

Article

Graph-Community-Enabled Personalized Course-Job Recommendations with Cross-Domain Data Integration

Guoqing Zhu *, Yan Chen and Shutian Wang

School of Maritime Economics and Management, Dalian Maritime University, Dalian 116026, China; chenyan_dlm@163.com (Y.C.); wangshutian77@163.com (S.W.)

* Correspondence: guoqing9677521@126.com

Abstract: With millions of students/employees browsing course information and job postings every day, the need for accurate, effective, meaningful, and transparent course and job recommender systems is more evident than ever. The current recommendation research has attracted wide attention in the academic and industrial areas. However, existing studies primarily focus on content analysis and user feature extraction of courses or jobs and fail to investigate the problem of cross-domain data integration between career and education. At the same time, it also fails to fully utilize the relations between courses, skills, and jobs, which helps to improve the accuracy of the recommendation. Therefore, this study aims to propose a novel cross-domain recommendation model that can help students/employees search for suitable courses and jobs. Employing a heterogeneous graph and community detection algorithm, this study presents the Graph-Community-Enabled (GCE) model that merges course profiles and recruiting information data. Specifically, to address the skill difference between occupation and curriculum, the skill community calculated by the community detection algorithm is used to connect curriculum and job information. Then, the innovative heterogeneous graph approach and the random walk algorithm enable cross-domain information recommendation. The proposed model is evaluated on real job datasets from recruitment websites and the course datasets from MOOCs and higher education. Experiments show that the model is obviously superior to the classical baselines. The approach described can be replicated in a variety of education/career situations.

Keywords: education; career; heterogeneous data/heterogeneous graph mining; information recommendation; cross-domain



Citation: Zhu, G.; Chen, Y.; Wang, S. Graph-Community-Enabled Personalized Course-Job Recommendations with Cross-Domain Data Integration. *Sustainability* **2022**, *14*, 7439. <https://doi.org/10.3390/su14127439>

Academic Editors: Sandro Serpa and Maria José Sá

Received: 12 April 2022

Accepted: 10 June 2022

Published: 17 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

What benefits may education provide in terms of career planning? Finding a satisfying career is a popular response that is both comfortable and obvious. Students'/employees' productivity, employability, and career satisfaction are all boosted by lifelong learning, which is the acquisition of knowledge for personal or professional goals. Students/employees are constantly exploring various educational opportunities to further their knowledge in order to achieve their career goals. Education ought to, in general, support the ecological system of employment [1], whereas the skills gap among academic studies, schooling, and industries needs to be narrowed [2].

With the rapid development of Internet technology, online resources have facilitated access to course and job information for students/employees. At the same time, it also brings the problem of "information overload", which confuses students and employers with the vast amount of online material available and prevents them from quickly identifying the most relevant courses and jobs. Recommender systems, which assist users in locating the most relevant items, are a promising method of filtering information. It will provide a series of specialized recommendations based on each user's individual needs and preferences.

In the domains of education and employment, recommender systems can assist students/employees in making better and more informed decisions, consequently influencing

their future. Many classical job/occupation and curriculum/education recommendation systems (JCRs) have been proposed, for instance, the CourseAgent system [3], the CourseRank system [4], and the CaPaR framework [5], etc. Existing JCRs, although meeting the requirements of some students/employees, have a limited impact. Furthermore, most algorithms in previous research used user-based models (UBM), content-based models (CBM), and collaborative filtering (CF). UBM focuses on analyzing learner or job seeker profiles, CBM concentrates on investigating course or job content features, and CF primarily examines course ratings, user learning history, and employment history. However, crucial implicit information that helps enhance the accuracy of the suggestion, such as the linkages between courses, skills, and jobs, is not properly exploited [6,7]. More sophisticated routes, such as *jobs-skills-skills-courses-courses*, can be used to link courses and jobs implicitly. According to this viewpoint, the linkage between courses, skills, and jobs may be far more complicated than the traditional CF approach. Furthermore, a large number of studies have been conducted on course recommendations and job recommendations [8], but rarely do they address the issue of cross-domain data integration between the two.

Therefore, this study proposes a novel Graph-Community-Enabled (GCE) approach to address the career-education cross-domain recommendation problem from the heterogeneous graph mining viewpoint. Two domains have three different nodes that interact through four types of relationships. The recommendation issue is therefore transformed into a graph-based random walk problem. Figure 1 depicts the integration of two disparate data sources, education and career into a heterogeneous network, employing skills as a bridge. However, due to the employment of various vocabulary, there are variations between work skills and course skills. We employ the Infomap algorithm to compute skill communities, which aids in the linking of job and course data. Finally, five meta-path features are manually constructed for recommendations based on a suitably indexed heterogeneous graph. A ranking hypothesis is represented by each feature. After that, using a random walk algorithm, we may deliver multiple customized courses and employment proposals by taking into account future professional aspirations or scheduled educational backgrounds.

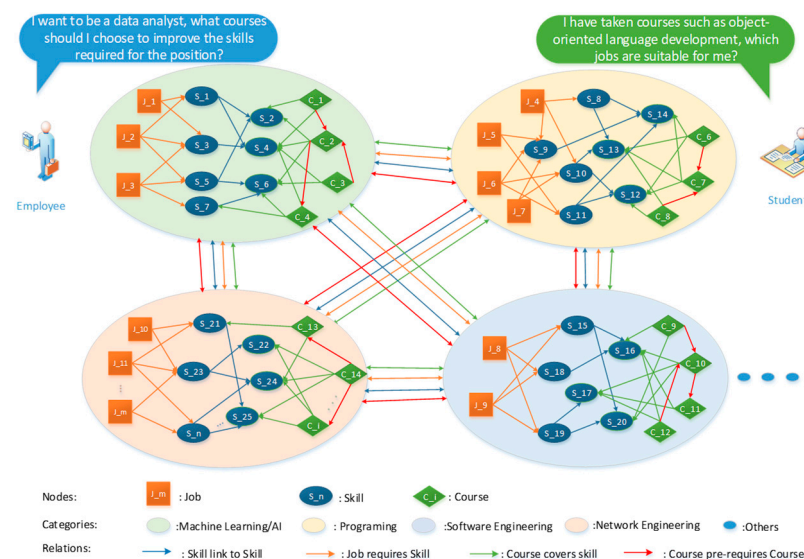


Figure 1. Heterogeneous Graph Index Schema. The *job* nodes are represented by orange squares, *skill* nodes are represented by blue ovals, and *course* nodes are represented by green rhombus. The *job-skill* (required by a job) relations are represented by the directed orange edges. The *skill* (required by a job)-*skill* (covered by a course) relations are represented via the directed blue edges. The *course-skill* (covered by a course) relations are represented by the directed green edges. The *course-course* represents the sequence of courses taken and is indicated by a red directed edge. The large ovals of different colors show the skill community the job and course belong to.

The significance and originality of this study lie in the integration and indexing of information from the job and education domains through heterogeneous networks and community detection to achieve cross-domain information recommendations. Experiments were performed utilizing course data from MOOCs and a university, as well as job advertisements from the IT industry, to illustrate that students/employees may benefit from this graph-based data integration. The findings suggest that the strategy is effective at generating curriculum recommendations based on pre-determined career objectives. The suggested approach is applicable to a variety of educational and occupational settings.

We note that an earlier version of this paper was presented at the International Conference [7], and it is also accessible on the arXiv (<https://arxiv.org/>, accessed on 11 April 2022) [8]. Our previous conference paper only conducted simple preliminary experiments on the course recommendation task and did not fully validate the proposed model. This manuscript introduces Word2Vec/Bert technology combined with a community detection algorithm to solve the skill difference problem in both course and job domains and improve the connection quality of heterogeneous graphs. In addition, multiple different meta-paths are designed for both course and job recommendation tasks according to the application scenarios of different types of users, while more data (i.e., MOOCs are added) and more baseline methods are used to analyze and validate the effectiveness of the proposed model, which enriches the experimental content. The details are presented in Section 3.

The structure of this paper is as follows: Section 2 reviews related literature; Section 3 details the process of data collection and the proposed community-based graph method; Section 4 discusses the results and provides our conclusions and suggestions for future work.

2. Related Work

2.1. Course Recommendation

In the area of course planning, recommendation systems have been widely used [6,9,10]. Courses were recommended to end-users in the majority of these studies based on feedback from other users [11,12], general user performance [13,14], or similarities across course content [15,16]. For instance, Nguyen et al. [17] employed sequential rule mining to find the optimum course for a pair of courses and grades. Recurrent neural networks were utilized by Morsy and Karypis [18] to suggest courses that could help students maintain and improve their GPAs. In general, it is rare for course recommendation systems to take the target occupation into account [1,19].

Likewise, systems for job recommendation have inspired widespread attention in the academic community in recent decades. Some studies have investigated job recommendations in terms of career pathways. Paparrizos et al. [20] trained machine learning models using prior work experience to predict candidates' next job transfer. Patel et al. [5] introduced CaPaR, a "career path recommendation" framework that mines users' work experience leveraging text mining and collaborative filtering approaches to make two sorts of suggestions to users: work and skill recommendations. To create career suggestions, some systems employed social networks [21]. Lu et al. [22] suggested a graph-based method for generating job recommendations based on the relationships between three entities in society (users, companies, and jobs). Prior job recommendation research has mostly focused on the work experience of users and ignored their educational background. Furthermore, a portion of the research employs graph-based methodologies; however, they are limited to one area of career.

2.2. Graph-Based Recommendation

The link analysis method in graph theory is used in the graph-based recommendation model to address the drawbacks (such as sparsity) of the classic methodology based on cooperation probability and to enhance recommendation accuracy. Early attempts to investigate graph-based recommendation techniques used homogeneous and bipartite graphs, with nodes representing items or users and edges representing similarities between items

or between users and items they evaluated [23]. Since heterogeneous graphs contain more node types and edge types, they are able to store richer semantic features compared to homogeneous networks. Recommendation research based on heterogeneous graphs has been widely used in several fields in recent years [24], including social recommendation [25–27], interest recommendation [28], friend recommendation [29], etc. Naturally, there have been numerous graph-based course and job recommender systems designed. For example, Bridges et al. [30] leveraged historical data on grades and enrollment to generate a directed graph, and then used that graph and a student's grades history to provide individualized recommendations about which course he should enroll in. By running a random walk on a Markov chain for each degree program throughout the course enrollment history, Polyzou et al. [31] recommended a shortlist of lessons to be enrolled for the next semester. Shalaby et al. [32] developed a graph-based method for real-time job recommendations that exploits the relationship between user–work interactions and recruitment information to arrive at a scalable solution. In general, existing graph-based course and job recommendation studies are limited to the education domain or the career domain only. In addition, to our knowledge, there are no graph-based studies on cross-domain information recommendations integrating educational and occupational data.

2.3. Cross-Domain Based Recommendation

The cross-domain recommender system is intended to enhance the target domain recommendation results in terms of accuracy and diversity by incorporating the profiles of users across multiple domains, sensing every user's characteristics intelligently, and fulfilling their requirements accurately. There are three main categories of cross-domain recommender algorithms: methods based on collaborative filtering relation, semantic relation, and deep learning. The collaborative filtering relationship mainly refers to the nearest neighbor relationship and the implicit semantic model of users or items. Semantic relation mainly refers to item attribute, tag information, semantic network (computer science) relation, association relation, etc. However, the recommendation performance of the same method varies in different cross-domain recommendation scenarios. Different solutions are usually required for different recommendation scenarios. According to the degree of overlap of users/items between domains, the application scenarios of the cross-domain recommendation system can be grouped into three categories: non-overlapping, partially overlapping, or fully overlapping users/items between domains [33].

In non-overlapping user/items scenarios, it is common to mine hidden common users/items or other hidden relationships between domains for migration learning. Using metadata as a bridge between domains, Fernández-Tobías et al. [34] constructed a cross-domain mixed matrix factorization model. Kuma et al. [35] used the LDA topic method to model the tag information of users and constructed the topic distribution space of user-profiles shared by different fields. Based on this space, users with similar preferences in different fields can be found to achieve cross-domain recommendations. For partially overlapping users/items, these shared users/items are often used as a bridge for information sharing and migration between domains. Jiang et al. [36] proposed a semi-supervised transfer learning method based on joint matrix decomposition, in which users with similar interest preferences in the source domain should have similar interest preferences in the target domain. Krishnamurthy et al. [37] trained user and item feature vectors based on the language model and transferred user feature information from a source domain to the target domain through a training knowledge transfer matrix with overlapping users as the bridge. In fully overlapping users/items scenarios, two domains are usually merged to turn cross-domain recommendations into single-domain recommendations. Jiang et al. [38] created a star-structured hybrid network that fused social network data and transferred knowledge from source domains to the target domain through a hybrid random walk method.

Notwithstanding a rich body of literature on cross-domain recommender systems, there is no specific design for career-education recommendation problems. Therefore, they cannot be directly applied to the problems studied in this study.

This study advances previous work by proposing a novel graph-based methodology that uses a heterogeneous graph combining educational and occupational data to recommend courses or occupations for students (or entry-level employees) in a hierarchical order based on their educational/occupational experience.

3. Research Methods

In this section, we will explore in-depth the proposed method, including data collection on education and occupation (Section 3.1), integrating and indexing two datasets through a heterogeneous graph with skills communities computed by Infomap. (Section 3.2), grading personalized curriculum and jobs using a graph-community-enabled cross-domain rating function that employs a random walk method (Section 3.3), and performing a case study (Section 3.4).

3.1. Data Collection

There are two types of data in the dataset collected in this study:

Courses/Education data were gleaned from the Luddy School of Informatics, Computing, and Engineering (SICE) at Indiana University Bloomington's (IUB) course enrollment log and course catalog (<https://luddy.indiana.edu/academics/courses/search/iub-fall-2019>, accessed on 11 April 2022). In total, these data include 188,881 course enrollment records from six departments (or fields) (Computer Science, Data Science, Informatics, Information and Library Science, Intelligent Systems Engineering, and Statistics) with 7824 students in 371 courses over 10 semesters in the 2016–2019 academic years. The collected curriculum data are structured. However, a significant feature not described in the original university curriculum data is the skills that a course teaches. In order to achieve the linkage of IUB-SICE courses and skills, we downloaded 957 MOOCs (<https://www.coursera.org/>, accessed on 11 April 2022) (each MOOC has a specific skill set, with a total of 1101 skills) in the same fields as IUB-SICE courses through web crawling (<https://www.webscraper.io/>, accessed on 11 April 2022). The Greedy Matching algorithm that is shown in Algorithm 1 was leveraged to extract the corresponding skills in IUB-SICE courses. The principle of the algorithm is to match from the right to the left of the text based on the dictionary, taking the length of the longest phrase in the dictionary as the length of the first matched text in each round, decreasing word by word from the left each time, scanning through the corresponding dictionary, and making the longest possible phrase that is matched as the best option. Specifically, the relevant skills from MOOCs as a skill-vocabulary V are greedily matched with the content C (course title and course description) of each IUB-SICE course, and the related skills of all courses of IUB-SICE are projected from MOOCs. After that, in this study, the course registration information of students who took less than five courses was removed. Courses with empty course descriptions and course skills were also removed. Finally, there were four features for any course included in the education dataset, namely course ID, course name, course description, and all associated skills. Overall, the education dataset consists of 266 university curricula, 957 MOOCs, and 1011 relevant skills.

The Job/Career dataset was composed of job postings scraped from Careerbuilder (<https://www.careerbuilder.com/>, accessed on 11 April 2022) in December 2019 using automated crawler technology. Trending IT job titles (<https://money.usnews.com/money/careers/slideshows/discover-the-best-technology-jobs>, accessed on 11 April 2022) were used as search terms. Redundant postings without the required skills were deleted, resulting in a total of 20,000 jobs and 1611 skills related to them listed in the final data. Finally, five characteristics were derived from job advertisements: Job ID, Job Title, Company, Location, and the set of needed Skills.

Algorithm 1. Greedy Matching Algorithm**Input:**the skill vocabulary V from MOOCs;the content C of each IUB-SICE course (course name+ course description).**Output:**the corresponding skill S of each IUB-SICE course.1: $skills \leftarrow []$;2: convert content C to list C_List ; convert vocabulary V to list V_List ;3: let the pointer P point to the end of C_List ;4: calculates the number of words from the pointer P to the beginning of C_List (that is, the length of the unsliced content) as n ;5: calculates the number of words in the longest phrase in V_List as m ;6: **while** P is not at the beginning of the C_List **do**7: **if** $n < m$ **then**8: $m = n$;9: **end**10: takes m words from the current P to the left of C_List as the phrase W 11: **if** W is in the V_List **then**12: adds W to $skills$;13: modifies the pointer P based on the length of W ;14: **else**15: removes one word from the left end of W ;16: **end**17: **end**18: **return** $skills$.**3.2. Skill Community Detection and Data Indexation Based on Heterogeneous Graph**

The key to designing this cross-domain recommendation system was to integrate data from the career domain and the education domain by building a heterogeneous graph. The essential link between the two domains was the range of skills needed for a career and the set of skills taught by each curriculum [1,6]. However, the skills stated in the courses use separate vocabularies from the skills listed in the jobs. Only 79 overlapped skills were detected in our dataset, which spanned career and curriculum data.

To overcome this problem and to link the two domains more closely, a community detection technique, the Infomap algorithm [39], was used for skill community partitioning in the candidate graph M . The Infomap approach works by simulating a random walker's m steps on a graph and indexing his random walk paths with a two-level codebook (a globally indexed codebook, one for each community). The following formula was used to construct a community division with the shortest description length of random walks:

$$L(\pi) = \sum_i^m q^i H(Q) + \sum_{i=1}^m p^i H(P^i), \quad (1)$$

where $L(\pi)$ represents a random walker's description length in the present community division π . q^i and p^i are the inter- and intra-group jump rates of the i_{th} community at each step. $H(Q)$ represents the frequency-weighted average length of codewords from the globally indexed codebook, while $H(P^i)$ represents the frequency-weighted average length of codewords from the i_{th} community codebook.

To begin, we generated a career network graph and an education network graph. All skill nodes were categorized into communities using Infomap (e.g., computer science community, data science community, and artificial intelligence community). The most similar vocational and educational communities (by counting the skills with similarity exceeding 0.9 via BM25/Word2Vec/Bert algorithm) were then combined with the related jobs and courses (as shown in Figure 1).

All educational and occupational data (skills, courses, and jobs) were combined into a single heterogeneous graph $G = (V, E)$ utilizing this approach. We define a node-type

mapping function $\tau : V \rightarrow O$ and an edge-type mapping function $\phi : E \rightarrow R$ in this graph, where each node $v \in V$ corresponds to a particular variable $\tau(v) \in O$ and each edge $e \in E$ corresponds to a particular relation $\phi(e) \in R$. Both the start and end objects type are the same if two linkages are of the identical relation type. Table 1 shows the descriptions of the nodes and relations.

Table 1. Nodes and Relations in the constructed heterogeneous graph.

Nodes and Relations	Description
C	The course nodes
J	The job nodes
S	The skill nodes
$C \xrightarrow{P} C$	The course to course edge via the pre-required relation
$C \xrightarrow{C} S$	The course to skill edge via the covered relation
$J \xrightarrow{r} S$	The job to skill edge via the required relation
$S \xrightarrow{1} S$	Skill to skill edge (skill-skill text similarity within each community based on BM25/Word2Vec/Bert).

The sum of the weights of outgoing connections of the same type for each node shown on the graph is equal to 1. For example, a link from C_i to C_j has a weight that is defined as $w(C_i \xrightarrow{P} C_j) = \frac{d(C_i \xrightarrow{P} C_j)}{d(C_i \xrightarrow{P} C)}$, in which $d(C_i \xrightarrow{P} C_j)$ is the count of students registered in a course C_j prior to registration in a course C_i , and in which $d(C_i \xrightarrow{P} C)$ is the sum of students registered in any course prior to registration in a course C_i . A link of $C_i \xrightarrow{C} S_j$ has a weight that is defined as $w(C_i \xrightarrow{C} S_j) = \frac{1}{d(C_i \xrightarrow{C} S)}$, where $d(C_i \xrightarrow{C} S)$ represents the sum count of skills that the course C_i taught. An edge of $J_i \xrightarrow{r} S_j$ has a weight that is defined as $w(J_i \xrightarrow{r} S_j) = \frac{d(J_i \xrightarrow{r} S_j)}{d(J_i \xrightarrow{r} S)}$, in which $d(J_i \xrightarrow{r} S_j)$ is the number of jobs J_i requiring skill S_j , where $d(J_i \xrightarrow{r} S)$ is the overall quantity of job J_i demanding any skill. The key to linking the whole heterogeneous graph internally is the linkage $S \xrightarrow{1} S$. For $S \xrightarrow{1} S$, the weight is the normalized similarity score among skills inside of each community (via BM25/Word2Vec/Bert). Similar skill pairs with noise do not affect the accuracy of the graph due to community restrictions.

The final network graph contained 23,845 nodes and 123,208 edges. There are 20,000 occupations, 266 university courses, 957 MOOCs, and 2622 skills (from jobs and courses) among them, with 73,560 $J \xrightarrow{r} S$, 11,155 $C \xrightarrow{P} C$, 10,641 $C \xrightarrow{C} S$, and 27,825 $S \xrightarrow{1} S$ linkages between them.

3.3. Community-Constrained Cross-Domain Recommendations with Heterogeneous Graph-Enabled

To generate educational/career recommendations, this section uses the Graph Community Enabled (GCE) cross-domain rating technique. We gathered the target candidate nodes for each query node in the graph and provide recommendations based on their ranking scores. A ranking function based on meta-paths [40] and community structure was used to calculate the ranking score of each candidate node. Briefly, a meta-path is a specific path connecting two entities. In this paper, meta-path refers to the relationship between query nodes and candidate nodes within a heterogeneous graph. Additionally, more than one different meta-path is generally included with the identical recommendation task (e.g., to suggest a curriculum to various users with different preferences). Furthermore, by changing the types of query nodes and suggestion nodes, the technique can be

applied to different recommendation tasks, such as proposing occupations to students or professionals.

The random walk algorithm is capable of efficiently catching complicated, higher-order, and indirect interactions between different types of nodes in a graph. It can successfully handle challenges in graph learning recommender systems (GLRS), particularly GLRS constructed on heterogeneous graphs, such as information propagation among different types of nodes [41]. Typically, random walk-based recommender systems start by having random walkers walk on a particular graph with a predefined probability of transfer at each step to simulate implicit preference or interaction propagation between users and/or items, and then rank these candidate nodes for recommendation using the probability of the walker landing on a node after a particular step. Therefore, a random walk-based metric is presented to measure the ranking scores of candidate nodes along meta-paths [40]. The following is a formula for it:

$$s(v_i^{(1)} \rightarrow v_j^{(l+1)}) = \sum_{t=v_i^{(1)} \rightarrow v_j^{(l+1)}} RW(t), \quad (2)$$

where v_i denotes the seed node, and v_j represents the query node of the candidate and in which t denotes a wander from v_i to v_j . Assuming $t = (v_{i1}^{(1)}, v_{i2}^{(2)}, \dots, v_{il+1}^{(l+1)})$, then the probability of the random walk is $RW(t) = \prod_j w(v_{ij}^{(j)}, v_{ij+1}^{(j+1)})$, where $w(v_{ij}^{(j)}, v_{ij+1}^{(j+1)})$ is the weight of the edge $v_{ij}^{(j)} \rightarrow v_{ij+1}^{(j+1)}$.

We explored course recommendation as a proof of concept and specified distinct meta-paths for the course recommendation task in three scenarios. In addition, for the job recommendation study, different meta-paths have also been designed for the two cases (see Table 2).

Table 2. Meta-path in the constructed heterogeneous graph.

	Scenarios	Meta-Paths	Ranking Hypothesis
Course Recommendation	1	$C_J \parallel J \xrightarrow{r} S \xrightarrow{l} S \xleftarrow{c} C_?$	The candidate courses in the community where the query job is located will be related to the query job if the skills covered by the courses are related to the skills required by the query job.
	2	$C_{/J} \parallel C_?^P \xrightarrow{P} C_?$	The pre-required course of the candidate courses not in the community where the query job is located will be related to the query job.
	3	$C_{/J} \parallel CP \xrightarrow{s} S \xleftarrow{c} C_?$	The candidate courses not in the community where the query job is located will be related to the job if they share similar skills as the taken course.
	3	$C_J \parallel J \xrightarrow{r} S \xrightarrow{l} S \xleftarrow{c} C \xleftarrow{P} C_?$	The candidate courses in the community where the query job is located will be related to the job if the skills of the pre-required courses of candidate courses are related to the skills required by the query job.

Table 2. Cont.

	Scenarios	Meta-Paths	Ranking Hypothesis
Job Recommendation	4	$\mathbb{C}_C \parallel C \xrightarrow{S} S \xrightarrow{J} J_?$	The candidate jobs in the community where the courses that have been taken will be related to the courses, if the skills covered for the query courses are related to the skills required by the job.
	5	$\mathbb{C}_{/J} \parallel C \xrightarrow{S} S \xrightarrow{J} J_?$	The candidate jobs not in the community where the current job is located will be related to the courses if the skills covered for the query courses are related to the skills required by the job.

Scenario 1: A freshman undergraduate or first-year graduate student with a career target (job node) is seeking educational advice (course node) to help him or her accomplish that goal.

The input to Scenario 1 is the student's target occupation (job node) and its output is the list of suggested curricula (course nodes). The associated meta-path functions are shown in Table 2. The queried job node is represented by J , and the candidate curriculum node is represented by $C_?$. These path functions gather the candidate courses connected to the career objective by skill traversing the relationship between jobs and courses. It is worth noting that the first function only retrieves for the objective community \mathbb{C}_J , which has the job J , whereas the second path function operates only for all pre-requisite courses $\mathbb{C}_?^P$ in the community $\mathbb{C}_{/J}$, which do not have the job J ;

Scenario 2: An employee/student has completed certain courses C^P and is seeking the potential courses $C_?$ to help him (or her) accomplish his (or her) career objective J . A new path function is added compared to the path function of Scenario 1: $\mathbb{C}_{/J} \parallel C^P \xrightarrow{S} S \xrightarrow{J} C_?$, in which C^P has a better likelihood of assisting in finding suitable courses $C_?$;

Scenario 3: An employee/professional is currently employed at job J and wishes to advance in his or her career. That is, he is searching for a course that will assist him in upskilling.

$\mathbb{C}_J \parallel J \xrightarrow{S} S \xrightarrow{J} S \xrightarrow{C} C \xrightarrow{P} C_?$ can be seen as the ranking function. Since the information demands of users are to upgrade their skills, the final stage will be $C \xrightarrow{P} C_?$, namely, basic courses (e.g., programming) can move on to more advanced courses (e.g., machine learning);

Scenario 4: A graduating undergraduate/graduate student is looking for a new job (job node). The input is the student's completed courses (course nodes), and its output is the list of suggested jobs (job nodes). The meta-path function for Scenario 4 is shown in Table 2. Where C denotes the queried course node, and $J_?$ represents the recommended job node. The path function gathers candidate jobs associated with courses already taken by traversing the relationship between jobs and courses through skills. It is worth noting that the function only works on the target community \mathbb{C}_C , which includes course C ;

Scenario 5: An employee currently has a job and has enrolled in some courses and is looking for a new career not related to the current job. This is similar to Scenario 4, but this function is executed only for community $\mathbb{C}_{/J}$, which does not contain the current job J .

In contrast to prior research, the community constraint of the random walk function described in this work is critical for efficiently reducing noise and improving recommendation accuracy.

3.4. Experimental Result

Experiments were run for Scenarios 1 to 5, respectively. Four university lecturers with at least 10 years of teaching experience and 10 years of industrial experience and 20 graduate students with at least 2 years of industrial experience were invited to perform a case study with the presented recommendation algorithm. Following [42], Mean Average Precision (MAP@K), Precision (P@K) and Normalized Discounted Cumulative Gain (NDCG@K) were used as the evaluation metrics. The performance of the ranking at a particular cutoff level was also estimated, considering only the top K courses returned as candidates in the experiment. We set K to 5, 10, 15, 20 and calculate all metrics. P@K(q) was computed as follows:

$$P@K(q) = \frac{\sum_{k=1}^K \text{acc}_q(k)}{K}, \quad (3)$$

where $\text{acc}_q(k)$ is a binary indicator function that returns 1 when the k-th recommendation is relevant for the q-th query and 0 otherwise. MAP@K is computed as the average AP@K(q) per query, where AP@K(q) is computed as follows:

$$AP@K(q) = \frac{1}{\min(R_q, K)} \sum_{k=1}^K P_q@k \times \text{Rel}_q(k), \quad (4)$$

where R_q is the total number of corresponding courses/jobs that appear in the q-th query, $P_q@k$ is the precision at rank k for the q-th query, and $\text{Rel}_q(k)$ is a binary indicator function that returns 1 when the k-th recommendation is relevant for the q-th query and 0 otherwise. NDCG@K(q) is computed as follows:

$$NDCG@K(q) = \frac{DCG@K(q)}{IDCG@K(q)} = \frac{\sum_{k=1}^K \frac{2^{\text{Rel}_q(k)-1}}{\log_2(k+1)}}{\sum_{k=1}^{|\text{Rel}_q|_K} \frac{2^{\text{Rel}_q(k)-1}}{\log_2(k+1)}}, \quad (5)$$

where $|\text{Rel}_q|_K$ represents the length of the recommendation list that are most relevant to the q-th query among the top k candidate recommendations.

3.4.1. Comparison with Baselines

This experiment evaluates our model Graph-Community-Enabled (GCE) against the Text-based and Graph-Enabled (GE) models. The comparison methods are given below.

- Text ranking features:

BM25 [43]: the text query is sent to the course/job text indexation, and relevant courses/jobs are fetched. Then, extracting the keyword information from each retrieved (and top-ranked) course content or job posting as a new query. In the end, matching precisely the words between the new query and each course/job document.

Word2Vec [44] and BERT [45]: firstly embedding the descriptions of a course and a job into two vectors, and then calculating the cosine similarity between the two vectors. Finally, ranking the relevant course/job related to the query according to similarity.

- Graph-Enabled (GE) ranking features:

GE(BM25): matching the words of skills between courses and work through BM25, linking the education network and work network. A ranking is then performed on the scores of each candidate node, which are derived from the meta-path ordering function based on the random walk algorithm.

GE(W2V): the skills of course and job are embedded through Word2vec, and the cosine similarity between them is calculated. Each course skill retains the top five job skills and establishes a skill graph connecting education and vocational fields. In addition, random walk algorithms are used to rank each candidate node's scores that come from the meta-path.

GE(Bert): except for the embedding of skills from courses and jobs being calculated through the Bert model, other processes are the same as GE (W2V).

- Graph-Community-Enabled (GCE) ranking features:

GCE(BM25): integrating the education-career heterogeneous graph by matching the words of skills between courses and work through **BM25**. Using the Infomap algorithm to detect the communities in the graph. Then, a random walk is run on the graph with the classified communities and the scores of each candidate are ranked.

GCE(W2V): the skills of course and job are embedded through Word2vec, and the cosine similarity between them is calculated. Each course skill retains the top five job skills and establishes a skill graph connecting education and vocational fields. Using the Infomap algorithm to detect the communities in the graph. Then, a random walk is run on the graph with the classified communities and the scores of each candidate are ranked

GCE(Bert): except that the embedding of skills from courses and jobs is calculated through the Bert model, other processes are the same as GCE(W2V).

3.4.2. Recommending Courses to User

For Scenario 1 to 3, participants tested five text queries (job title, e.g., “Database Administrator”, “Java Developer”, and “Data Scientist”, etc.) for each scenario. Additionally, they entered the courses (course name, e.g., “Database Concepts”, “Programming Language Principles”, “Introduction to Data Analysis and Mining”, etc.) that had been taken for scenario 2. Then, the results of each recommendation were rated as “relevant” or “not relevant” by each respondent against the query terms. Combining each participant’s feedback, we calculated the recommendation performance of each scenario. In Tables 3–5, the performance of multiple recommendation algorithms is reported, including overall and top-ranked course recommendation performances. Figure 2a–c show the top-20 courses’ performances of the graph-based recommendation approaches in three scenarios.

Table 3. Result of Course Recommendation for Scenario 1.

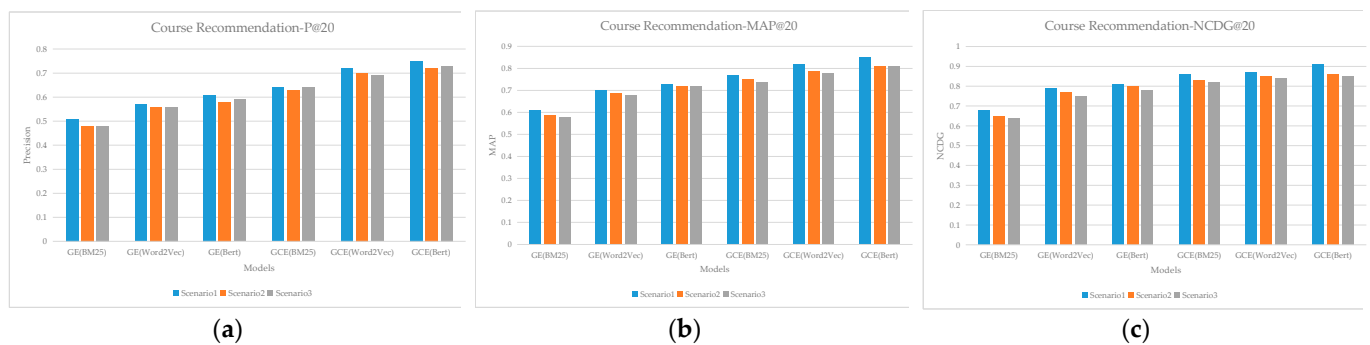
Method	P@5	MAP@5	NCDG@5	P@10	MAP@10	NCDG@10	P@15	MAP@15	NCDG@15	P@20	MAP@20	NCDG@20
BM25	0.38	0.52	0.58	0.43	0.55	0.61	0.45	0.57	0.64	0.48	0.58	0.65
Word2Vec	0.41	0.55	0.61	0.48	0.57	0.65	0.48	0.6	0.67	0.5	0.62	0.68
Bert	0.48	0.62	0.73	0.5	0.63	0.74	0.51	0.65	0.77	0.55	0.66	0.79
GE(BM25)	0.39	0.53	0.58	0.48	0.56	0.62	0.49	0.59	0.65	0.51	0.61	0.68
GE(W2V)	0.43	0.65	0.74	0.52	0.64	0.76	0.55	0.69	0.78	0.57	0.70	0.79
GE(Bert)	0.51	0.67	0.75	0.57	0.69	0.77	0.58	0.71	0.79	0.61	0.73	0.81
GCE(BM25)	0.55	0.71	0.79	0.59	0.73	0.82	0.61	0.76	0.85	0.64	0.77	0.86
GCE(W2V)	0.63	0.75	0.81	0.66	0.78	0.85	0.68	0.81	0.87	0.72	0.82	0.87
GCE(Bert)	0.67	0.78	0.83	0.69	0.81	0.89	0.72	0.83	0.9	0.75	0.85	0.91

Table 4. Result of Course Recommendation for Scenario 2.

Method	P@5	MAP@5	NCDG@5	P@10	MAP@10	NCDG@10	P@15	MAP@15	NCDG@15	P@20	MAP@20	NCDG@20
BM25	0.39	0.51	0.55	0.44	0.54	0.59	0.46	0.56	0.63	0.47	0.57	0.64
Word2Vec	0.4	0.54	0.58	0.47	0.55	0.62	0.48	0.58	0.65	0.5	0.60	0.66
Bert	0.46	0.61	0.69	0.49	0.64	0.71	0.52	0.67	0.74	0.54	0.67	0.75
GE(BM25)	0.37	0.51	0.57	0.41	0.54	0.59	0.45	0.57	0.63	0.48	0.59	0.65
GE(W2V)	0.42	0.61	0.69	0.49	0.62	0.72	0.54	0.68	0.75	0.56	0.69	0.77
GE(Bert)	0.5	0.65	0.72	0.54	0.69	0.75	0.56	0.70	0.78	0.58	0.72	0.80
GCE(BM25)	0.53	0.68	0.74	0.57	0.72	0.78	0.60	0.73	0.82	0.63	0.75	0.83
GCE(W2V)	0.59	0.73	0.77	0.62	0.75	0.79	0.67	0.78	0.83	0.70	0.79	0.85
GCE(Bert)	0.61	0.75	0.80	0.66	0.77	0.83	0.70	0.79	0.85	0.72	0.81	0.86

Table 5. Result of Course Recommendation for Scenario 3.

Method	P@5	MAP@5	NCDG@5	P@10	MAP@10	NCDG@10	P@15	MAP@15	NCDG@15	P@20	MAP@20	NCDG@20
BM25	0.37	0.52	0.54	0.42	0.56	0.58	0.46	0.58	0.61	0.47	0.59	0.63
Word2Vec	0.39	0.53	0.56	0.48	0.58	0.60	0.51	0.6	0.63	0.53	0.62	0.64
Bert	0.47	0.63	0.65	0.50	0.62	0.68	0.53	0.65	0.70	0.55	0.66	0.72
GE(BM25)	0.38	0.5	0.53	0.41	0.53	0.59	0.47	0.56	0.63	0.48	0.58	0.64
GE(W2V)	0.43	0.64	0.66	0.51	0.63	0.70	0.55	0.66	0.73	0.56	0.68	0.75
GE(Bert)	0.5	0.66	0.69	0.55	0.68	0.74	0.58	0.70	0.77	0.59	0.72	0.78
GCE(BM25)	0.52	0.69	0.71	0.58	0.71	0.76	0.64	0.72	0.81	0.64	0.74	0.82
GCE(W2V)	0.61	0.72	0.75	0.64	0.75	0.78	0.68	0.77	0.82	0.69	0.78	0.84
GCE(Bert)	0.63	0.76	0.79	0.66	0.79	0.82	0.71	0.80	0.83	0.73	0.81	0.85

**Figure 2.** Comparison of graph-based course recommendation methods. (a) Course Recommendation Precision@20; (b) Course Recommendation MAP@20; (c) Course Recommendation NCDG@20.

The results show that the proposed GCE method performs clearly better than other baselines, and GCE(Bert) works best. Besides, the text-only ranking model performs poorly compared to graph-based ranking methods, with the exception of GE (BM25), because it cannot capture the higher-order relationships across job and course domains using only job and course descriptions. Meanwhile, the GE approaches' performance (MAP, Precision and NCDG) of the top-20 courses underperformed the GCE method. The GCE method performed better in Scenario 1 than in the other two scenarios. Compared with the other two scenarios, the length of the meta-path for Scenario 1 is shorter, and the structure is simple. As pointed out in Sun et al. [46], shorter meta-paths have more information than longer ones because longer meta-paths link more remote objects (with lower semantic relevance).

3.4.3. Recommending Jobs to User

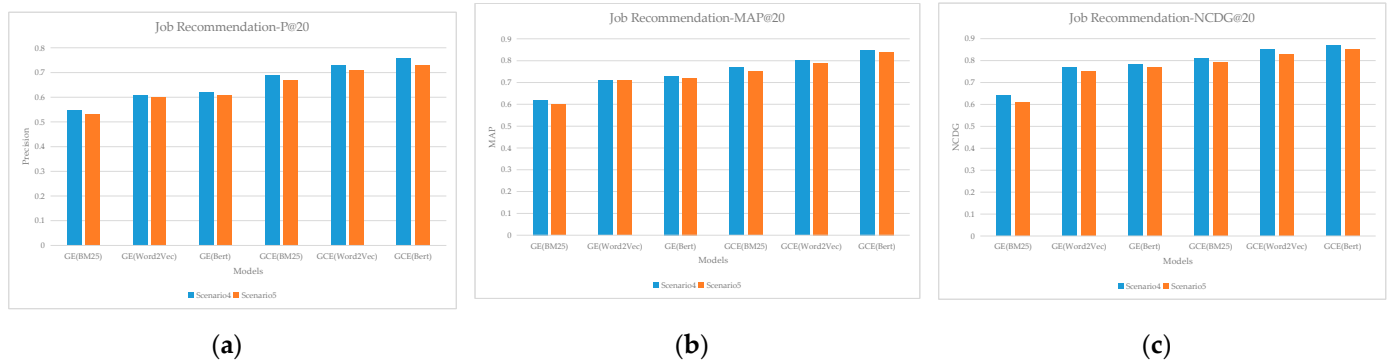
Similar to the course recommendation test, for Scenarios 4 and 5, each participant tested five text queries (course name, e.g., "Database Concepts", "Programming Language Principles", "Introduction to Data Analysis and Mining") for each scenario and additionally entered a current job (job title, e.g., "Database Administrator", "Java Developer", and "Data Scientist") for Scenario 5. Tables 6 and 7 show the results of different recommendation methods. Figure 3a–c shows the top-20 job performance of the graph-based recommendation approaches in two scenarios. Based on these tables, the results are similar to course recommendations, and the GCE method outperforms the baselines on the job recommendation task. GE(BM25) still works worst, as the skills from courses and jobs were represented in different vocabularies, and only a few skills overlap. In addition, from the data in Figure 3a–c, it is apparent that the performance of GCE in Scenario 4 is better than in Scenario 5 and is obviously superior to other methods.

Table 6. Result of Job Recommendation for Scenario 4.

Method	P@5	MAP@5	NCDG@5	P@10	MAP@10	NCDG@10	P@15	MAP@15	NCDG@15	P@20	MAP@20	NCDG@20
BM25	0.35	0.48	0.51	0.51	0.55	0.56	0.52	0.57	0.59	0.53	0.58	0.61
Word2Vec	0.38	0.58	0.62	0.52	0.59	0.65	0.54	0.62	0.69	0.54	0.64	0.7
Bert	0.41	0.62	0.65	0.54	0.63	0.68	0.58	0.66	0.72	0.59	0.67	0.73
GE(BM25)	0.36	0.5	0.54	0.49	0.57	0.58	0.53	0.61	0.62	0.55	0.62	0.64
GE(W2V)	0.42	0.64	0.68	0.56	0.65	0.72	0.59	0.69	0.76	0.61	0.71	0.77
GE(Bert)	0.49	0.68	0.71	0.59	0.69	0.73	0.62	0.73	0.77	0.62	0.73	0.78
GCE(BM25)	0.52	0.7	0.73	0.63	0.72	0.77	0.67	0.75	0.80	0.69	0.77	0.81
GCE(W2V)	0.61	0.73	0.77	0.67	0.75	0.79	0.71	0.79	0.83	0.73	0.80	0.85
GCE(Bert)	0.65	0.75	0.78	0.71	0.79	0.83	0.73	0.84	0.86	0.76	0.85	0.87

Table 7. Result of Job Recommendation for Scenario 5.

Method	P@5	MAP@5	NCDG@5	P@10	MAP@10	NCDG@10	P@15	MAP@15	NCDG@15	P@20	MAP@20	NCDG@20
BM25	0.34	0.49	0.50	0.52	0.54	0.54	0.53	0.56	0.57	0.53	0.57	0.59
Word2Vec	0.36	0.58	0.61	0.53	0.58	0.63	0.55	0.60	0.67	0.56	0.62	0.69
Bert	0.4	0.61	0.63	0.55	0.62	0.67	0.56	0.65	0.70	0.58	0.66	0.71
GE(BM25)	0.34	0.49	0.52	0.48	0.56	0.55	0.51	0.59	0.59	0.53	0.60	0.61
GE(W2V)	0.44	0.64	0.66	0.57	0.66	0.70	0.58	0.69	0.73	0.60	0.71	0.75
GE(Bert)	0.47	0.66	0.69	0.58	0.67	0.72	0.60	0.71	0.76	0.61	0.72	0.77
GCE(BM25)	0.51	0.69	0.72	0.64	0.7	0.75	0.66	0.73	0.78	0.67	0.75	0.79
GCE(W2V)	0.6	0.72	0.75	0.67	0.76	0.78	0.69	0.78	0.81	0.71	0.79	0.83
GCE(Bert)	0.63	0.75	0.76	0.69	0.78	0.82	0.72	0.82	0.84	0.73	0.84	0.85

**Figure 3.** Comparison of graph-based job recommendation methods. (a) Job Recommendation Precision@20; (b) Job Recommendation MAP@20; (c) Job Recommendation NCDG@20.

In summary, the GCE approach described in this paper has a better chance for cross-domain recommendation than other traditional methods and can serve the individualized needs of students/professionals. It is also worth noting that we just employed a simple graph ranking model in this experiment. To increase recommendation performance in the future, we will continue to investigate more complicated graph ranking models and learn ranking algorithms.

4. Discussion and Conclusions

In this study, we present a novel model GEC that combines heterogeneous graphs with community detection techniques, aimed at integrating and indexing education and career data according to the group of skills, to implement a variety of information retrieval/recommendation tasks to satisfy students' and job seekers' diverse information requirements.

We chose 24 people to conduct five separate experiments for each scenario on real-world work and course datasets. In other words, 120 experiments were conducted for each scenario. Since the input of each experiment was different, the experimental results were also different. The same inputs were also tested in this study and the results were the same. The experimental results shown in this paper are the average value after all tests.

The experimental results show that, among the three text ranking models in all scenarios, the Bert model works best, followed by Word2Vec, BM25 with the worst results. The text similarity algorithm based on semantic matching outperformed the non-semantic matching similarity algorithm. Similarly, the experimental performance of the three GE baseline models is similar to the three text ranking methods mentioned above, and the recommended performance is GE(Bert), GE(W2V), and GE(BM25) in descending order. The GE baseline methods are better than the text ranking methods overall, except GE(BM25). This is because, based on BM25 algorithm, the course skill nodes are poorly matched with the job nodes, resulting in a poorly linked job network graph and course network graph, which affects the random walk ranking performance. The performance of the proposed model GCE(Bert) is the best, followed by GCE(W2V) and finally GCE(BM25). The performance of GCE(Bert) (i.e., the mean of P@20, MAP@20 and NCDG@20) is higher than that of BM25, Word2Vec, Bert with ((0.25)(0.21)(0.16), respectively. It is also higher than the performance of GE(Bert), GE(W2V), and GE(BM25), which improved by (0.23)(0.13)(0.11), respectively. This validates the effectiveness of the model proposed in this paper. In addition, it is interesting to note that in all five cases of this study, the GCE model works best for students who have not taken a course or have no work experience (Scenarios 1 and 4). A possible explanation for this might be that the information input by such users is refined and has a low impact on the ranking of candidate recommendations.

The present results are significant in at least two major respects. First, in the view of data science, it is necessary to cross-traverse occupational data and educational data. It is also a prospective way to integrate data through community detection and data fusion. Second, by crossing curriculum and job domains, this study helps to eliminate the “Skills Gap” between the students/employees and the employers.

However, this approach is limited in that there are almost no overlapping skills between jobs and courses. Specifically, among the 1011 skills that are recognized from the course data, there are only 79 skills that can be mapped to the skill vocabulary set out in recruitment advertisements. Another limitation of this study is that the experiment used only one real dataset (education and work dataset) to validate the model. Although most of the related studies [47–50] also used only one dataset, in order to further validate the model generalizability, new datasets will be considered for future studies.

Going forward, there are three areas we will be working on with this project. First, we want to improve the quality of heterogeneous graphs: (1) unifying skill terms and skill classifications by using external knowledge bases, e.g., Wikipedia; and (2) adding new node types and edges, e.g., company, (course) instructor, and salary information; (3) to further improve the recommendation performance, we will try to study a more elaborate heterogeneous graph ranking model. Finally, we intend to leverage user feedback to train models and improve recommendation performance.

Author Contributions: Conceptualization, investigation, software, validation, formal analysis, data curation writing—original draft preparation, G.Z.; supervision, Y.C. and S.W.; methodology, writing—review and editing, G.Z., Y.C. and S.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Grant Numbers 71271034).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding authors.

Acknowledgments: The authors will thank Xiaozhong Liu for his great help in writing assistance and data providing. Thanks also to IUB for the data support.

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

References

1. Majidi, N. A Personalized Course Recommendation System Based on Career Goals. Ph.D. Dissertation, Memorial University of Newfoundland, St. John's, NL, Canada, 2018.
2. Börner, K.; Scrivner, O.; Gallant, M.; Ma, S.; Liu, X.; Chewing, K.; Wu, L.; Evans, J.A. Skill discrepancies between research, education, and jobs reveal the critical need to supply soft skills for the data economy. *Proc. Natl. Acad. Sci. USA* **2018**, *115*, 12630–12637. [[CrossRef](#)] [[PubMed](#)]
3. Farzan, R.; Brusilovsky, P. Social navigation support in a course recommendation system. In Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Dublin, Ireland, 21–23 June 2006; pp. 91–100.
4. Parameswaran, A.; Venetis, P.; Garcia-Molina, H. Recommendation systems with complex constraints: A course recommendation perspective. *ACM Trans. Inf. Syst.* **2011**, *29*, 20. [[CrossRef](#)]
5. Patel, B.; Kakuste, V.; Eirinaki, M. Capar: A career path recommendation framework. In Proceedings of the 2017 IEEE Third International Conference on Big Data Computing Service and Applications (BigDataService), San Francisco, CA, USA, 6–9 April 2017; pp. 23–30.
6. Li, N.; Naren, S.; Gao, Z.; Xia, T.; Börner, K.; Liu, X. Enter a job, get course recommendations. In Proceedings of the iConference 2017, Wuhan, China, 22–25 March 2017; pp. 118–122.
7. Zhu, G.; Kopalle, N.A.; Wang, Y.; Liu, X.; Jona, K.; Börner, K. Community-based data integration of course and job data in support of personalized career-education recommendations. In Proceedings of the 83rd Annual Meeting of the Association for Information Science and Technology, Pittsburgh, PA, USA, 22 October–1 November 2020; p. e324.
8. Zhu, G.; Kopalle, N.A.; Wang, Y.; Liu, X.; Jona, K.; Börner, K. Community-based data integration of course and job data in support of personalized career-education recommendations. *arXiv* **2020**, arXiv:2006.13864. [[CrossRef](#)]
9. Manouselis, N.; Drachler, H.; Vuorikari, R.; Hummel, H.; Koper, R. *Recommender Systems in Technology Enhanced Learning. Recommender Systems Handbook*; Springer: Boston, MA, USA, 2011; pp. 387–415.
10. Linden, G.; Smith, B.; York, J. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet Comput.* **2003**, *7*, 76–80. [[CrossRef](#)]
11. Li, X.; Chang, S.K. A personalized e-learning system based on user profile constructed using information fusion. In Proceedings of the 11th International Conference on Distributed Multimedia Systems (DMS), Banff, AB, Canada, 5–7 September 2005; pp. 109–114.
12. Tan, H.; Guo, J.; Li, Y. E-learning recommendation system. In Proceedings of the 2008 International Conference on Computer Science and Software Engineering, Wuhan, China, 12–14 December 2008; pp. 430–433.
13. Ray, S.; Sharma, A. A collaborative filtering based approach for recommending elective courses. In Proceedings of the International Conference on Information Intelligence, Systems, Technology and Management, Gurgaon, India, 10–12 March 2011; pp. 330–339.
14. Thai-Nghe, N.; Drumond, L.; Krohn-Grimberghe, A.; Schmidt-Thieme, L. Recommender system for predicting student performance. *Procedia Comput. Sci.* **2010**, *1*, 2811–2819. [[CrossRef](#)]
15. Ghauth, K.I.; Abdullah, N.A. Measuring learner's performance in e-learning recommender systems. *Australas. J. Educ. Technol.* **2010**, *26*, 764–774. [[CrossRef](#)]
16. Ghiani, G.; Manni, E.; Romano, A. Training offer selection and course timetabling for remedial education. *Comput. Ind. Eng.* **2017**, *111*, 282–288. [[CrossRef](#)]
17. Nguyen, H.Q.; Pham, T.T.; Vo, V.; Vo, B.; Quan, T.T. The predictive modeling for learning student results based on sequential rules. *Int. J. Innov. Comput. Inf. Control* **2018**, *14*, 2129–2140.
18. Morsy, S.; Karypis, G. Learning Course Sequencing for Course Recommendation. Available online: <https://conservancy.umn.edu/handle/11299/216025> (accessed on 27 June 2020).
19. Ma, X.; Ye, L. Career goal-based e-learning recommendation using enhanced collaborative filtering and prefixspan. *Int. J. Mob. Blended Learn.* **2018**, *10*, 23–37. [[CrossRef](#)]
20. Paparrizos, I.; Cambazoglu, B.B.; Gionis, A. Machine learned job recommendation. In Proceedings of the fifth ACM Conference on Recommender Systems, Chicago, IL, USA, 23–27 October 2011; pp. 325–328.
21. Diaby, M.; Viennet, E.; Launay, T. Toward the next generation of recruitment tools: An online social network-based job recommender system. In Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013), Niagara Falls, ON, Canada, 25–28 August 2013; pp. 821–828.
22. Lu, Y.; El Helou, S.; Gillet, D. A recommender system for job seeking and recruiting website. In Proceedings of the 22nd International Conference on World Wide Web, Rio de Janeiro, Brazil, 13–17 May 2013; pp. 963–966.
23. 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](#)]
24. Ma, M.; Na, S.; Wang, H.; Chen, C.; Xu, J. The graph-based behavior-aware recommendation for interactive news. *Appl. Intell.* **2022**, *52*, 1913–1929. [[CrossRef](#)]
25. Zhang, C.; Wang, Y.; Zhu, L.; Song, J.; Yin, H. Multi-graph heterogeneous interaction fusion for social recommendation. *ACM Trans. Inf. Syst.* **2021**, *40*, 1–26. [[CrossRef](#)]
26. Salamat, A.; Luo, X.; Jafari, A. HeteroGraphRec: A heterogeneous graph-based neural networks for social recommendations. *Knowl. Based Syst.* **2021**, *217*, 106817. [[CrossRef](#)]
27. Liu, H.; Zheng, C.; Li, D.; Zhang, Z.; Lin, K.; Shen, X.; Wang, J. Multi-perspective social recommendation method with graph representation learning. *Neurocomputing* **2022**, *468*, 469–481. [[CrossRef](#)]

28. Yan, D.; Xie, W.; Zhang, Y. Heterogeneous information network-based interest composition with graph neural network for Recommendation. *Appl. Intell.* **2022**, 1–15. [[CrossRef](#)]
29. Huang, M.; Jiang, Q.; Qu, Q.; Chen, L.; Chen, H. Information fusion oriented heterogeneous social network for friend recommendation via community detection. *Appl. Soft Comput.* **2022**, *114*, 108103. [[CrossRef](#)]
30. Bridges, C.; Jared, J.; Weissmann, J.; Montanez-Garay, A.; Spencer, J.; Brinton, C.G. Course recommendation as graphical analysis. In Proceedings of the 52nd Annual Conference on Information Sciences and Systems, (CISS), Princeton, NJ, USA, 21–23 March 2018; pp. 1–6.
31. Polyzoou, A.; Nikolakopoulos, A.N.; Karypis, G. Scholars walk: A markov chain framework for course recommendation. In Proceedings of the 12th International Conference on Educational Data Mining (EDM 2019), Montreal, QC, Canada, 2–5 July 2019; pp. 396–401.
32. Shalaby, W.; AlAila, B.; Korayem, M.; Pournajaf, L.; AlJadda, K.; Quinn, S.; Zadrozny, W. Help me find a job: A graph-based approach for job recommendation at scale. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; pp. 1544–1553.
33. Chang, W.; Zhang, Q.; Fu, C.; Liu, W.; Zhang, G.; Lu, J. A cross-domain recommender system through information transfer for medical diagnosis. *Decis. Support Syst.* **2021**, *143*, 113489. [[CrossRef](#)]
34. Fernández-Tobías, I.; Cantador, I.; Tomeo, P.; Anelli, V.W.; Di Noia, T. Addressing the user cold start with cross-domain collaborative filtering: Exploiting item metadata in matrix factorization, user model. *User Model. User-Adapt. Interact.* **2019**, *29*, 443–486. [[CrossRef](#)]
35. Kumar, A.; Kumar, N.; Hussain, M.; Chaudhury, S.; Agarwal, S. Semantic clustering-based cross-domain recommendation. In Proceedings of the 2014 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), Orlando, FL, USA, 9–12 December 2014; pp. 137–141.
36. Jiang, M.; Cui, P.; Yuan, N.J.; Xie, X.; Yang, S. Little is much: Bridging cross-platform behaviors through overlapped crowds. In Proceedings of the the Thirtieth AAAI Conference on Artificial Intelligence (AAAI), Phoenix, AZ, USA, 12–17 February 2016; pp. 13–19.
37. Krishnamurthy, B.; Puri, N.; Goel, R. Learning vector-space representations of items for recommendations using word embedding models. *Procedia Comput. Sci.* **2016**, *80*, 2205–2210. [[CrossRef](#)]
38. Jiang, M.; Cui, P.; Chen, X.; Wang, F.; Zhu, W.; Yang, S. Social recommendation with cross-domain transferable knowledge. *IEEE Trans. Knowl. Data Eng.* **2015**, *27*, 3084–3097. [[CrossRef](#)]
39. Rosvall, M.; Bergstrom, C.T. Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci. USA* **2008**, *105*, 1118–1123. [[CrossRef](#)] [[PubMed](#)]
40. Liu, X.; Yu, Y.; Guo, C.; Sun, Y. Meta-path-based ranking with pseudo relevance feedback on heterogeneous graph for citation recommendation. In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, Shanghai, China, 3–7 November 2014; pp. 121–130.
41. Wang, S.; Hu, L.; Wang, Y.; He, X.; Sheng, Q.Z.; Orgun, M.A.; Cao, L.; Ricci, F.; Philip, S.Y. Graph learning based recommender systems: A review. In Proceedings of the 30th International Joint Conference on Artificial Intelligence, Virtual Event. Montreal, QC, Canada, 19–26 August 2021; pp. 4644–4652.
42. Guo, C.; Liu, X. Automatic Feature Generation on Heterogeneous Graph for Music Recommendation. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, 9–13 August 2015; pp. 807–810.
43. Robertson, S.; Zaragoza, H. The probabilistic relevance framework: BM25 and beyond. *Inf. Retr.* **2009**, *3*, 333–389. [[CrossRef](#)]
44. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 3111–3119.
45. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
46. Sun, Y.; Han, J.; Yan, X.; Yu, P.S.; Wu, T. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proc. VLDB Endow.* **2011**, *4*, 992–1003. [[CrossRef](#)]
47. 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**, *415*, 84–95. [[CrossRef](#)]
48. Gong, J.; Wang, S.; Wang, J.; Feng, W.; Peng, H.; Tang, J.; Yu, P.S. Attentional graph convolutional networks for knowledge concept recommendation in moocs in a heterogeneous view. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, 25–30 July 2020; pp. 79–88.
49. Pardos, Z.A.; Jiang, W. Designing for serendipity in a university course recommendation system. In Proceedings of the Tenth International Conference on Learning Analytics & Knowledge, Frankfurt, Germany, 23–27 March 2020; pp. 350–359.
50. Almaleh, A.; Aslam, M.A.; Saeedi, K.; Aljohani, N.R. Align my curriculum: A framework to bridge the gap between acquired university curriculum and required market skills. *Sustainability* **2019**, *11*, 2607. [[CrossRef](#)]