

# Auto-Construction of Course Knowledge Graph based on Course Knowledge

Peng Zhu<sup>a,b</sup>, Wei Zhong<sup>c,\*</sup>, and Xianming Yao<sup>d</sup>

<sup>a</sup>*School of Information Science and Technology, Yunnan Normal University, Kunming, 650500, China*

<sup>b</sup>*Key Laboratory of Educational Informatization for Nationalities of Ministry of Education, Yunnan Normal University, Kunming, 650500, China*

<sup>c</sup>*Archives Center of Yunnan Normal University, Kunming, 650500, China*

<sup>d</sup>*School of Information Engineering, Qujing Normal University, Qujing, 655011, China*

---

## Abstract

Course knowledge informationization increases the difficulties to present intrinsic connections among knowledge points in a visual way. This paper presents research on the auto-construction of course knowledge graphs based on knowledge graphs. Detailed questions regarding the auto-construction of course knowledge graphs are illustrated through a series of mature and reliable technical means. Additionally, this paper studies the application of course knowledge graph visualized navigation, providing new methods of course knowledge information construction and decreasing the learning costs for new knowledge.

*Keywords:* knowledge graph; course knowledge; ontology pattern; knowledge extraction

(Submitted on June 11, 2019; Revised on July 7, 2019; Accepted on August 12, 2019)

© 2019 Totem Publisher, Inc. All rights reserved.

---

## 1. Introduction

Education has always been a vital topic in the development of the national society, economy, and politics. The traditional one-way teaching-accepting mode has been blamed universally due to the lack of diversity of teaching methods and the insufficiency of teaching equipment. The information society has brought a new mode of education [1]. The concepts of fragmented learning [2], distant education [3], and mobile learning [4] have been pioneered, and they have dramatically improved traditional learning methods and students' learning initiatives. The birth of the Internet generated a suite of new learning methods and platforms [5-6], which, unrestricted by time or space, make learning accessible at all times and all places.

However, these learning modes only convey traditional learning resources to the Internet. Though vivid media-rich content is added and the interactivity of the teaching process is improved, students need to understand knowledge points and sum up the correlation among knowledge points by themselves to internalize the content of courses, so that a holistic set of self-understanding of the course content can be formed. Such a process is similar to the traditional one to an extent. Therefore, Internet learning remains in the phase of content learning. It lacks a depth of knowledge points and a correlation among knowledge points, so the learning process cannot be optimized significantly.

The birth of semantic net [7] brought new opportunities to the current teaching methods. The emphasis of the Internet (the predecessor of semantic net) is the interconnection of webpages, while the emphasis of semantic net is the interconnection of knowledge. This change of connection allows semantic net to get through to the intention of users so that more suitable content can be pushed, which endows new thought to Internet education. Doubtlessly, the problems in traditional education will be solved and the study efficiency will be improved if the textual learning in a traditional course (webpage interconnection) is transferred into the learning of knowledge points and their interrelation (knowledge interconnection).

The knowledge graph [8] is one of the most important products of semantic net, fulfilling the transferring from

\* Corresponding author.

E-mail address: zhongwei@ynnu.edu.cn

webpage interconnection to knowledge interconnection. Knowledge graphs consist of nodes, which usually refer to concepts and entities. Sides, which express the semantic relations between concepts and entities, are also known as knowledge. The interconnection of innumerable nodes and sides forms the abstract description to the objective world, which objectively demonstrates our cognition to the world—the knowledge. Course knowledge is an integrated whole with organic connections. There are synonymies, brotherhood, antecedent relations, and subsequent relations between course knowledge and nodes of knowledge graphs. The theory of knowledge graph is effective in the construction of course knowledge, which is also the course knowledge graph, providing an efficient technical method for the construction of course knowledge and its relations.

The goals of this paper are to promote the development of the construction technology of course knowledge graphs, solve the problems in traditional teaching, and enhance the possibility of effective learning. This paper studies the auto-construction of course knowledge graphs based on knowledge graphs, analyzes problems as well as the train of thought of construction of course knowledge graphs, builds a course knowledge graph and its visualization application with a study of program design courses, and provides reference to future research.

## 2. The General Framework of Course Knowledge Graph Construction

During the construction of a course knowledge graph, the semi-structured online community webpages are adopted in this paper, bonding the Internet webpages retrieved from the search engines as the data resources of knowledge graph construction. The construction is divided into three major sections, which deliberate the full-automatic course knowledge graph construction and its application.

From the semi-structured text to the extraction of knowledge, and the discrete knowledge points to the auto-construction of the course knowledge graph and the deep application, the overall process contains many complicated research tasks and technical proposals. According to the analysis of knowledge graphs and combining the features of course knowledge, three sections of a course knowledge graph are divided as shown in Figure 1.

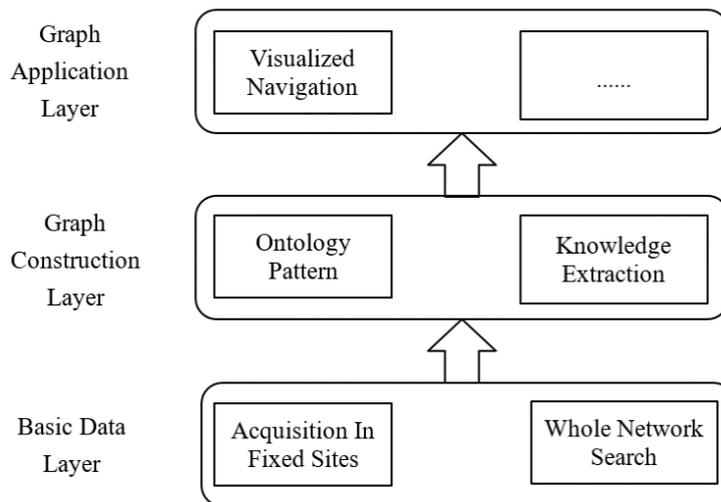


Figure 1. Overall framework of knowledge graph construction

(1) The basic data layer: It forms the data foundation of knowledge graph construction, including the semi-structured data acquisition in the fixed sites and the unstructured data through the whole network retrieval. The basic data layer offers data support to the upper layer for building the ontology pattern and extracting the knowledge of course knowledge graphs.

(2) The graph construction layer: It completes the task of construction of course knowledge graphs. Among these, the ontology pattern performs the top layer task of construction of the course knowledge graph, while knowledge extraction completes the extraction from the specific entities and their relations. The graph construction layer acquires data from the basic data layer and provides the inquiry function of structured data to the upper layer by completing the knowledge graph construction with a certain algorithm.

(3) The graph application layer: It is established on the data foundation provided by the interfaces of the graph construction layer and is used for completing the concrete applications, such as knowledge graph navigation services, depth

questions and answers, and knowledge inferences. This paper focuses on knowledge graph navigation services.

### 3. The Construction of Course Knowledge Graph Corpus

Corpus is the dominant foundation of course knowledge graph construction. This paper divides corpus construction into two methods: fixed sites acquisition and whole network retrieval.

Acquisition in fixed sites: As stated above, several network course platforms or specific communities offering course resources have emerged in the era of the Internet, and they involve many valuable pieces of literature and documents for public learning. These platforms are featured with fixed webpage structure, from which valuable information can be fully explored based on the heuristic rule. The feature of the semi-structured data in fixed site acquisition helps extract the structured knowledge effectively. Making use of web crawler technology, this paper crawls the relevant learning resources from W3C School [9].

Whole network retrieval: As its name implies, it crawls all the webpages from the Internet through crawler technology and extracts the interested resources through the supervised algorithm. However, due to the large volume of the Internet, the workload is high. Therefore, a compromise, which is to crawl the relevant webpages automatically through Internet searching engines and download them to the local file system as the resources of the basic data layer, is adopted. Although the data searched through whole network retrieval is also semi-structured, it is impossible to uniformly model all the webpages to extract their semi-structured data due to the randomness of the webpage structure. Thus, the resource extraction mainly occurs from the free text of the text segment retrieved from the whole network and is saved on the basic data layer.

There is a total of 215,823 relevant webpages are captured in this paper, including four popular themes. Among them, "C Language" contains 38,111, "Python Language" contains 45,098, "Java Language" contains 44,876 and "PHP Language" contains 87,756. The basic data space occupies 9,823M of space. Generally speaking, this dataset meets the demand of course knowledge graph construction.

### 4. The Ontology Pattern Construction of Course Knowledge Graph

The ontology pattern is the top pattern in the knowledge graph and defines the top concept, the attribute features of concepts, and the hierarchical relations among concepts in the knowledge graph. All of the entities and their relations in the knowledge graph are derived from the ontology pattern. Therefore, ontology pattern construction is a priority in knowledge graph construction.

The ontology pattern stems from ontology and describes the concepts and interrelation among them. The ontology is composed of concept-word reference (terminology) and hierarchical relations. According to these two aims, the auto-construction of ontology pattern is researched in this paper.

#### 4.1. The Auto-Extraction of Terminology based on the Heuristic Rule

In the construction of domain ontology, terminology normally refers to nominal phrases without the inclusion of other part-of-speech temporarily. It can be seen that the auto-construction of ontological pattern in courses can be implemented by constructing the rule of composition of nouns, whereas not all nouns belong to the concept. Nouns are composed of general nouns and proper nouns. General nouns are the terminologies constantly refined from the objective world during the development of language and belong to the category of terminology. Meanwhile, proper nouns are ordinarily used to modify an individual concept. The nouns or phrases that the individual concepts correspond to are generally entities. Thus, general nouns should be considered when domain-specific terms are extracted.

In NLP [10] (the Tokenizer of Beijing Institute of Technology), the tab of general nouns is "n" and the tabs of proper nouns are "nr", "ns", and "nz", representing names, toponymies, organization names, and proper names, respectively. The tabs "t" and "m", which stand for time and quantity, and tab "q", which matches the former two and represents units, are also helpful to this paper.

The rules of parts of speech are fully used to extract the terminologies in the course domain in this paper. On the basis of the analysis of textual resources of course domains, the rules in Table 1 are applied to extract domain terminologies after extracting the date from text segments, reducing noise, and labeling the particles and parts-of-speech.

In Table 1, three rules of the composition are defined. All "n"s in the rules stand for the common nouns that are behind

participles and in the parts-of-speech of "n". The words composed by the single common noun are extracted directly as domain-specific terms, as are the nominal phrases composed by multiple common nouns. Because the function in the text of gerunds "vn" is similar to nouns, they are extracted as domain-specific terms. Additionally, the adjuncts of the parts-of-speech of "a", "ag", and "b" can be extracted as domain-specific terms if they are closely followed by nouns. Meanwhile, these parts-of-speech can pose between two nouns, but they should be ended with nouns or gerunds. Through such lists of examples, parts of the course domain-specific terms are extracted in accordance with the corresponding rules, and the rules are viewed as feasible from the data of examples.

Table 1. Simulation parameters of the signals

No.	Rules	Examples	Instruction
1	N	Language, program	Single noun
2	N+n+	Data class, system management	Combined noun
3	N vn+n vn	Access right, exception handling	A combined noun with the gerund
4	A ag b+n vn vg+ a ag b+n vn vg	A unique identification, relative path	A combined noun with gerund and modifier

#### 4.2. The Construction of Concept Hierarchical Relations based on the Association Rules

Strictly speaking, terminologies can be used as concepts directly after extracting the terms of program design courses, so that the footing stone of the construction of concept hierarchical relations is prepared. The concept hierarchical relation is a taxonomic relation, describing the relations between whole and part and even the relations in inheritance. The construction of hierarchical relations integrates discrete concepts into an organic entirety to lay a foundation for knowledge inference, depth questions and answers, and more.

We adopt a method on the basis of association rules [11] in this paper to study the construction of hierarchical relations of program design courses. A basic hypothesis of association rules is the following: if a document in which concept  $i$  appears contains all the documents in which concept  $j$  appears, concept A is the upper seat concept of concept B. This theory can be defined by Equation (1).

$$R(\eta_{ij}) = \frac{S_{\eta_{ij}}}{S_{\eta_i} + S_{\eta_j}} \times \theta \quad (1)$$

In Equation (1),  $S_{\eta_{ij}}$  is the document collection of the co-occurrence of concept  $i$  and concept  $j$ .  $S_{\eta_i}$  is the document collection containing the occurrence of concept  $i$ ,  $S_{\eta_j}$  is the document collection containing the occurrence of concept  $j$ . The scalar  $\theta$  adjusts the distribution range of association strength. If  $R(\eta_{ij})$  is greater than 1, concept  $i$  is the upper seat concept of concept  $j$ . If it is between interval estimation (0, 1), parts of the common documents appear in both concepts. Hence, the hyponymy between the concepts can be inferred.

On account of association rules, the concept with a higher TF-IDF value is selected as the primitive concept after calculating the importance of each concept through TF-IDF. Then, the concept hierarchical relation of program design courses is constructed from top to bottom through the process of iteration. Eventually, the construction of ontological pattern is accomplished.

### 5. Knowledge Extraction of Course Knowledge Graph

Ontology patterns constitutes the core of knowledge graphs, yet for knowledge graphs to take roots and better serve other applications, instances corresponding to concepts in ontology patterns and the semantic relations among them must be extracted to enrich the connotation of concepts. Thus, the task of knowledge extraction of knowledge graphs is formed. The knowledge ultimately extracted is stored as a form of a triple: (entity 1, relationship, entity 2). Based on the different extents of structuring of data sources, knowledge extraction expressed in the form of triples is divided into two types of data sources in this paper: structured and unstructured.

#### 5.1. Knowledge Extraction of Structured Data Sources

A webpage is semi-structured data; however, for specific sites or communities, its structure is basically fixed. Therefore, it can be modeled, regarded as structured data, and then the data can be directly extracted and stored in knowledge triples.

We use webpages in W3School as the data source in this paper, and each webpage is a detailed explanation for a specific concept. `<title></title>` in HTML declares the concept to be illustrated in the webpage, while the first-level heading `<h1>` is also used for illustrating. The concept in the rest of the content is explained by sub-headings, with each of them declared by secondary headings `<h2>`. Moreover, all tables and instances include certain tags. Hence, heuristic rules can be built to extract corresponding content to fill in the triple, which will be stored directly as knowledge.

(1) Knowledge extraction of tables: A table consists of a header and data entries. It also includes secondary headings of the table to illustrate the purpose of the table. Accordingly, we can use the secondary headings as "entity 1" in the triple, the text of the table header as the name of "relationship", and the data in each row of the table as "entity 2" after the concatenation operation. Thus, the extraction of the table's data can be realized, as shown in Figure 2.

#### C Variables

Variables are merely names of operable memory area of programs. Each variable in C has a specific type.  
The name of a variable is composed of letters, numbers and underscore characters.  
It has to begin with a letter or an underline.  
Several basic types of variables:

Type	Description
char	Normally an 8-bit byte (one byte). It's an integer type.
int	The most natural value of integers for machines.
float	Single precision floating point number.
double	Double precision floating point number.
void	Indicate the absence of types.

Figure 2. Instance of table knowledge extraction

Figure 2 is an illustration of the data type of entity "c variable" in the data source. The range of the table's title is below the tag `<h1>`. The table includes two columns of "type" and "description". Meanwhile, the table also includes five rows of data records, such as char and int. Based on the heuristic rule built above, five entity relationships can be extracted: (C variable, type, char-normally an 8-bit byte, one byte, integer type), (C variable, type, the most natural value of integers for machines), (C variable, type, for machines, single precision floating point value), (C variable, type, for machines, double precision floating point value), and (C variable, type, for machines, indicates the absence of types).

For the knowledge extraction of code blocks, a code block is an explanation of the specific usage scenario of a concept. It has great reference value. It has a specific form in HTML, generally included in tag `<code></code>`. Furthermore, code blocks are also included in specific sub-headings. Thus, the knowledge extraction of code blocks can be realized by extracting corresponding tags, as shown in Figure 3.

```
#include <stdio.h>
// variable declaration
extern int a, b;
extern int c;
extern float f;

int main ()
{
    /* variable definition */
    int a, b;
    int c;
    float f;
    /* actual initialization */
    a = 10;
    b = 20;
    c = a + b;
    printf("value of c : %d in", c);
    f=70.0/3.0;
    printf("value of f : %f\n", f);
    return 0;
}
```

Figure 3. Instance of table knowledge extraction

Figure 3 demonstrates the code block instance used in the trial. According to the rule mentioned above, triple

knowledge can be extracted. (C variable, instance, #include <stdio. H> //variable declaration extern int a, b; extern int c; ...)

## 5.2. Knowledge Extraction of Structured Data Sources

Knowledge extraction of structured triples is relatively simple, and extraction strategies can be developed according to the features of the structure. Yet for unstructured data, it is not easy to extract triple knowledge, and techniques of natural language must be employed synthetically. The unstructured data mentioned here mainly include data of fixed sites with tables and code blocks removed and the natural-language-expressed free text obtained by extracting the text, which is retrieved by entire web retrieval.

Given the characteristic of randomness and ambiguity of natural language, it is not easy to extract its fixed structure. Techniques such as noise reduction, sentence segmentation, word segmentation, and part-of-speech tagging are also commonly used. Meanwhile, although the natural language has a linear chain structure, it has problems such as long-distance dependencies. The words in a sentence are not only related to the words before and after it, but also possibly related to the words at the beginning and end of the sentence. This brings a great challenge to knowledge extraction. Taking "Variable definition assigns a data type and includes the list of one or several variables of this type" as a case, the knowledge triple can be relatively easily obtained (variable definition, assign, a data type). However, there exists another knowledge triple (variable definition, include, list of one or several variables of this type). "Entity 1" of the second knowledge triple is located at the beginning of the example sentence, and here the long-distance dependency exists.

For the extraction of knowledge triples with long distance dependency, it is an effective strategy to use dependency parsing natural language processing. Dependency parsing reveals the syntactic structure by parsing the dependency relationship among the components in a linguistic unit. Intuitively, dependency parsing recognizes the grammatical components such as "subject, verb, object" and "attribute, adverbial, complement" in a sentence, and it parses the relationship among the components. By obtaining the subject, verb, and object in a sentence, the knowledge triple expressed by the verb is obtained. Then, the knowledge triple is more apprehensible by supplementing the attribute, adverbial, and complement. As a result, dependency parsing is an effective method for the extraction of knowledge triples.

The knowledge triple extraction algorithm based on dependency parsing is shown in Algorithm 1. It can be seen from Algorithm 1 that the key of knowledge triple extraction based on dependency parsing is to perform the dependency parsing, then obtain the subject and object of each predicate in sequence and supplement the attribute, adverbial, and complement. Then, the extraction task of the knowledge triple is completed.

---

### Algorithm 1: Knowledge Triple Extraction Algorithm based on Dependency Parsing

**Step 1** Pre-processing. Execute tasks such as noise reduction and sentence segmentation.

**Step 2** Dependency parsing. Execute dependency parsing on the LTP platform of the Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology, and obtain the results.

**Step 3** Obtain the predicate list. Scan all predicates in the sentence in the order of part-of-speech of verbs "v" defined by the LTP platform.

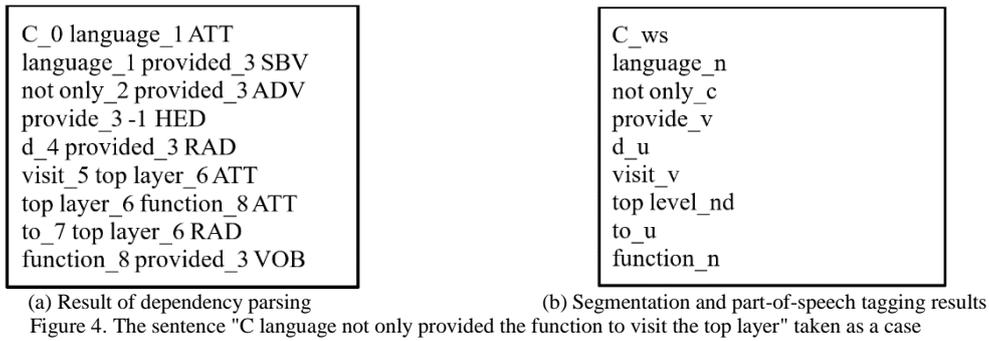
**Step 4** Obtain the subject, verb, and object. For each predicate in the predicate list, look up the subject of the predicate according to the subject-verb relationship (SBV), and then look up the object of the verb according to the verb-object relationship (VOB), indirect object relationship (IOB), and prepositional object relationship (FOB). Then, build the knowledge triple (subject, verb, object).

**Step 5** Supplement the attribute, adverbial, and complement. According to the attribute-center relationship (ATT), adverbial-center structure (ADV), and verb-complement structure (CMP), supplement the attribute, adverbial, and complement of the subject and the verb and object in the knowledge triple, making the knowledge triple more apprehensible.

---

The sentence "C language not only provided the function to visit the top layer" is the focus of Figure 4. The result of dependency parsing is demonstrated in Figure 4(a), and the segmentation and part-of-speech tagging results of the example sentence are shown in Figure 4(b).

According to Algorithm 1, finish performing the dependency parsing, and simultaneously obtain the predicate list (provide, visit) after obtaining the part-of-speech tagging result. Then, the subject "language" of the verb "provide" is obtained according to the algorithm, the object "function" is obtained, and the triple is constructed (language, provide, function). In accordance with the algorithm, the verb "visit" does not have links to subjects or objects, so "visit" is abandoned. Finally, the attribute, adverbial, and complement of the triple (language, provide, function) is supplemented, (C language, provided, function to visit the top layer) is obtained, the knowledge triple is output, and the extraction of knowledge is accomplished.



### 6. Visualized Navigation Service for Program Design Courses based on Course Knowledge Graph

The knowledge graph built in previous chapters has stored rich semantic knowledge with triples as units. In order to bring its value into full play, relevant tools built at the graph application layer, which is the top layer of knowledge graph, must be employed. In previous studies, numerous applications such as knowledge navigation [12], deepQA [13], information retrieval [14], and knowledge reasoning [15] have been covered. Starting with the auto-construction of knowledge graph of program design courses, this paper studies the most basic knowledge visualized navigation service in these applications and provides the technical premise for the deep application of program design courses. Meanwhile, this paper also pays great attention to the general trend of education informatization. By complying with the development of times and taking program design courses as the research emphasis, it provides systematic knowledge service of knowledge interconnection to the educational audience in order to achieve the goals of improving educational methods and promoting learning efficiency.

The architecture of the knowledge graph visualized navigation service complies with the classic MVC pattern. The storing of model layer data is realized by employing the NEO4J database. The controller is implemented by using Java in the business logic processing and JSP in the view layer. The visualized knowledge navigation interface of the graph structure is constructed by the graph visualization render engine. The operation result is shown in Figure 5.

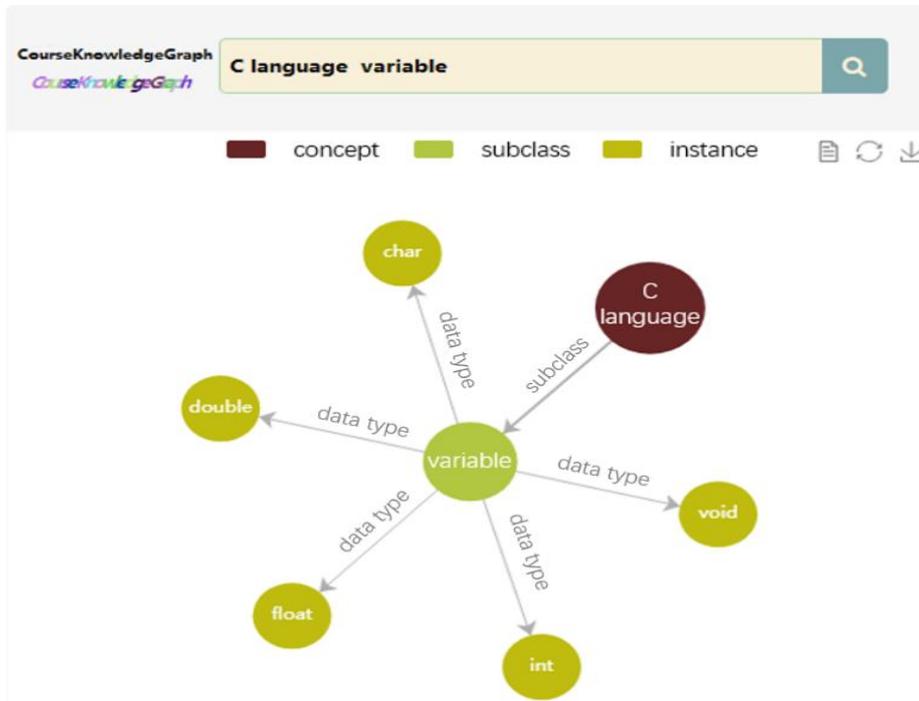


Figure 5. Knowledge graph navigation service for program design courses

When users click on a knowledge point, related knowledge will be listed, from which the users can find the knowledge they need. More detailed resources can be obtained by clicking on relevant links, as shown in Figure 6.

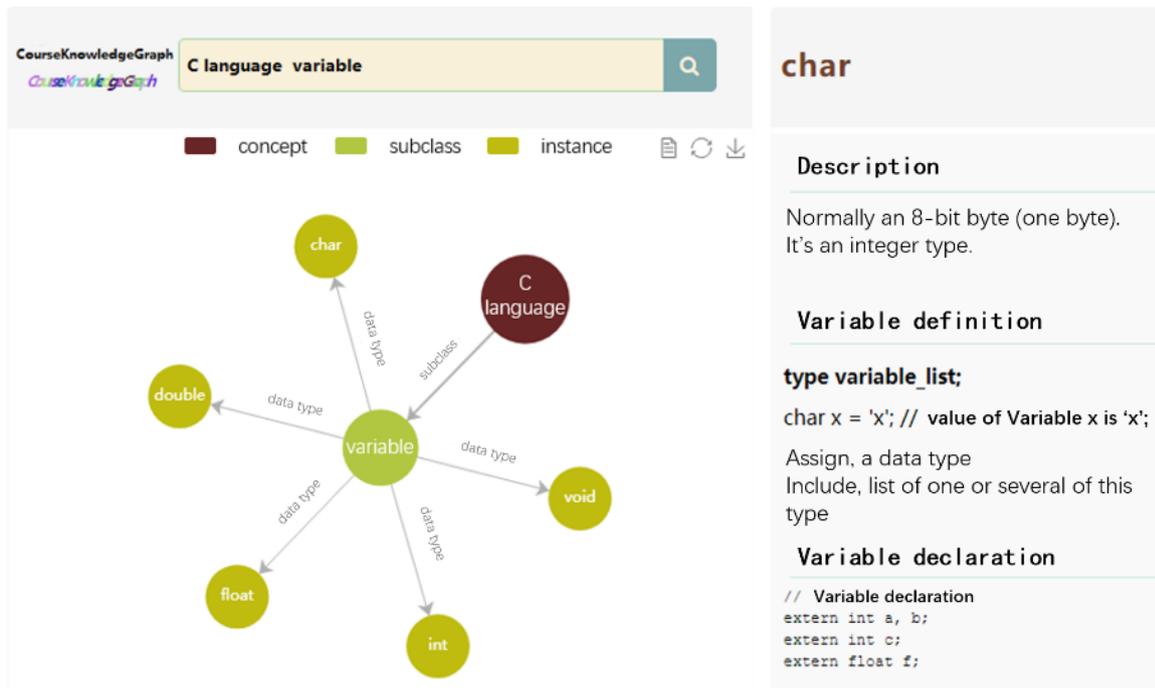


Figure 6. Knowledge graph navigation service for program design courses

## 7. Conclusions

This paper focused on the general trend of education informatization. With the help of knowledge graph as a useful tool, we conducted a key study on the auto-construction of course knowledge graph, taking program design courses as the research field. To a certain extent, we changed the current situation of webpage interconnection in internet education. We investigated knowledge interconnection and a knowledge visualized navigation service in the application, improving knowledge graph services for program design courses. Meanwhile, further studies could be carried out. Research on the auto-construction of knowledge graphs could be more refined, such as in the aspects of attribute extraction and knowledge fusion. In applied research on knowledge graphs, learning path recommendation, learning results assessment, and audience transfer learning assistance could also be conducted. These will be the key research fields in the future.

## Acknowledgements

The research is supported by the National Nature Science Fund Project (No. 61562093), Key Project of Applied Basic Research Program of Yunnan Province (No. 2016FA024), Program for Innovative Research Team (in Science and Technology) in University of Yunnan Province, and Starting Foundation for Doctoral Research of Yunnan Normal University (No. 2017ZB013).

## References

1. H. Gao, H. W. Pan, and Z. G. Lu, "Exploration on the Teaching Reform of Computer Major under the Background of 'Internet +'," *Software Engineering*, Vol. 21, No. 2, pp. 56-59, February 2018
2. C. B. Wang, X. P. Li, F. N. Zhao, and L. Zhang, "Study on Fragmented Learning in the Era of Big Data," *E-Education Research*, Vol. 270, No. 10, pp. 26-30, October 2015
3. L. Chen, S. Y. Lin, and Q. H. Zheng, "Opportunities and Challenges of Distance Education in China in the Era of 'Internet +'," *Modern Distance Education Research*, Vol. 139, No. 1, pp. 3-10, January 2016
4. Y. F. Li, M. J. Wang, and J. J. Wang, "Mobile Learning System and Its Related Learning Model," *Open Education Research*, Vol. 18, No. 1 pp. 152-158, February 2012
5. "Coursera," (<https://www.coursera.org/>, last accessed on February 22, 2019)
6. "Netease Open Classes," (<http://open.163.com/ted/>, last accessed on February 22, 2019)
7. T. Berners-Lee, "WWW: Past, Present, and Future," *Computer*, Vol. 29, No. 10, pp. 69-77, October 1996
8. H. Uszkoreit and F. Xu, "From Strings to Things SAR-Graphs: A New Type of Resource for Connecting Knowledge and Language," in *Proceedings of the International Conference on NLP & Dbpedia*, Vol. 1064, pp. 109-117, January 2013
9. "W3school," (<https://m.w3school.cn/>, last accessed on February 22, 2019)
10. "NLPIR Chinese Word Segmentation System," (<http://ictclas.nlpir.org/>, last accessed on February 22, 2019)

11. W. J. Cai, X. H. Zhang, J. Q. Zhu, and Y. Y. Zhu, "Survey of Association Rule Generation," *Computer Engineering*, Vol. 27, No. 5, pp. 31-33, May 2001
12. J. P. Qiu and H. Lv, "Visualization Analysis of Research on Knowledge Network based on Mapping Knowledge Domains," *Information Science*, Vol. 31, No. 12, pp. 3-8, December 2013
13. Z. Y. Du, Y. Yang, and L. He, "Question Answering System of Electric Business Field based on Chinese Knowledge Map," *Computer Applications and Software*, Vol. 34, No. 5, pp. 153-159, May 2017
14. L. Shao, "Research and Application of Knowledge Graph-based Search Engine," School of Computer Science & Engineering, March 2016
15. Y. B. Wu, D. H. Zhu, X. G. Liao, D. Zhang, and K. Y. Lin, "Knowledge Graph Reasoning based on Paths of Tensor Factorization," *Pattern Recognition and Artificial Intelligence*, Vol. 30, No. 5, pp. 473-480, May 2017

**Peng Zhu** is a master's student in the Key Laboratory of Educational Informatization for Nationalities (Ministry of Education) at Yunnan Normal University. His research interest is the informatization of ethnic education resources.

**Wei Zhong** is an associate professor in the Key Laboratory of Educational Informatization of Nationalities (Ministry of Education) at Yunnan Normal University. She specializes in education informationization in language teaching. Her researches focus on utilizing new technology in language teaching, especially educational technology in multilingual contexts.

**Xianming Yao** graduated from Kunming University of Science and Technology with a master's degree and works as an information engineering lecturer at Qujing Normal University. His current research interests include knowledge graph construction, domain ontology, and automatic question answering systems.