the recommendations.

b. If Emily engages more with "Inheritance" resources, the graph continues to offer advanced topics like "Polymorphism" and "Exception Handling" with tailored content and challenges.

6. Evaluation

The assessment stage of software holds importance in gauging its effectiveness and user acceptance. In our scenario, the evaluation period spanned an entire academic semester. Students engaged with the e-learning software within the framework of their obligatory postgraduate course focused on object-oriented programming using the Java language.

Sample

The participants involved in the study are currently pursuing their postgraduate studies (a conversion course in informatics and applications), focusing on informatics and computer engineering at a public university situated in the capital city of the country (as outlined in Table 1). The experiment involved a total of 100 students, who were evenly divided into two groups by the evaluators and university professors. The first group, referred to as Group A, utilized the software with the incorporated knowledge graph-based recommender system. The second group, known as Group B, used an alternative version of the system with identical subject matter and interface. The distinction between the two versions lay in the method of providing recommendations for educational content. Group B solely relied on contextual signals to guide the recommendations. The system adopted by Group B, serving as the conventional version in our study, gathered a list of learner attributes to tailor its personalized recommendations. Specifically, it assessed the learner′s prior knowledge level. Furthermore, it used the learner′s learning style preferences. Moreover, it took into consideration the learner′s current learning goal. Once these learner attributes were gathered, the system employed a set of predefined metrics to evaluate them. It assigned weights to each attribute based on perceived significance. Let us consider a learner with a strong background in Java programming, a preference for visual learning materials, and a specific goal of excelling in Java programming skills. In this scenario, the educational content recommendation system might apply rules that assign a higher weight to the learner′s prior knowledge attribute, recognizing their expertise in Java. Additionally, it would consider the learner′s preference for visual aids, giving it due importance. Furthermore, to align with the learner′s specific goal of excelling in Java programming, the system would prioritize recommending content directly related to Java programming, such as video tutorials, interactive coding exercises, and Java programming materials. The rules, for this example, are as follows:

- If the learner′s prior knowledge level in Java exceeds a threshold value (e.g., priorKnowledgeLevel > 7), assign a higher weight (e.g., priorKnowledgeWeight = 0.8) to the prior knowledge attribute. This rule recognizes and acknowledges the learner′s expertise in Java programming.
- If the learner′s preferred learning style is "visual" (e.g., learningStylePreference.equals ("visual")), give weight (e.g., learningStyleWeight = 0.5) to the visual learning style preference. This rule emphasizes the importance of visual aids in the learner′s educational materials.
- If the learner′s specific learning goal is "Mastering Java programming", prioritize the recommendation of content directly related to Java programming. This content may include video tutorials, interactive coding exercises, Java programming materials, and relevant resources.

The system used by Group B was selected as the most suitable choice for comparison for several key reasons: (i) The system involves the prevailing practices in the field of educational content recommendation. Many existing e-learning platforms utilize similar approaches, focusing on learner attributes such as prior knowledge, learning styles, and goals to generate personalized recommendations; (ii) by comparing our proposed approach against the system used by Group B, we ensure a realistic evaluation that aligns with the expectations and experiences of both learners and educators. This comparison provides insights into the efficacy of our novel approach within the existing educational technology landscape; (iii) by evaluating our approach against a system that solely relies on contextual signals (used by Group B), we can clarify the advantages and contributions of incorporating knowledge graph-based recommender systems enriched with cosine similarity. This comparison sheds light on how our method enhances personalized content recommendation beyond conventional approaches.

Table 1. Characteristics of the population.

Characteristics	Group A	Group B
Average age	25	26
Gender	23 female, 27 male	22 female, 28 male
Demographics	The learners come from diverse backgrounds and geographical locations.	
Computer use	Adequate background in computer-related tasks	
Prior knowledge level in computer programming	The learners share a common academic year and have successfully completed preceding programming courses.	
Motivation	The learners enrolled the Java programming course and wanted to attain a commendable grade.	

Initially, the evaluation of the system encompasses three key dimensions: user interaction, recommender system efficiency, and its impact on learning outcomes [50,51]. In light of this, a questionnaire employing a 10-point Likert scale was administered to students. This questionnaire featured two queries addressing the assessment of the first dimension, followed by three queries for gauging the second and third dimensions, respectively (as outlined in Table 2). To ascertain the questionnaire's reliability, Cronbach's alpha was applied to the sample dataset. With a calculated value of 0.93, it is evident that the scale's structure and the specific sample maintained a notably high level of internal consistency.

Table 2. Questionnaire.

Aspect	No.	Questions
User experience (UX)	Q.1	Rate the user interface of the software.
	Q.2	Rate the learning experience following your recent interaction with the software.
Recommender system effectiveness (RE)	Q.3	Did the educational materials align with your existing knowledge level?
	Q.4	Did the educational materials align with your preferred learning style?
	Q.5	Did the educational materials align with your current learning goal?
Impact on learning (IOL)	Q.6	Did you observe the software aiding in enhancing your understanding of Java programming?
	Q.7	Are you interested in utilizing this platform for other courses?
	Q.8	Would you recommend the software to your friends for their use?

Each learner provided responses to the questionnaire upon its distribution at the conclusion of the academic semester.

For a better presentation of the results, the students' responses were grouped according to the relevant aspect of the questions they concerned. Additionally, the 10-point Likert scale responses underwent conversion into three distinct classifications:

- Limited grading: spanning values from 1 to 3;
- Mediocre grading: encompassing values between 4 and 7;
- Substantial Grading: ranging from 8 to 10.

In Figure 2, the results concerning the answers of students in Groups A and B are presented.

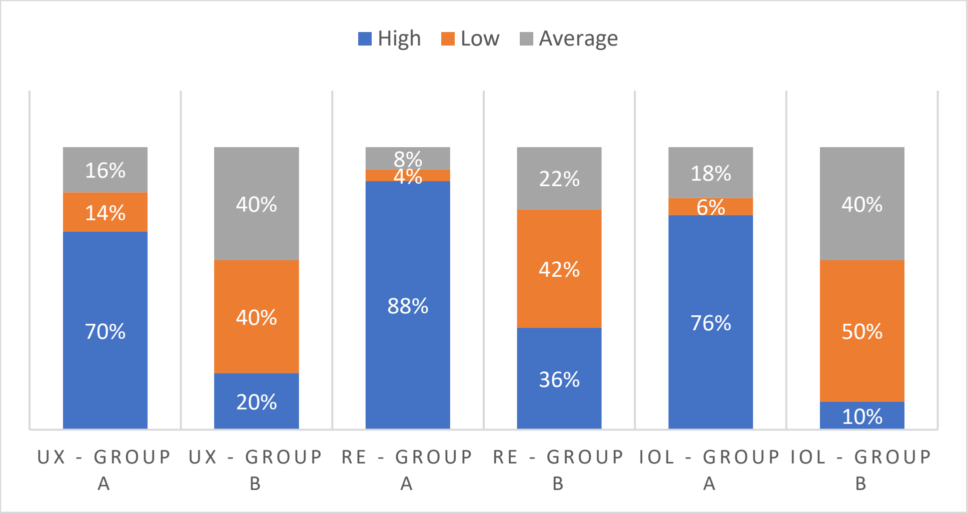

Figure 2. Questionnaire results.

In terms of the user experience facet, the outcomes from participants in Group A revealed a notably elevated rating (70%), while those from Group B yielded a more modest 20% rating. This disparity underscores the more positive experience encountered by Group A, likely attributable to the personalized and relevant learning content recommended by the presented knowledge graph-based recommender system. Furthermore, a percentage of 88% of Group A students affirmed the efficacy of the recommender system, which stands in stark juxtaposition to the meager 36% affirmation among Group B students. This variance was expected, given that our presented recommender system incorporates our novel approach utilizing a knowledge graph-based path method. This method enables sophisticated reasoning for the provision of learning activity recommendations, setting it apart from Group B's approach, which primarily relies on contextual signals. Lastly, in terms of impact on learning, Group A registered responses at 76%, while Group B's responses were limited to 10%. This difference serves to accentuate the critical significance of our knowledge graph-based recommender system, shedding light on its profound and constructive influence on the educational process.

To further explore the impact of the recommendation system on learners, a statistical hypothesis test (*t*-test) was employed to compare the innovative system (used by Group A) with its conventional counterpart (Group B). This *t*-test was executed for questions Q3–Q5, with the results of the analysis displayed in Table 3.

Table 3. The *t*-test findings.

	Q.3		Q.4		Q.5	
	Group A	**Group B**	**Group A**	**Group B**	**Group A**	**Group B**
Mean	8.2	4.98	8.88	5.16	8.64	5.24
Variance	5.510204	4.02	2.638367	10.38204	2.520816	9.32898
Observations	50	50	50	50	50	50
Pooled variance	4.765102		6.510204		5.924898	
Hypothesized mean difference	0		0		0	
df	98		98		98	
t-Stat	7.375471		7.2898		6.984068	
P(T ≤ t) one-tailed	2.66×10^{-11}		4.02×10^{-11}		1.73×10^{-10}	
P(T ≤ t) two-tailed	1.660551		1.660551		1.660551	

From the outcomes garnered, it can be deduced that there exists a statistically noteworthy distinction in the means between the two trials pertaining to Q.3, Q.4, and Q.5. To elaborate, it was observed that the software utilized by Group A exhibited a notably superior performance in suggesting educational content that aligns with students' knowledge levels compared to the conventional system used by Group B (Q.3: t-stat ≈ 7.37, $p < 0.05$). Moreover, a significant variance was found in the adequacy of the alignment with preferred learning styles (Q.4) between Group A (mean = 8.88, variance $\approx$ 2.63) and Group B (mean = 5.16, variance $\approx$ 10.38), with t-stat ≈ 7.28 and $p \approx 4.02 \times 10^{-11}$. The same divergence was observed for the degree of alignment with current learning goals (Q.5) between Group A (mean = 8.64, variance $\approx$ 2.52) and Group B (mean = 5.24, variance = 9.32), with t-stat ≈ 6.98 and $p \approx 1.73 \times 10^{-10}$.

When the literature outlined in Section 2 is analyzed, it is evident that knowledge graph-based recommender systems have been relatively underutilized in the field of e-learning. When our approach is contrasted with those previously employed in e-learning software's recommender systems, a distinctive advantage emerges. Specifically, our method integrates user attributes, educational entities, and their intricate relationships. This integration harnesses the latent potential for tailored learning content recommendations by incorporating learner attributes such as PKL, LSP, and CLG. By combining these learner characteristics, this integration has the potential to generate personalized learning content recommendations. The resulting graph structure, involving both learner attributes and educational entities, holds a broad range of personalized information, ensuring that content recommendations correspond not only to the learner's knowledge level but also to their learning style and current learning goals. This integration of learner attributes and educational entities within the graph forms a novel and impactful feature, leading to a personalized learning environment.

Finally, our approach distinguishes itself from traditional content recommendation methods, such as collaborative filtering and content-based systems, in several ways. Unlike collaborative filtering, which relies mainly on user-user interactions, our system combines knowledge graph-based recommendations and contextual signals, allowing for a more comprehensive understanding of learners' needs and preferences. Additionally, while content-based systems primarily focus on matching content attributes to user profiles, our approach takes into account the complicated relationships between learners, educational entities, and their characteristics. This rationale enables to offer tailored recommendations that go beyond simple content matching, enhancing the relevance and effectiveness of the suggested learning materials.

7. Conclusions

In conclusion, this study presents an innovative technique of recommending educational material to learners using knowledge graphs, while taking into consideration their prior knowledge level, their learning style as well as their current learning goal. The process of recommendation is further enhanced with the use of cosine similarity.

The above technique has been incorporated into an intelligent tutoring system for learning Java, and it was evaluated. Our findings underscore the significance of the presented technique. The personalized recommendations stemming from the synergy of knowledge graphs and contextual signals foster an intellectually enriching environment.

While our paper is centered around recommending Java programming content, our approach demonstrates potential applicability to various subjects and domains. The modularity of our system architecture enables the incorporation of domain-specific attributes and features, making it adaptable to different educational contexts. This adaptability extends the potential generalizability of our approach to diverse learning domains.

In the design of a recommender system, addressing potential algorithmic bias and ensuring diversity in content recommendations is of great importance. In view of the above, our system incorporates a balanced representation of educational resources to prevent overemphasis on certain materials. Also, we employ a hybrid approach that combines both knowledge graph-based recommendations and contextual signals, reducing the risk of creating "filter bubbles" by introducing diverse pathways for learners.

Our approach incorporates automated techniques for extracting relationships from educational resources, reducing the need for manual intervention. Additionally, we have designed our system architecture to accommodate scalability, allowing for the integration of larger educational datasets while ensuring efficient performance.

The presented approach holds practical implications that extend to diverse learning styles, offering adaptable learning experiences. Educators can gain valuable insights into students' progress, tailoring teaching strategies effectively. Its applicability spans disciplines, and collaboration with domain experts can fine-tune course-specific recommendations.

Limitations of this research include the need for consistent updates and upkeep of the knowledge graph due to the ever-changing nature of educational content and learner inclinations. Consequently, any deficiencies or inaccuracies within the graph's depiction may result in less-than-optimal recommendations.

Future research plans include the utilization of machine learning to automatically link entities mentioned in various sources to existing entities within the knowledge graph as well as to predict potential relationships between entities based on patterns and existing relationships within the graph.

Author Contributions: Conceptualization, C.T., A.K., P.T., D.K.K. and S.B.; methodology, C.T., A.K., P.T., D.K.K. and S.B.; software, C.T., A.K., P.T., D.K.K. and S.B.; validation, C.T., A.K., P.T., D.K.K. and S.B.; formal analysis, C.T., A.K., P.T., D.K.K. and S.B.; investigation, C.T., A.K., P.T., D.K.K. and S.B.; resources, C.T., A.K., P.T., D.K.K. and S.B.; data curation, C.T., A.K., P.T., D.K.K. and S.B.; writing—original draft preparation, C.T., A.K., P.T., D.K.K. and S.B.; writing—review and editing, C.T., A.K., P.T., D.K.K. and S.B.; supervision, C.T., A.K., P.T., D.K.K. and S.B. 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. Sundermann, C.V.; Domingues, M.A.; Sinoara, R.A.; Marcacini, R.M.; Rezende, S.O. Using Opinion Mining in Context-Aware Recommender Systems: A Systematic Review. *Information* **2019**, *10*, 42. [CrossRef]
2. López, M.B.; Alor-Hernández, G.; Sánchez-Cervantes, J.L.; Paredes-Valverde, M.A.; del Pilar Salas-Zárate, M. EduRecomSys: An Educational Resource Recommender System Based on Collaborative Filtering and Emotion Detection. *Interact. Comput.* **2020**, *32*, 407–432. [CrossRef]
3. Jena, K.K.; Bhoi, S.K.; Malik, T.K.; Sahoo, K.S.; Jhanjhi, N.Z.; Bhatia, S.; Amsaad, F. E-Learning Course Recommender System Using Collaborative Filtering Models. *Electronics* **2023**, *12*, 157. [CrossRef]
4. Deschênes, M. Recommender systems to support learners' Agency in a Learning Context: A systematic review. *Int. J. Educ. Technol. High Educ.* **2020**, *17*, 50. [CrossRef]
5. Elahi, M.; Starke, A.; El Ioini, N.; Lambrix, A.A.; Trattner, C. Developing and Evaluating a University Recommender System. *Front. Artif. Intell.* **2022**, *4*, 796268. [CrossRef] [PubMed]
6. Urdaneta-Ponte, M.C.; Mendez-Zorrilla, A.; Oleagordia-Ruiz, I. Recommendation Systems for Education: Systematic Review. *Electronics* **2021**, *10*, 1611. [CrossRef]
7. Yang, Y.; Huang, C.; Xia, L.; Li, C. Knowledge Graph Contrastive Learning for Recommendation. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'22), Association for Computing Machinery, Madrid, Spain, 11–15 July 2022; pp. 1434–1443. [CrossRef]
8. 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]
9. 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]
10. 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]
11. 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.

12. 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.
13. Gupta, M.; Thakkar, A.; Aashish; 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.
14. 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]
15. 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.
16. 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.
17. 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.
18. 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]
19. 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]
20. 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]
21. 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]
22. 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]
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. 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.
26. 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]
27. 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]
28. Angelis, S.; Kotis, K.; Spiliotopoulos, D. Semantic Trajectory Analytics and Recommender Systems in Cultural Spaces. *Big Data Cogn. Comput.* **2021**, *5*, 80. [CrossRef]
29. 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]
30. 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]
31. 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]
32. Sousa, D.; Couto, F.M. Biomedical Relation Extraction with Knowledge Graph-Based Recommendations. *IEEE J. Biomed. Health Inform.* **2022**, *26*, 4207–4217. [CrossRef] [PubMed]
33. 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]

34. 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]
35. 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]
36. Lully, V.; Laublet, P.; Stankovic, M.; Radulovic, F. Enhancing explanations in recommender systems with knowledge graphs. *Procedia Comput. Sci.* **2018**, *137*, 211–222. [CrossRef]
37. Lee, C.; Han, D.; Han, K.; Yi, M. Improving Graph-Based Movie Recommender System Using Cinematic Experience. *Appl. Sci.* **2022**, *12*, 1493. [CrossRef]
38. 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]
39. 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.
40. 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]
41. 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]
42. 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.
43. 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.
44. 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.
45. 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.
46. Hua, S.; Chen, W.; Li, Z.; Zhao, P.; Zhao, L. Path-Based Academic Paper Recommendation. In *Web Information Systems Engineering—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]
47. Troussas, C.; Krouska, A. Path-Based Recommender System for Learning Activities Using Knowledge Graphs. *Information* **2023**, *14*, 9. [CrossRef]
48. Troussas, C.; Krouska, A.; Sgouropoulou, C.; Voyiatzis, I. Ensemble Learning Using Fuzzy Weights to Improve Learning Style Identification for Adapted Instructional Routines. *Entropy* **2020**, *22*, 735. [CrossRef] [PubMed]
49. Fleming, N.D.; Mills, C. Not Another Inventory, Rather a Catalyst for Reflection. *Improv. Acad.* **1992**, *11*, 137–155. [CrossRef]
50. 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]
51. Alepis, E.; Troussas, C. M-learning programming platform: Evaluation in elementary schools. *Informatica* **2017**, *41*, 471–478.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.