knowledge

MDPI

Article

Recommendations for Smoothing the Transition from Education to Career: A Heterogeneous Knowledge Graph Architecture for Career-Motivated Explainable Course Recommendation

Jacob Striebel [1,*], Rebecca Myers [1], Tatiana Ringenberg [2], Patrick C. Shih [1] and Xiaozhong Liu [3]

1 Luddy School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN 47405, USA
2 Department of Computer and Information Technology, Purdue University, West Lafayette, IN 47907, USA
3 Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA 01609, USA
* Correspondence: jstrieb@iu.edu

Abstract

Complexity science studies systems in which properties and behaviors emerge at meso- and macroscales that are difficult to predict and model by observing the properties and behaviors exhibited by the system's components at smaller scales. The set of relationships that exist among post-secondary school curricula and job markets is one example of such a system. Prior work has undertaken the challenge of modeling this system for several purposes, one of which has been to develop retrieval and ranking algorithms in the education–career domain. A particular emergent property that is closely bound up with this prior work, and that is the focus of the present work, is the salience of a course with respect to a specific objective. The specific objective that we are interested in here is career usefulness, which means that our overall task is to rank order courses based on their usefulness in helping a student to obtain and perform a specific job. One aspect of this overall task that remains understudied concerns how it can best be performed in an interpretable manner and whether existing interpretable methods that may be applied to it, such as text-based similarity measures and document-ranking functions, represent workable solutions or whether an approach involving more detailed modeling of the underlying complex system may prove more effective. The purpose of this article is to answer this question, and, in order to do this, most of this article's content is devoted to the latter kind of approach, because the former kind is described in detail in the existing literature. The specific approach of the latter kind that we investigate is based on, first, developing a heterogeneous knowledge graph model of the overall complex system, and, second, developing a procedure that quantifies salience using the strength of the skill-dependency chains that link a course to a specified job. To evaluate our methods, we perform a human subjects study in which we leverage the domain expertise of fifty participants. The results of the study demonstrate that the latter approach produces career-motivated course recommendations, as well as accompanying explanations, which systematically exceed those produced by the former approach, in terms of both their quality and usability.

Keywords: information retrieval; recommender systems; explainable AI; course recommendation; career-based education; knowledge graphs; complexity science; knowledge representation

Check for updates

Academic Editor: Jose María Merigo

Received: 26 September 2025
Revised: 14 December 2025
Accepted: 9 January 2026
Published: 9 February 2026

1. Introduction

Post-secondary educational institutions—such as universities, vocational schools, and community colleges—are intended to provide curricula that help students to train for and eventually transition into their chosen careers [1–3]. This can be a challenging objective to achieve, however, because the set of relationships that exist among post-secondary school curricula and job markets is large, difficult to model, and evolves in real time [4–8]. These challenges contribute to the situation today in which mismatches are common between jobs that graduates are best prepared to perform and jobs that are in demand [9,10].

In the face of these challenges, knowledge representation engineering and complexity science can provide theoretical bases from which to approach modeling the education–career domain [11–15]. In this work, we are concerned with modeling this domain to solve the specific problem of quantifying the salience of a course in terms of its usefulness for obtaining and performing a specific job [16]. A key practical motivation for solving this problem is to allow for rank orderings of courses to be automatically defined, which can then be used by an application to perform the task of career-motivated course recommendation [17]. Delivering a course-salience quantification procedure within an application can benefit both students, who are trying to choose courses that align with their career goals, and curriculum developers, who are trying to better align courses and instructional strategies with job markets [18,19].

Furthermore, because choosing courses and developing curricula that align with job markets are such challenging and important tasks, one desirable property of a recommender system designed to help perform these tasks is trustworthiness [20]. And, while there exist multiple strategies for promoting trust in a recommender system, one strategy that is studied in the literature and that we further investigate in this work is to employ an interpretable architecture in the recommender system which enables a textual explanation to be generated alongside each recommended item that describes why the item is recommended [21,22].

This is all to say that the information retrieval problem that we are concerned with in this work can be described, from a user's perspective, as career-motivated explainable course recommendation, and then, from another perspective focusing more on the underlying implementation that will be required, the problem can be described as one of quantifying the salience of a given course in terms of its usefulness in obtaining and performing a specific job, and doing this in a way that is naturally interpretable so that there is a straightforward method that can be used to faithfully explain why a given course has been determined to have a certain salience.

One significant gap that is present in our existing knowledge of this problem concerns how effectively different interpretable information retrieval architectures may address the problem and, in particular, whether existing interpretable methods that can be applied to computing course salience, such as text-based similarity measures and document-ranking functions, represent workable solutions, or whether a more complex interpretable architecture that models rich heterogeneous relationships among jobs and courses using a knowledge graph might prove more effective. This question concerning the relative merits of these two classes of architectures for addressing the problem represents the main research question that it is the purpose of this work to investigate.

In order to investigate and answer this question, the present work is focused mainly on the second class of architectures, because the first class has been described in detail in the existing literature. Regarding the second class, our goal is to develop a method that is especially fitted to the problem that we have outlined above, and that effectively balances recommendation and explanation performance. Then, in the experiment that we describe in Section 4 below, we benchmark our novel method against existing methods of the first class.

The novel course-salience quantification method that we propose in this work requires, first, the construction of a network model of the real-world complex system that is defined by the set of relationships that exist among post-secondary school curricula and job markets. Then, using this model, our algorithm quantifies salience using the strength of the skill-dependency chains which link a course to the specified job of interest. Because our algorithm uses explicit skill-dependency paths in a heterogeneous network, or knowledge graph, our algorithm is more interpretable than many alternatives, such as embedding-based approaches [23,24]. The interpretability of our algorithm enables any application that implements it to generate an explanation alongside each course recommendation, which accurately describes why the course was chosen. Our work is therefore partially motivated by, and seeks to contribute to, the research field that studies explainable recommendation methods [21,22].

The combination of our network model of the complex system and our course recommendation procedure is summarized in Figure 1. (A complete description of the network schema and salience quantification procedure is provided in Section 3 below).

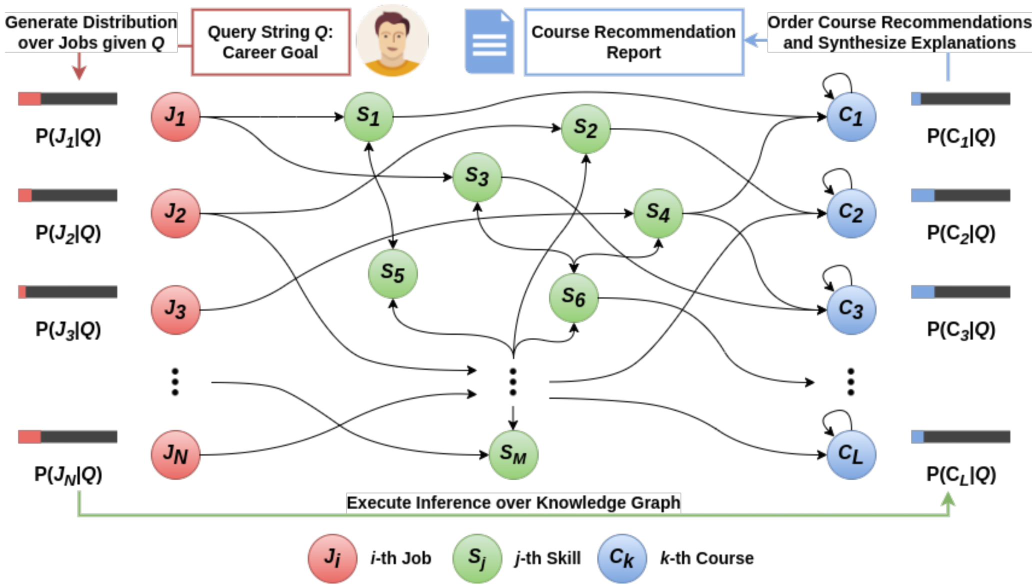


Figure 1. Education–career network and course recommendation procedure. To obtain a ranked list of career-motivated course recommendations, first, a user inputs search terms that describe their career goal, which we call search query Q. A document-ranking function is then applied to Q to generate a probability distribution over a set of candidate job advertisements J. This is followed by a random walk simulation from the jobs in J, across a network of skills S, toward a set of course syllabi C. The courses in C are ordered for recommendation by their probabilities of being the terminal step in the walk.

Our procedure for quantifying course salience is general and can in practice be applied to any subset of the education–career domain for which a knowledge graph (KG) can be constructed that conforms to our standard schema [25]. To evaluate the effectiveness of our framework (our proposed KG schema and salience quantification procedure), we conduct a case study. The area that we choose for the case study is the microelectronics industry, because it is an area which is critical to the economy and to society but also for which there are currently skilled worker shortages in the United States and elsewhere [26–29].

In our case study we leverage the expertise of fifty participants. Studies in the education–career domain have previously used surveys to evaluate information retrieval and ranking algorithms [30,31], and, in our own case study, we likewise employ a web-based survey instrument to evaluate our methods. Our survey allows each participant to select the microelectronics education–career subfield in which they have expertise, and then interactively rate course recommendations and explanations that are generated by our model and the baselines.

We analyzed the results of the case study in order to draw conclusions about the complex system modeling decisions and the algorithm design decisions that have the greatest effect on the quality and usability of the generated course salience scores and the generated explanations. And, by doing this, we are able to answer our research question, which asks whether existing interpretable methods that can be applied to computing course salience, such as text-based similarity measures and document-ranking functions, represent effective solutions to the problem, or whether our proposed method that models rich heterogeneous relationships among jobs and courses using a knowledge graph is more effective in terms of the quality and usability of the generated course salience scores and the generated explanations that it produces.

The remainder of this article is divided into the following five sections: Section 2: "Related Work", Section 3: "Methods", Section 4: "Evaluation", Section 5: "Results", and Section 6: "Discussion". A preliminary version of this work was presented as a short paper at iConference 2023 [32].

2. Related Work

The present section is divided into the following three subsections: Section 2.1: "Approaches to Course Recommendation", Section 2.2: "Challenges in Course Recommendation", and Section 2.3: "Knowledge Graph Architectures in Recommendation".

2.1. Approaches to Course Recommendation

Search and recommendation systems have been popularized in the education domain both due to the growing number of institutions adopting online learning and due to the development of new, entirely online educational platforms [33]. Platforms like edX, Coursera, and Udemy, which host massive open online courses (MOOCs), each have built-in course search and recommendation systems [34]. These systems are often designed to connect students to courses and other learning resources in a personalized manner [35,36]. Much of the research conducted on search and recommendation in education has focused on recommending learner pathways to students based on course content and students' interests [37–41]. Furthermore, recommendation systems have been proposed for e-learning personalization to accommodate differences in learning motivation and style [42,43], as well as methods designed to broaden or reinforce existing student knowledge [44]. Systems have been developed to support students learning English as a second language [45,46], some of which seek to recommend study materials that are at the appropriate difficulty level for a specific student [47].

In course recommendation systems, collaborative filtering (CF) is often used to recommend courses to a student based on the courses that were chosen by other students who have similar learner profiles [23,48], and, recently, CF approaches have demonstrated improvements through the incorporation of contextualized embeddings produced by large language models (LLMs) into course representations [15]. Also, in recent years, the proliferation of the use of generative LLMs in a number of information retrieval tasks (beyond only course recommendation) has occurred, and one of the advantages of generative LLMs in this context is their ability to synthesize information from multiple documents into a single response [49–51]. There has also been work on leveraging generative LLMs to perform document ranking, which is more applicable to the task of course recommendation [52].

Relatively sophisticated course recommendation systems have been previously constructed that take into account several types of constraints, such as pre- and corequisite constraints and scheduling constraints [53,54].

https://doi.org/10.3390/knowledge6010005

However, several significant challenges continue to confront the tasks of course search and recommendation. One core challenge, which the present work aims to address and which is described in greater detail in the next subsection, is the problem of ensuring that the design of an information retrieval architecture is well aligned with how a user is able to articulate their information need and, further, ensuring that the objective which is optimized to generate recommendations corresponds well with the conceptual objective of the system, which in this case is that a course should be recommended for a given input job because taking that course will help a student to obtain and perform the specified job.

2.2. Challenges in Course Recommendation

The most straightforward technique for performing course search is to use traditional query-based information retrieval (illustrated in Figure 2). However, if a user would like to find courses in a manner other than by providing search terms that directly describe the courses of interest, there are several other styles of course search and recommendation that have been studied which seek to overcome this inherent limitation of query-based IR that requires a user to know the terms contained in the syllabi of the target courses in advance.

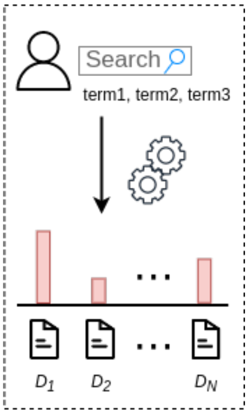


Figure 2. Query-based information retrieval. In query-based information retrieval, a document-ranking function is applied to a set of search terms to yield a probability distribution over a collection of target documents. When query-based IR is used for course search, the target documents are course syllabi, and the search terms provided by the user must relatively directly describe the course of interest. However, in many cases, a student will not be able to directly describe the course that they are looking for, rather, they will only be able to provide an indirect description, for example, in terms of what career the course should prepare them for.

These alternative approaches to course recommendation must be designed, first, by identifying a criterion by which courses should be chosen for recommendation and, second, by successfully operationalizing said criterion. Although the second of these two steps is the more challenging, a course recommendation system can fail if the first step is not given proper consideration, as observed in many instances in the contemporary literature. Consider that when designing a product or media recommendation system, the recommendation criterion likely should be engagement or interest: If a user buys a recommended product or watches a recommended movie, that likely should be considered a success [55]. However, in course recommendation, engagement by itself likely should not be the goal: A student should not take a course only because they may like it, but rather because the course will contribute to the student's achievement of some higher-level personal or professional goal [56,57]. Nevertheless, collaborative filtering architectures are often applied uncritically to the task of course recommendation, with the result being systems optimized to achieve user engagement rather than to help students achieve their goals [23].

Considering that the design objective of a course recommendation system should be to identify courses that will best help a student realize their professional goals, we can observe that this objective implies an interesting situation in which a user's information need is decoupled from the description that they are able to provide of that information need (illustrated in Figure 3). That is, a student's information need is course recommendations, but since their objective is to obtain a list of courses that will help them to achieve a specific career goal, then it is clear that the system input should be the relevant career goal, perhaps in the form of a job title. However, prior work [58] has found that the vocabulary that is used in a job advertisement (including the job title) is often distinct from the vocabulary that is used in the syllabi of courses that are most relevant to performing the job (this situation is illustrated in Figure 4).

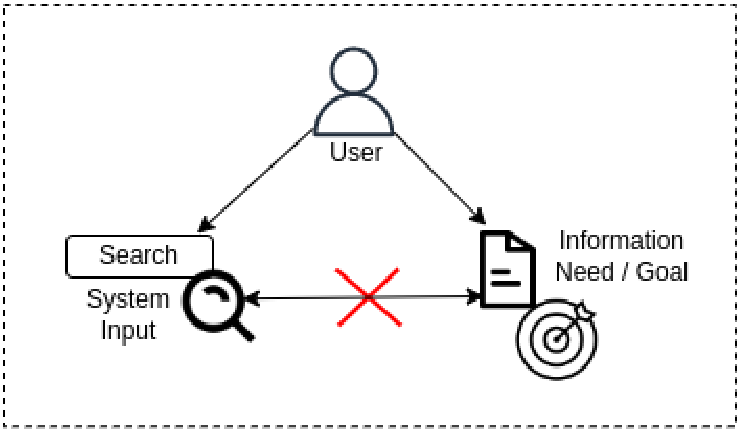


Figure 3. System input–information need decoupling problem. Illustration of decoupling between a user's information need and the terms in which they are able to describe their information need. In the case of career-motivated course search, the information need (a course recommendation) is only indirectly related to the terms in which the user is prepared to describe the information need (a description of a career goal).

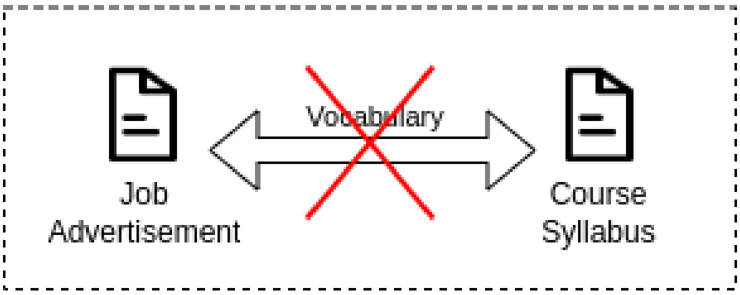


Figure 4. Vocabulary disagreement problem. Illustration of the vocabulary disagreement problem in which courses relevant to certain jobs are often described in very different terms than those jobs [58]. The vocabulary disagreement problem adds a second layer of challenges to the system input–information need decoupling problem that already occurs in career-motivated course search.

One approach to bridging the vocabulary gap from job advertisements to course syllabi then is to use an intermediate knowledge representation, such as a network of skills [2,59]. Ready-made skill networks are available, for example, from the O*NET OnLine platform [60] and the U.S. Bureau of Labor Statistics [61]. In spite of the challenges associated with career-motivated course recommendation in which educational trajectories—from courses toward careers—are modeled using skill pathways, there are several investigations on this topic that have suggested successful strategies for performing this style of modeling [38,40,62].

However, one research gap that remains when using an intermediate knowledge representation in a course recommendation system concerns how the knowledge representation can be most effectively integrated into the overall recommendation system. This research gap is described in greater detail in the following subsection, including the solutions that we investigate in the present work to address this gap.

2.3. Knowledge Graph Architectures for Recommendation

There is a significant body of work that has sought to make use of explicit, highly structured knowledge representations when performing information retrieval tasks such as course search and recommendation [17,63]. Existing knowledge representations often take the form of an inventory of entities, where each entity has multiple types of attributes, some of which may consist of relations to other entities, in which case the knowledge representation can be considered to be, or contain, a heterogeneous network or knowledge graph (KG) [64].

Contemporary approaches to recommendation that employ knowledge graphs often have, as their first step, a process of learning a vector representation for each entity in the relevant KG [65–68]. Recommendation of target entities (e.g., courses) can then be performed based on their proximity to source entities of interest (e.g., jobs) in the learned vector space [31,69,70]. There has been much recent work on trying to learn better vector representations for entities (nodes) in a knowledge graph, such as by using meta-paths which encode graph topological information including node and edge type [71–75]. However, while continuous feature representations of nodes in a knowledge graph can be extremely useful in improving the performance of a task like recommendation, attempts to make them transparent and explainable have tended to be post hoc; that is, the way explanations are generated using them has usually been independent of the recommendation procedure so that generated explanations do not necessarily describe the underlying recommendation process [76]. Therefore, one solution that has been suggested for exploiting the benefits of a knowledge graph data representation while also allowing for faithful explanation is to conduct inference directly over the graph itself [21,58], which is the direction taken in the present work.

In addition to using explicit reasoning over a knowledge graph to promote explainability, this approach has also been employed simply to help enforce a specific desired recommendation criterion: e.g., in career-motivated course recommendation to help enforce that recommended courses will help a student to perform their specified target career. This is done by modeling, within the KG, the skills required by specific jobs and taught by specific courses. A shortcoming of existing work [30], however, has frequently been a lack of richness in skill modeling within the KG, in which job–skill and skill–course relationships are represented, but interskill relationships are omitted. This is a problem because interskill relationships are needed to effectively overcome the "vocabulary disagreement problem" that is described in the previous subsection and illustrated in Figure 4. The knowledge graph that we describe and analyze in the present work models rich interskill relationships.

In summary, when it comes to leveraging a KG data structure as part of the overall architecture of a recommender system, and in particular a recommender system intended to perform the task of career-motivated explainable course recommendation, one aspect that is underdeveloped in the literature is how the KG can be best integrated into the overall recommender system architecture in a way that supports accurate explanation production while minimizing any sacrifice in performance on the corresponding recommendation task. The approach that we propose in this work—in which we integrate a KG into our recommender system that models rich interskill relationships among jobs and courses, and

https://doi.org/10.3390/knowledge6010005

which is described in complete detail in the following section—is intended to contribute to addressing this aspect, which is underdeveloped in the existing literature.

3. Methods

In the present section, we describe our proposed framework for career-motivated explainable course recommendation; this section is divided into the following two subsections: Section 3.1: "Knowledge Graph" and Section 3.2: "Recommendation and Explanation".

3.1. Knowledge Graph

The present subsection describes the architecture and construction of the knowledge graph that we use to generate course recommendations and explanations; this subsection is divided into the following three minor sections: Section 3.1.1: "Knowledge Graph Schema", Section 3.1.2: "Knowledge Graph Construction", and Section 3.1.3: "Modeling Edge Salience".

3.1.1. Knowledge Graph Schema

Network data structures are well suited to encode, store, and retrieve information in a variety of complex real-world problem domains [59,77–79]. The heterogeneous network, or knowledge graph (KG), that we employ in this work is designed to model relationships between jobs and courses through an intermediate subnetwork of skills. The connections between jobs and courses that are provided by skills are used to predict the usefulness of completing a particular course to performing a particular job. The KG schema that we use in this work is based on others that have recently been used successfully in the education–career domain [16,17,58]. Specifically, we define our KG as

$$G = (V,\ E,\ \tau,\ \sigma), \tag{1}$$

where V is a set of vertices, E is a set of edges, τ is an edge type function, and σ is an edge salience function (each schema component is also defined in Table 1). The set V is defined in terms of a set of jobs J, a set of courses C, and a set of skills S (see Table 2). In particular J, C, and S form a partition of V; that is J, C, and S are mutually disjoint subsets of V and their union is equal to V. The definitions of E and τ are given in the next minor section (Section 3.1.2: "Knowledge Graph construction"), and the definition of σ is given in the minor section following the next (Section 3.1.3 "Modeling Edge Salience").

Table 1. Schema components. Summary of the knowledge graph schema components.

Name	Description
V	The set of knowledge graph vertices
E	The set of knowledge graph edges
τ	The edge type function
σ	The edge salience function

Table 2. Data sets. Summary of the basic data sets used to construct the knowledge graph.

Name	Description
J	A collection of job advertisements
C	A collection of course syllabi
S	A collection of interrelated skill terms

3.1.2. Knowledge Graph Construction

The edges in E are ordered pairs of vertices of the form (v, u), indicating a directed link from $v \in V$ to $u \in V$. We equivalently denote an edge (v, u) in the more suggestive notation $v \to u$. In order to encode the meaning of edges, each edge in E is assigned a type. Formally, we define an edge type mapping function $\tau : E \to \mathfrak{T}$, where $\mathfrak{T}$ is the set of edge types, so that $\tau(v \to u) = \mathfrak{t}$ means that $\mathfrak{t}$ is the type of edge $v \to u$. We write $\tau(v \to u) = \mathfrak{t}$ equivalently as $v \xrightarrow{\mathfrak{t}} u$. We define $\mathfrak{T}$ as containing five edge types. In particular,

$$\mathfrak{T} = \{\texttt{req}, \texttt{sub}, \texttt{sup}, \texttt{tby}, \texttt{cse}\}, \tag{2}$$

where `req` stands for "requires", `sub` stands for "is subset of", `sup` stands for "is superset of", `tby` stands for "is taught by", and `cse` stands for "course self-edge". The `req` type applies to edges from jobs to skills. The `sub` and `sup` types apply to edges from skills to skills. The `tby` type applies to edges from skills to courses. And the `cse` type applies to the edges that map each course to itself.

Each edge in E is characterized by exactly one edge type in $\mathfrak{T}$. In particular, τ maps E to $\mathfrak{T}$ in the following manner:

- $j \xrightarrow{\texttt{req}} s$ when $j \in J$, $s \in S$, and the skill term s appears in the job posting j;
- $s \xrightarrow{\texttt{sub}} r$ when $s, r \in S$ and the skill s is a subclass of the skill r;
- $s \xrightarrow{\texttt{sup}} r$ when $s, r \in S$ and the skill s is a superclass of the skill r;
- $s \xrightarrow{\texttt{tby}} c$ when $s \in S$, $c \in C$ and the skill term s appears in the syllabus of course c;
- $c \xrightarrow{\texttt{cse}} c$ when just the condition $c \in C$ is met (course self-edges are used for bookkeeping).

If one of these five situations characterizes two vertices $u, v \in V$, then $u \to v \in E$. But if none of these five situations apply, then $u \to v \notin E$.

The knowledge graph construction procedure that we have described is intended to produce a KG which models rich interskill relationships among jobs and courses by including edges of types `sub` and `sup` in the KG. And we have chosen this particular construction procedure, which includes edges of these types, so that we will be able to answer our main research question which asks whether modeling rich interskill relationships among jobs and courses is a more effective solution for our task, or whether modeling only direct textual relationships between job advertisements and course syllabi, as described in Section 4 below, is a more effective solution.

3.1.3. Modeling Edge Salience

In addition to each edge $v \to u \in E$ having a type, our recommendation procedure requires an additional piece of information, which is the strength or importance (salience) of the relationship that connects v to u. To quantify an edge's salience, we assign a weight to each edge using an edge salience function defined as $\sigma : E \to \mathbb{Q}^+$, which is a piecewise function that is made up of five subfunctions, each corresponding to one edge type in $\mathfrak{T}$. σ maps a KG edge $v \to u \in E$ to a positive rational in the following manner:

- If the type of $v \to u$ is `req`, then $\sigma(v \to u)$ equals the number of times skill term u appears in job posting v normalized by the total times u appears in any job posting in J.
- If the type of $v \to u$ is `tby`, then $\sigma(v \to u)$ equals the number of times skill term v appears in syllabus u normalized by the total times v appears in any syllabus in C.

- If the type of $v \to u$ is `sub`, then $\sigma(v \to u)$ equals the mean of the weights of the edges outgoing from v, i.e.,

$$\sigma(v \to u) = \frac{\sum_{\{c: c \in C \wedge v \to c \in E\}} \sigma(v \to c)}{|\{c : c \in C \wedge v \to c \in E\}|} . \tag{3}$$

- If the type of $v \to u$ is `sup`, then $\sigma(v \to u)$ also equals the mean of the weights of the edges outgoing from v.
- If the type of $v \to u$ is `cse`, then $\sigma(v \to u)$ equals 1.

The formulas used to define the edge salience function for each edge type are intended to parallel common methods used in information retrieval to define the importance of a textual relationship, such as by using term frequency divided by the total number of documents in the corpus in which the term appears (TF–IDF) for the `req` and `tby` edges. Then, in the cases of the `sub` and `sup` edges, the averaging method employed is intended to balance the likelihood of transitioning from a skill node to a different skill node versus transitioning from a skill node to a course node.

The salience of each edge, as defined by σ, is used in the next subsection as an important component to specify the transition probabilities in the recommendation inference procedure. And the reason that we define the edge salience function to serve as a component within the overall KG recommendation inference procedure is to answer our main research question, which asks whether the heterogeneous knowledge graph approach to our task, as described in this section, exhibits more desirable behavior than existing IR approaches.

3.2. Recommendation and Explanation

The present subsection describes the algorithms which operate over the knowledge graph described in the previous subsection and which together with the KG constitute our system for career-motivated explainable course recommendation (see Figure 5 for an illustration of the system); the present subsection is divided into the following four minor sections: Section 3.2.1: "Random Walk Simulation", Section 3.2.2: "Course Recommendation", Section 3.2.3: "Explanation Procedure", and Section 3.2.4: "Explanation Template".

3.2.1. Random Walk Simulation

The recommendation method relies on a random walk simulation in which an edge $v \to u \in E$ with a greater salience (with salience given by $\sigma(v \to u)$) corresponds to a larger probability that a simulated walker will choose to traverse the edge. Executing a random walk over a knowledge graph in the manner we describe has previously been shown to lend itself well to explainable recommendation [80,81].

To precisely execute the simulation, the probability that a random walker on any vertex v at timestep t will have transitioned to vertex u at timestep $t+1$ must be defined. This can be done by normalizing each edge's weight, as assigned by the edge-salience function σ, by the sum of the weights of all edges outgoing from the edge of interest's source vertex; that is, for each $v, u \in V$, if $v \to u \in E$, then the probability of transitioning onto vertex u during any timestep, given that the walker had just been on vertex v, is computed by

$$P(u|v) = \frac{\sigma(v \to u)}{\sum_{\{w: v \to w \in E\}} \sigma(v \to w)} , \tag{4}$$

or, if $v \to u \notin E$, then $P(u|v) = 0$.

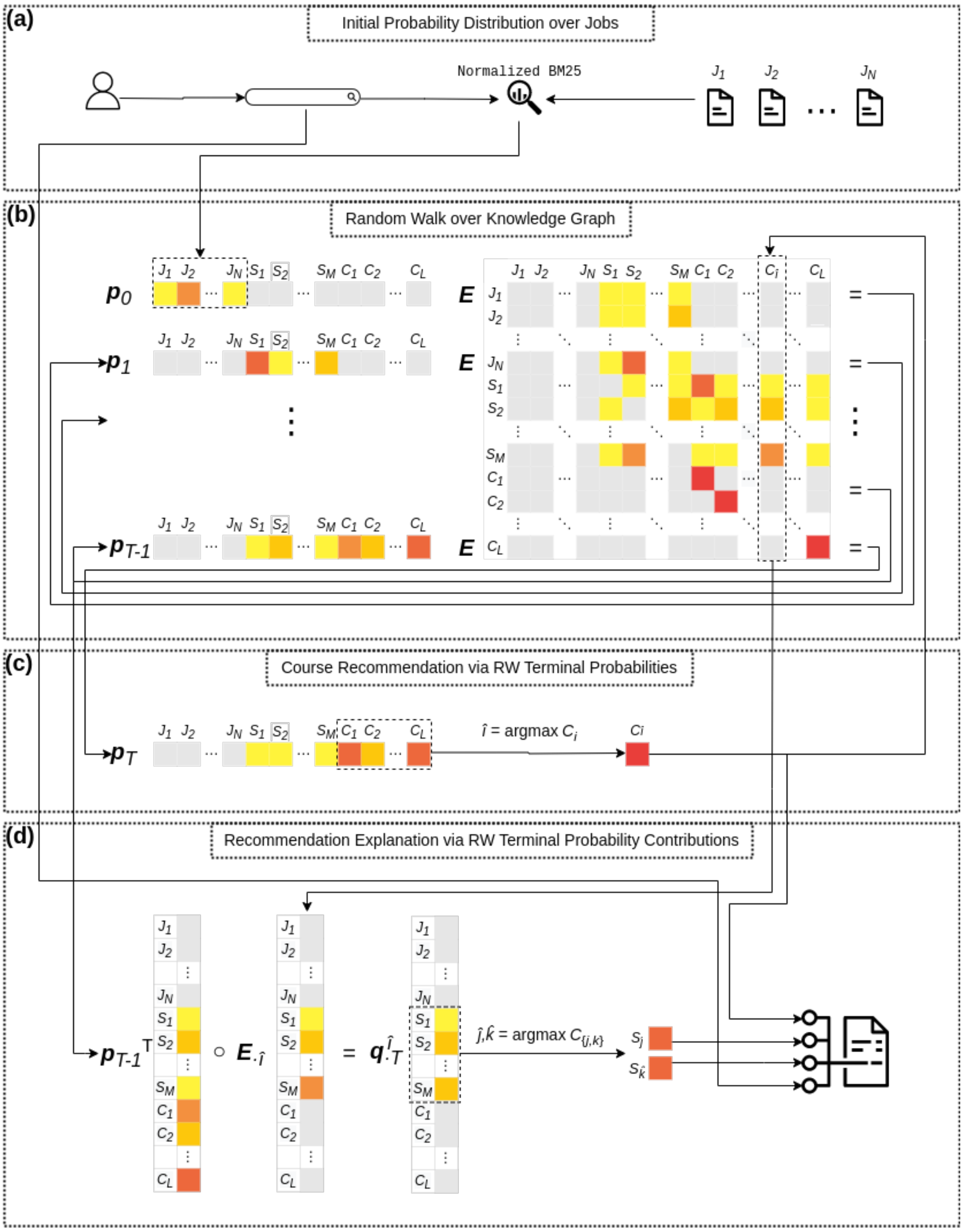


Figure 5. System architecture in detail. (**a**) The course recommendation procedure takes as input a user-supplied description of a career goal. The supplied search terms are then given as input to the BM25 document-ranking function in order to generate a probability distribution over the collection of job advertisements, and the distribution is denoted as p_0. (**b**) The probability distribution over job advertisements p_0 is then used as the initial probability distribution in a random walk simulation over the knowledge graph that starts from the jobs and traverses the skills network toward the courses. (**c**) The resulting output of the random walk simulation is the vector p_T, which is a probability distribution over all nodes in the graph after the terminal step. The course that has the highest probability of being the terminal state in the random walk is selected for recommendation. (**d**) The two skills that contributed the most to the recommended course being chosen serve as an explanation for why the recommended course was selected by the system.

All the information needed to conduct the random walk simulation is contained in G and the probability distribution we have just specified. However, in order both to describe the simulation more compactly and to suggest how the simulation is implemented, we define another object $\boldsymbol{E}$ which is a $|V| \times |V|$ transition probability matrix that is populated as follows. First, an arbitrary ordering is defined for the elements of V such that the i-th vertex in the ordering is denoted V_i. With this ordering in place, $\boldsymbol{E}_{ij}$ is given by $P(V_j|V_i)$, for all $0 \leq i, j < |V|$.

Once $\boldsymbol{E}$ is populated, the only item that is still needed in order to conduct the random walk is an initial probability distribution over the vertices in V. This initial distribution will be 0 for the vertices that correspond to courses and skills; the distribution's probability will be spread over the job vertices in V based on how closely each job corresponds to the user-supplied career goal. We define a row vector $\boldsymbol{p}_0$ to represent the initial probability distribution, in which each entry $\boldsymbol{p}_{0i}$ corresponds to the probability of the random walk beginning at vertex V_i. If $V_i \in C$ or $V_i \in S$, then $\boldsymbol{p}_{0i} = 0$; otherwise, $V_i \in J$ and $\boldsymbol{p}_{0i}$ is given as the normalized BM25 score of the job posting V_i calculated using the user-supplied name of their job of interest as the query string (see Figure 5a).

Having defined the initial probability distribution, the probability of a random walker being at any vertex in V after timestep t, for $0 < t \leq T$ (where T is the final timestep), is given by

$$\boldsymbol{p}_t = \boldsymbol{p}_{t-1}\boldsymbol{E} \tag{5}$$

See Figure 5b for an illustration of the random walk procedure through which $\boldsymbol{p}_T$ is recursively computed.

The random walk simulation that we have described serves as a mechanism by which course salience can be inferred, where a course is judged more salient with respect to a particular job if it is more densely connected to that job in the KG and less salient if it is less densely connected. Explanations for why highly salient courses are recommended can then be given in terms of the skills that are most likely to be traversed in the random walk which connect the job of interest to the course.

3.2.2. Course Recommendation

Finally, once $\boldsymbol{p}_T$ is obtained, the best course recommendation is given by $V_{\hat{\imath}}$, where

$$\hat{\imath} = \underset{\{i:V_i \in C\}}{\operatorname{argmax}} \, \boldsymbol{p}_{Ti} \,. \tag{6}$$

See Figure 5c for an illustration of this method for selecting the best course recommendation.

In addition to $V_{\hat{\imath}}$ being the single most relevant course, the ordering of the probabilities in $\boldsymbol{p}_T$ also define an ordering of all courses in C, so that all courses in C can be listed according to their relevance to the selected job of interest.

3.2.3. Explanation Procedure

Once a recommended course $V_{\hat{\imath}}$ is identified, the skills that most strongly link the course to the job of interest can also be found. The idea here is to determine the skill vertices that contributed the most probability to $V_{\hat{\imath}}$ being the terminal state in the random walk. This can be done, first, by defining a column vector $\boldsymbol{q}^{\hat{\imath}}_{\cdot T}$ of length $|V|$, where $\boldsymbol{q}^{\hat{\imath}}_{jT}$, which is the j-th entry in the vector, equals the probability of a random walker being in state V_j after step $T-1$ and transitioning to state $V_{\hat{\imath}}$ during the final step T of the random walk. $\boldsymbol{q}^{\hat{\imath}}_{\cdot T}$ is calculated as the element-wise product of the row vector containing the probabilities of being at any vertex at step $T-1$ transposed and the column vector containing the probabilities of transitioning into $V_{\hat{\imath}}$ from any vertex; that is,

$$\boldsymbol{q}^{\hat{\imath}}_{\cdot T} = {\boldsymbol{p}_{T-1}}^{\mathrm{T}} \circ \boldsymbol{E}_{\cdot\hat{\imath}} \,, \tag{7}$$

where $\boldsymbol{E}_{\cdot\hat{\imath}}$ is the $\hat{\imath}$-th column in the transition probability matrix $\boldsymbol{E}$ and contains the probabilities of transitioning from V_j into $V_{\hat{\imath}}$ for all $0 \leq j < |V|$.

Once $\boldsymbol{q}^{\hat{\imath}}_{\cdot T}$ is obtained, the skill that contributed the most probability to the recommended course $V_{\hat{\imath}}$ during the final timestep T is given by $V_{\hat{\jmath}}$, where

$$\hat{\jmath} = \underset{\{j:V_j \in S\}}{\operatorname{argmax}} \, \boldsymbol{q}^{\hat{\imath}}_{jT} \,. \tag{8}$$

In addition to $V_{\hat{\jmath}}$ being the skill that most strongly links $V_{\imath}$ to the originally selected job, the probabilities given in $\boldsymbol{q}^{\hat{\imath}}_{\cdot T}$ also define an ordering of all skills in S which encodes how strongly each skill contributed to $V_{\hat{\imath}}$'s recommendation. See Figure 5d for an illustration of this method for selecting two skill terms that explain a recommended course.

3.2.4. Explanation Template

Once a course has been selected for recommendation and its most relevant skill terms have been identified using the algorithms described in the preceding three minor sections, the question still remains of how to best present this information to a user. One common approach is to employ an explanation template [82,83], which is the direction taken in the present work. In particular, we use an explanation template that integrates the user-specified job of interest, an individual recommended course, and the course's two most relevant skill terms (see the far right side of Figure 5d for an illustration of how these four items are collected for presentation to the user).

It is easiest to describe our explanation template with an example. Therefore, with respect to a particular course recommendation, if we assume that the user-selected job of interest is "Electrical Engineer", the recommended course is "Real Time Bluetooth Networks Shape the World", and the two skills taught by the course that are judged most relevant to the job are "Display Technology" and "Distributed Computing Architecture", then the following explanation will be generated:

> In "Real Time Bluetooth Networks Shape the World" you will gain skills in "Display Technology" and "Distributed Computing Architecture" which are important for the job "Electrical Engineer".

The manner in which this particular example recommendation-plus-explanation would be displayed in our survey instrument, which is used in the evaluation of our system, is shown below in Section 4.3. Employing an explanation template of this type is intended to help a user to better judge the usefulness of course recommendations and to intelligently choose among a set of recommendations.

4. Evaluation

The present section describes the evaluation protocol that we designed and carried out in order to assess the proposed course recommendation framework, which we described in the previous section (Section 3: "Methods"). The current section is divided into the following three subsections: Section 4.1: "Data sets", Section 4.2: "Baselines", and Section 4.3: "Survey instrument".

4.1. Data Sets

As described in Section 3 above, the knowledge graph that we employ in this work is based on three data sets: a set of jobs advertisements J, a set of course syllabi C, and a set of skill terms S. In the previous section, in addition to outlining the construction of a knowledge graph G (which is built from J, C, and S), we presented a larger framework (which employs G) for performing career-motivated explainable course recommendation. The framework may be applied to any career domain; however, in order to focus the scope of our work—to make data collection, and hence system evaluation, feasible—we chose to construct the instances of J, C, and S, with which we would perform our system evaluation, by using data specific to one career domain. In particular, we focused on the microelectronics industry, because it is critical to the economy and to society, yet it represents an area of the labor market that continues to evolve rapidly and therefore may be difficult to align coursework with [26]. Furthermore, the microelectronics industry continues to grow worldwide, but especially in the United States where it has recently seen increased investment [27,28].Therefore, an investigation into information retrieval methods to aid in microelectronics education is currently particularly topical.

We do emphasize, however, that focusing our case study only on the microelectronics domain could be interpreted as a limitation of the present work, because it is possible that the relative performance of the different methods tested in our experiments could be different in another domain.

But, nevertheless, as stated above, choosing one domain is helpful in that it limits the scope of the present research, and, furthermore, keeping domain as a control variable in our experiments is a methodological decision which increases our confidence that the results we obtain are representative of the one domain which we do test. This would not be the case if we collected data sets of the same sizes as our microelectronics data sets but in which the samples in the data sets were drawn from many different domains.

For this project we collected four data sets: a jobs data set J, a courses data set C, and two skills data sets S_0 and S_1 (each data set is summarized in Table 3; we address the purpose of representing S using two separate skill sets below). All four data sets were collected during the summer of 2021.

Table 3. Data sets. Summary of the four data sets collected for and used in the present work.

Data Set	Contents	Download
J	1277 microelectronics industry job postings	File S1
C	1269 MOOC syllabi	File S2
S_0	695 technical skill terms manually chosen	File S3
S_1	12,133 technical skill terms automatically chosen	File S4

The jobs data set J was collected from the Indeed employment website and contains 1277 job postings for positions within the U.S. microelectronics industry. These job advertisements are intended to represent the wide variety of positions that are currently being hired for within the microelectronics industry and include openings for technicians, engineers, and managers.

The courses data set C was collected from three providers of massive open online courses (MOOCs): Coursera edX and Udemy The courses data set contains 1269 syllabi, of which 27% are on microelectronics topics and 21% are on science and technology topics not directly microelectronics related. To ensure a robust explainable recommendation method, the model must be capable of filtering out noise; therefore, the remaining 52% of the syllabi were for courses on random topics outside of microelectronics, science, and technology.

The skills data set S_1 contains 12,133 terms which are the names of all Wikipedia articles and categories that are within four steps of the Electrical Engineering category in Wikipedia's article–category graph. Thus the construction of the skills data set S_1 follows existing work [70,84] that has sought to automatically extract skills, and skill networks, from Wikipedia. In contrast to the automatically obtained S_1, we also experimented with a second skills data set S_0 which was constructed manually, and contains 695 skills which are related to the microelectronics industry.

The procedure of knowledge graph construction described in the previous section (Section 3: "Methods") requires exactly three data sets, J, C, and S; however, in the preceding paragraph we have described the collection of four data sets, J, C, S_0, and S_1. The reason that we collected two skills data sets with different properties is so that we can construct two knowledge graphs and compare the effectiveness of their different properties for performing the course recommendation and explanation tasks: We construct one KG that uses J, C, and S_0 (which is named KGS_0), and a second that uses J, C, and S_1 (which is named KGS_1). When constructing the first KG using skill data set S_0, there are no edges of type `sub` or `sup`. The result is a simpler graph topology. The second KG, which is constructed using skill data set S_1, takes its `sub`- and `sup`-type edges from Wikipedia's article–category structure.

One motivation for pursuing this particular setup is so that one question, in addition to our main research question, can be answered, which asks whether a simpler graph topology with more specific skill terms is better suited to our problem of career-motivated explainable course recommendation, or whether a KG that is more richly heterogeneous and has more, but less specific, skill terms is more effective.

4.2. Baselines

In order to evaluate our proposed explainable recommendation method, we compare it to two alternatives.

The first is the cosine similarity (CsSm) metric, where the two items that are compared are the course syllabi in C and the full text of the job post of interest. The features that are compared are the non-stopwords in the user-selected job post and in each of the course syllabi. The most frequently occurring non-stopwords in the syllabus of a recommended course serve as the explanation terms used to fill the explanation template.

The second alternative that we compare our system to is the widely used document-ranking function BM25, which takes a query string as input and assigns a score to each document in a corpus based on the document's relevance to the query string. We use the user-selected job title to serve as the query string, and the documents that are scored against the query string are the course syllabi in C. The BM25 baseline, like the CsSm baseline, uses the most frequently occurring non-stopwords in the syllabus of a recommended course as that course's explanation terms.

The motivation for selecting these two baseline architectures is that they are commonly used information retrieval methods which can be employed to model direct relationships between jobs and courses, and therefore are explainable. One potential limitation, however, of these architectures being applied to the particular IR problem that we study in this work is that the extent to which they may be able to overcome the *vocabulary disagreement problem*, which is described in detail above, is uncertain. And one component of the main research question that this overall work is intended to answer asks to what extent this is in fact the case.

4.3. Survey Instrument

Studies in the education–career domain have previously used surveys both to collect data for use in the construction of IR systems and to evaluate IR systems after they are built [30,31]. In the present work we employ a dynamic survey instrument for the latter purpose. The instrument that we designed and deployed is general enough to evaluate any explainable course search system that ranks courses based on their usefulness to a specified career, and that can explain its recommendations in terms of skills taught by a recommended course that are required by the specified career.

https://doi.org/10.3390/knowledge6010005

We distributed the survey by electronic mail on 22 November 2021 to all students who were at that time enrolled in a degree program within the Worcester Polytechnic Institute's School of Engineering. We accepted the first fifty responses ($N = 50$), and the fiftieth, final response was received two days after the survey was distributed. Informed consent was obtained electronically from each participant before the administration of the survey, and all participants were at least eighteen years old. Upon completion of the survey, each participant was compensated with a twenty dollar Amazon gift card. Prior to distributing the survey, we received ethics approval for this study from the Indiana University Human Research Protection Program. Approval was granted on 11 November 2021, and the study was assigned protocol number 13453.

One limitation of our evaluation is that our evaluators are undergraduate and graduate engineering students, which means that many of them will have direct experience with microelectronics education and the microelectronics job market, but they will inherently have less experience than microelectronics scientists and engineers.

We use the survey instrument which we describe in this subsection to evaluate four competing systems: two systems employing knowledge graph–based inference procedures and two baseline systems, one employing cosine similarity and the other using the BM25 document-ranking function.

In order to complete the system evaluation survey, each participant is first asked to provide several items of demographic information (Figure 6). The participant is then asked to specify the career goal for which they would like to receive course recommendations (Figure 7). The dynamic survey then retrieves the top ten course recommendations for the specified career goal that are generated by each of the four systems under evaluation. These forty courses are randomly shuffled together and presented to the participant one by one. For each course, the participant is asked to rank its relevance to the originally specified career, as well as the adequacy of the accompanying explanation; recommendation relevance and explanation adequacy are both rated on a four-point scale (Figure 8). Following the completion of these eighty questions, the next set of questions that the participant is asked concern the usability of the recommendations and explanations generated by each of the four systems under evaluation. The usability questions are divided into four sections, one for each system. Each section is prefaced with the top three course recommendations (with their explanations) that were generated by the relevant system (Figure 9), and these are followed by eight questions which assess the usability of the recommendations and six questions which assess the usability of the explanations; the fourteen usability questions are presented to the survey participant in random order (Figure 10).

In summary, the overall goal of the survey instrument's design is to allow our main research question, which is described in detail above in Section 1, to be answered. Answering this question requires a comparison to be made among methods that model direct relationships among jobs and courses, and the particular method that is described above in Section 3 which supports the modeling of rich interskill relationships among jobs and courses. And, in order to compare these methods, the reason that human evaluators are needed is that in the newly collected data sets which are used in the evaluation gold annotations that give the salience of each course with respect to each job are not present.

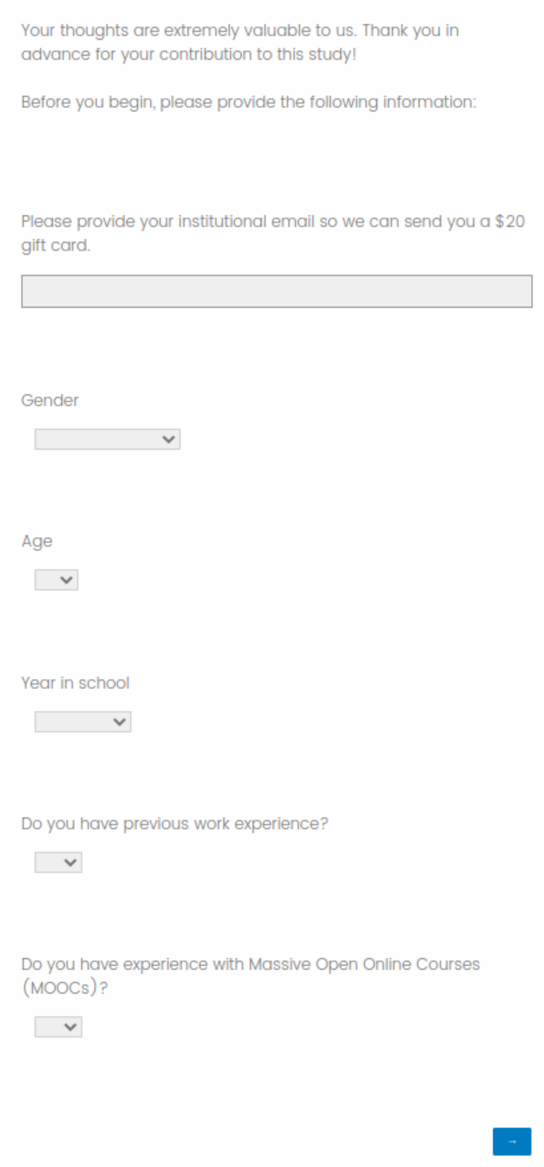


Figure 6. Survey instrument's demographic questions. The first screen in the survey asks the participant to answer several demographic questions, including providing gender, age, and year in school. The participant is also asked if they have previous work experience and if they have previous experience with massive open online courses (MOOCs).

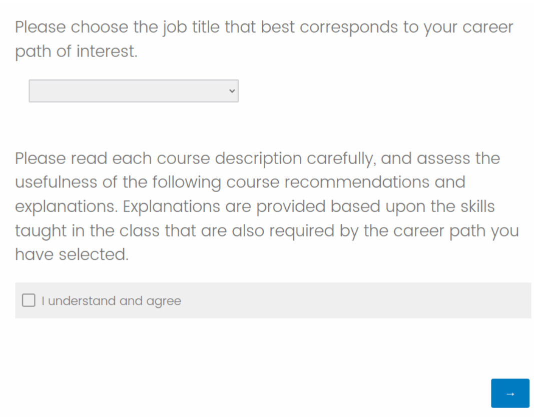


Figure 7. Survey instrument's career-goal question. The second screen in the survey asks the participant to input the career goal for which they would like course recommendations to be retrieved.

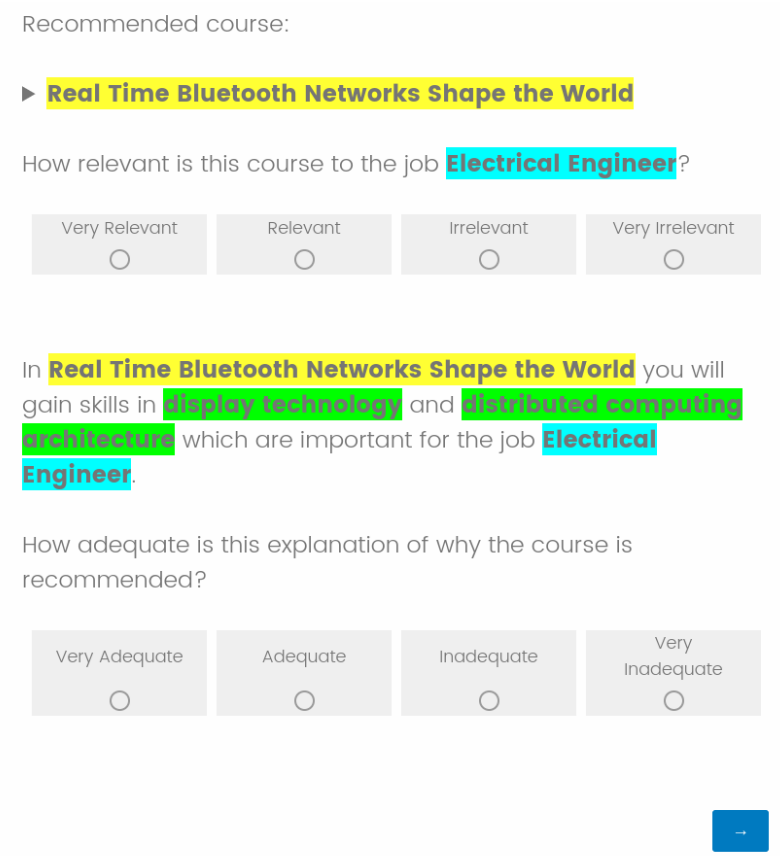


Figure 8. Survey instrument recommendation relevance and explanation adequacy questions. For each course recommendation and its accompanying explanation generated with respect to the career goal specified by the participant on the second screen of the survey, the participant is asked to rate the recommendation's relevance and the explanation's adequacy.

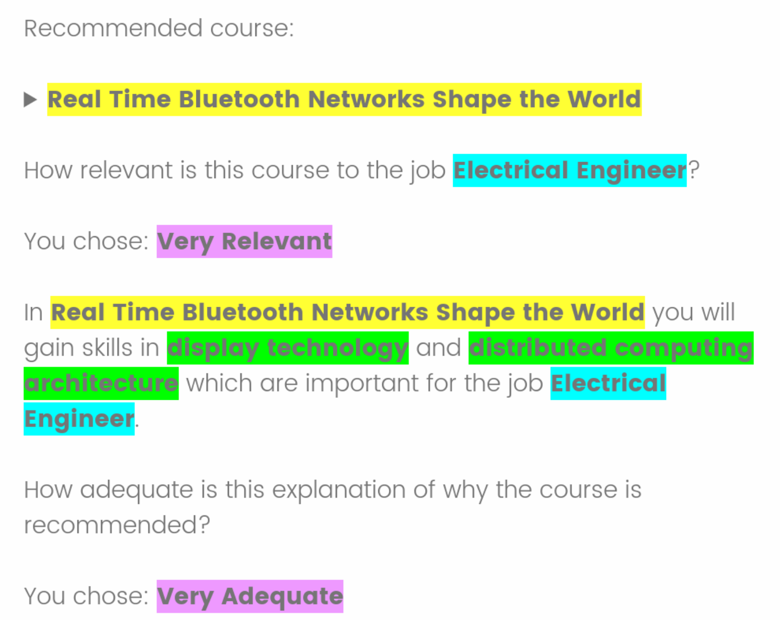
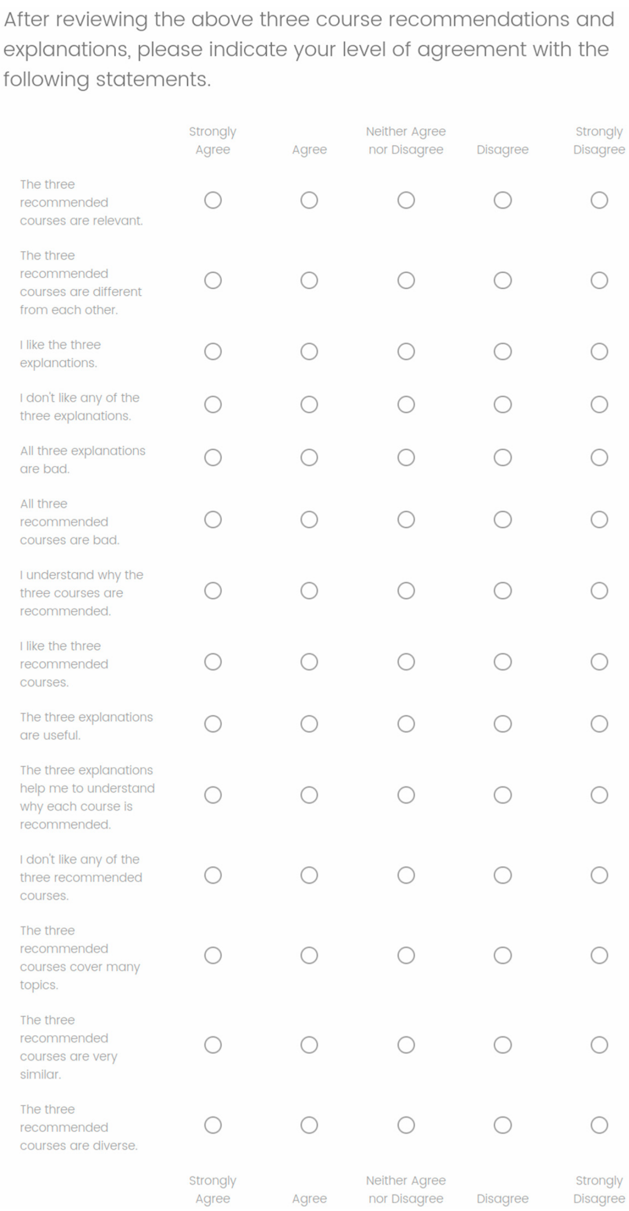


Figure 9. Survey instrument usability questions preface. The usability part of the survey is divided into four sections corresponding to the four systems under evaluation. Each section is prefaced by the top three ranked courses chosen by the relevant system, and then the subsequent usability questions are posed with respect to the three given recommendations. This figure shows one of three preface recommendations with its explanation.


After reviewing the above three course recommendations and explanations, please indicate your level of agreement with the following statements.

	Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree
The three recommended courses are relevant.	○	○	○	○	○
The three recommended courses are different from each other.	○	○	○	○	○
I like the three explanations.	○	○	○	○	○
I don't like any of the three explanations.	○	○	○	○	○
All three explanations are bad.	○	○	○	○	○
All three recommended courses are bad.	○	○	○	○	○
I understand why the three courses are recommended.	○	○	○	○	○
I like the three recommended courses.	○	○	○	○	○
The three explanations are useful.	○	○	○	○	○
The three explanations help me to understand why each course is recommended.	○	○	○	○	○
I don't like any of the three recommended courses.	○	○	○	○	○
The three recommended courses cover many topics.	○	○	○	○	○
The three recommended courses are very similar.	○	○	○	○	○
The three recommended courses are diverse.	○	○	○	○	○
	Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree

Figure 10. Survey instrument usability questions. The survey poses usability questions with respect to the course recommendations, and their accompanying explanations, which are produced by each of the four systems under evaluation. For each system, eight questions are posed concerning the usability of the generated course recommendations (adapted from [85]) and six concerning the usability of the explanations (adapted from [86]). These fourteen usability questions are presented to the survey participant in random order, such as illustrated at left.

5. Results

In the current section we present the results of the evaluation, which was carried out according to the evaluation protocol described in the previous section (Section 4). As stated in the protocol, the evaluation was conducted via a user study in which fifty students participated. The fifty participant responses (which are the basis of our study's results) are provided as a supporting information file (File S5).

From each participant, at the start of the user study, we collected a profile consisting of six items of information. We summarize this participant profile information in Table 4. Using this information, we assess several between-subject demographic factors with respect to the perceived usability of both the recommendations and the explanations produced by each system. The results of this assessment are a component of the full study results reported below.

Table 4. Summary of study participant profiles.

	Attribute	Frequency	Fraction
Gender	Male	34	0.68
	Female	13	0.26
	Prefer not to answer	3	0.06
Age	18	8	0.16
	19	11	0.22
	20	9	0.18
	21	8	0.16
	22	5	0.10
	⋮	⋮	⋮
	24	3	0.06
	25	1	0.02
	26	2	0.04
	⋮	⋮	⋮
	31	1	0.02
	⋮	⋮	⋮
	34	1	0.02
	⋮	⋮	⋮
	51	1	0.02
Year in school	Freshman	9	0.18
	Sophomore	16	0.32
	Junior	6	0.12
	Senior	10	0.20
	Master's	6	0.12
	PhD	3	0.06
Work experience	Yes	42	0.84
	No	8	0.16
Experience with MOOCs	Yes	8	0.16
	No	42	0.84
Job of interest	Electrical Engineer	39	0.78
	Hardware Design Engineer	8	0.16
	Systems Engineer	3	0.06

The remaining part of the present section is divided into the following two subsections: Section 5.1: "Recommendation and Explanation Performance" and Section 5.2: "Recommendation and Explanation Usability". Each subsection reports and compares the results for each of the four systems (i.e., BM25, CsSm, KGS_0, and KGS_1).

5.1. Recommendation and Explanation Performance

To assess system *performance*, evaluators scored each of the top ten course recommendations, and the accompanying explanations, that were generated by each of the four systems. The performance results for the recommendation task and for the explanation task are presented in Table 5 and Table 6, respectively.

Table 5. Recommendation performance. BM25 refers to the widely used document-ranking function, CsSm stands for cosine similarity, KGS_0 stands for knowledge graph with manually created skills data set, and KGS_1 stands for knowledge graph with skills extracted from Wikipedia. Prec is an abbreviation for precision, and NDCG stands for normalized discounted cumulative gain. The @5 and @10 suffixes indicate that a metric was calculated using the top 5 or 10 course recommendations generated by each system for each input job title.

	Prec@5	**Prec@10**	**NDCG@5**	**NDCG@10**
CsSm	0.57	0.53	0.49	0.48
BM25	0.76	0.80	0.67	0.69
KGS_0	0.89	0.89	**0.75**	**0.75**
KGS_1	**0.91**	**0.90**	0.73	0.74

Table 6. Explanation performance. Abbreviations are defined in the caption of Table 5.

	Prec@5	**Prec@10**	**NDCG@5**	**NDCG@10**
CsSm	0.42	0.49	0.44	0.46
BM25	0.32	0.38	0.36	0.40
KGS_0	0.70	0.71	0.64	0.63
KGS_1	**0.75**	**0.73**	**0.69**	**0.68**

Focusing first on the recommendation task, we observe that, for each evaluation metric, the KG-based systems outperform the text-based systems. Between the two KG systems, however, there is not necessarily an overall best performer: the KGS_1 system outperforms KGS_0 for precision, but, for NDCG (normalized discounted cummulative gain), the roles are reversed with KGS_0 doing better than KGS_1. This result suggests that a larger, more general and more richly connected knowledge graph (KGS_1) may be better at ranking many relevant courses highly, but without necessarily placing extremely relevant courses at the very top of the ranking; on the other hand, a knowledge graph that is manually created for a specific recommendation domain (KGS_0) may be more effective at pushing a few highly relevant courses to the top of the ranking but may also permit a greater number of less relevant courses to appear high in the ranking (without pushing them to the very top).

Turning our attention from the recommendation task to the explanation task, we observe that, for each evaluation metric, the KG-based systems again outperform the text-based systems. This result suggests that a KG architecture is more suited to the task of explainable recommendation than are architectures based on direct textual similarity. Furthermore, where in the case of recommendation performance the KGS_1 system achieved higher precision but the KGS_0 system achieved higher NDCG, in the case of explanation performance the KGS_1 system outperformed the KGS_0 system in terms of *both* precision *and* NDCG. This suggests that a larger, more general and more richly connected knowledge graph may be more useful than a smaller, domain-specific, manually constructed KG in applications that require explanation.

These performance results for the recommendation task and for the explanation task when considered together provide an answer to the first part of our main research question which concerns quality of outputs of the tested methods (while the second part of our main research question concerns usability of outputs and is answered in the following subsection). The answer to the first part of our research question, in brief, is that the quality of the outputs of the methods which model direct relationships between jobs and courses is lower than the quality of the outputs of the methods which model rich interskill relationships between jobs and courses. This means that modeling rich interskill relationships in the way described in this work increases explanation performance without sacrificing performance on the underlying career-motivated course recommendation task. The significance and contributions of these results are described below in Section 6.

5.2. Recommendation and Explanation Usability

While, to assess system performance, evaluators scored each of the top ten course recommendations, and the accompanying explanations, that were generated by each of the four systems; to assess system *usability*, evaluators additionally reported their level of agreement with a series of statements related to the usability of the top three recommendations and explanations that were generated by each of the four systems. The usability results for the recommendation task are reported in Table 7, and the usability results for the explanation task are reported in Table 8. These results are elaborated in the following two minor sections: Section 5.2.1: "Recommendation Usability" and Section 5.2.2: "Explanation Usability". Then, the final minor section of this subsection briefly summarizes the complete usability results: Section 5.2.3: "Usability Summary".

Table 7. Recommendation usability.

Recommendation Usability Proposition	Average Level of Agreement			
	CsSm	BM25	KGS_0	KGS_1
"I like the three recommended courses."	0.42	0.58	0.70	0.74
"The three recommended courses are relevant."	0.45	0.68	0.72	0.73
"I don't like any of the three recommended courses."	0.55	0.33	0.30	0.27
"All three recommended courses are bad."	0.47	0.35	0.28	0.26
"The three recommended courses are different from each other."	0.69	0.59	0.66	0.62
"The three recommended courses cover many topics."	0.62	0.63	0.62	0.71
"The three recommended courses are diverse."	0.62	0.54	0.60	0.63
"The three recommended courses are very similar."	0.45	0.56	0.52	0.46
	Chronbach's α			
	0.77	0.82	0.65	0.85

Table 8. Explanation usability.

Explanation Usability Proposition	Average Level of Agreement			
	CsSm	BM25	KGS_0	KGS_1
"I understand why the three courses are recommended."	0.44	0.60	0.69	0.71
"The three explanations help me to understand why each course is recommended."	0.46	0.41	0.58	0.56
"I like the three explanations."	0.48	0.40	0.54	0.52
"The three explanations are useful."	0.46	0.40	0.55	0.57
"I don't like any of the three explanations."	0.52	0.58	0.40	0.45
"All three explanations are bad."	0.52	0.58	0.42	0.39
	Chronbach's α			
	0.90	0.88	0.84	0.93

5.2.1. Recommendation Usability

Of the eight statements that compose the recommendation-usability survey instrument, four have a clear desirability polarity. Of those four statements, two are clearly desirable for a course recommendation. And, in the results, we observe that, for those two statements, the four explainable recommendation systems have the same ordering in terms of average level of agreement. In particular, the ordering is, from greatest to least average level of agreement: KGS_1, KGS_0, BM25, CsSm. Furthermore, this ordering is exactly reversed for the two statements that are clearly undesirable. These usability results tend to agree with the performance results for the course recommendation task.

The four remaining statements that compose the recommendation-usability survey instrument are not intrinsically desirable or undesirable, but are related to the variability of the course recommendations that are generated by a system. In contrast to the the single ordering of systems that is observed for each of the first four usability statements, we here observe that each of the final four statements has a unique system ordering. This result implies that an evaluator's perception of the variability of recommended courses may be more subjective than their perception of attributes of recommended courses that have clear desirability polarity. In the results for the final four statements, some general trends do, however, appear: For example, CsSm and KGS_1 tend to appear near the top of the orderings, BM25 at the bottom, and KGS_0 in the middle.

In addition to tabulating the average level of agreement to the eight course-recommendation usability propositions, we also performed several statistical tests to further analyze recommendation usability. A mixed-model ANOVA with a Greenhouse–Geisser correction was used to evaluate associations between recommendation usability, recommender system, and three demographic factors: gender, year in school, and previous experience with MOOCs.

The results of the mixed model ANOVA showed a significant main effect of recommender system on course-recommendation usability (i.e., $F(2.164, 69.24) = 6.38$, $p = 0.002$, ${\eta_p}^2 = 0.17$). Main effects of between-subject demographic factors were not significant. The lack of significant main effects for the between-subject variables indicated that the recommendation usability did not vary significantly across the levels of each of the factors. Additionally, no significant interactions were found for the mixed-model ANOVA with respect to recommendation usability.

Post hoc analyses were performed for the main effect of recommender system on recommendation usability in order to compare the performance of the four systems. Post hoc tests using the Bonferroni correction showed the KGS_0 algorithm ($M = 3.84$, $SD = 0.63$) and KGS_1 algorithm ($M = 3.94$, $SD = 0.73$) to have the highest recommendation usability. There was no significant difference in the mean scores of recommendation usability between the KGS_0 algorithm and the KGS_1 algorithm ($p = 1.00$). The performance of the BM25 algorithm ($M = 3.58$, $SD = 0.73$) was not significantly lower than the KGS_0 algorithm ($p = 0.199$). However, the KGS_1 algorithm had a significantly higher recommendation usability than the BM25 algorithm ($p = 0.008$). Both the KGS_0 algorithm ($p = 0.001$) and KGS_1 algorithm ($p < 0.001$) had significantly higher recommendation usability than the CsSm algorithm ($M = 2.85$, $SD = 0.90$). Finally, the BM25 algorithm and CsSm algorithm did not differ significantly based on recommendation usability ($p = 0.121$).

5.2.2. Explanation Usability

Each of the six statements that compose the explanation-usability survey instrument has a clear desirability polarity. Of those statements, the first four are clearly desirable for a generated explanation to exhibit. We observe that, for those four statements, the orderings of the recommender systems by average level of agreement (to those statements) exhibit several patterns. In each of the orderings, KGS_1 and KGS_0 appear in the first half and BM25 and CsSm appear in the second half. This pattern also appears in reverse in the orderings for the two statements that are clearly undesirable. These observations suggest that the KGS_1 and KGS_0 recommender systems may produce more usable explanations than the BM25 and CsSm systems, even if it is not clear which system in these two groups is superior within its own group.

In addition to tabulating the average level of agreement to the six explanation usability propositions, we also performed several statistical tests to further analyze explanation usability. A mixed-model ANOVA with a Greenhouse–Geisser correction was used to evaluate associations between usability of explanations, explainable recommender system, and three demographic factors: gender, year in school, and previous experience with MOOCs.

The results of the mixed-model ANONA showed a significant main effect of explainable recommender system on explanation usability (i.e., $F(3, 96) = 3.57$, $p = 0.017$, ${\eta_p}^2 = 0.10$). As in the assessment of recommendation usability, between-subject demographic factors of gender, year in school, and previous experience with a MOOC showed no significant main effects; explanation usability did not vary significantly across the levels of each of the between-subject factors. Additionally, no interactions were significant in the mixed-model ANOVA of explanation usability.

Post hoc analyses were performed for the main effect of system on explanation usability in order to compare the performance of the four explainable recommender systems. Post hoc analyses with a Bonferroni correction showed that the KGS_0 algorithm ($M = 3.35$, $SD = 0.79$) and the KGS_1 algorithm ($M = 3.34$, $SD = 1.04$) did not differ significantly in explanation usability ($p = 1.00$). However, the KGS_1 algorithm had a significantly higher explanation usability score than both the CsSm algorithm ($M = 2.87$, $SD = 0.94$, $p = 0.087$) and the BM25 algorithm ($M = 2.77$, $SD = 0.96$, $p = 0.003$). The KGS_0 algorithm had a significantly higher explanation usability score than the BM25 algorithm ($p = 0.022$) but did not differ significantly from the CsSm model ($p = 0.141$). Finally, the CsSm algorithm and the BM25 algorithm did not differ significantly ($p = 1.00$).

5.2.3. Usability Summary

The usability results for the recommendation task and for the explanation task, when considered together, provide an answer to the final part of our main research question which asks (independent of the *quality* of the model outputs that are generated during the evaluation) how can the *usability* of the generated outputs, and therefore the usability of the different models, be compared and characterized. The answer to this part of the research question, in brief, is that the outputs of the methods which model direct relationships between jobs and courses are less usable than are the outputs of the methods which model rich interskill relationships between jobs and courses. This means that modeling rich interskill relationships in the manner described in this work increases both recommendation and explanation usability. The significance and contributions of these results are described in the discussion section below (Section 6).

6. Discussion

The present section is divided into three subsections: Section 6.1 “Contributions”, Section 6.2: “Limitations”, and Section 6.3: “Future Work”.

6.1. Contributions

In much of the prior work on search-based information retrieval, a user must provide search terms that directly describe the information item that they are looking for. On the other hand, there also exists work on recommendation systems that do not require search terms as input but rather employ a collaborative filtering algorithm applied to an existing user profile [23]. The goal of collaborative filtering algorithms is to maximize user engagement, which makes them suitable for an application such as media recommendation [55,87–89]. There are, however, some applications of recommender systems in which simply holding a user's attention should not be the system's main objective. Course recommendation is one such application, since, although it is desirable that a student be interested in a recommended course, likely even more important is that a recommended course helps a student to achieve some higher-level goal, such as a career goal [24,56].

The present work is partially motivated by this observation and hence has focused on developing a system that incorporates aspects of both existing course search systems and course recommendation systems into a framework for ranking courses based on their usefulness to achieving a specific career goal. The usefulness of a course to achieving a career goal is quantified in terms of the skills taught by the course which are either directly required by the specified career goal or are related to other skills that are required by the career goal. The key data-architectural component of our system is a heterogeneous network, or knowledge graph (KG), that models relationships among courses and the skills that they teach, and jobs and the skills that they require. By modeling our problem using a KG, we were able to develop and apply a novel network analysis procedure (to quantify course salience with respect to any job), which is a general approach that has also been taken to complex systems problems in a variety of other domains [90–92].

And, when we undertake the task of modeling the career–education domain using a KG, the main goal that we have sought to achieve is to answer one main research question, which is described in detail in Section 1 above, but, to summarize, the question asks (in terms of our particular recommendation and explanation tasks): are existing interpretable methods that model direct relationships between jobs and courses more effective, or is modeling indirect interskill relationships more effective, where effectiveness refers to the quality and usability of the generated outputs.

As described in Section 5 above, we found that the latter approach proved more effective on the recommendation and explanation tasks in terms of both quality and usability. A theoretical implication of these results is that contextualizing job listings and course syllabi within a network of skills is one effective solution to the problem of the frequent lack of direct overlap of vocabulary between job postings and the syllabi of courses that are useful to performing those jobs [58], which can be described as the *vocabulary disagreement problem*. This theoretical implication of the results also leads to a practical implication which is that, when an information retrieval system is being developed to perform the task of career-motivated explainable course recommendation (for a practical purpose such as helping students to align their coursework with their career goals), it follows that employing the KG methods that we have investigated in this work will improve the quality and usability of the IR system outputs. Furthermore, the implications of this work are in line with recent work in other complex systems domains where heterogeneous-network analyses have yielded results which could not be obtained by simpler and more direct approaches [77,93,94].

In light of our observation of the effectiveness of skill networks in our experiments, we have investigated what properties of these networks are most important for improving the performance and usability of course recommendations and their explanations. In particular, one question that we were able to answer, in addition to our main research question, was: what are the different influences of a more domain-specific and hand-tailored skills data set (i.e., S_0) versus a less specific but larger and more richly heterogeneous skills data set (i.e., S_1) when constructing a KG that will be used in career-motivated explainable course recommendation?

Some of the performance and usability measures rank both the larger and more heterogeneous KG (i.e., KGS_1) and the smaller but domain-specific KG (i.e., KGS_0) as generally comparable. The one area, however, where KGS_1 clearly exceeds KGS_0 is on its performance for the explanation generation task. This result implies that explainability is one dimension where using computational methods to create a larger and more richly heterogeneous knowledge graph yields higher-quality and more usable outputs.

6.2. Limitations

One limitation of the present work is that the evaluation is done only for the subset of the entire education–career domain which corresponds to microelectronics-related courses and jobs. This may be interpreted as a limitation because it is possible that the relative performance of the different methods tested in our experiments could be different in another domain. Nevertheless, choosing one domain is helpful in that it limits the scope of the present research, and, furthermore, keeping domain as a control variable in our experiments can increase our confidence that the results we obtain are representative of the one domain which we do test. This would not be the case if we collected data sets of the same sizes as our microelectronics data sets but in which the samples in the data sets were drawn from many different domains. Furthermore, the methods that we describe in Section 3 above are agnostic to domain, and testing additional domains may be done in future work.

Another limitation of the present work is that the human evaluators who took part in the study are undergraduate and graduate engineering students, which means that while many of them will have direct experience with microelectronics education and with the microelectronics job market, as a group they will inherently have less experience than scientists and engineers who are currently working in the microelectronics field. Therefore it remains an open question the extent to which the results of our experiments would be different if the evaluators had greater experience in the microelectronics domain.

6.3. Future Work

In the future, we plan to investigate improvements to our network analysis procedure that will allow for a larger number of constraints to be taken into consideration when generating course recommendations. Currently, the only input that the network analysis procedure accepts is a career goal; however, the procedure may be more useful if it were able to accept and make use of more information such as a student's current education level and skill proficiencies [34]. This information would allow courses to be recommended that teach skills which are relevant to a student's career goal and which the student has not yet mastered. To streamline the collection of this additional information from a user during a future user study, another planned improvement is to design a frontend that can perform a short interactive evaluation of a user, before the application advances to the main search and recommendation interface. From this evaluation, the user's competence in the skills that are most relevant to their career goal will be estimated [7]. Then, with a greater understanding of a student's current competencies, course recommendation will be better informed.

This line of future work will further develop upon the theoretical and practical advancements that have been made in the present work in the field of career-motivated explainable course recommendation, and in the broader fields of complexity science and knowledge representation.

Supplementary Materials: The following supporting information can be downloaded at https://www.mdpi.com/article/10.3390/knowledge6010005/s1, File S1. Job data set J. Structure and usage described in Section 3: "Methods"; specific content described in Section 4.1: "Data sets". File S2.

Course data set C. Structure and usage described in Section 3: "Methods"; specific content described in Section 4.1: "Data sets". File S3. Skill data set S_0. Structure and usage described in Section 3: "Methods" for a general skill data set S; specific content of S_0 described in Section 4.1: "Data sets". File S4. Skill data set S_1. Structure and usage described in Section 3: "Methods" for a general skill data set S; specific content of S_1 described in Section 4.1: "Data sets". File S5. Evaluator responses. Structure described in Section 4.3: "Survey instrument"; content described in Section 5: "Results".

Author Contributions: Conceptualization: J.S., R.M., T.R., P.C.S., and X.L.; data curation: J.S. and R.M.; formal analysis: J.S., R.M., and T.R.; funding acquisition: X.L.; investigation: J.S. and R.M.; methodology: J.S., R.M., T.R., P.C.S., and X.L.; project administration: P.C.S. and X.L.; resources: J.S., R.M., and T.R.; software: J.S., R.M., and T.R.; supervision: P.C.S. and X.L.; validation: J.S., R.M., and T.R.; visualization: J.S.; writing—original draft: J.S., R.M., and T.R.; writing—review and editing: J.S. All authors have read and agreed to the published version of the manuscript.

Funding: We acknowledge support from the U.S. Department of Defense [Contract No. W52P1J-22-9-3009]. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Department of Defense or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation here on.

Institutional Review Board Statement: The study was conducted in accordance with the Declaration of Helsinki, and approved by the Institutional Review Board of Indiana University Bloomington on 11 November 2021 as protocol number 13453.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The original contributions presented in this study are included in the article and Supplementary Materials. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Radermacher, A.; Walia, G.; Knudson, D. Investigating the Skill Gap between Graduating Students and Industry Expectations. In *Proceedings of the Companion Proceedings of the 36th International Conference on Software Engineering (ICSE), Hyderabad, India, 31 May–7 June 2014*; Association for Computing Machinery: New York, NY, USA, 2014; pp. 291–300. [CrossRef]
2. Börner, K.; Scrivner, O.; Gallant, M.; Ma, S.; Liu, X.; Chewning, 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]
3. Igoche, B.I.; Matthew, O.; Olabanji, D. A Structural Causal Model Ontology Approach for Knowledge Discovery in Educational Admission Databases. *Knowledge* **2025**, *5*, 15. [CrossRef]
4. Martinez, W. How science and technology developments impact employment and education. *Proc. Natl. Acad. Sci. USA* **2018**, *115*, 12624–12629. [CrossRef] [PubMed]
5. Papoutsoglou, I. Minding the Gap between Labor and Educational Markets by Enhancing Standards with Semantic Web Techniques. 2019. Available online: https://www.openarchives.gr/aggregator-openarchives/edm/rep_ihu/000180-11544_29310 (accessed on 1 July 2021).
6. Zalat, M.M.; Hamed, M.S.; Bolbol, S.A. The experiences, challenges, and acceptance of e-learning as a tool for teaching during the COVID-19 pandemic among university medical staff. *PLoS ONE* **2021**, *16*, e0248758. [CrossRef]
7. Badie, F.; Rostomyan, A. Competency Mapping as a Knowledge Driver in Modern Organisations. *Knowledge* **2025**, *5*, 13. [CrossRef]
8. Dolgonosov, B.M. Modeling the Knowledge Production Function Based on Bibliometric Information. *Knowledge* **2025**, *5*, 7. [CrossRef]
9. Johnson, B.; Ulseth, R. Development of professional competency through professional identity formation in a PBL curriculum. In Proceedings of the 2016 IEEE Frontiers in Education Conference (FIE), Erie, PA, USA, 12–15 October 2016; pp. 1–9. [CrossRef]

10. World Economic Forum; Centre for the New Economy and Society Boston Consulting Group (BCG). *Towards a Reskilling Revolution: Industry-Led Action for the Future of Work*; World Economic Forum: Cologny, Switzerland, 2019. Available online: https://www3.weforum.org/docs/WEF_Towards_a_Reskilling_Revolution.pdf (accessed on 1 July 2021).
11. Cherifi, H. Introducing PLOS Complex Systems. *PLoS Complex Syst.* **2024**, *1*, e0000007. [CrossRef]
12. Emmert-Streib, F.; Cherifi, H.; Kauffman, S.; Yli-Harja, O. Moving beyond simulation and learning: Unveiling the potential of complexity data science. *PLoS Complex Syst.* **2024**, *1*, e0000002. [CrossRef]
13. Neal, Z.P.; Cadieux, A.; Garlaschelli, D.; Gotelli, N.J.; Saracco, F.; Squartini, T.; Shutters, S.T.; Ulrich, W.; Wang, G.; Strona, G. Pattern detection in bipartite networks: A review of terminology, applications, and methods. *PLoS Complex Syst.* **2024**, *1*, e0000010. [CrossRef]
14. Nigam, A.; Roy, A.; Singh, H.; Waila, H. Job Recommendation through Progression of Job Selection. In Proceedings of the 2019 IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS), Singapore, 19–21 December 2019; pp. 212–216. [CrossRef]
15. Zhou, W.; Zhou, X.; He, J.; Nie, Y.; Chen, Q.; Kang, G. Online Course Recommendation by Exploring User Interaction and Course Description. In Proceedings of the 2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Tianjin, China, 8–10 May 2024; pp. 2973–2978. [CrossRef]
16. Zhu, G.; Chen, Y.; Wang, S. Graph-Community-Enabled Personalized Course-Job Recommendations with Cross-Domain Data Integration. *Sustainability* **2022**, *14*, 7439. [CrossRef]
17. Li, N.; Suri, N.; Gao, Z.; Xia, T.; Börner, K.; Liu, X. Enter a job, get course recommendations. In Proceedings of the iConference 2017 Proceedings, Seville, Spain, 16–18 November 2017; Volume 2, pp. 118–122.
18. Lourenço, A.A.; Paiva, M.O. Academic Performance of Excellence: The Impact of Self-Regulated Learning and Academic Time Management Planning. *Knowledge* **2024**, *4*, 289–301. [CrossRef]
19. Tsiakmaki, M.; Kostopoulos, G.; Kotsiantis, S. Exploiting the Regularized Greedy Forest Algorithm Through Active Learning for Predicting Student Grades: A Case Study. *Knowledge* **2024**, *4*, 543–556. [CrossRef]
20. Zhang, T.; Qin, Y.; Li, Q. Trusted Artificial Intelligence: Technique Requirements and Best Practices. In Proceedings of the 2021 International Conference on Cyberworlds (CW), Caen, France, 28–30 September 2021; pp. 303–306. [CrossRef]
21. Zhang, Y.; Chen, X. Explainable recommendation: A survey and new perspectives. *Found. Trends Inf. Retr.* **2020**, *14*, 1–101. [CrossRef]
22. Liu, X.; Yu, T.; Xie, K.; Wu, J.; Li, S. Interact with the Explanations: Causal Debiased Explainable Recommendation System. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining (WSDM '24), Merida, Mexico, 4–8 March 2024*; Association for Computing Machinery: New York, NY, USA, 2024; pp. 472–481. [CrossRef]
23. 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*; Association for Computing Machinery: New York, NY, USA, 2020; pp. 79–88. [CrossRef]
24. Darmani, A.; Behkamal, B. SkillBridge: A Job-Course Matching Framework Using Pre-Trained Language Models and Graph Representation. In Proceedings of the 2025 11th International Conference on Web Research (ICWR), Tehran, Iran, 16–17 April 2025; pp. 235–242. [CrossRef]
25. Li, C.; Xian, X.; Ai, X.; Cui, Z. Representation Learning of Knowledge Graphs with Embedding Subspaces. *Sci. Program.* **2020**, *2020*, 4741963. [CrossRef]
26. Bonnaud, O.; Fesquet, L.; Bsiesy, A. Skilled manpower shortage in microelectronics: A challenge for the French education microelectronics network. In Proceedings of the 2019 18th International Conference on Information Technology Based Higher Education and Training (ITHET), Magdeburg, Germany, 26–27 September 2019; pp. 1–5. [CrossRef]
27. Boylan, B.M.; McBeath, J.; Wang, B. US–China relations: Nationalism, the trade war, and COVID-19. *Fudan J. Humanit. Soc. Sci.* **2021**, *14*, 23–40. [CrossRef]
28. Wu, X.; Zhang, C.; Du, W. An Analysis on the Crisis of "Chips shortage" in Automobile Industry—Based on the Double Influence of COVID-19 and Trade Friction. *J. Phys. Conf. Ser.* **2021**, *1971*, 012100. [CrossRef]
29. Demetroulis, E.A.; Papadogiannis, I.; Wallace, M.; Poulopoulos, V.; Antoniou, A. CORE: Cultivation of Collaboration Skills via Educational Robotics. *Knowledge* **2025**, *5*, 9. [CrossRef]
30. Baneres, D.; Conesa, J. A Life-long Learning Recommender System to Promote Employability. *Int. J. Emerg. Technol. Learn. (iJET)* **2017**, *12*, 77–93. [CrossRef]
31. Dawson, N.; Williams, M.A.; Rizoiu, M.A. Skill-driven recommendations for job transition pathways. *PLoS ONE* **2021**, *16*, e0254722. [CrossRef]
32. Striebel, J.; Myers, R.; Lui, X. Career-Based Explainable Course Recommendation. In Proceedings of the iConference 2023, Online, 13–17 March 2023. [CrossRef]

33. Shanto, S.S.; Jony, A.I. Interpretable Ensemble Learning Approach for Predicting Student Adaptability in Online Education Environments. *Knowledge* **2025**, *5*, 10. [CrossRef]
34. Kaur, R.; Gupta, D.; Madhukar, M. Learner-Centric Hybrid Filtering-Based Recommender System for Massive Open Online Courses. *Int. J. Perform. Eng.* **2023**, *19*, 324. [CrossRef]
35. Deschênes, M. Recommender systems to support learners' Agency in a Learning Context: A systematic review. *Int. J. Educ. Technol. High. Educ.* **2020**, *17*, 1–23. [CrossRef]
36. Klašnja-Milićević, A.; Vesin, B.; Ivanović, M. Social tagging strategy for enhancing e-learning experience. *Comput. Educ.* **2018**, *118*, 166–181. [CrossRef]
37. Klašnja-Milićević, A.; Vesin, B.; Ivanović, M.; Budimac, Z. E-Learning personalization based on hybrid recommendation strategy and learning style identification. *Comput. Educ.* **2011**, *56*, 885–899. [CrossRef]
38. Pardos, Z.A.; Tang, S.; Davis, D.; Le, C.V. Enabling Real-Time Adaptivity in MOOCs with a Personalized Next-Step Recommendation Framework. In *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale (L@S '17), Cambridge, MA, USA, 20–21 April 2017*; Association for Computing Machinery: New York, NY, USA, 2017; pp. 23–32. [CrossRef]
39. Dai, Y.; Asano, Y.; Yoshikawa, M. Course Content Analysis: An Initiative Step toward Learning Objection Recommendation Systems for MOOC Learners. In Proceedings of the 9th International Conference on Educational Data Mining, Raleigh, NC, USA, 29 June–2 July 2016; pp. 347–352.
40. Yu, R.; Jiang, D.; Warschauer, M. Representing and Predicting Student Navigational Pathways in Online College Courses. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale (L@S '18), London, UK, 26–28 June 2018*; Association for Computing Machinery: New York, NY, USA, 2018. [CrossRef]
41. Chu, T.S.; Ashraf, M. Artificial Intelligence in Curriculum Design: A Data-Driven Approach to Higher Education Innovation. *Knowledge* **2025**, *5*, 14. [CrossRef]
42. Bourkoukou, O.; El Bachari, E. E-learning personalization based on collaborative filtering and learner's preference. *J. Eng. Sci. Technol.* **2016**, *11*, 1565–1581.
43. Reategui, E.; Boff, E.; Campbell, J. Personalization in an interactive learning environment through a virtual character. *Comput. Educ.* **2008**, *51*, 530–544. [CrossRef]
44. Mangina, E.; Kilbride, J. Evaluation of keyphrase extraction algorithm and tiling process for a document/resource recommender within e-learning environments. *Comput. Educ.* **2008**, *50*, 807–820. [CrossRef]
45. Hsu, C.K.; Hwang, G.J.; Chang, C.K. A personalized recommendation-based mobile learning approach to improving the reading performance of EFL students. *Comput. Educ.* **2013**, *63*, 327–336. [CrossRef]
46. Wang, P.Y.; Yang, H.C. Using collaborative filtering to support college students' use of online forum for English learning. *Comput. Educ.* **2012**, *59*, 628–637. [CrossRef]
47. Hsu, C.K.; Hwang, G.J.; Chang, C.K. Development of a reading material recommendation system based on a knowledge engineering approach. *Comput. Educ.* **2010**, *55*, 76–83. [CrossRef]
48. Ahmad, H.K.; Qi, C.; Wu, Z.; Muhammad, B.A. ABiNE-CRS: Course recommender system in online education using attributed bipartite network embedding. *Appl. Intell.* **2023**, *53*, 4665–4684. [CrossRef]
49. Gienapp, L.; Scells, H.; Deckers, N.; Bevendorff, J.; Wang, S.; Kiesel, J.; Syed, S.; Fröbe, M.; Zuccon, G.; Stein, B.; et al. Evaluating Generative Ad Hoc Information Retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), Washington, DC, USA, 14–18 July 2024*; Association for Computing Machinery: New York, NY, USA, 2024; pp. 1916–1929. [CrossRef]
50. Hersh, W. Search still matters: Information retrieval in the era of generative AI. *J. Am. Med Informatics Assoc.* **2024**, *31*, ocae014. [CrossRef]
51. Najork, M. Generative Information Retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), Taipei, Taiwan, 23–27 July 2023*; Association for Computing Machinery: New York, NY, USA, 2023; p. 1. [CrossRef]
52. Li, Y.; Yang, N.; Wang, L.; Wei, F.; Li, W. Learning to rank in generative retrieval. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 26–27 February 2024; Volume 38, pp. 8716–8723.
53. Jing, X.; Tang, J. Guess You like: Course Recommendation in MOOCs. In *Proceedings of the International Conference on Web Intelligence, Leipzig, Germany, 23–26 August 2017*; Association for Computing Machinery: New York, NY, USA, 2017; pp. 783–789. [CrossRef]
54. Parameswaran, A.; Venetis, P.; Garcia-Molina, H. Recommendation Systems with Complex Constraints: A Course Recommendation Perspective. *ACM Trans. Inf. Syst.* **2011**, *29*, 1–33. [CrossRef]
55. Drivas, I.C.; Kouis, D.; Kyriaki-Manessi, D.; Giannakopoulou, F. Social Media Analytics and Metrics for Improving Users Engagement. *Knowledge* **2022**, *2*, 225–242. [CrossRef]

56. da Silva, F.L.; Slodkowski, B.K.; da Silva, K.K.A.; Cazella, S.C. A systematic literature review on educational recommender systems for teaching and learning: Research trends, limitations and opportunities. *Educ. Inf. Technol.* **2023**, *28*, 3289–3328. [CrossRef]
57. Varveris, D.; Saltas, V.; Tsiantos, V. Exploring the Role of Metacognition in Measuring Students' Critical Thinking and Knowledge in Mathematics: A Comparative Study of Regression and Neural Networks. *Knowledge* **2023**, *3*, 333–348. [CrossRef]
58. 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. *Proc. Assoc. Inf. Sci. Technol.* **2020**, *57*, e324. [CrossRef]
59. Anderson, K.A. Skill networks and measures of complex human capital. *Proc. Natl. Acad. Sci. USA* **2017**, *114*, 12720–12724. [CrossRef] [PubMed]
60. ONETOnLine. Development NCfON. Available online: https://onetonline.org (accessed on 1 July 2021).
61. U.S. Bureau of Labor Statistics. *Occupational Outlook Handbook 2015*; Technical Report; US Department of Labor: Washington, DC, USA, 2015.
62. Wang, Y.; Liu, X.; Chen, Y. Analyzing cross-college course enrollments via contextual graph mining. *PLoS ONE* **2017**, *12*, e0188577. [CrossRef]
63. Bridges, C.; Jared, J.; Weissmann, J.; Montanez-Garay, A.; Spencer, J.; Brinton, C.G. Course recommendation as graphical analysis. In Proceedings of the 2018 52nd Annual Conference on Information Sciences and Systems (CISS), Princeton, NJ, USA, 21–23 March 2018. [CrossRef]
64. Ezaldeen, H.; Bisoy, S.K.; Misra, R.; Alatrash, R. Semantics aware intelligent framework for content-based e-learning recommendation. *Nat. Lang. Process. J.* **2023**, *3*, 100008. [CrossRef]
65. Grover, A.; Leskovec, J. Node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016*; Association for Computing Machinery: New York, NY, USA, 2016; pp. 855–864. [CrossRef]
66. Kong, X.; Mao, M.; Wang, W.; Liu, J.; Xu, B. VOPRec: Vector representation learning of papers with text information and structural identity for recommendation. *IEEE Trans. Emerg. Top. Comput.* **2018**, *9*, 226–237. [CrossRef]
67. Pennington, J.; Socher, R.; Manning, C. GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543. [CrossRef]
68. Zhang, H.; Shen, X.; Yi, B.; Wang, W.; Feng, Y. KGAN: Knowledge Grouping Aggregation Network for course recommendation in MOOCs. *Expert Syst. Appl.* **2023**, *211*, 118344. [CrossRef]
69. Gogoglou, A.; Bruss, C.B.; Nguyen, B.; Sarshogh, R.; Hines, K.E. Quantifying Challenges in the Application of Graph Representation Learning. In Proceedings of the 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), Miami, FL, USA, 14–17 December 2020; pp. 1519–1526. [CrossRef]
70. Bernabé-Moreno, J.;Tejeda-Lorente, Á.; Herce-Zelaya, J.; Porcel, C.; Herrera-Viedma, E. An automatic skills standardization method based on subject expert knowledge extraction and semantic matching. *Procedia Comput. Sci.* **2019**, *162*, 857–864. [CrossRef]
71. Dong, Y.; Chawla, N.V.; Swami, A. Metapath2vec: Scalable Representation Learning for Heterogeneous Networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017*; Association for Computing Machinery: New York, NY, USA, 2017; pp. 135–144. [CrossRef]
72. Hao, P.; Li, Y.; Bai, C. Meta-relationship for course recommendation in MOOCs. *Multimed. Syst.* **2023**, *29*, 235–246. [CrossRef]
73. Nguyen, D.; Malliaros, F.D. BiasedWalk: Biased Sampling for Representation Learning on Graphs. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; pp. 4045–4053. [CrossRef]
74. Seo, S.; Oh, B.; Lee, K.H. Reliable Knowledge Graph Path Representation Learning. *IEEE Access* **2020**, *8*, 32816–32825. [CrossRef]
75. Wang, Z.; Li, L.; Li, Q.; Zeng, D. Multimodal Data Enhanced Representation Learning for Knowledge Graphs. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019. [CrossRef]
76. Ai, Q.; Zhang, Y.; Bi, K.; Croft, W.B. Explainable Product Search with a Dynamic Relation Embedding Model. *ACM Trans. Inf. Syst.* **2020**, *38*, 1–29. [CrossRef]
77. Cau, E.; Morini, V.; Rossetti, G. Trends and topics: Characterizing echo chambers' topological stability and in-group attitudes. *PLoS Complex Systems* **2024**, *1*, e0000008. [CrossRef]
78. Tarrade, L.; Chevrot, J.P.; Magué, J.P. How position in the network determines the fate of lexical innovations on Twitter. *PLoS Complex Syst.* **2024**, *1*, e0000005. [CrossRef]
79. Wei, H.; Li, F.; Miao, W. Autonomous and ubiquitous in-node learning algorithms of active directed graphs and its storage behavior. *PLoS Complex Syst.* **2024**, *1*, e0000019. [CrossRef]
80. Wang, X.; Wu, H.; Hsu, C.H. Mashup-oriented api recommendation via random walk on knowledge graph. *IEEE Access* **2018**, *7*, 7651–7662. [CrossRef]
81. Zheng, L.; Liu, S.; Song, Z.; Dou, F. Diversity-Aware Entity Exploration on Knowledge Graph. *IEEE Access* **2021**, *9*, 118782–118793. [CrossRef]

82. Wang, N.; Wang, H.; Jia, Y.; Yin, Y. Explainable Recommendation via Multi-Task Learning in Opinionated Text Data. In *Proceedings of the The 41st International ACM SIGIR Conference on Research Development in Information Retrieval (SIGIR '18), Ann Arbor, MI, USA, 8–12 July 2018*; Association for Computing Machinery: New York, NY, USA, 2018; pp. 165–174. [CrossRef]
83. Tao, Y.; Jia, Y.; Wang, N.; Wang, H. The FacT: Taming Latent Factor Models for Explainability with Factorization Trees. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'19), Paris, France, 21–25 July 2019*; Association for Computing Machinery: New York, NY, USA, 2019; pp. 295–304. [CrossRef]
84. Kivimäki, I.; Panchenko, A.; Dessy, A.; Verdegem, D.; Francq, P.; Bersini, H.; Saerens, M. A Graph-Based Approach to Skill Extraction from Text. In Proceedings of the TextGraphs-8 Graph-based Methods for Natural Language Processing, Seattle, WA, USA, 18 October 2013; pp. 79–87.
85. Knijnenburg, B.P.; Willemsen, M.C.; Gantner, Z.; Soncu, H.; Newell, C. Explaining the user experience of recommender systems. *User Model. User-Adapt. Interact.* **2012**, *22*, 441–504. [CrossRef]
86. Pu, P.; Chen, L.; Hu, R. A user-centric evaluation framework for recommender systems. In Proceedings of the Fifth ACM Conference on Recommender Systems, Chicago, IL, USA, 23–27 October 2011; pp. 157–164.
87. Alharbe, N.; Rakrouki, M.A.; Aljohani, A. A collaborative filtering recommendation algorithm based on embedding representation. *Expert Syst. Appl.* **2023**, *215*, 119380. [CrossRef]
88. Li, J.; Wang, M.; Li, J.; Fu, J.; Shen, X.; Shang, J.; McAuley, J. Text Is All You Need: Learning Language Representations for Sequential Recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), Long Beach, CA, USA, 6–10 August 2023*; Association for Computing Machinery: New York, NY, USA, 2023; pp. 1258–1267. [CrossRef]
89. Li, L.; Lian, J.; Zhou, X.; Xie, X. Ada-Retrieval: An Adaptive Multi-Round Retrieval Paradigm for Sequential Recommendations. *Proc. AAAI Conf. Artif. Intell.* **2024**, *38*, 8670–8678. [CrossRef]
90. Ezaki, T.; Imura, N.; Nishinari, K. Synergistic integration of fragmented transportation networks: When do networks (not) synergize? *PLoS Complex Syst.* **2024**, *1*, e0000017. [CrossRef]
91. van Oort, P.A.J.; Fonteijn, H.M.J.; Hengeveld, G.M. DARTS: Modelling effects of shocks on global, regional, urban and rural food security. *PLoS Complex Syst.* **2024**, *1*, e0000006. [CrossRef]
92. Park, M.; Tabatabaee, Y.; Ramavarapu, V.; Liu, B.; Pailodi, V.K.; Ramachandran, R.; Korobskiy, D.; Ayres, F.; Chacko, G.; Warnow, T. Well-connectedness and community detection. *PLoS Complex Syst.* **2024**, *1*, e0000009. [CrossRef]
93. Acevedo, A.; Wu, Y.; Traversa, F.L.; Cannistraci, C.V. Geometric separability of mesoscale patterns in embedding representation and visualization of multidimensional data and complex networks. *PLoS Complex Syst.* **2024**, *1*, e0000012. [CrossRef]
94. Moore, T.R.; Pachucki, M.C.; Economos, C.D. Determinants and facilitators of community coalition diffusion of prevention efforts. *PLoS Complex Syst.* **2024**, *1*, e0000004. [CrossRef]

https://doi.org/10.3390/knowledge6010005