Knowledge-Based Systems 195 (2020) 105618

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A learning path recommendation model based on a multidimensional knowledge graph framework for e-learning☆

Daqian Shi [a], Ting Wang [a], Hao Xing [a], Hao Xu [a,b,c,d,*]

[a] College of Computer Science and Technology, Jilin University, China
[b] School of Management, Jilin University, China
[c] Department of Computer Science and Technology, Zhuhai College of Jilin University, China
[d] Symbol Computation and Knowledge Engineer of Ministry of Education, Jilin University, China

A R T I C L E I N F O

Article history:
Received 8 May 2019
Received in revised form 26 December 2019
Accepted 3 February 2020
Available online 6 February 2020

Keywords:
Learning path recommendation
Knowledge graph
e-learning
Learning needs

A B S T R A C T

E-learners face a large amount of fragmented learning content during e-learning. How to extract and organize this learning content is the key to achieving the established learning target, especially for non-experts. Reasonably arranging the order of the learning objects to generate a well-defined learning path can help the e-learner complete the learning target efficiently and systematically. Currently, knowledge-graph-based learning path recommendation algorithms are attracting the attention of researchers in this field. However, these methods only connect learning objects using single relationships, which cannot generate diverse learning paths to satisfy different learning needs in practice. To overcome this challenge, this paper proposes a learning path recommendation model based on a multidimensional knowledge graph framework. The main contributions of this paper are as follows. Firstly, we have designed a multidimensional knowledge graph framework that separately stores learning objects organized in several classes. Then, we have proposed six main semantic relationships between learning objects in the knowledge graph. Secondly, a learning path recommendation model is designed for satisfying different learning needs based on the multidimensional knowledge graph framework, which can generate and recommend customized learning paths according to the e-learner's target learning object. The experiment results indicate that the proposed model can generate and recommend qualified personalized learning paths to improve the learning experiences of e-learners.

1. Introduction

With the popularity and improvement of information search technology, increasingly more learners choose self-learning through the Internet, which is a well-known method for e-learning [1]. Compared with the traditional teacher-centered learning style, e-learning has advantages: the learning target can be flexibly searched and viewed by learners anywhere, not only in class. However, since most of the learning content on the network is fragmented, how to systematically and efficiently learn multiple knowledge objects in a specific field has always been a problem for e-learning.

Researchers have realized that the order of learning objects has a great impact on learning quality [2,3]. Methods for organizing fragmented learning content by learning paths to guide learners have gradually been accepted. This can effectively reduce the time required to collect learning materials and improves learning efficiency. At the same time, a high-quality learning path is helpful for improving the learners' understanding of the learning content. Therefore, how to generate high-quality learning paths is an issue of concern for the research community.

Recent research [4–8] has focused on generating high-quality customized learning paths to satisfy e-learners. These clustering-based learning path generation methods often collect redundant or irrelevant learning objects because they ignore potential dependencies between learning objects. Other work [9–12] used knowledge graphs for applying these dependencies to the learning path recommendation model and has achieved some success. However, they only built the simplest relationship to link learning objects but did not explore further by building more complex semantics in the relationship. These works did not fully exploit the connectivity of the knowledge graph to link learning objects using various semantic relationships. Therefore, they can only generate

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.knosys.2020.105618.

* Corresponding author at: College of Computer Science and Technology, Jilin University, China.
E-mail address: xuhao@jlu.edu.cn (H. Xu).

https://doi.org/10.1016/j.knosys.2020.105618

single learning paths that cannot satisfy different learning needs in for e-learning today.

This paper proposes a learning path recommendation model based on the knowledge graph (KG) to satisfy diverse learning needs. To improve the learning path recommendation model by using semantic relationships, an innovative knowledge graph framework is designed for storing and presenting learning objects. We construct a multidimensional knowledge graph by describing the learning object as the hierarchical classes of "basic knowledge", "algorithm" and "task". Based on this type of knowledge graph framework, learning objects from different subjects and curricula can be appropriately merged and organized. We have also constructed more complex and logical semantic relationships between these learning objects. Based on this framework, we have introduced a learning path generation algorithm and a learning path recommendation algorithm in the model. The learning path generation algorithm generates relationship constraints based on an individual's learning need, and then decides how to generate all possible learning paths. To recommend a logical learning path from the set of all possible learning paths, we proposed a learning path recommendation algorithm for scoring learning paths, in which variables and their weighted coefficients consider the different learning path preferences of the e-learner.

In this paper, we introduce an innovative model for learning path recommendation that is expandable and reusable. Therefore, our model can be promoted and used in more fields of e-learning to improve learning efficiency. At the same time, this research will also help researchers develop more effective learning path recommendation methods for improving e-learning.

This paper is organized as follows. Section 2 introduces related work on learning path recommendation. Section 3 presents some basic definitions and describes how the multidimensional KG was built (including the KG structure and data collection). Section 4 presents the details of the algorithms in the learning path recommendation model. Section 5 provides the structure of the e-learning system based on the proposed model. Section 6 evaluates the learning path recommendation model. Feedback from participants was used to verify the quality of the recommended learning paths. Section 7 presents the conclusions and future research.

2. Related work

Researchers have proposed various methods for improving learning efficiency; constructing systems [13–15] for recommending learning materials to e-learners is a leading-edge research field. Durand et al. [9] have proved that the order of learning objects is important for learners and introduced a graph-theory-based system that can generate one-way learning paths by linking all necessary learning objects through the relationship. Chen's research [16] noted that learning the prior and posterior knowledge is helpful in understanding the current learning object, which also supports the idea that learning objects should be sorted into learning paths. Therefore, for systematic and efficient e-learning, individuals should study learning objects in the suggested learning path order to achieve their learning target.

Based on this context, researchers have tried to explore the dependency of learning content in e-learning systems. Chen et al. [17] found that prerequisite relations among concepts play an important role in e-learning and proposed an effective data-driven solution for prerequisite classification. Pan et al. [18] developed a graph-based propagation algorithm to order the concepts based on the learned representations of course concepts. At the same time, some researchers [19–21] tried to obtain concept prerequisite relation from course dependencies by novel approaches like directed graphs. These works show great potential in exploring the dependency on learning content in the e-learning environment. But they still lack the consideration of the concrete classification of relations/dependencies between learning contents, which is important for effectively organizing learning content for e-learners.

Data mining was widely used by researchers for organizing learning contents into learning paths. Chen [22] constructed a genetic-based, personalized e-learning system that can generate appropriate learning paths by mining the individual learner's pretest and learning performance data. Dwivedi et al. [23] improved Chen's approach by constructing learning paths using a variable-length genetic algorithm (VLGA) that includes the learning path record from predecessors. Chen et al. [24] proposed an improved ant colony optimization algorithm (ACO) based on coordinate system to recommend learning path. Bendahmane et al. [25] presented a competence based approach (CBA) derived from learning data, learner's expectations. In this approach, learners were clustered and traced, and finally obtained proper learning paths. Hsieh and Wang [26] developed an e-learning system using a data mining approach that can create a relationship hierarchy of learning materials to generate a suitable learning path. With these data-mining-based methods, learners no longer need to waste time organizing the learning content, but there are still some problems: (1) Updating new data in this type of learning system will be difficult because these systems are not self-adaptive, so the learning path has to be generated every time data updates are performed. (2) These systems sometimes generate redundant results when identical learning content exists.

As a popular research domain, knowledge graphs have been used recently for learning path construction since knowledge graphs can avoid ambiguities in learning content descriptions. Encouraged by this characteristic, some researchers [27–29] tried to build learning systems for learning path recommendation based on knowledge graphs, and successfully solved problems (1) and (2). Wan et al. [30] provided a learner-oriented learning recommendation method for generating learning paths based on knowledge, in which knowledge units are represented by nodes, and the relationships between knowledge units are represented by the node connections. Ouf et al. [31] proposed the framework for smart e-learning ecosystem using knowledge graph and SWRL. Shmelev et al. [32] proposed a method that combined the genetic approach and knowledge graph technology to arrange the learning objects in a sequence. Chu et al. [33] built an e-learning system based on the concept map that can generate learning paths according to the relationships in the concept map. Subsequently, Zhu et al. [10] introduced a learning path recommendation method using preset learning scenarios, considering that learners need different learning paths in different scenarios. They proposed a learning path generation method that requires specifying starting and ending nodes.

The above research studies focus on how to generate good learning paths using knowledge graphs. These studies choose to regard the learning objects as nodes, and relationships as paths in a learning path. However, these knowledge-graph-based methods did not fully exploit the connectivity of the knowledge graph. The relationships between learning objects should have logical relationship names that represent specific semantics, and not just display a connection line. At the same time, these studies note that because e-learners may require different learning paths for the same learning object, a good learning path recommendation model should generate learning paths that satisfy diverse learning needs in different scenarios. Based on the related studies described above, we will design a specific knowledge graph for the learning path recommendation model, and try to recommend logical learning paths that satisfy the diverse learning needs of individuals.

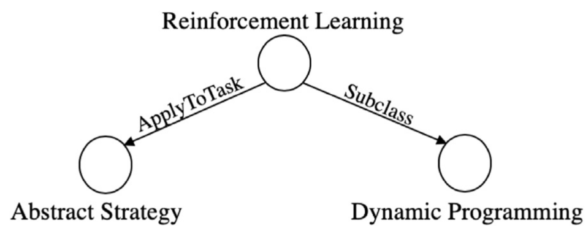

Fig. 3.1. Example of relationships between learning objects.

3. Knowledge graph design

We have designed a knowledge graph framework in order to better organize learning objects and to logically express semantic relationships between learning objects. Based on this framework, we build a knowledge graph that has been applied to our learning path recommendation model by taking partial machine-learning knowledge as a use case.

3.1. Definition of terminology

To better explain the content that follows, we have defined some relevant terminology that will be used in this paper.

- Learning object (LO): A learning object is the meta-learning material unit, which includes the basic information (name, description, etc.) and the learning link. Every learning object can be studied independently, for example, "Sigmoid_Function" is a learning object that belongs to the "basic knowledge" class.
- Relationship (RE): The relationship refers to the semantic dependencies of the relationships between different learning objects. For example, Fig. 3.1 presents the relationships between three learning objects.
- Knowledge graph (KG): A knowledge graph is a type of directed graph in which learning objects are represented by nodes and relationships are represented by the node connections. A knowledge graph is defined as follows:

$$KG = (LO, RE) \tag{1}$$

  In which: *LO* is the set of all the learning objects in the knowledge graph, and *RE* is the set of all the relationships in the knowledge graph.
- Learning need: The learning need is how the learner expects to learn the target learning object. For example, when we have the target learning object "Region_Based_CNN", then "what is the prior knowledge of Region_Based_CNN" and "how to achieve Region_Based_CNN by basic knowledge" are two different learning needs. In this study, we have pre-established a set of six learning needs as shown in Table 3.1. In the experiment, we use labels (like #1) to represent different learning needs.
- Learning path (LP): A learning path is the sequence of learning objects that is generated for achieving the specific learning goal. For example, Fig. 3.2 shows the learning path for solving entity recognition.
- Target learning object: A target learning object is what the individual intends to learn in the current learning session. In the learning path recommendation, the target learning object is always the last node of the learning path.

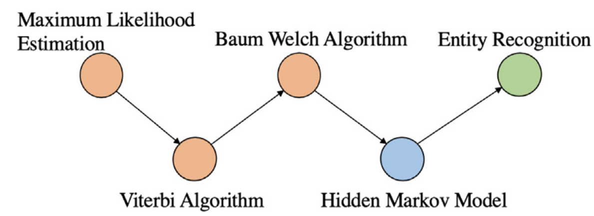

Fig. 3.2. Example of the learning path.

Table 3.1
Learning needs and corresponding labels.

Label	Type of learning need
#1	Requesting the prior knowledge.
#2	Requesting the algorithm prototype.
#3	Requesting the position of the learning object in the hierarchy.
#4	Requesting the single learning object.
#5	Requesting the learning path from algorithm to target.
#6	Requesting the learning path from basic knowledge to target.

3.2. The framework of the knowledge graph

In this study, we proposed a multidimensional knowledge graph framework for representing knowledge. According to a previous study [34], dividing educational resources into multiple classes helps learners to efficiently understand learning profiles, and enables them to logically organize and memorize knowledge. Therefore, we decided to separate different learning objects into three classes:

- Basic Knowledge
  This class contains all the necessary basic knowledge learning objects for supporting algorithms, such as "naïve Bayes" and "Bayesian statistics", which is the bottom level of this knowledge graph framework.
- Algorithm
  The center of this framework is the "algorithm" class that contains all the algorithms related to the specific knowledge field (e.g., "Bayes classifier"). An algorithm is the method to achieve/solve a particular task.
- Task
  This class contains all the practical tasks such as "natural language processing" and "sentiment analysis" in the machine learning field. The "task" class is the top level of this knowledge graph framework. E-learners can understand what practical tasks they can achieve/solve after they have learned the basic knowledge/algorithm.

Fig. 3.3 shows an example of a multidimensional KG framework. In this framework, each class consists of a hierarchy and the corresponding instances of the learning object. The hierarchy represents the knowledge structure of the current class, while the learning objects are the meta-learning materials that are inserted into the hierarchy and connected by semantic relationships. To better express the semantic relationships between learning objects, we have designed multiple relationships and divided them into intraclass relationships and interclass relationships. Intraclass relationships enable intraclass learning objects to be connected to each other. On the other hand, interclass relationships provide connections between learning objects from different classes. Details of the relationships are shown in Table 3.2. These relationships between learning objects connected by semantics are fundamental for generating relationship constraints in the learning path generation algorithm.

The one-dimensional knowledge graph framework can only present learning objects in a single class and has no specific

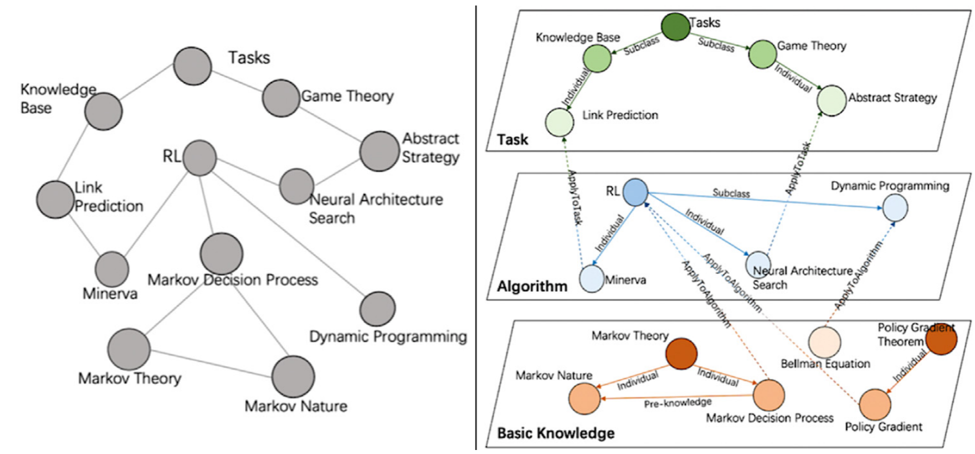

Fig. 3.3. (left) One-dimensional KG framework in previous work [10,35]. (right) Multidimensional KG framework in our project, where dotted lines represent inter-class relationships, solid lines represent intra-class relationships, and the nodes with different colors represent learning objects in different classes.

Table 3.2
Designed relationships in the knowledge graph.

	Type	Description
Subclass	Intra-class	The relationship indicates that the current LO has a subclass.
Individual	Intra-class	The relationship indicates that the current LO has an individual.
Pre-knowledge	Intra-class	The relationship indicates that the current LO (basic knowledge) has prior knowledge that should be learned first.
Ori-algorithm	Intra-class	The relationship indicates that the current LO (algorithm) was improved from an original algorithm.
ApplyToAlgorithm	Inter-class	The relationship indicates that the current LO (basic knowledge) can be applied in the target LO (algorithm).
ApplyToTask	Inter-class	The relationship indicates that the current LO (algorithm) can be applied in the target LO (task).

relationships between learning objects. Our knowledge graph provides multiple classes for learning objects and strengthens the connections between interclass learning objects. In this context, the knowledge graph is able to show how to practically apply the learned knowledge (basic knowledge-algorithm, algorithm-task), which can expand the e-learners' understanding of the learned knowledge and enable them to grasp the practical use of theoretical knowledge.

3.3. Data collection

In this study, we have built a knowledge graph based on machine learning field knowledge as the use case. The knowledge graph consists of 675 learning objects and 1033 relationships, where two interclass relationships and four intraclass relationships were set to ensure the diversity of connections between the learning objects.

During the data collection process, we applied data mining methods and expert knowledge validation to collect learning objects for the knowledge graph. The first step is to crawl for data; several educational websites were crawled for information about algorithms and tasks, which was then stored as structured data. After removing duplicates, we collected 392 algorithm objects and 89 task objects and established the corresponding relationships between these objects. Then, textbooks (on artificial intelligence and mathematics) were crawled for information about basic knowledge to produce a list of named entities of basic knowledge. A named entity recognition model was trained using this list to recognize basic knowledge in algorithms and to extract the relationship between the target basic knowledge and the current algorithm.

To this point, raw data with all the learning objects and relationships has been obtained. However, since the data from the network is not always reliable, the data need to be processed using expert knowledge. In this step, we manually proofread the information about learning objects and corrected wrong relationships. The procedure of manually proofread mainly includes (a) 2 annotators determine whether the learning objects and relationships are correct in raw data; (b) compare the results from annotators, we regard the agreements with correct labels as "correct data", agreements with incorrect labels and all disagreements as "incorrect data"; (c) the "incorrect data" is checked and annotated by a super-annotator which is the expert. For the procedure (b), we use Cohen's kappa coefficient [36] to verify the reliability of the annotation from inter-annotators, the result is 0.63 which means the annotations are substantial agreement. A knowledge graph was ultimately built that contained 225 basic knowledge objects, 361 algorithm objects, 89 task objects, and 1033 relationships. The accuracy of the learning object information (96.7%) and relationships (94%) proves that our knowledge graph reliably supports the learning path recommendation. For detailed introducing our knowledge graph, Fig. 3.3 (right) abstractly shows an example of the constructed knowledge graph, which contains 2 tasks, 3 algorithms, 4 basic knowledge and 5 different types of relationships. At the same time, a more concrete example of related algorithms for the specific task "Entity Recognition" is shown in Fig. 3.4, part of basic knowledge was also presented for algorithms respectively in the example.

4. Learning path recommendation model

Based on the knowledge graph, we can start designing and developing the learning path recommendation model. This section first presents how to generate all the possible learning paths according to the specific learning need, and then presents how to recommend an appropriate learning path from these possible learning paths. The structure of the recommendation model and detailed algorithms will be introduced at the end.

4.1. Learning path generation algorithm

In order to make our proposed learning path recommendation model more user-friendly, we design an algorithm based

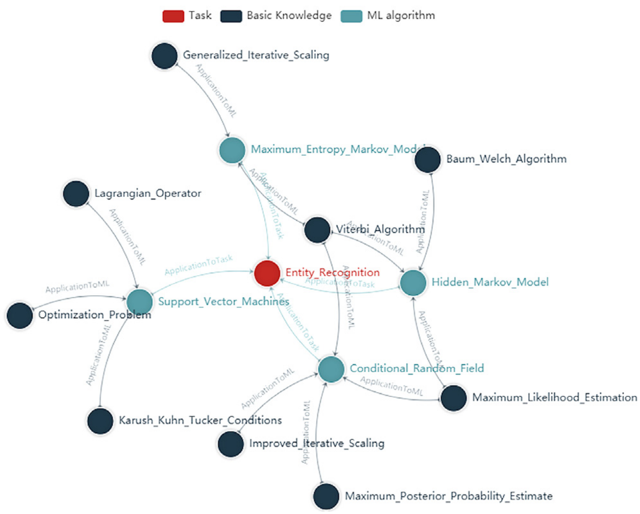

Fig. 3.4. Example of constructed knowledge graph.

on the multidimensional knowledge graph to generate the possible learning path through the e-learner's target learning object and the learning need. This algorithm replaces the learning path generation algorithm [10] which using the starting and ending (target) learning objects. Because most of the e-learners only know the target learning object but not the starting node in general. In this algorithm, the semantic relationships in the multidimensional knowledge graph will be used as constraints for generating the learning path.

The first step of this algorithm is to calculate the relationship constraints ϕ based on the learning need. The relationship constraints refer to a set of relationships that may appear in the current learning path. A relationship outside the constraints is not allowed in the learning path. According to the learning need N_u, all the corresponding relationship constraints $\phi = (\alpha, \beta, \gamma, \delta \ldots)$ will be output by $getRelationConstraint(N_u)$. It is worth noting that $\phi \sqsubseteq RE$. The second step of the algorithm is to generate the learning path according to the relationship constraints ϕ. Learning path generation will start with the target learning object, and search for the next learning object connected by the relationship under the constraints. Then, the search continues from the connected learning object. When the current learning object has no connected learning object, this is the start (the first learning object) of the current learning path. Based on this, the algorithm will start a greedy search from the target learning object until it generates all the possible learning paths P. The detailed algorithmic process is presented in algorithm 3.

4.2. Learning path recommendation algorithm

Based on the e-learner's input, the learning path recommendation model may generate many possible learning paths. However, the model should not output multiple learning paths all at once as the result recommended to the e-learner. Therefore, the optimal learning path needs to be selected for the e-learner. In this context, we have designed a learning path recommendation algorithm as shown in Eq. (2). The target of $max_{i=0\ldots k}Score(P_i)$ is to output the recommended learning path P_b with the highest score.

$$max_{i=0\ldots k}Score(P_i) = max_{i=0\ldots k}\sum_{j=1}^{n}(w_j * f_j(P_i)) \tag{2}$$

In Eq. (2), P_i is a specific learning path which needs to be scored. k is the total number of these learning paths. n is the total number of features in feature set. f is a function for calculating the corresponding feature of P_i. And w_j is the weight of the feature $f_j(P_i)$. Considering the model needs to quantize the learning path in order to select the one with the highest score. According to the previous work [37,38] on quantizing algorithms, we propose a feature set (F) which are publication time (f_1), citation count (f_2), search frequency (f_3), the impact of the publisher (f_4) and the impact of the author (f_5). Table 4.1 shows the detail about these features, including the formal expression of the feature and the description of the parameters in the expression.

Research [39–41] present that e-learners concern on the novelty, authority and popularity of the research. In this paper, we propose these three directions as the learning preference of e-learner. Specific learning preference represents the characteristic of the learning path the e-learner more concern with. The weighting method (W) was applied to constrain features in order to satisfy different learning preferences. We have provided learning preference options where each weight (w_i) corresponds to a particular feature (f_i). Details of weight distributions are as follows:

1. Novelty: Learner prefers to solve a problem using novel algorithms. When $w_1 = 1$, $w_2 = 0$, $w_3 = 0$, $w_4 = 0$, $w_5 = 0$, the feature publication time is considered and the learning path with the most current algorithms will be selected.
2. Popularity: Learner prefers to learn famous algorithms. When weights are set as $w_1 = 0$, $w_2 = 0.5$, $w_3 = 0.5$, $w_4 = 0$, $w_5 = 0$, means the selected learning path is famous in the research community and well-known in the public.
3. Authority: Learner prefers to use algorithms from authoritative resources. When $w_1 = 0$, $w_2 = 0$, $w_3 = 0$, $w_4 = 0.5$, $w_5 = 0.5$, the selected learning path considers the impact of publishers and authors which represent the authority of publishers and authors.

$$calculateW(lp) = \frac{\sum_{i=1}^{n} W(lp_i)}{n} \tag{3}$$

In this context, e-learners can select learning preference to generate learning path according to their needs. At the same time, we propose a function for calculating the weight distribution when e-learners have multiple learning preferences, as shown in Eq. (3) . The target of $calculateW(lp)$ is to output the weight distribution. lp is a set of selected learning preferences from e-learner and lp_i represents the specific learning preference. W is the function for calculating corresponding weights for lp_i. n is the number of learning preferences in lp.

According to the learning path recommendation algorithm above, our model can recommend the most appropriate learning path among multiple learning paths.

4.3. Learning path recommendation model based on knowledge graph

The structure of the learning path recommendation model is presented in Fig. 4.1. The input query Q from the learner should be a query sentence in natural language. The output of the model should be the recommended learning path.

The details of the algorithms in the model are presented below:

1. The main procedures of how to generate the recommended learning path according to the learner's query are shown in algorithm 1.

Table 4.1
Features for quantizing algorithms.

Features	Correlated factors	Parameters
Publication time	$f_1(P_i) = \frac{T_p - E(T_p)}{T_c - E(T_p)}$	T_p is the publishing year of the algorithm. $E(T_p)$ represents the earliest publishing year for algorithms in the KG. T_c is the current year.
Citation count	$f_2(P_i) = \frac{C_c}{Max(C_c)}$	C_c is the citiation count of the algorithm. $Max(C_c)$ represents the maximum C_c in the KG.
Search frequency	$f_3(P_i) = \frac{F_s}{Max(F_s)}$	F_s is the search frequency of an algorithm in Google trends [38], the interval of F_s is [0,1]. $Max(F_s)$ represents the maximum F_s in the KG.
Impact of publisher	$f_4(P_i) = \frac{H_p - Min(H_p)}{Max(H_p) - Min(H_p)}$	H_p is the H5-index [37] of the publisher. $Min(H_p)$ and $Max(H_p)$ represent the minimum and maximum H5-index value.
Impact of author	$f_5(P_i) = \frac{H_a - Min(H_a)}{Max(H_a) - Min(H_a)}$	H_a is the H5-index of the algorithm's authors. $Min(H_a)$ and $Max(H_a)$ represent the minimum and maximum H5-index value of the author.

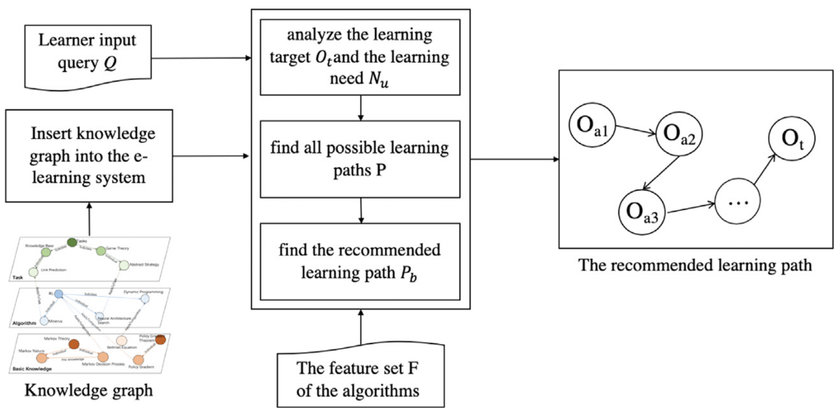

Fig. 4.1. The structure of the learning path recommendation model.

2. The learner's input query will be sent to the model for extracting the target learning object and learning need in algorithm 2. The raw input query Q needs to be converted into the token set by preprocessing. This step includes removing stop words, tokenization, lemmatization and some other basic natural language preprocessing methods. Then, the model needs to recognize the target learning object and its class. We have established a list of frequent requirements that will be used to recognize the query keywords in step 4. Lastly, the learning need will be obtained from the class of the target learning object and the query keywords.
3. In algorithm 3, the model will traverse the knowledge graph to find the learning path set that contains all the eligible learning paths. In step 1, the function $\phi = getRelationConstraint(N_u)$ will constrain the type of relationship to be used in this round, according to the learning need N_u. Steps 2–14 aim to construct an iterative algorithm for extracting all the possible learning paths for set P, based on the target learning object O_t.
4. Algorithm 4 is the learning path scoring algorithm we introduced in the previous section. In this algorithm, the model will ultimately output the recommended learning path with the highest score.

5. E-learning system design

The quality of learning content is the most important part of the learning path recommendation, the learning quality can only be guaranteed when the learning content is accurate. However, the way in which learning content is presented to e-learners is also an important factor in determining the quality of learning. An e-learning system has been built based on our learning path recommendation model for providing the e-learning service. This section introduces the design of the e-learning system.

Algorithm 1 Learning path recommendation algorithm based on KG. $P_b = RecommendationPath(G, Q, F)$.

Input:
The knowledge graph $G = (LO, RE)$;
The learner's query Q;
The feature set F;
Output:
The recommended learning path P_b.

1. $O_t, N_u = RequireAnalyze(Q)$; {analyze the learner's target learning object O_t and learning need N_u.}
2. $P = GetPossiblePaths(G, O_t, N_u)$; {get the learning path set P which includes all possible learning paths.}
3. $P_b = GetMaxScorePath(P, lp)$; {select the recommended learning path P_b with the highest score.}
4. **return** P_b;

Fig. 5.1 shows the structure of the e-learning system. The system consists of three parts: a graph database, the website backend, and the website interface.

- Graph database: To enable the website to access data, we stored the knowledge graph in a graph database that can be easily operated from the website backend. The graph database contains all the learning objects and relationships, which will be called when generating learning paths and searching learning objects.
- Website backend: The website backend is responsible for building the function and outputting the target result to the website interface. After receiving the input query from the text description window, the semantic processing unit will analyze the learning requirement from the input query. Then, the kernel algorithm unit will return the possible learning path set P and the recommended learning path P_b.

Algorithm 2 Input processing: extract the target learning object and analyze learning need. $O_t, N_u = RequireAnalyze(Q)$.

Input:
The learner's query Q;
Output:
The target learning object O_t;
The learner's learning need N_u;

1. $T = preprocessing(Q)$; {convert raw query Q to the token set T.}
2. $O_t = getTarget(T)$; {obtain the target learning object O_t from T.}
3. $c = getClass(O_t)$; {get the class of the target learning object O_t.}
4. $K = getQueryK(T)$; {extract the learner's query keywords set K from T.}
5. $N_u = calculateN(c, K)$; {calculate the learner's learning need N_u by c and K.}
6. **return** O_t, N_u;

Algorithm 3 Generate possible learning path set. $P = GetPossiblePaths(G, O_t, N_u)$.

Input:
The knowledge graph G;
The target learning object O_t;
The learner's learning need N_u;
Output:
The possible learning path set P;

1. $\phi = getRelationConstraint(N_u)$; {generate a set of relationship constraints $\phi = (\alpha, \beta, \gamma, \delta ...)$.}
2. **function** *findPath*(O_t)
3. $R = getRelations(O_t)$; {determine the relationships R that are directly related to O_t.}
4. **if** $\{r|r \in R, r \in \phi\}$ not exists **then**
5. $P.add(p)$; {Put the possible learning path p into learning path set P.}
6. **else**
7. **for all** $r \in R$ **do**
8. **if** $r \in \phi$ **then**
9. $O_i = getO(O_t, r)$; {get the learning object O_i related to O_t with r.}
10. $p.addRelation(r, O_i)$; {put r and O_i into the possible learning path set p.}
11. recursively apply *findPath*(O_i);
12. **end if**
13. **end for**
14. **end if**
15. **end function**
16. **return** P;

- Website interface: As a window for interaction with e-learners, the website interface consists of a text-description window and a graph-display window. The possible learning path set P will be presented in the graph-display window, and the target recommended learning path and guidance will be presented in the text-description window.

As shown in Fig. 5.1, the proposed e-learning system combines dynamic graphic display and text guidance for presenting the learning content. We applied the text description to succinctly demonstrate the recommended learning content. Furthermore, the target learning object and its related learning object set were presented through the operable graph-display window to maximize the possibility of discovering potential learning content. In the dynamic graph-display window, a mouse-click on a learning object will extend to more learning objects. The mouse-click will also display the introductory information of the corresponding target learning object. Based on these functions, the system fully exploits the connectivity of the knowledge graph when presenting the learning content, so that e-learners can explore more potential learning content while using our learning path recommendation system.

Algorithm 4 Recommended the learning path P_b with the highest score. $P_b = GetMaxScorePath(P, lp)$.

Input:
The possible learning path set P;
The learner's learning preference lp;
Output:
The recommended learning path P_b;

1. **for all** $P_i \in P$ **do**
2. $F = calculateF(P_i)$; {calculate the corresponding feature set F of the algorithms in the learning path P_i.}
3. $W = calculateW(lp)$; {generate a weight distribution W according to lp, each feature f_j has a corresponding weight w_j.}
4. $Score(P_i) = calculateScore(W, F)$; {the score $s = \sum_{j=1}^{n} w_j * f_j(P_i)$, $w_j \in W, f_j \in F$.}
5. **end for**
6. $P_b = max_{i=0...k} Score(P_i)$; {recommend the learning path P_b with the highest score.}
7. **return** P_b;

6. Evaluation and analysis

This section describes how we evaluate the learning path recommendation model, which includes a description of the experiment process and the experiment data analysis.

6.1. Experiment process description

This learning path recommendation model aims to output the target learning content according to e-learner's input query. The quality of output learning content should be evaluated by the e-learner. Therefore, experiments are designed using the extrinsic evaluation method to evaluate the learning path recommendation model. In the experiment, we will verify whether (i) the learning path generation algorithm based on the multidimensional knowledge graph is better (higher search success rate and more user-friendly) than the algorithm based on the one-dimensional knowledge graph; (ii) e-learners are satisfied with the recommended learning path.

The experiment process consists of three parts, as shown in Fig. 6.1.

Control experiment:

We have designed a control experiment to verify the hypothesis (i). Two learning path recommendation models are used in the control experiment. The test model uses our learning path generation algorithm based on the multidimensional knowledge graph; this algorithm can generate the learning path by the target learning object. The control model uses the learning path generation algorithm based on the one-dimensional knowledge graph, which generates the learning path using the starting and ending learning objects. The control model was developed to simulate the state-of-the-art learning path generation method [10].

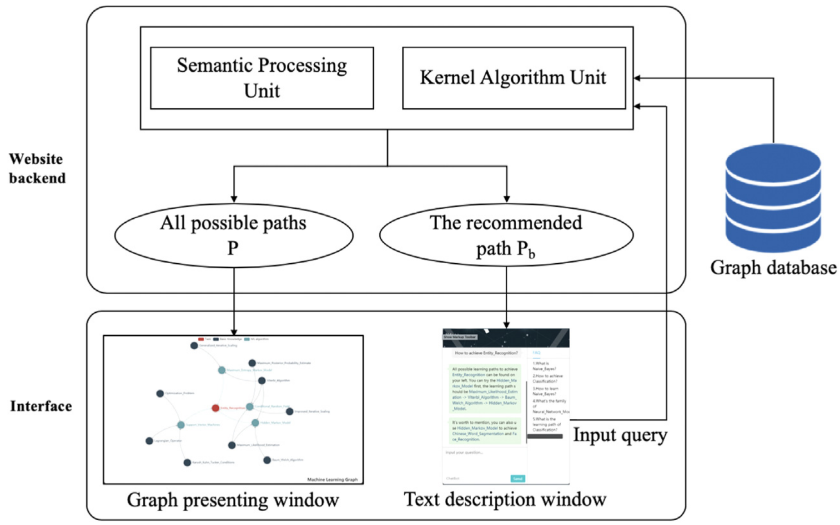

Fig. 5.1. The structure of the e-learning system.

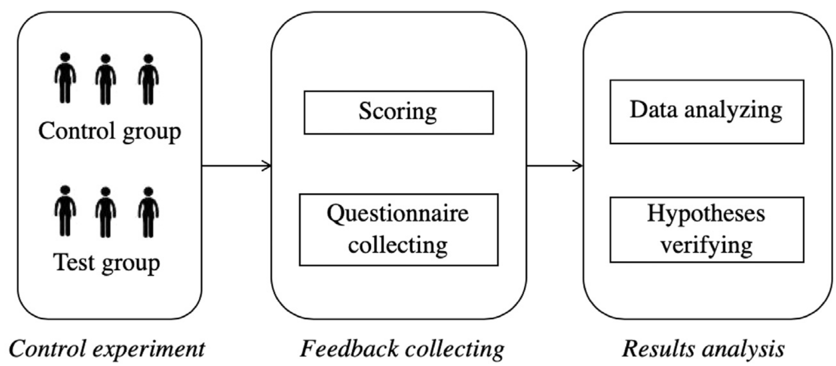

Fig. 6.1. Experiment process of verifying learning path recommendation model.

Participants will be divided into test and control groups without telling which group they are in. Then, they will be asked to use the corresponding model to conduct 10 learning path searches based on their own learning interests.

Feedback collecting:

Participants are asked to record and score their search results, in which feedbacks from the test group will be used to verify the hypothesis (ii). The score represents the participants' satisfaction with the recommended learning path. We use the Likert scale to quantify participant satisfaction. Five points represents very satisfied; four points represents satisfied; three points represents generally satisfied; two points represents unsatisfied; and one point represents very unsatisfied. Table 6.1 shows some examples of scored results. It noteworthy that example 3 presents a failure case that did not yield a target learning path based on the input query.

Next, the participants are asked to complete a questionnaire to collect basic participant information and feedback on system usability. The feedback should represent participant satisfaction with the learning experience, also rated using the Likert scale.

Results analysis:

Based on the data collected from the feedback, we will evaluate our learning path recommendation model through a data analysis. The expected performance of the model will be verified in this step.

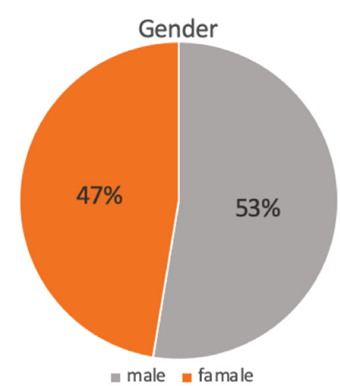

Fig. 6.2. Participants' gender information.

6.2. Analysis of experiment results

The experiment included 251 participants and generated 2440 qualified experiment result records; 1470 records were from the test group, and 970 records were from the control group. Figs. 6.2 and 6.3 show some basic information about the 251 participants, we can find that participants are balanced distributed in gender and location. It is worth to mention that all participants were randomly divided into two groups with a similar gender ratio and the ages of all participants are between 19 to 26 years old.

Table 6.1
Examples of participants' feedback.

	Learning need	Target learning object	Recommended learning path	Score
Example 1	#2	Faster RCNN	CNN, RCNN, Fast RCNN, Faster RCNN	5
Example 2	#6	Entity Recognition	Maximum Likelihood Estimation, Viterbi Algorithm, Baum Welch Algorithm, Hidden Markov Model, Entity Recognition	4
Example 3	#5	3D Reconstruction	Query failed	0

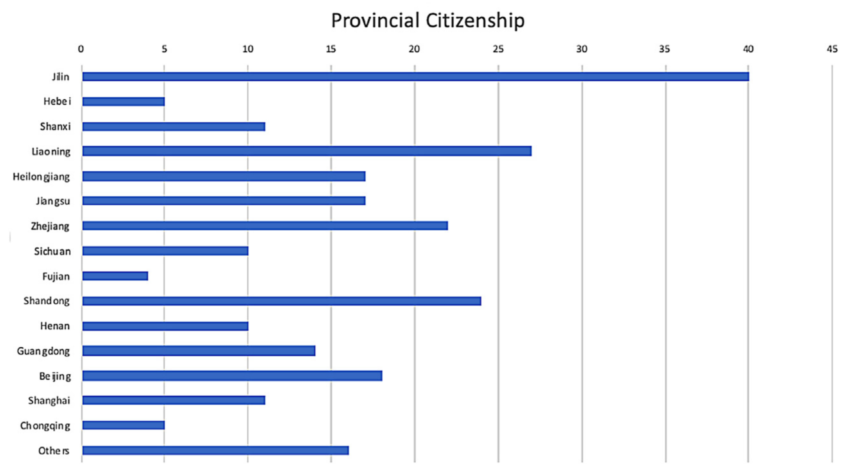

Fig. 6.3. Participants' provincial citizenship information.

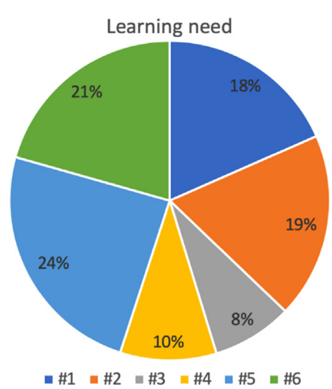

Fig. 6.4. Distribution of records containing different learning needs.

Fig. 6.4 shows the distribution of records containing different learning needs; the different colors represent the six learning needs. Learning needs #5 and #6 represent the largest portion of the pie chart, which means that e-learners are more concerned about applying knowledge learned to practical tasks. At the same time, e-learners are less likely to directly search for the target learning object (#4) and are also not much concerned about the position of the learning object in the hierarchy (#3). We are inspired by this analysis that shows that by using the learning path recommendation model, e-learners favor learning the evolution of the target knowledge (#1 and #2) and the application of the target knowledge (#5 and #6). Subsequent studies on learning path recommendation should focus on these two points.

To verify the hypothesis (i), we need to calculate the success rate based on the learning paths returned by the two models. During the experiment, all failed queries and incorrectly returned learning paths were counted as failures; the remaining results were counted as successes. Fig. 6.5 shows the success rate of the learning paths returned by the two models, labels on the horizontal axis represent different learning needs. Since learning need #4 only requires a single learning object, we did not include it in this statistic. According to Fig. 6.5, the learning path generation success rate of the experiment group is significantly higher than that of the control group.

We found that although the success rate of the control model is lower, it can still generate correct learning paths for learning needs #1, #2, #5, and #6. However, it cannot generate learning paths for #3. The reason is that, unlike learning path generation for #1, the starting node is unknown while the ending node is known. When the model needs to present the hierarchy for the current learning object in learning need #3, either the starting node or the ending node is unknown. However, there is no significant decrease in the success rate of the test model under #3. This indicates that by using the semantic relationship, the learning path generation algorithm based on the multidimensional knowledge graph is robust and will not be affected by a missing learning object in the learning path.

When aggregating the basic data about the participants collected from the questionnaire, we found that about half of the participants who successfully generated learning paths using the test model do not have the relevant learning experience. However, most of the successful participants in the control group have relevant learning experience. This indicates that the test model is user-friendly for inexpert e-learners who need guidance.

The above evaluation verifies the hypothesis (i): the learning path generation algorithm based on the multidimensional knowledge graph is better than the algorithm using one-dimensional knowledge graph. This indicates that generating learning paths through semantic relationships is more user-friendly and efficient. Therefore, it proves that the multidimensional knowledge graph framework we designed is more reasonable for generating learning paths.

To verify the hypothesis (ii), we have counted the participants' satisfaction with the recommended learning path results. Table 6.2 shows some statistics of participants' records in the test group. We find that the average total satisfaction score is 3.9. The statistical median of the satisfaction score is 4. These results prove that most participants are satisfied with the recommended

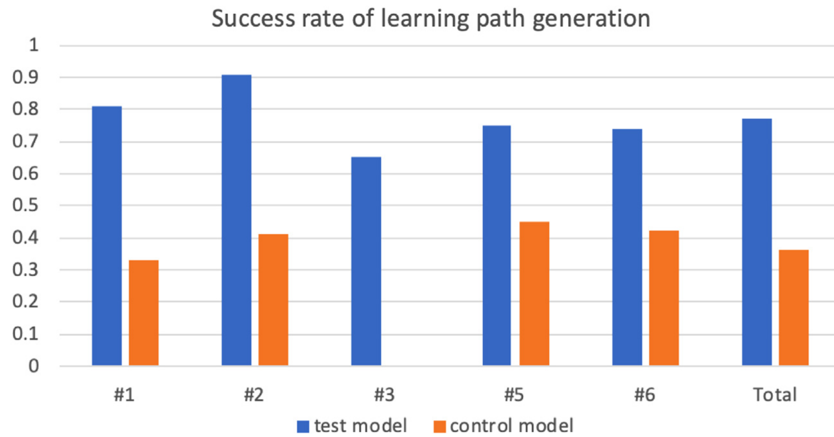

Fig. 6.5. Success rate of learning path generation under different learning needs.

Table 6.2
Statistics of participants' records in the test group.

Learning need	Average satisfaction score	Median satisfaction score	Number of records	Average length of learning path
#1	4.2	4	242	3.4
#2	4.3	4	246	2.6
#3	2.7	2	107	7.7
#4	4.7	5	130	1
#5	3.7	4	390	3.9
#6	3.6	4	355	4.4
Total	3.9	4	1470	3.8

learning path in general. When looking at the satisfaction scores for different learning needs, all the scores are over 3.6 except for learning need #3; the scores for #1, #2, and #4 are even above 4.0. This indicates that, in most cases, participants are satisfied with the selected learning paths for different learning needs. Therefore, this proves that our learning path recommendation model can recommend appropriate learning paths according to the learning need of e-learners.

It is worth noting that the relationship constraints of learning needs #1 and #2 are fewer than those for #3, #5, and #6. Fig. 6.5 shows that the test model is good at generating learning paths with fewer relationships, but the success rate decreases when generating learning paths with more relationships. Table 6.2 presents a similar trend as Fig. 6.5. The satisfaction score is higher for learning needs with simple relationship constraints such as #1, #2, and #4 (#4 has no relationship constraint). However, the satisfaction score decreases when the relationship constraints become complex, especially for learning need #3 which has the lowest average satisfaction score. At the same time, we can also find that the average satisfaction score and median satisfaction score decrease when the length of the learning path increases according to Table 6.2. This suggests that our learning path recommendation model still needs to improve in its ability to generate appropriate learning paths within the complex relationship constraints and longer length.

7. Conclusion and future work

In this paper, we presented a learning path recommendation model based on a multidimensional knowledge graph that resulted in e-learners that are more satisfied with the learning paths recommended for his learning needs. We designed a multidimensional knowledge graph with diverse relationships between learning objects for presenting learning content. Based on our knowledge graph, we proposed a learning path generation algorithm which can generate the learning path by the target learning object. This algorithm has been proven to be more efficient in generating suitable learning paths than previous algorithms. Furthermore, we proposed a weighted coefficient scoring method for selecting the target learning path considering the e-learner's learning needs. In conclusion, our learning path recommendation model can generate satisfactory learning paths for e-learners and improves their learning efficiency.

This paper introduced a method for efficiently recommending suitable learning paths for e-learners that can be expanded and used in any other e-learning knowledge field. However, there are still some limitations to be noted. When applying this model to another knowledge field, the quality of data collection will affect the results of the learning path recommendation. At the same time, future research can focus on how to improve the success rate of generating learning paths with complex relationship constraints, which can improve the quality of the recommended learning path.

CRediT authorship contribution statement

Daqian Shi: Conceptualization, Methodology, Software, Validation, Formal analysis, Data curation, Writing - original draft, Writing - review & editing. **Ting Wang:** Methodology, Data curation, Writing - original draft. **Hao Xing:** Software, Data curation. **Hao Xu:** Conceptualization, Writing - original draft, Writing - review & editing.

Acknowledgments

This work is supported by the Development of Sicence and Technology of Jilin Province, China (No. 20170203002GX, 20170101006JC), the National Natural Science Foundation of China (No. 61472159, 71620107001, 71232011).

References

[1] Marc J. Rosenberg, Rob Foshay, E-learning: Strategies for delivering knowledge in the digital age, Perform. Improv. 41 (5) (2002) 50–51.
[2] Cristobal Romero, Sebastian Ventura, Data Mining in E-Learning, Vol. 4, WIT Press, 2006.
[3] Norsham Idris, Norazah Yusof, Puteh Saad, et al., Adaptive course sequencing for personalization of learning path using neural network, Int. J. Adv. Soft Comput. Appl. 1 (1) (2009) 49–61.
[4] Tiffany Y. Tang, Gordon G. McCalla, Data mining for contextual educational recommendation and evaluation strategies, in: Handbook of Educational Data Mining, 2010, p. 257.

[5] Daniela Godoy, Analía Amandi, Link recommendation in e-learning systems based on content-based student profiles, in: Handbook of Educational Data Mining, CRC Press, 2010, pp. 273–286.
[6] Chungho Su, Designing and developing a novel hybrid adaptive learning path recommendation system (ALPRS) for gamification mathematics geometry course, Eurasia J. Math. Sci. Technol. Educ. 13 (6) (2017) 2275–2298.
[7] Chengling Zhao, Zhihui Chen, Zhifang Huang, Recommendation algorithm and application of adaptive learning path, China Educ. Technol. 8 (2015) 85–91.
[8] Vincent Tam, Edmund Y. Lam, S.T. Fung, A new framework of concept clustering and learning path optimization to develop the next-generation e-learning systems, J. Comput. Educ. 1 (4) (2014) 335–352.
[9] Guillaume Durand, Nabil Belacel, François LaPlante, Graph theory based model for learning path recommendation, Inform. Sci. 251 (2013) 10–21.
[10] Haiping Zhu, Feng Tian, Ke Wu, Nazaraf Shah, Yan Chen, Yifu Ni, Xinhui Zhang, Kuo-Ming Chao, Qinghua Zheng, A multi-constraint learning path recommendation algorithm based on knowledge map, Knowl.-Based Syst. 143 (2018) 102–114.
[11] Imane Kamsa, Rachid Elouahbi, Fatima El Khoukhi, Touria Karite, Hayat Zouiten, Optimizing collaborative learning path by ant's optimization technique in e-learning system, in: 2016 15th International Conference on Information Technology Based Higher Education and Training (ITHET), Institute of Electrical and Electronics Engineers, 2016, pp. 1–5.
[12] Eugenijus Kurilovas, Inga Zilinskiene, Valentina Dagiene, Recommending suitable learning scenarios according to learners' preferences: An improved swarm based approach, Comput. Hum. Behav. 30 (2014) 550–557.
[13] Chun Fu Lin, Yu-chu Yeh, Yu Hsin Hung, Ray I. Chang, Data mining for providing a personalized learning path in creativity: An application of decision trees, Comput. Educ. 68 (2013) 199–210.
[14] Ebru Özpolat, Gözde B. Akar, Automatic detection of learning styles for an e-learning system, Comput. Educ. 53 (2) (2009) 355–367.
[15] Mei-Hua Hsu, A personalized English learning recommender system for ESL students, Expert Syst. Appl. 34 (1) (2008) 683–688.
[16] Chih-Ming Chen, Ontology-based concept map for planning a personalised learning path, Br. J. Educ. Technol. 40 (6) (2009) 1028–1058.
[17] Chen Liang, Jianbo Ye, Shuting Wang, Bart Pursel, C. Lee Giles, Investigating active learning for concept prerequisite learning, in: Thirty-Second AAAI Conference on Artificial Intelligence, Association for the Advancement of Artificial Intelligence, 2018.
[18] Liangming Pan, Xiaochen Wang, Chengjiang Li, Juanzi Li, Jie Tang, Course concept extraction in moocs via embedding-based graph propagation, in: Proceedings of the Eighth International Joint Conference on Natural Language Processing, 2017, pp. 875–884.
[19] Chen Liang, Jianbo Ye, Zhaohui Wu, Bart Pursel, C Lee Giles, Recovering concept prerequisite relations from university course dependencies, in: Thirty-First AAAI Conference on Artificial Intelligence, Association for the Advancement of Artificial Intelligence, 2017.
[20] Liangming Pan, Chengjiang Li, Juanzi Li, Jie Tang, Prerequisite relation learning for concepts in moocs, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017, pp. 1447–1456.
[21] Hanxiao Liu, Wanli Ma, Yiming Yang, Jaime Carbonell, Learning concept graphs from online educational data, J. Artificial Intelligence Res. 55 (2016) 1059–1090.
[22] Chih-Ming Chen, Intelligent web-based learning system with personalized learning path guidance, Comput. Educ. 51 (2) (2008) 787–814.
[23] Pragya Dwivedi, Vibhor Kant, Kamal K. Bharadwaj, Learning path recommendation based on modified variable length genetic algorithm, Educ. Inf. Technol. 23 (2) (2018) 819–836.
[24] Mengyuan Chen, Mingwen Tong, Chunmiao Liu, Meimei Han, Ying Xia, Recommendation of learning path using an improved ACO based on novel coordinate system, in: 2017 6th International Congress on Advanced Applied Informatics, IEEE, 2017, pp. 747–753.
[25] Mohamed Bendahmane, Brahim El Falaki, Mohammed Benattou, Individualized learning path through a services-oriented approach, in: Europe and MENA Cooperation Advances in Information and Communication Technologies, Springer, 2017, pp. 95–102.
[26] Tung-Cheng Hsieh, Tzone I. Wang, A mining-based approach on discovering courses pattern for constructing suitable learning path, Expert Syst. Appl. 37 (6) (2010) 4156–4167.
[27] Tzone I. Wang, Kun Hua Tsai, Ming-Che Lee, Ti Kai Chiu, Personalized learning objects recommendation based on the semantic-aware discovery and the learner preference pattern, Educ. Technol. Soc. 10 (3) (2007) 84–105.
[28] Hsiao-Chien Tseng, Chieh-Feng Chiang, Jun-Ming Su, Jui-Long Hung, Brett E Shelton, Building an online adaptive learning and recommendation platform, in: International Symposium on Emerging Technologies for Education, Springer, 2016, pp. 428–432.
[29] Matteo Gaeta, Francesco Orciuoli, Pierluigi Ritrovato, Advanced ontology management system for personalised e-learning, Knowl.-Based Syst. 22 (4) (2009) 292–301.
[30] Shanshan Wan, Zhendong Niu, A learner oriented learning recommendation approach based on mixed concept mapping and immune algorithm, Knowl.-Based Syst. 103 (2016) 28–40.
[31] Shimaa Ouf, Mahmoud Abd Ellatif, Shaimaa Ezzat Salama, Yehia Helmy, A proposed paradigm for smart learning environment based on semantic web, Comput. Hum. Behav. 72 (2017) 796–818.
[32] Vadim Shmelev, Maria Karpova, Alexey Dukhanov, An approach of learning path sequencing based on revised bloom's taxonomy and domain ontologies with the use of genetic algorithms, Procedia Comput. Sci. 66 (2015) 711–719.
[33] Kuo-Kuang Chu, Chien-I Lee, Rong-Shi Tsai, Ontology technology to assist learners' navigation in the concept map learning system, Expert Syst. Appl. 38 (9) (2011) 11293–11299.
[34] Fausto Giunchiglia, Biswanath Dutta, Vincenzo Maltese, Faceted lightweight ontologies, in: Conceptual Modeling: Foundations and Applications, Springer, 2009, pp. 36–51.
[35] Yanru Zhu, Peng Wang, Yaqin Fan, Ying Chen, Research of learning path recommendation algorithm based on knowledge graph, in: Proceedings of the 6th International Conference on Information Engineering, Association for Computing Machinery, 2017, p. 11.
[36] Mary L. McHugh, Interrater reliability: the kappa statistic, Biochem. Medica 22 (3) (2012) 276–282.
[37] Gyula Mester, Rankings scientists, journals and countries using h-index, Interdiscip. Descr. Complex Syst. 14 (1) (2016) 1–9.
[38] Hyunyoung Choi, Hal Varian, Predicting the present with google trends, Econ. Rec. 88 (2012) 2–9.
[39] You-Na Lee, John P. Walsh, Jian Wang, Creativity in scientific teams: Unpacking novelty and impact, Res. Policy 44 (3) (2015) 684–697.
[40] J.B. Arbaugh, Learning to learn online: A study of perceptual changes between multiple online course experiences, Internet Higher Educ. 7 (3) (2004) 169–182.
[41] Lynn Silipigni Connaway, Joyce Kasman Valenza, Christopher Cyr, Tara Tobin Cataldo, Amy G Buhler, Ixchel M Faniel, Rachael Elrod, Randy A Graff, Samuel R Putnam, Brittany Brannon, et al., Authority, context and containers: Student perceptions and judgments when using google for school work, in: World Library and Information Congress, International Federation of Library Associations and Institutions, 2017.