applied sciences

MDPI

Article

Research on Learning Resource Recommendation Based on Knowledge Graph and Collaborative Filtering

Yanmin Niu *, Ran Lin and Han Xue

College of Computer and Information Science, Chongqing Normal University, Chongqing 401331, China; lllinran@126.com (R.L.); xuehan0228@163.com (H.X.)
* Correspondence: niuym@cqnu.edu.cn

Abstract: This study aims to solve the problem of limited learning efficiency caused by information overload and resource diversity in online course learning. We adopt a recommendation algorithm that combines knowledge graph and collaborative filtering, aiming to provide an application that can meet users' personalized learning needs and consider the semantic information of learning resources. In addition, this article collects and models implicit data in online courses and compares the impact of video and text learning resources on user learning needs under different weights in order to deeply understand the different contributions of video and text learning resources to meeting learning needs. The experimental results show that the video high-weight experimental group performs better than the text high-weight experimental group; students tend to prefer video resources. This experiment can help students cope with the challenges brought by numerous types of learning resources and provide personalized and high-quality learning experiences for learners. At the same time, adjusting and innovating teaching models for teachers has great reference value.

Keywords: recommendation system; knowledge map; collaborative filtering; implicit data

check for updates

Citation: Niu, Y.; Lin, R.; Xue, H. Research on Learning Resource Recommendation Based on Knowledge Graph and Collaborative Filtering. *Appl. Sci.* **2023**, *13*, 10933. https://doi.org/10.3390/app131910933

Academic Editor: Tobias Meisen

Received: 8 August 2023
Revised: 24 September 2023
Accepted: 28 September 2023
Published: 2 October 2023

1. Introduction

With the rapid development of artificial intelligence and the continuous promotion of digital transformation in education, technology also plays a crucial role in the field of education, as it helps to improve the quality of education [1,2]. Traditional search engines cannot meet the resource retrieval needs of students [3]. Contemporary learners are striving to solve a common dilemma: the task of identifying the most suitable learning materials among numerous online resources. Often, they find themselves overwhelmed by information overload, which hampers their learning effectiveness [4]. In the face of this 'information overload' quandary, the question of furnishing personalized learning resource recommendations has emerged as a critical concern [5]. However, high dropout rates and low retention rates have emerged in online learning environments [6,7]. In MOOC, the relevance of courses can affect the dropout rate of users [8]. Therefore, it is more necessary to intelligently recommend courses with complementary knowledge rather than alternative courses. Intelligent algorithms and technologies can filter out irrelevant learning resources and automatically recommend relevant resources to learners based on their personalized preferences, enabling them to acquire knowledge more effectively and promote personal development [9].

The need to suggest preferences in virtual environments is considered an important component of online education [10]. Learners usually have different learning needs and preferences, which can affect their learning efficiency [11,12]. This article aims to explore methods to improve the effectiveness of learning resource recommendations, provide more personalized suggestions, enhance students' learning experience, and ultimately improve learning outcomes. The inspiration for this experiment comes from the concept of "one person, one model", which reduces the burden of a single model trying to adapt to all scenes [13]. It analyzes the learning resources that are most suitable for this specific group in the current learning environment. Based on real-world conditions, we investigated which resource types are most advantageous among various learning materials and whether different resource types exhibit different learning preferences.

Recommendation algorithms, as an effective means to solve the problem of "information overload," have become a hot topic in the academic community and have been widely applied [14]. It can effectively solve the problem of user information overload, making it possible to recommend projects related to users [15]. Learning resource recommendation systems, as intelligent tools, can analyze learners' personalized characteristics and the associations between learning resources, thereby providing accurate learning recommendations to help learners acquire knowledge more efficiently. Wang et al. [16] believe that learning needs are not only related to current abilities but also to future development plans. They jointly model current abilities and future goals and, through a unique mechanism for identifying needs, obtain the development needs of different learners, thereby recommending the most suitable course. Lv et al. [17] proposed a knowledge point recommendation method based on a weighted knowledge graph, where each node represents a knowledge point and is weighted based on students' mastery level. Wan et al. [18] designed a learner impact model (LIM) to capture learners' interpersonal relationships, optimized the learner impact model using intuitionistic fuzzy logic, and found the most suitable learner through self-organization. Zhu et al. constructed a heterogeneous graph of teaching evaluation and transformed student information, interpersonal relationships, and evaluation texts into a network structure, aggregating multimodal and heterogeneous course information into a graph-based network [19]. A learning object recommendation model based on heterogeneous learning behavior and knowledge graph is proposed, which eliminates the influence of noise and obtains complex interaction behavior by using an attentive composition-based graph convolutional network (ACGCN) and dense feature-based operation-aware network (DFOAN) [20].

In conventional recommendation systems, the common practice is to utilize explicit rating data provided by users [21]. However, in the realm of online course learning, explicit rating actions are infrequent, rendering the application of explicit feedback to personalized learning resource recommendations challenging. In contrast, implicit feedback, such as user interactions like clicks and browsing behavior on items [22], can be effectively captured and analyzed. Implicit feedback can customize recommendations based on each interaction [23]. In their course activities, students leave behind traces that implicitly express their preferences. This type of data is relatively easy to gather and can be leveraged as implicit feedback. To explore the utilization of implicit data in recommendation systems, researchers have delved into various dimensions. Chen et al. [24] concentrated on implicit content within multimedia interactions and introduced a novel attention mechanism called Attentive Collaborative Filtering (ACF) into collaborative filtering algorithms to incorporate user behavior information and preferences. Zhou et al. [25] underscored the significance of user micro-behavior, which offers a deeper and more granular understanding of users. They put forth a recommendation framework for modeling the sequence and impact of micro-behavior, abbreviated as RIB. Meanwhile, Xu et al. [26] posited that learners' learning behavior can provide deeper insights into their knowledge mastery. They introduced a learning behavior-oriented knowledge tracing (LBKT) model and investigated three typical learning behaviors: speed, attempts, and hints. By analyzing these learning behaviors, it becomes possible to infer the most indicative knowledge state.

While knowledge graph-based recommendation systems excel in comprehension, they exhibit limited adaptability to the personalized needs of learners. On the other hand, collaborative filtering-based recommendation systems can provide personalized learning suggestions, but they often encounter data sparsity and cold start issues, making it difficult to obtain the best recommendations [27,28]. Researchers have uncovered that the integration of knowledge graphs into recommendation systems can enhance recommendation performance, augment interpretability, and bolster personalization [29]. Gu et al. [30]

amalgamated the RippleNet model with collaborative filtering, taking into account both movie attribute data and user-movie interaction data. This approach seeks to strike a balance between considering item attributes and user interactions. In a similar vein, Feng et al. [31], building upon collaborative filtering, crafted item knowledge graphs and user knowledge graphs to enhance information richness. This involved enriching the matrix with item feature data and extracting user preferences to enhance recommendation accuracy. Li et al. [32] integrated the knowledge graph into the task of ocean forecasting data while considering the correlation of user behavior among forecasters.

Building upon the aforementioned background, this article integrates knowledge graph information into the recommendation system by constructing a knowledge graph of the course on the basis of collaborative filtering recommendations. In addition, construct an implicit feedback model, mine students' implicit feedback data, and design different preference rules to apply to the recommendation process.

The primary contributions of this article are outlined as follows:

(1) This study integrates user learning behavior with the semantic information of learning resources, leveraging the strengths of both knowledge graphs and collaborative filtering algorithms.

(2) With the aid of students' learning records, our study centers on examining the influence of video and graphic learning resources on students. Building upon previous experiments, we delve deeply into the effects of resource types to elucidate their impact on students' learning outcomes and personalized requirements.

In the contemporary digital era, an array of learning resources is available, yet video and graphic resources stand as predominant forms. The focus in online learning environments is to recommend personalized learning materials based on the specific characteristics of learners [33,34]. This experiment aims to assess the efficacy of these two resource types in catering to diverse learning preferences among students. Such insights can aid educators in making informed decisions regarding resource selection and optimization, ultimately enhancing personalized learning experiences. Through an extensive investigation of the impact of these resource categories, this study aspires to offer actionable insights for the education sector, fostering more effective development and utilization of learning resources.

2. Methods

2.1. Collaborative Filtering Algorithm

Collaborative filtering algorithm is the most widely used recommendation technology [35], mainly divided into two methods: user-based collaborative filtering and item-based collaborative filtering. Item-based collaborative filtering focuses on the similarity between items. It analyzes user behavior data, calculates the similarity matrix between objects, and recommends other items that are similar to their past preferences to users. User-based collaborative filtering focuses on the similarity between users, searching for other users with similar preferences to the target user through their historical behavior. Based on the preferences of these similar users, recommend items that the target user may be interested in.

In recommendation systems, user feedback data are commonly categorized into explicit data and implicit data [36,37]. Explicit data are direct feedback provided by users, such as explicit ratings. On the other hand, implicit data are indirect behavioral signals generated by users in their daily activities, such as clicks, comments, and viewing history. Implicit feedback, when compared to its explicit counterpart, offers a more accurate reflection of users' actual behavior and yields richer user behavioral insights. These implicit feedback behaviors often convey users' item preferences, even in the absence of explicit ratings, enabling the inference of user interests and preferences.

2.2. The Knowledge Graph-Based Recommendation Algorithm

Knowledge graph is a knowledge representation format based on a graph structure used to capture semantic relationships between knowledge entities. It consists of interconnected entities, attributes, and relationships, typically represented as triples (h, r, t), where 'h' represents the head entity, 'r' signifies the relationship, and 't' denotes the tail entity. The foundational TransE model, pioneered by Bordes et al. [38], involves vectorizing entities and relationships in a vector space, aiming to make the vector addition of the head entity 'h' and the relationship 'r' approximately equal to the vector of the tail entity 't'. However, the TransE model exhibits limitations and is better suited for one-to-one relationships, struggling to perform well in scenarios involving one-to-many, many-to-one, or many-to-many relationships [39]. In response, subsequent models like TransH [40], TransR [41], and TransD [42] were introduced to address these shortcomings.

In typical recommendation systems, each user is associated with multiple items, which can introduce certain challenges when employing conventional models. The TransD model represents an advancement over the TransE model and specifically addresses the management of many-to-many relationships. It excels at resolving the correspondence issue between users and items. The TransD model introduces two projection matrices: one for mapping entities and another for relationships. These matrices are determined based on the specific entities and relationships in the system. The model maps entities and relationships into distinct vector spaces and employs trainable transformation matrices to transition them from one vector space to another. This innovative approach empowers the TransD model to effectively handle diverse types of relationships while preserving consistent entity distances. The structural representation of the TransD model is illustrated in Figure 1.

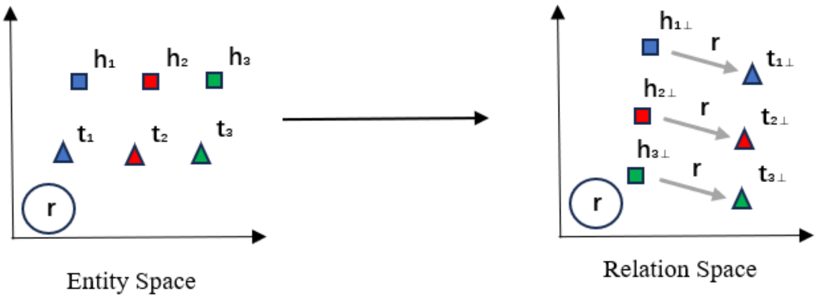


Figure 1. Structure of the TransD Model.

In this model, each head entity, relationship, and tail entity is represented by two vectors. One vector encapsulates its intrinsic semantics, while the other vector contributes to the formation of a matrix within the mapping space. The projection matrices are denoted as M_{rh} and M_{rt}, serve the purpose of projecting the head entity and tail entity from the entity space into the relation space. These projections can be derived from Equations (1) and (2), with I representing the identity matrix.

$$M_{rh} = r_p h_p^{\top} + I^{m\times n} \tag{1}$$

$$M_{rt} = r_p t_p^{\top} + I^{m\times n} \tag{2}$$

In the formula, h_p, r_p, and t_p represent the mapping vectors of each vector, and the projection matrix can consider both entities and relationships. Initialize the projection matrix by adding the identity matrix I.

$$h_{\mathrm{i}\perp} = M_{rh}h \tag{3}$$

$$t_{\mathrm{i}\perp} = M_{rt}\mathrm{t} \tag{4}$$

Using the projection matrix, the projection of the head entity vector h and tail entity vector t can be obtained from Formulas (3) and (4). Two projection matrices can be applied to the head entity vector h_{i} (i = 1, 2, 3...) and tail entity vector t_{i} (i = 1, 2, 3...) respectively, ultimately obtaining the projection vectors of each entity vector in the relational space, such as $h_{\mathrm{i}\perp}$(i = 1, 2, 3. . .) and $t_{\mathrm{i}\perp}$(i = 1, 2, 3. . .).

3. Combining Knowledge Graph and Collaborative Filtering Recommendation Algorithm

The algorithm described in this article proceeds as follows: Initially, it analyzes course information and constructs a course knowledge graph. Using the knowledge graph embedding method, it converts chapter content into vector representations and calculates the similarity between chapters. Subsequently, implicit user data are collected, and preference rules are applied to calculate chapter similarity. Finally, the algorithm combines collaborative filtering and knowledge graph similarity to achieve accurate and personalized recommendation results.

The algorithm's framework, depicted in Figure 2, consists of three main parts.

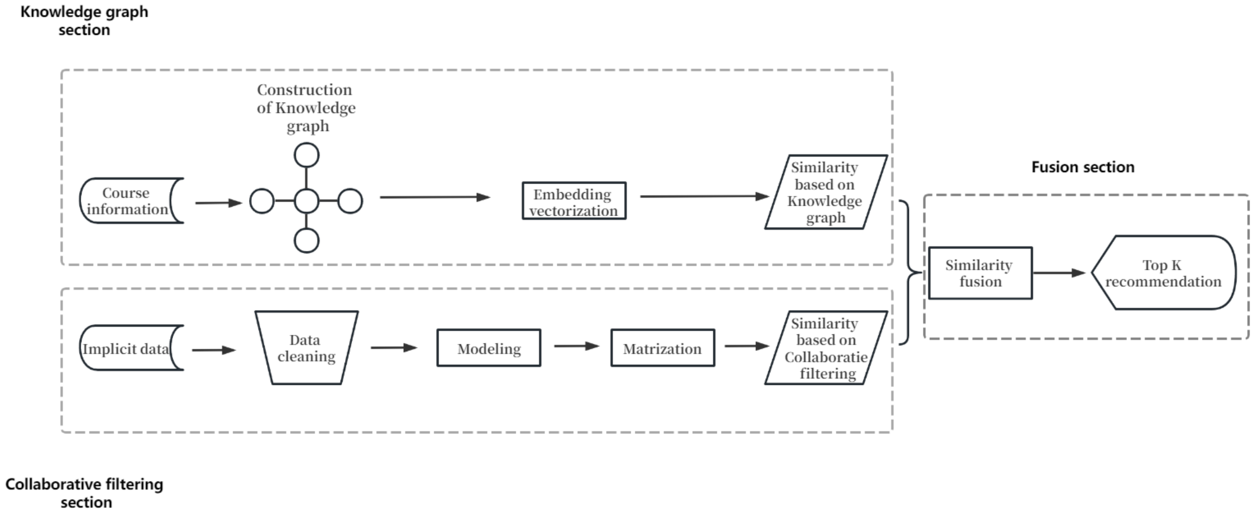


Figure 2. Algorithm framework diagram.

The first part focuses on the knowledge graph. It begins by gathering course information and constructing a knowledge graph for the course. This knowledge graph is used to vectorize chapter content through knowledge graph embedding, ultimately determining the similarity of the knowledge graph section.

The second part is dedicated to collaborative filtering. It starts by collecting implicit user data within the course and performing data cleaning to remove redundant information. An implicit feedback model is then established based on preference rules. This part concludes with a data matrix to calculate the collaborative filtering section's similarity.

The third and final part involves fusion. It combines the similarities obtained from the previous two sections, applying different weights for Top-K recommendations.

3.1. Knowledge Graph Similarity Calculation

3.1.1. Construction of Knowledge Graph

In the realm of education, the course knowledge graph stands as a significant application of knowledge graph technology in course content. Constructing the course knowledge graph enables the structured representation of course knowledge points, culminating in the formation of a cohesive knowledge system. Among the various approaches to knowledge graph construction, manual methods excel in precisely determining knowledge points and relationships. Consequently, this paper employs manual construction methods to establish the knowledge graph.

During the construction phase, data are sourced from online learning platforms, encompassing course materials such as slides, micro-videos, and text materials. These teaching materials define the course structure, organizing the knowledge graph into three layers. The first layer, denoted as 'L,' represents course modules and is established based on the selected courses. The second layer, 'C,' signifies chapter modules and is structured around the chapter names extracted from online courses. The third layer, 'K,' represents knowledge point modules. Each chapter necessitates a clear definition of relevant knowledge points to ensure accuracy. Within the course resources, fundamental chapters and knowledge points exhibit clear logical order relationships. Based on the course's characteristics and content, five types of relationships—namely "belongs to, associates with, precedes, contains, and synonym"—are defined. Figure 3 illustrates the conceptual structure of the knowledge graph.

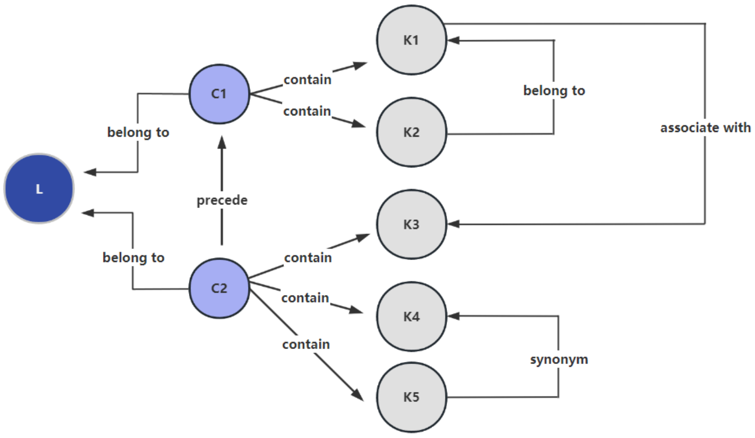


Figure 3. Structure concept map of knowledge graph.

Knowledge extraction constitutes the central step in the manual construction of the course knowledge graph. Entities and relationships are meticulously extracted, guided by predefined knowledge points. In the entity extraction phase, a thorough review of video and text resources from each chapter is conducted to identify concepts closely related to the chapter's content. Knowledge point entities are then extracted and linked to the previously defined knowledge points, encompassing concepts such as "process" and "scheduling". In the relationship extraction phase, knowledge graph relationships among knowledge points are deduced based on their logical connections. For instance, the knowledge point "scheduling algorithm" may entail a predecessor relationship, such as the "scheduling concept". To ensure the precision of the knowledge system, knowledge extraction encompasses the acquisition of knowledge points from various textbooks and data sources. Due to the disparities among sources, the same knowledge point entity may exhibit variations, necessitating entity alignment to eliminate ambiguities. For instance, "process control" in one textbook and "process management" in another can be harmonized into a standardized entity: "process control".

3.1.2. Knowledge Graph Representation Learning

In this paper, the training of the knowledge graph utilizes the TransD model. The TransD model employs a margin-based ranking loss function to compute the distance between real relationships and vector representations. It employs the backpropagation algorithm to iteratively update the vector representation values. This iterative process aims to attain optimal vector representations. The loss function for this model is expressed as follows:

$$L = \sum_{\varepsilon \epsilon \Delta} \sum_{\varepsilon' \epsilon \Delta'} \left[\gamma + f_r(\varepsilon') - f_r(\varepsilon)\right]_+ \tag{5}$$

ε represents positive triplets and ε' represents the corresponding negative triplets. Δ is the set of positive triplets and Δ' is the set of negative triplets. γ is the margin value that separates positive and negative triplets. $[\gamma + f_r(\varepsilon') - f_r(\varepsilon)]_+$ represents setting elements less than 0 to zero while keeping the values of elements greater than 0 unchanged.

Based on the pre-constructed knowledge graph triplets, such as <Chapter A, Precedes, Chapter B> and <Chapter B, Contains, Knowledge Point C>, the entities in the knowledge graph are embedded into a high-dimensional vector space. For each entity and relationship, two vectors are initialized, one in the entity space representing the entity's position and another in the relationship space representing the relationship's position. A loss function is defined, and the stochastic gradient descent algorithm is used to minimize the loss function. Through batch training, vector representations for each chapter entity are obtained. Next, the entity vectors are used to calculate the similarity between entities. Different similarity metrics can be used for similarity calculation. In this paper, cosine similarity is chosen as the similarity calculation method. It measures the similarity by calculating the cosine value of the angle between two vectors. The calculation formula is as follows (Equation (6)):

$$cos(x,y) = \frac{x \cdot y}{\|x| \times |y\|} \tag{6}$$

After obtaining the similarity between vectors, the semantic vectors obtained from the knowledge graph can be transformed into a similarity matrix. The chapter set is defined as $Y = \{Y_1, Y_2, Y_3, \ldots, Y_v\}$.

Indeed, when the cosine similarity approaches 1, it signifies that the directions of the two vectors are nearly identical, and the angle between them is close to 0 degrees. This implies that they can be regarded as highly similar. The notation $simA(x,y)$ represents the similarity between chapters achieved through the knowledge graph method.

3.2. Collaborative Filtering Similarity Calculation

3.2.1. Implicit Feedback Model

This paper builds an implicit feedback model based on learning information from online courses. The learning information includes the user's video-watching progress and text-reading progress for each chapter. According to this feature, it assumes that user i has a preference value V_{ij} for the video content of chapter j and a preference value P_{ij} for the text content. In the implicit feedback, the overall preference value M_{ij} for this chapter is calculated using V_{ij} and P_{ij}. The formula is defined as follows:

In this paper, an implicit feedback model is constructed by utilizing learning information obtained from online courses. This learning information includes the user's progress in watching videos and reading text materials for each chapter. Given this context, it assumes that user i holds a preference value V_{ij} for the video content of chapter j and a preference value P_{ij} for the text content of chapter j. Within the framework of implicit feedback, the combined preference value M_{ij} for this chapter is computed by considering both V_{ij} and P_{ij}. The formula for this calculation is defined as follows:

$$V_{ij} = \frac{N_v}{A_v} \tag{7}$$

$$P_{ij} = \frac{N_p}{A_p} \tag{8}$$

$$M_{ij} = V_{ij} * r_v + P_{ij} * r_p \tag{9}$$

In the formula, N_v represents the total duration of videos that user i has watched in chapter j, while N_p represents the total duration of text content that user i has studied in chapter j. A_v signifies the total duration of videos available in chapter j, and A_p denotes the total amount of text content within chapter j. Furthermore, r_v represents the predetermined preference ratio for video content while r_p indicates the predetermined preference ratio for text content.

In the information age, video and text formats constitute the primary multimedia learning resources. These multimedia formats integrate visual, auditory, and textual components to offer learners vivid, intuitive, and diverse learning experiences, which exert a profound influence on learning outcomes. To assess the significance of these two formats in learning resource recommendations, this paper formulates the following preference rules:

(1) Video Dominance

In this configuration, the weight r_v for video resources is established as 1, while the weight r_p for text resources is designated as 0.5. This weight assignment is intended to underscore the significance and relatively higher priority attributed to video resources in learning resource recommendations.

(2) Text Dominance

In this configuration, the weight r_p for text resources is defined as 1, while the weight r_v for video resources is assigned a value of 0.5. This weight arrangement is intended to highlight the significance and relatively higher priority assigned to text resources in learning resource recommendations.

3.2.2. Item based Collaborative Filtering

Assuming there are u users and v chapters in the dataset. The user set can be represented as $X = \{X_1, X_2, X_3, \ldots, X_u\}$, and the chapter set can be represented as $Y = \{Y_1, Y_2, Y_3, \ldots, Y_v\}$. Based on Equation (9) and the preference rules set in this paper, we can form the user behavior matrix $M_{\mathrm{u\times v}}$.

In Table 1, the user behavior matrix, M_{ij} represents the user i's preference level for chapter j. For example, if user i has a video learning progress of 70% and a text learning progress of 20% in chapter j. When using the video-dominant preference rule, $M_{ij} = 0.7 * 1 + 0.2 * 0.5 = 0.8$. When using the text-dominant preference rule, $M_{ij} = 0.7 * 0.5 + 0.2 * 1 = 0.55$. Under the two preference rules, different user behavior matrices are formed.

Table 1. User behavior matrix.

	X_1	X_2	$X_3,$	…	X_u
Y_1	M_{11}	M_{21}	M_{31}	…	M_{u1}
Y_2	M_{12}	M_{22}	M_{32}	…	M_{u2}
Y_3	M_{13}	M_{23}	M_{33}	…	M_{u3}
…	…	…	…	…	…
Y_v	M_{1v}	M_{2v}	M_{3v}	…	M_{uv}

In item-based collaborative filtering, using the user behavior matrix, recommendations are made based on the user's historical behavior. To achieve this, the similarity between chapters is computed using cosine similarity, resulting in the similarity matrix for the collaborative filtering method. When the similarity approaches 1, it signifies a higher frequency of occurrence or a stronger connection between these two chapters in user behavior data. This suggests that users tend to select these two chapters together. The notation $simB(x,y)$ represents the chapter-to-chapter similarity obtained through the collaborative filtering algorithm.

3.3. Similarity Fusion

In the similarity fusion method, linear models are favored for their interpretability and scalability. Utilizing a linear model allows for an intuitive understanding of the influence of various features on the recommendation results. Hence, this paper opts for a linear model. The similarity obtained from the knowledge graph, $simA(x,y)$, is integrated with the similarity derived from item-based collaborative filtering, $simB(x,y)$, through a weighted fusion approach. The optimal outcome is attained by iteratively adjusting the weights. The computational formula is presented in Equation (10):

$$sim = (1-\alpha) \cdot simA(x,y) + \alpha \cdot simB(x,y) \tag{10}$$

The parameter α signifies the fusion weight ratio, with $\alpha \in [0, 1]$ governing the balance between item-based collaborative filtering and TransD in the fusion process. By varying the value of α, we can assess how different fusion ratios influence the performance of the recommendation algorithm and identify the optimal α value. Once the similarity results are obtained, a Top-K sorting is conducted to produce the final recommendation list.

4. Analysis of Experimental Data and Results

4.1. Experimental Dataset

The data used in this study are collected from the learning resource data and learner learning behavior records within the municipal high-quality course "Introduction to Computer" on the Chongqing Smart Education Platform. This dataset includes both video and text resources for each chapter, as well as the learning progress of each user.

After extracting concepts and establishing relationships based on the content of the course chapters and knowledge points, the extracted concepts and established relationships are integrated into the knowledge graph. Figure 4 provides a visual representation of the knowledge graph for the course.

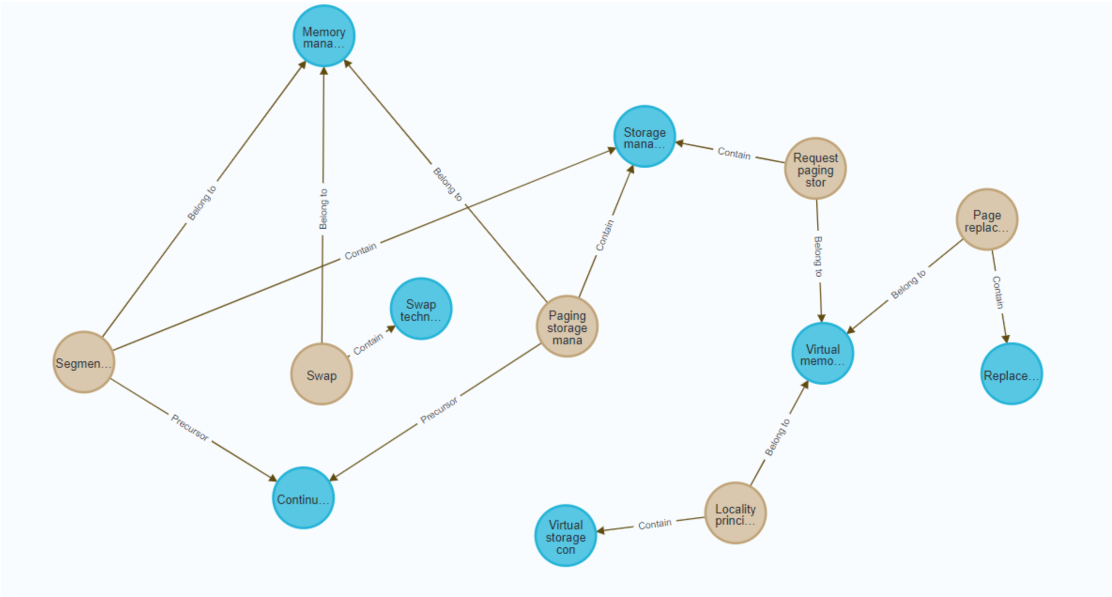


Figure 4. Knowledge graph of course.

4.2. Evaluating Indicator

In this study, courses chosen by users were categorized as positive, whereas courses not selected were deemed negative. The system's performance is assessed using Precision, Recall, and F1 scores. The parameters within the confusion matrix can be employed to elucidate Precision and Recall. The confusion matrix is presented in Table 2.

Table 2. Error matrix.

	User Selected	**User not Selected**
Successfully recommended	TP	FP
Failed recommendation	FN	TN

Accuracy represents the proportion of courses that students actually choose among all recommended courses.

The recall rate represents the proportion of courses successfully recommended by the recommendation system among all students' actual choices.

The F1 value considers both to make the recommended results more balanced.

4.3. Fusion Proportion Experiment and Result Analysis

In this experiment, collaborative filtering and knowledge graphs are integrated, with their contributions weighted by the parameter α, which governs the balance between these two components in the recommendation process. The fusion ratio α is a variable used to comprehensively evaluate recommendation performance under various fusion ratios. To ensure the rigor of the experimental outcomes, two sets of experimental designs are employed, aligned with the preference rules outlined in this study. This approach ensures that collaborative filtering and knowledge graph results, generated under the two sets of preference rules, are weighted separately to yield the experimental evaluation metrics. The results are subsequently subjected to detailed analysis and comparison.

In both sets of experiments, the value of K is set to 16 for Top-K recommendations, and the embedding dimension of TransD is set to 100. The range of α is [0, 1], starting from 0 with a step size of 0.1, gradually increasing to 1. Each set of experiments is repeated 10 times, and the average values of each metric are calculated. The experimental results under different fusion weights α are shown in Table 3.

Table 3. Fusion ratio experimental results.

α	Video Dominance			Text Dominance		
	Precison	Recall	F1	Precison	Recall	F1
$\alpha = 0$	0.5567	0.2788	0.3716	0.5568	0.2788	0.3716
$\alpha = 0.1$	0.6300	0.3646	0.4619	0.6288	0.3617	0.4592
$\alpha = 0.2$	0.6823	0.4157	0.5166	0.6802	0.4061	0.5086
$\alpha = 0.3$	0.7062	0.4381	0.5407	0.6940	0.4301	0.5311
$\alpha = 0.4$	0.7355	0.4753	0.5774	0.7274	0.4636	0.5663
$\alpha = 0.5$	0.7564	0.5278	0.6217	0.7490	0.5138	0.6095
$\alpha = 0.6$	0.7795	0.5473	0.6431	0.7759	0.5399	0.6367
$\alpha = 0.7$	0.7672	0.5183	0.6187	0.7607	0.5052	0.6072
$\alpha = 0.8$	0.7629	0.5036	0.6067	0.7554	0.4959	0.5988
$\alpha = 0.9$	0.7572	0.4913	0.5959	0.7514	0.4861	0.5903
$\alpha = 1$	0.7507	0.4776	0.5838	0.7471	0.4717	0.5783

Upon analyzing the experimental results within the video-dominant group, it becomes evident that precision, recall, and F1-score exhibit an upward trend as the fusion weight ratio α increases, with video content being given higher priority. The three performance metrics reach their peak values when $\alpha = 0.6$, after which they begin to gradually decline.

Similarly, from the analysis of the text-dominant group results, the performance metrics also show an increasing-then-decreasing trend. When $\alpha = 0.6$, the experimental performance is optimal.

By comprehensively analyzing the results from both groups, it can be concluded that under the conditions of $K = 16$ and the embedding dimension of 100, selecting $\alpha = 0.6$, which corresponds to a fusion ratio of 6:4, yields the best recommendation performance.

4.4. Algorithm Comparison

In order to verify the practicality of the methods used in this article, four other recommended methods were selected for comparison, namely TransD, ItemCF, CKE, and KGAT. In order to ensure the comprehensiveness of the experimental comparison, the baseline method selected in this article includes traditional recommendation algorithms and more advanced recommendation algorithms. Its introduction is as follows:

(1) TransD [42]: A model used for knowledge graph representation learning aimed at capturing semantic associations between entities and relationships.

(2) ItemCF [43] (item-based collaborative filtering): Object-based collaborative filtering that analyzes items that users have liked in the past and recommends other items that are similar to these items.

(3) CKE [44] (Collaborative Knowledge Base Embedding): Based on knowledge graph embedding, a multi-layer neural network is used to integrate the knowledge base.
(4) KGAT [45] (Knowledge Graph Attention Network): It combines graph convolutional networks and attention mechanisms to capture complex associations in the knowledge graph.

ItemCF-Text represents ItemCF using text dominant preference rules in this article. ItemCF-Video represents ItemCF using video dominant preference rules in this article. In the CKE algorithm, the interaction event that occurs in the dataset is denoted as 1; otherwise, it is denoted as 0. In the KGAT algorithm, each user's interaction with an item is considered a positive instance. The method used in this article α Set to 0.6

The experimental results are shown in Table 4.

Table 4. Experimental comparison results.

Model	*F*1@13	*F*1@14	*F*1@15	*F*1@16	*F*1@17	*F*1@18
TransD	0.3048	0.3291	0.3497	0.3716	0.3909	0.4104
ItemCF-Text	0.4383	0.4859	0.5315	0.5783	0.5879	0.6008
ItemCF-Video	0.4596	0.4918	0.5437	0.5808	0.5902	0.6086
CKE	0.4729	0.5135	0.5511	0.6077	0.6193	0.6322
KGAT	0.5088	0.5451	0.5842	0.6224	0.6476	0.6596
Text Dominance	0.5546	0.5858	0.6089	0.6367	0.6638	0.6785
Video Dominance	0.5734	0.6005	0.6193	0.6431	0.6637	0.6877

Analyzing the experimental results can lead to the following conclusions:

(1) Compared with other models, the method proposed in this paper shows a certain improvement in performance. This indicates that the method used in this article relies more on the specific interaction data during the learning process. By introducing implicit feedback models, this article can better capture the feature representations of students' learning progress and resource types, especially when setting weights for different resource types. This means that we have a better understanding of student's needs and learning environments and can more accurately recommend appropriate learning resources for them.
(2) Compared to the KGAT and CKE methods, the method proposed in this paper slightly improves the F1 value. This may be because the task of this article focuses more on considering the interaction between users and resources, which can more easily adapt to the set preference rules. In addition, the complexity of the method proposed in this article is relatively low, which may also contribute to performance improvement.
(3) Compared with ItemCF and TransD, regardless of the K value, the mixed method exhibits better performance than using either method alone. This emphasizes the suitability of combining these two methods for recommendation in this article, providing better performance for recommendation systems.

In summary, in this experiment, a hybrid method combining implicit feedback models, knowledge graphs, and collaborative filtering has been proven to be more suitable and suitable for the current experimental environment. This method takes into account students' learning preferences and fully utilizes existing learning interaction data, providing students with more intelligent and practical learning resources.

4.5. Preference Comparison Experiment and Result Analysis

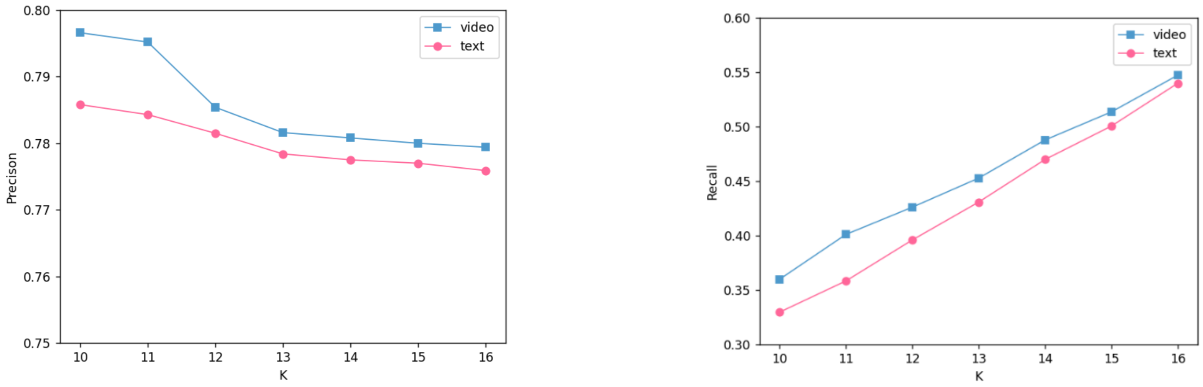
To investigate the influence of different types of learning resources in the recommendation process, we conducted a comparison between the video-dominant and text-dominant scenarios while maintaining the fusion ratio determined in the previous experiments. Within the specified fusion ratio condition, Top-K recommendations were performed to assess the impact of the two resource types. We also varied the value of K to compare experimental results across different recommendation quantities. In both sets of experiments,

K ranged from 10 to 16, α was set to 0.6, and TransD's embedding dimension remained at 100. Each experiment was repeated 10 times, and the average values were computed. The results for different values of K are detailed in Table 5.

Table 5. Preference comparison experiment results.

K	Video Dominance			Text Dominance		
	Precison	Recall	F1	Precison	Recall	F1
K = 10	0.7966	0.3599	0.4958	0.7858	0.3297	0.4645
K = 11	0.7952	0.4011	0.5332	0.7843	0.3584	0.4919
K = 12	0.7854	0.4261	0.5525	0.7815	0.3960	0.5256
K = 13	0.7816	0.4528	0.5734	0.7784	0.4308	0.5546
K = 14	0.7808	0.4878	0.6005	0.7775	0.4699	0.5858
K = 15	0.7800	0.5136	0.6193	0.7770	0.5006	0.6089
K = 16	0.7794	0.5473	0.6430	0.7759	0.5399	0.6367

Figure 5a,b show the precision and recall of the experiment, respectively.



(**a**) (**b**)

Figure 5. (**a**) Comparison of precision under different preference rules; (**b**) Comparison of recall rates under different preference rules.

Based on the above results, we can observe that as the K value increases, the accuracy under the two preference rules shows a decreasing trend, while the recall rate shows an increasing trend. Increasing the k value usually increases the diversity of recommendation results. This means that the system is more inclined to recommend various types of content to meet the diverse needs that users may have, which can improve recall rates. But, capturing more related projects may also lead to the introduction of unrelated projects, thereby reducing accuracy. The comparison of the results shown in the two images shows that assigning high-weight values to video resources has achieved good results in recommendation performance, both in terms of accuracy and recall. This indicates that compared to text resources, video resources play a more important role in course recommendation, and in this course, video resources can better meet users' learning needs and preferences.

In the experiment in Figure 5a, a sharp decrease in the accuracy curve was observed when the recommended quantity K increased from 11 to 12. When increasing the k value, the system may have already considered most of the relevant courses. When more course options are introduced, there may be differences in the relevance between these additional courses and the specific needs of the user. This sharp decrease may be due to one or more additional recommended courses not matching the user's interests, resulting in a decrease in accuracy. Meanwhile, when increasing the number of recommendations, users may face information overload issues as they need to evaluate more course options. This may make it more difficult for users to make wise choices, thereby affecting accuracy. However, as the experiment progresses, the accuracy curve stabilizes. Users gradually adapt to the recommendations provided by the system and begin to provide positive feedback on more courses suggested by the system.

In future education and teaching, teachers can harness technology and educational innovations to integrate multimedia resources and online learning platforms into their teaching methodologies. Emphasizing the use of video resources can offer students immersive and engaging educational content and experiences. Schools can provide the necessary technical support and training to assist teachers in effectively incorporating video resources into their teaching methods. This approach not only enhances students' personalized learning experiences but also boosts the effectiveness of teaching for educators.

5. Conclusions

The goal of this study is to deeply explore the hidden resources present in online courses and use a recommendation algorithm that combines knowledge graph representation learning with collaborative filtering. It constructs a course knowledge map and utilizes the knowledge map embedding method to perform weighted fusion with item-based collaborative filtering algorithms. Nowadays, many online learning platforms provide a large amount of learning resources, but learners often find it difficult to find resources that are suitable for their needs and learning styles. This issue is considered one of the bottlenecks in online learning, and in previous learning resource recommendations, the impact of different resource types has not been explored. In order to optimize learning resource recommendation strategies and improve learners' learning outcomes and needs, two preference rules, video-dominant and text-dominant, were set up to study their impact on recommendation results. A key finding is that there are differences in the impact of different types of learning resources on learners. The experimental results show that the higher the video weight, the better the results, highlighting the importance of video resources in learners' learning process. The importance of research lies not only in addressing the resource recommendation problem in online learning but also in its potential impact. In the field of education, this study provides valuable insights for educators, encouraging them to adopt video resources more widely and actively explore innovative teaching methods. Teachers can consider emphasizing the production and use of video content to provide a richer learning experience.

Although progress has been made in this study, there are still some shortcomings. Currently, there are challenges in the relevance of user needs and their information processing capabilities. When the diversity of user needs is high, some less relevant courses may be taken into account. At the same time, the course evaluation mechanism also needs to be improved, and more user feedback data needs to be added to reduce the recommendation of lower quality or unevaluated courses. In future work, a course screening mechanism will be added to eliminate low-quality or poorly evaluated courses in order to improve the quality of the recommendation list. Introduce user feedback, such as ratings or comments, to further improve recommendations. The experiment will be conducted on larger and more representative datasets, such as MOOCCube, with a focus on exploring the feasibility of the algorithm [46].

Author Contributions: Conceptualization, form analysis, methodology, supervision, visualization, writing—review and editing, Y.N. Conceptualization, form analysis, software, methodology, visualization, writing—review and editing, R.L. Methodology, supervision, writing—review and editing, H.X. All authors have read and agreed to the published version of the manuscript.

Funding: Chongqing Normal University 2023 Higher Education Teaching Reform Research Project (202344).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Dahdouh, K.; Dakkak, A.; Oughdir, L.; Ibriz, A. Large-scale e-learning recommender system based on Spark and Hadoop. *J. Big Data* **2019**, *6*, 2. [CrossRef]
2. Fu, Q.K.; Lin, C.J.; Hwang, G.J.; Zhang, L. Impacts of a mind mapping-based contextual gaming approach on EFL students' writing performance, learning perceptions and generative uses in an English course. *Comput. Educ.* **2019**, *137*, 59–77. [CrossRef]
3. Cui, L.Z.; Guo, F.L.; Liang, Y.J. Research overview of educational recommender systems. In Proceedings of the 2nd International Conference on Computer Science and Application Engineering, Hohhot, China, 22–24 October 2018; pp. 1–7.
4. Lee, Y.; Choi, J. A review of online course dropout research: Implications for practice and future research. *Educ. Technol. Res. Dev.* **2011**, *59*, 593–618. [CrossRef]
5. Kong, L.; Ma, H.; Peng, Y.; Tang, W. Hybrid Recommendation of Personalized MOOC Resources: A User Context-aware Approach. In Proceedings of the 2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Hangzhou, China, 4–6 May 2022; pp. 537–542.
6. Chiappe-Laverde, A.; Hine, N.; Martínez-Silva, J.A. Literature and Practice: A Critical Review of MOOCs. *Comunicar* **2015**, *22*, 9–17. [CrossRef]
7. Martín-Monje, E.; Castrillo, M.D.; Mañana-Rodríguez, J. Understanding online interaction in language MOOCs through learning analytics. *Comput. Assist. Lang. Learn.* **2018**, *31*, 251–272. [CrossRef]
8. Feng, W.; Tang, J.; Liu, T.X. Understanding dropouts in MOOCs. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 517–524. [CrossRef]
9. Lu, J.; Wu, D.; Mao, M.; Wang, W.; Zhang, G. Recommender system application developments: A survey. *Decis. Support Syst.* **2015**, *74*, 12–32. [CrossRef]
10. Pecori, R. A virtual learning architecture enhanced by fog computing and big data streams. *Future Internet* **2018**, *10*, 4. [CrossRef]
11. Buder, J.; Schwind, C. Learning with personalized recommender systems: A psychological view. *Comput. Hum. Behav.* **2012**, *28*, 207–216. [CrossRef]
12. Joy, J.; Raj, N.S. An ontology model for content recommendation in personalized learning environment. In Proceedings of the Second International Conference on Data Science, E-Learning and Information Systems, Dubai, United Arab Emirates, 2–5 December 2019; pp. 1–6.
13. Liu, Z.; Cheng, M.; Li, Z.; Liu, Q.; Chen, E. One Person, One Model—Learning Compound Router for Sequential Recommendation. In Proceedings of the 2022 IEEE International Conference on Data Mining (ICDM), Orlando, FL, USA, 28 November–1 December 2022; pp. 289–298.
14. Huang, L.; Jiang, B.; Lv, S.; Liu, Y.; Li, D. A Review of Recommender Systems Based on Deep Learning. *J. Comput. Sci.* **2018**, *7*, 1619–1647.
15. Bollen, D.; Knijnenburg, B.P.; Willemsen, M.C.; Graus, M. Understanding choice overload in recommender systems. In Proceedings of the Fourth ACM Conference on Recommender Systems, Barcelona, Spain, 26–30 September 2010; pp. 63–70.
16. Wang, C.; Zhu, H.; Zhu, C.; Zhang, X.; Chen, E.; Xiong, H. Personalized employee training course recommendation with career development awareness. In Proceedings of the Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; pp. 1648–1659.
17. Lv, P.; Wang, X.; Xu, J.; Wang, J. Intelligent personalised exercise recommendation: A weighted knowledge graph-based approach. *Comput. Appl. Eng. Educ.* **2021**, *29*, 1403–1419. [CrossRef]
18. Wan, S.; Niu, Z. A hybrid e-learning recommendation approach based on learners' influence propagation. *IEEE Trans. Knowl. Data Eng.* **2019**, *32*, 827–840. [CrossRef]
19. Zhu, Y.; Lu, H.; Qiu, P.; Shi, K.; Chambua, J.; Niu, Z. Heterogeneous teaching evaluation network based offline course recommendation with graph learning and tensor factorization. *Neurocomputing* **2020**, 84–95. [CrossRef]
20. Zhu, Y.; Lin, Q.; Lu, H.; Shi, K.; Liu, D.; Chambua, J.; Wan, S.; Niu, Z. Recommending learning objects through attentive heterogeneous graph convolution and operation-aware neural network. *IEEE Trans. Knowl. Data Eng.* **2021**. [CrossRef]
21. Pereira, M.; Etemad, E.; Paulovich, F. Iterative learning to rank from explicit relevance feedback. In Proceedings of the 35th Annual ACM Symposium on Applied Computing, Brno, Czech Republic, 30 March–3 April 2020; pp. 698–705.
22. Wen, H.; Yang, L.; Estrin, D. Leveraging post-click feedback for content recommendations. In Proceedings of the 13th ACM Conference on Recommender Systems, Copenhagen, Denmark, 16–20 September 2019; pp. 278–286.
23. Himeur, Y.; Alsalemi, A.; Al-Kababji, A.; Bensaali, F.; Amira, A.; Sardianos, C.; Dimitrakopoulos, G.; Varlamis, I. A survey of recommender systems for energy efficiency in buildings: Principles, challenges and prospects. *Inf. Fusion* **2021**, *72*, 1–21. [CrossRef]
24. Chen, J.; Zhang, H.; He, X.; Nie, L.; Liu, W.; Chua, T.S. Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan, 7–11 August 2017; pp. 335–344.
25. Zhou, M.; Ding, Z.; Tang, J.; Yin, D. Micro behaviors: A new perspective in e-commerce recommender systems. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, Marina Del Rey, CA, USA, 5–9 February 2018; pp. 727–735.
26. Xu, B.; Huang, Z.; Liu, J.; Shen, S.; Liu, Q.; Chen, E.; Wu, J.; Wang, S. Learning Behavior-oriented Knowledge Tracing. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Long Beach, CA, USA, 6–10 August 2023; pp. 2789–2800.

27. Sharif, N.; Afzal, M.T. Recommendation approaches for e-learners: A survey. In Proceedings of the 7th International Conference on Management of computational and collective intElligence in Digital EcoSystems, Caraguatatuba, Brazil, 25–29 October 2015; pp. 137–141.
28. Klašnja-Milićević, A.; Ivanović, M.; Vesin, B.; Budimac, Z. Enhancing e-learning systems with personalized recommendation based on collaborative tagging techniques. *Appl. Intell.* **2018**, *48*, 1519–1535. [CrossRef]
29. Xian, Y.; Fu, Z.; Muthukrishnan, S.; De Melo, G.; Zhang, Y. Reinforcement knowledge graph reasoning for explainable recommendation. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 285–294.
30. Gu, Y.; Zhang, Y.; Yang, H. Movie recommendation algorithm based on movie attributes and interaction information. *J. Nanjing Univ. Sci. Technol.* **2022**, *46*, 177–184.
31. Feng, X.; Yang, Q. Collaborative filtering algorithm with information enhancement based on knowledge graph. *J. Jiangxi Norm. Univ. Nat. Sci.* **2022**, *04*, 386–393.
32. Li, Z.; Gao, D.; Liu, X.; Wu, J. Knowledge Graph-Based Ocean Numerical Forecast Data Recommendation Algorithm. *Comput. Eng. Des.* **2023**, *05*, 1385–1391.
33. Essalmi, F.; Ayed LJ, B.; Jemni, M.; Graf, S. A fully personalization strategy of E-learning scenarios. *Comput. Hum. Behav.* **2010**, *26*, 581–591. [CrossRef]
34. Dwivedi, P.; Bharadwaj, K.K. e-Learning recommender system for a group of learners based on the unified learner profile approach. *Expert Syst.* **2015**, *32*, 264–276. [CrossRef]
35. Alharbe, N.; Rakrouki, M.A.; Aljohani, A. A collaborative filtering recommendation algorithm based on embedding representation. *Expert Syst. Appl.* **2023**, *215*, 119380. [CrossRef]
36. Aljunid, M.F.; Dh, M. An efficient deep learning approach for collaborative filtering recommender system. *Procedia Comput. Sci.* **2020**, *171*, 829–836. [CrossRef]
37. Mandal, S.; Maiti, A. Explicit feedback meet with implicit feedback in GPMF: A generalized probabilistic matrix factorization model for recommendation. *Appl. Intell.* **2020**, *50*, 1955–1978. [CrossRef]
38. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 2787–2795.
39. Wang, H.; Lun, B.; Zhang, X.; Sun, X. Multi-modal entity alignment based on joint knowledge representation learning. *Control. Decis.* **2020**, *35*, 2855–2864.
40. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge graph embedding by translating on hyperplanes. In Proceedings of the AAAl Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014; Volume 28.
41. Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X. Learning entity and relation embeddings for knowledge graphcompletion. In Proceedings of the AAAl Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; Volume 29.
42. Ji, G.; He, S.; Xu, L.; Liu, K.; Zhao, J. Knowledge graph embedding via dynamic mapping matrix. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long papers); Association for Computational Linguistics: Beijing, China, 2015; pp. 687–696.
43. Bellogín, A.; Cantador, I.; Díez, F.; Castells, P.; Chavarriaga, E. An empirical comparison of social, collaborative filtering, and hybrid recommenders. *ACM Trans. Intell. Syst. Technol. TIST* **2013**, *4*, 14. [CrossRef]
44. Zhang, F.; Yuan, N.J.; Lian, D.; Xie, X.; Ma, W.Y. Collaborative knowledge base embedding for recommender systems. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 353–362.
45. Wang, X.; He, X.; Cao, Y.; Liu, M.; Chua, T.S. Kgat: Knowledge graph attention network for recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 950–958.
46. Yu, J.; Luo, G.; Xiao, T.; Zhong, Q.; Wang, Y.; Feng, W.; Luo, J.; Wang, C.; Hou, L.; Li, J.; et al. MOOCCube: A large-scale data repository for NLP applications in MOOCs. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics; Association for Computational Linguistics: Beijing, China, 2020; pp. 3135–3142.

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.