[1]Yong Xu

[2,*]Vladimir Y. Mariano

[3]Mideth Abisado

[4]Alexander A. Hernandez

Semi-automatic Knowledge Graph Construction Based on Deep Learning

Abstract: - The paper studied course knowledge graph in teaching resources and curriculum knowledge management tasks from the perspective of knowledge management. Considering that the course of Python Language Programming itself has formed a relatively complete knowledge system and knowledge point structure, the paper adopted a top-down approach to build the knowledge graph. Firstly, the paper obtained different types of course-related corpus and data from different sources, and then constructed the ontology layer of Python programming course. At the ontology level, the paper defined the concept type, relation type and attribute type of the course domain respectively. Considering the completeness of knowledge points in the curriculum domain knowledge graph, the paper extracted all entities, relationships, attributes, and its values from the curriculum corpus using a semi-automatic extraction method that takes into account both accuracy and efficiency based on the modeling results of the ontology layer. Then they were transformed into triples in the form of < entity, relationship, entity > or < entity, attribute, attribute value > to build data layer of knowledge graph. Finally, visualization of triplet data was realized through Neo4j graph database.

Keywords: Python Programming, Knowledge Graph, Entity Recognition, Attribute Extraction, Graph Database.

I. INTRODUCTION

Nowadays, information technology in education has received unprecedented attention and rapid development, which has brought new technical support and development opportunities for curriculum teaching reform in the new environment [1-3].

The concept of Knowledge Graph (KG) was proposed in May 2012, which is a structured graph network knowledge representation method [4]. With powerful semantic processing and knowledge storage capabilities, knowledge graph can standardize the organization of valuable information based on massive original corpus and help users to find the hierarchy and logical relationship between knowledge through visualization technology [5]. On the other hand, knowledge graph has powerful knowledge reasoning ability, which can meet the needs of deep knowledge service in various situations [6]. Knowledge graph technology can rationalize and organize massive curriculum teaching resources by standardized means and improve the efficiency of curriculum activity [7].

Many scholars from the world study curriculum knowledge graph [8, 9]. Li Jiazhe et al. used natural language processing technology to draw a curriculum knowledge map for junior middle school Chinese textbooks, aiming at the problems of unclear knowledge points and lack of connections among knowledge points in the existing junior middle school Chinese textbooks, and visualized the text knowledge map through a web page[10]; Yang et al. built knowledge graph of high school mathematics curriculum and applied it to intelligent question answering task to realize automatic answer of high school mathematics curriculum question[11].

To sum up, the existing studies on curriculum knowledge graph mainly focus on subject categories, lack the analysis and integration of a single course knowledge point, and have the problem of coarse-grained knowledge point division, and mainly focus on compulsory education courses, and there are few studies on the construction of knowledge graph for basic courses.

Python Language Programming course was considered the research object. Firstly, ontology modeling was carried out in combination with expert domain knowledge. Ontology concept was divided into 8 categories, ontology relationship analysis into 5 categories, and entity attributes into 18 categories. On this basis, the triplet information was constructed [12]. Then, Neo4j was used to construct knowledge graph of Python Language Programming [13].

[1] National University, Manila, Philippines; Anhui University of Finance & Economics, Bengbu, China
[2] National University, Manila, Philippines
[3] National University, Manila, Philippines
[4] National University, Manila, Philippines
*Corresponding author: Vladimir Y. Mariano

II. RELATED WORKS

Knowledge Graph was originally applied to semantic retrieval, which is a structured knowledge representation method used to accurately describe knowledge in the objective world. Knowledge graph also was viewed as a knowledge base that stores objective entities and its relationships through a graph network. The knowledge graph includes two components, data layer and ontology layer. Ontology layer defines concepts, relationships, attribute types, expression norms and other contents in the constructed knowledge graph. The definition of the ontology layer has constraints and norms on the data layer, which stores triples in the form of < entity, relationship, entity > or < entity, attribute, attribute value > that are extracted from corpus according to ontology layer constraints and can be visually displayed through graph databases (such as Neo4j, etc.) [14].

Knowledge graph is the result of development on the basis of many related technologies, including semantic network, ontology, knowledge base, semantic web, etc. It is a network structure that describes and represents the human knowledge system. In the early 1970s, with the development of intelligent question answering, the construction of knowledge base has gradually become a key issue in this field. In 1980, researchers in the field of artificial intelligence introduced the philosophical concept "ontology" into the task of domain knowledge characterization, which defines the types of terms in a specific field and the types of relations between terms. The Semantic Internet provides a lot of resources and data for the construction of a new generation of knowledge base, and a series of Semantic web and knowledge base fusion projects, such as DBpedia and FreeBase, are derived. The search engine service based on knowledge graph officially launched by Google in 2012 is a milestone achievement and product in the development of knowledge graph.

A. Ontology Modeling of Knowledge Graph

The term noumenon originates from the field of philosophy, which mainly refers to the abstract generalization of objective things in the real world and the revelation of their essence. Ontology in computer science is a shared conceptual model used to describe domain knowledge [15]. Ontology not only contains the basic concepts in a certain subject field and divides the categories and levels of the concepts, but also defines the relationships between different concepts and the basic attributes of concepts, which can be regarded as a set of recognized terms in a specific field.

Ontology defines the scope, types and relationships of concepts in the domain by defining the metadata in the domain. If the whole knowledge graph is regarded as a knowledge tree, the ontology is the trunk of this knowledge tree, which limits the "growth range" of domain knowledge. It is the basis and constraint of knowledge graph construction. After the ontology modeling is completed, we can extract the entity data that is instantiated in a specific domain and focuses on actual business. At the same time, based on the attribute category and relationship category defined in the ontology model, we can determine the relevant attributes of the entity and its relationship and other entities, so as to integrate the massive data into the knowledge graph which fully covers domain knowledge. Therefore, the ontology can be regarded as the upper mode of knowledge graph, which can be regarded as the concrete expression of the ontology model.

B. Knowledge Graph Construction Method

We have three ways to get knowledge graph: bottom-up, top-down, hybrid [16]. Bottom-up construction method emphasizes the breadth of the scope of knowledge and extracts a large number of triples from open Internet corpus through various data mining technologies to fill the data layer, then screens and summarizes the fragmented knowledge points, and abstracts the concepts with certain generalities step by step. Finally, the top pattern layer is formed [17]. Top-down construction method emphasizes the depth and precision of knowledge. First, it is necessary to refer to the existing mature knowledge system in the industry and use professionals and experts to first model the ontology layer and pattern layer according to industry rules or industry experience, define concept types, relationship types and attribute names, and gradually refine concepts and relationships downward to build a top-down "concept tree". Finally, high quality instantiated entity information is extracted from the domain data and the concept tree is filled.

The construction generally includes data acquisition, knowledge extraction [18], knowledge fusion [19] and knowledge storage. Data acquisition means obtaining initial corpus needed for knowledge graph. Knowledge extraction includes named entity identification, relationship extraction, attribute extraction on the obtained structured and unstructured data, and obtain triplet information, so as to construct corresponding knowledge graph "subgraph". In order to resolve the conflicts between data from different sources, knowledge fusion is needed, and the fusion process mainly includes entity disambiguation and entity alignment, in which entity disambiguation is mainly to eliminate the influence of "polysemous word" on semantic understanding, while entity alignment is the process of "normalization" of entities with the same or similar connotations. At present, there are two main storage methods of knowledge graph: RDF and graph database.

III. CONSTRUCTION OF KNOWLEDGE GRAPH OF PYTHON LANGUAGE PROGRAMMING COURSE

A. Main Idea

The paper considered the Python Language Programming course, aiming at teaching resources and curriculum knowledge management tasks from the perspective of knowledge management, and studied construction method of curriculum knowledge graph. Due to small data scale in a single curriculum area and the fact that the curriculum itself has formed a complete knowledge system, the paper used top-down approach to construct the knowledge graph. Considering that curriculum domain knowledge graph usually has high requirements for the integrity of knowledge points, the comprehensiveness of knowledge system in ontology layer and the academic rigor of entity data, the paper adopted a manual and algorithmic method that takes into account both accuracy and efficiency to obtain the information of concept types, relationship types and attribute types.

Firstly, we obtained relevant corpus from classic textbooks and reference learning materials. Then, ontology layer was constructed by combining domain expert knowledge, and the conceptual entity, relationship and attribute of the curriculum domain were defined in detail. Then, all entities, relations, attributes and corresponding values were extracted from the curriculum corpus by hand and converted into format triples to obtain corresponding data layer. Finally, the triplet data would be visualized through Neo4j.

B. Data Set

Based on some classical textbooks, reference materials, exercises and other sources, this paper obtains the basic corpus data to construct the knowledge map of this course. The corpus sources are described in Table 1.

Table 1: Corpus Sources

Source	Selected corpus	Type
Chinese Mooc	Python related courses web text	Unstructured data
Classic course material	Tsinghua University Press, python Language Programming 3rd Edition	Semi-structured data
	Shanghai Jiao Tong University Press, Python Language Programming Tutorial	Semi-structured data
Syllabus	Python terms	Semi-structured data
Python newbie tutorial page	Knowledge and terms in tabular form	Structural data

(1) Website knowledge of the course in MOOCs of Chinese universities. Firstly, the text corpus of "Python Language Programming" and "Python Programming Basics" courses in Mooc of Chinese universities has been crawled by Scrapy framework and saved in text format through manual selection and summary. Among them, the emphasis is on the text of the question-and-answer area after class.

(2) Summarize and organize the electronic document corpus of the course materials of Python Language Programming published by Tsinghua University Press and Shanghai Jiao Tong University Press respectively and save them in text format.

(3) Refer to the "Python Language Programming Course Syllabus" published by Peking University, mainly to obtain semi-structured texts related to professional terms and teaching requirements.

(4) Obtain tabular structured data containing Python related terms from the "Python Novice Tutorial" website.

Finally, data from different sources are integrated to obtain 20,046 corpora of course text and 605 corpora of structured terms, which together serve as the corpus for constructing curriculum knowledge map in this paper.

C. Curriculum Ontology Modeling

At present, ontology modeling is commonly used in the academic and practical fields to construct the pattern layer of knowledge graph. Firstly, this paper investigates the syllabus of Python Programming Language in a large number of ordinary universities to determine the scope of Python syntax knowledge involved in this study; Then, based on the existing knowledge system and the glossary of terms in the field of Python programming language, the concept type, the relationship type between concepts and the concept attribute type are defined in the field.

1) Ontology concept type definition

Since the course of Python Language Programming has a complete knowledge system, this paper refers to some university syllabus of Python Language Programming and course materials of Python Language Programming to sort out and summarize related concepts and knowledge points and divide the types of fine-grained knowledge points and concepts. This paper designs a total of three levels of concept categories and gives examples of the first two levels of concept categories, as shown in Table 2.

Table 2: Ontology Concept Modeling

No.	Primary class	Secondary class(part)
1	environment	Anaconda, Pycharm and Spyder
2	grammar	Basic concept, sentence et al
3	package	Numpy, Pandas et al
4	objectOrient	Class, Object et al
5	frame	Django, Flask, Scrapy et al
6	debug	Unit testing, Quality guarantee et al
7	advanced	database programming, Network programming et al
8	others	Other content not yet included.

Among them, there are eight categories of first-level concepts. The category of "programming environment" contains knowledge related to Python programming environment, such as environment construction and Python program code editing. The "basic syntax" contains the knowledge of identifiers, expressions, and data types involved in Python programming. The "Python package" contains commonly used packages such as Pandas, Numpy, and so on.

The above eight first-level concept categories cover all the basic grammar of Python, which could be used to build a complete basic knowledge system of the course. At the same time, each level category can be extended to the lower level to obtain fine-grained concept (knowledge point) subcategories, and finally build a complete course ontology model.

2) Ontology relation type definition

The relationship between the course knowledge points reflects the logical structure between the knowledge points, which may help the learners better the knowledge of the course. The paper analyzes the relationship types of the knowledge points, and defines them as the following categories, which is shown in Table 3.

Table 3: Ontological Relation

No.	Relation	Definition	Example
1	Extend	B belongs to A subclass of A.	Operator - relational operator.
2	Component	Concept B is A necessary component of concept A.	Loop structure -range object.
3	Equivalent	Different names for the same concept.	1<2, 2>1.
4	Instance	Entity a is an element of concept A.	Built-in function -eval() function.
5	OtherRelation	Other undefined relationship types.	

3) Entity attribute type definition

By setting different attribute types for the concepts in the ontology, the nature of a certain concept or entity can be explained in more detail from different angles and aspects, so as to help learners understand and master the connotation of this knowledge point more comprehensively. Considering this reason, we use top-down approach to sort out attribute types in the domain ontology, which is shown in Table 4.

Table 4: Entity Attribute Type

No.	Attribute Type	Example
1	Definition	Definition of immutable sequence: An immutable sequence means that after a sequence is created, its elements cannot be changed.
2	GrammarRule	Identifier writing rules: Identifiers must start with a letter or underscore "_" followed by any sequence of letters, numbers, or underscores.
3	Usage	The upper() method converts lowercase letters in a string to uppercase.
4	Feature	The characteristics of the list are: (1) variable order (2) can store any type of data, support nesting, and the value can be repeated
5	Operation	File operations: (1) Open the file. (2) Read or write files. (3) Close the file.
6	Create	Create list variables with an assignment statement.
7	opening	Open the file with the WITH statement.
8	Call	Call a function in the form of a function name (argument list).
9	Access	Access list elements by list name [index].
10	Close	Use the shortcut Ctrl+C to terminate the current program.
11	Trasfer	Using the list() function, the individual characters in the converted string become list elements in turn.
12	Modify	The method for modifying the value of a list element is: list name [index] = new value.
13	New	The append() method appends the specified new element to the end of the specified list.
14	Delete	List element deletion method: del list name [index].
15	Sort	sort () method.
16	Content	Member operators include in and not in.
17	Consist	Loop variables, loop bodies, and loop termination conditions.
18	otherAttribue	Other attributes not annotated.

D. Information Extraction

In the knowledge graph based on graph database, corpus is organized as the form of (entity, relation, entity) or (entity, attribute, attribute value) triplet. This paper extracts knowledge triplet according with the established ontology model. Since there are a lot of entities in the course field of Python Language Programming as professional terms, this paper adopts semi-automatic knowledge extraction method to obtain entity, relationship and attribute data from the original course corpus.

1) Named entity recognition

Entity recognition automatically determines entity boundaries and category names by extracting key entity words (such as personal names, place names, terms, etc.) from data corpus, an example is shown in **Error! Reference source not found.**. Entity recognition task is mainly regarded as a sequence annotation task [20]. For a given input text sequence $X = [x_1, x_2, \dots x_n]$, sequence features are extracted through the sequence annotation model to predict the corresponding label sequence $Y = [y_1, y_2, \dots y_n]$, where y_i belongs to Y, Y is a limited set of labels developed manually.

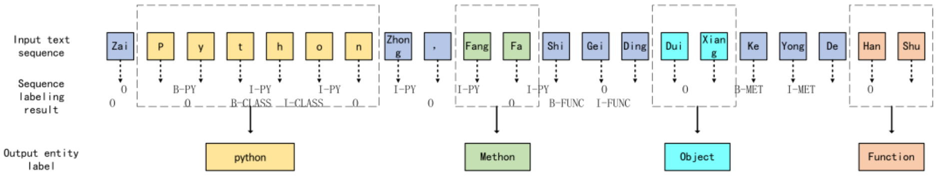

Figure 1: Named Entity Recognition Diagram.

In the paper, an NER algorithm based on LSTM [21] and CRF [22] is constructed. The bidirectional LSTM layer captures the input sequence features; ATT layer is a self-attention mechanism. By calculating the importance weight of each character vector to the text sequence, the model can improve the ability to extract key information. At last, we standardize and constrain the output label sequence through CRF layer to ensure rationality and correctness of the sequence labeling. The workflow is described in Figure 2.

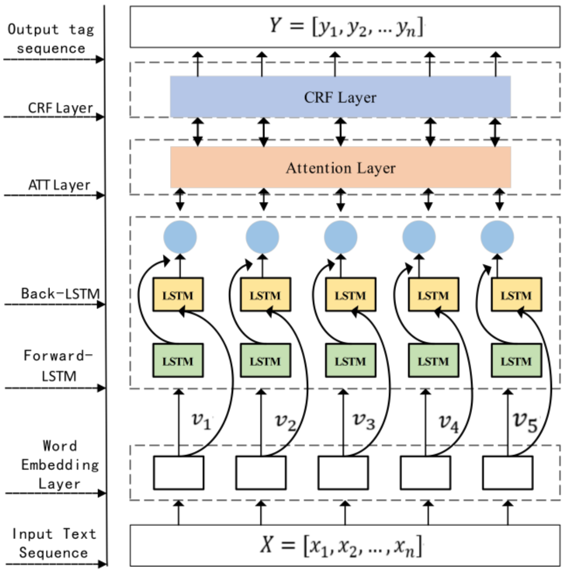

Figure 1: BiLSTM-ATT-CRF

Embedding layer converts user's question text into a Word vector through the Word embedding operation, which is used as the input information of the subsequent BiLSTM layer. For a given input text sequence $X = [x_1, x_2, \dots, x_n]$(where it is a single character in the text sequence), the word embedding vector $V = [v_1, v_2, \dots v_n]$ is obtained through the embedding layer.

BiLSTM layer is composed of forward and reverse LSTM layer respectively, which realizes the rejection and supplement of input information through three different "gate" structures, so that the model has long-distance memory ability. At the same time, the problems of "gradient explosion" and "gradient descent" which are easy to appear in the traditional RNN model are alleviated. Cell state structure is explained in Figure 2.

Calculation formulas of single-layer LSTM layer are follows, where f_t i_t and o_t are intermediate state of the forgetting gate,input gate and output gate respectively, where h_{t-1} and h_t are output of previous unit, v_t is input of current unit respectively, C_{t-1} and C_t represent states of previous unit and current unit respectively, W_f^v, W_i^v,

W_c^v, W_o^v, W_f^h, W_i^h, W_i^h and W_o^h are the weight matrices in different door structures respectively, b_h^f, b_h^i, b_h^c and b_h^o are the offset values in different door structures respectively.

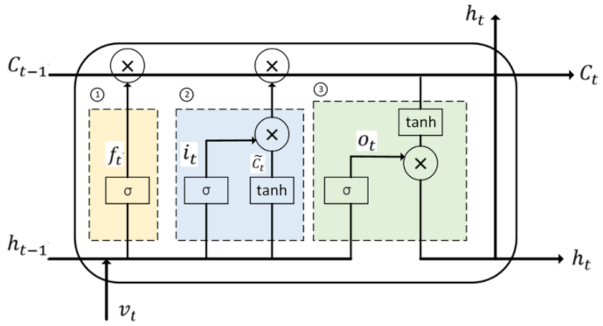

Figure 2: Cell State Structure of LSTM Model.

$$f_t = \sigma\left(W_f^v \cdot v_t + W_f^h \cdot h_{t-1} + b_h^f\right) \tag{1}$$

$$i_t = \sigma\left(W_i^v \cdot v_t + W_i^h \cdot h_{t-1} + b_h^i\right) \tag{2}$$

$$\widetilde{C_t} = \tanh(W_c^v \cdot v_t + W_c^h \cdot h_{t-1} + b_h^c\ ) \tag{3}$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \tag{4}$$

$$o_t = \sigma(W_o^v \cdot v_t + W_o^h \cdot h_{t-1} + b_h^o) \tag{5}$$

$$h_t = o_t * \tanh(C_t) \tag{6}$$

Finally, the forward and backward LSTM outputs are connected together as outputs of BiLSTM layer. Compared with LSTM, Bi-LSTM has the characteristics of high precision and low dependence on word vectors, and stronger processing ability of long sequence context information, which can capture more comprehensive and complete sequence features and improve the accuracy of sequence annotation tasks and named entity recognition tasks.

$$\overrightarrow{h_t} = \mathrm{LSTM}\left(\overrightarrow{h_{t-1}}, v_t\right) \tag{7}$$

$$\overleftarrow{h_t} = \mathrm{LSTM}\left(\overleftarrow{h_{t-1}}, v_t\right) \tag{8}$$

$$H = \overrightarrow{h_t} \oplus \overleftarrow{h_t} \tag{9}$$

ATT layer calculates the attention weight of the output of BiLSTM layer by introducing the self-attention mechanism, that is, when predicting a certain target word, it automatically obtains the semantic information of different positions in the original sentence and assigns a weight to the semantic information of each position. The sequence vector output by BiLSTM layer is $\mathrm{H} = [h_1, h_2, h_3, \dots h_t]$, and the weight matrix obtained by ATT layer is shown as equation 3.10, 3.11, 3.12, where tanh is nonlinear activation function, α is weight matrix of importance of the obtained character vector in relation to the whole text sequence, and w^T the transpose of the parameter vector learned by training.

$$M = tan\,h(H) \tag{10}$$

$$\alpha = softmax(w^T M) \tag{11}$$

$$r = H\alpha^T \tag{12}$$

CRF layer can automatically learn the sequence relationship and dependency relationship between sequence tags from the initial annotation corpus, filter the errors that do not meet the reality and do not meet the entity annotation rules in the output label combination in BiLSTM layer, and finally a reasonable label combination $Y = [y_1, y_2, \dots y_n]$ is got.

2) Relation extraction

In this paper, an expert manual extraction method is used to label the relationship between entities. The extraction of inter-entity relationship is mainly based on five relationship types defined in ontology relationship modeling: "inheritance", "component", "equivalence", "instance" and "other". Due to the constraints and restrictions of first-level concept categories in ontology modeling, entities under different concept categories are relatively independent. Therefore, this paper first marks the relationships of entities under the same concept category, that is, first extracts "intra-class relationships", and develops targeted extraction strategies for different relationship types.

For the "inheritance" relation, it corresponds to the "parent class" and "subclass" relation under the same concept category. Based on the top-level concept category, the ontology concept model can be searched from the top down in the subcategory to find out whether the subcategory concept or entity is included. For the "component" relation, a bottom-up extraction method can be adopted, that is, to determine whether a low-level entity is a component of its upper-level concept. For "equivalence" relations, the main way to determine is to check whether a concept or entity has a common synonym. For instance, relations, it mainly occurs in the concepts of "function", "method", "parameter", "variable", "Python library" and their sub-concept categories, which usually contain many instantiated entities. However, the relationship among the grammar in Python courses is complicated, and it is inevitable that there will be "inter-class correlation" between concepts or entities under different categories. In the last stage of relation extraction, this special situation will be summarized and sorted out, and finally all the triples marked (entities, relations, entities) will be summarized.

3) Attribute extraction

Entity attribute extraction consists of two main steps: First determine the type of attribute an entity has, and then populate it with the appropriate attribute value. In the essay, a semi-automatic extraction method is used which fill the entity attribute content. Firstly, candidate attribute types are set for each entity based on manual screening, and then some entities are queried to obtain corresponding attribute values based on structured Python course corpus data. For the attribute values that cannot be queried in structured corpus, the text similarity calculation algorithm based on TF-IDF is used to automatically extract the possible attribute content from the unstructured and semi-structured Python course corpus.

By combining manual screening and TF-IDF-based text similarity calculation method, this paper extracts a total of 949 attribute values from structured corpus, unstructured corpus and semi-structured corpus, and constructs the answer library of the intelligent question answering system in this study.

IV. KNOWLEDGE GRAPH VISUALIZATION

The triplet data of Python Language Programming course obtained in this paper is small and stored locally, so LOAD CSV method is used to import triplet data in batches to complete the construction of Python course knowledge graph. Some visual screenshots of knowledge graph are shown as Figure 4.

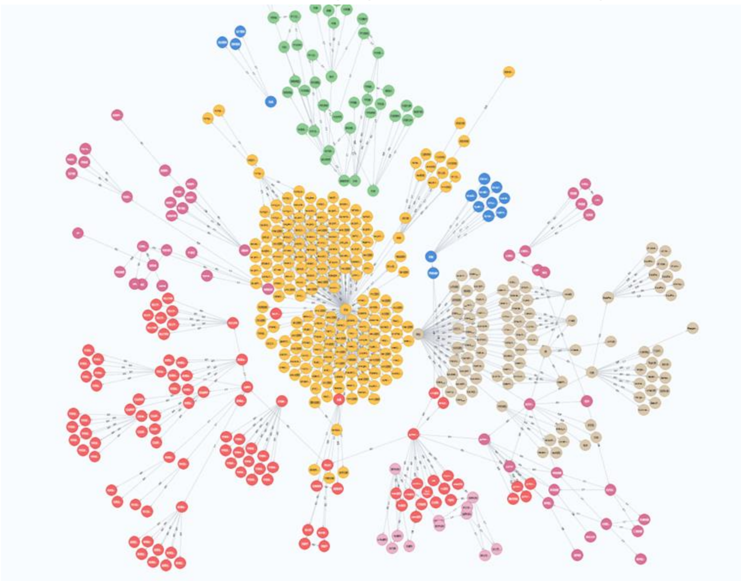

Figure 4: Python Curriculum Knowledge Graph Results Display

V. CONCLUSIONS

As a highly structured knowledge management and knowledge representation tool, knowledge graph can realize the integration, induction and visual display of knowledge in a certain field. With the continuous advancement of "education informatization" in our country, there is an urgent need for high-quality knowledge base and resource base in the field of curriculum education. In previous studying, we introduced knowledge graph into curriculum teaching. By taking Python Language Programming, where ontology model is first constructed to define the entity type, relationship type and attribute type in the field. A knowledge graph for Python course is constructed based on Neo4j graph database.

The shortcomings of this paper include that the degree of automation of corpus entity and relation extraction is not high, so the corpus triplet established based on semi-automatic method is not comprehensive enough. The next research work will focus on the joint entity and relation extraction method. Secondly, knowledge graph has not been fully studied here. The next step is to further test the accuracy of knowledge graph used in intelligent question and answering.

ACKNOWLEDGMENT

The work was supported by the key Research and Development Project of Anhui Province under Grant No. 2022O07020001.

REFERENCES

[1] V. Moscato, M. Postiglione, and G. Sperlí, "Few-shot Named Entity Recognition: Definition, Taxonomy and Research Directions," ACM Trans. Intell. Syst. Technol., vol. 14, no. 5, pp. 1–46, 2023, doi: 10.1145/3609483.

[2] T. Knez and S. Zitnik, "Event-Centric Temporal Knowledge Graph Construction: A Survey," MATHEMATICS, vol. 11, no. 23, DEC 2023, Art no. 4852, doi: 10.3390/math11234852.

[3] R. Quintero, E. Mendiola, G. Guzmán, M. Torres-Ruiz, and C. G. Sánchez-Mejorada, "Algorithm for the Accelerated Calculation of Conceptual Distances in Large Knowledge Graphs," MATHEMATICS, vol. 11, no. 23, DEC 2023, Art no. 4806, doi: 10.3390/math11234806.

[4] D. Daza, D. Alivanistos, P. Mitra, T. Pijnenburg, M. Cochez, and P. Groth, "BioBLP: a modular framework for learning on multimodal biomedical knowledge graphs," J. Biomed. Semant., vol. 14, no. 1, DEC 8 2023, Art no. 20, doi: 10.1186/s13326-023-00301-y.

[5] M. Boudin, G. Diallo, M. Drance, and F. Mougin, "The OREGANO knowledge graph for computational drug repurposing," SCIENTIFIC DATA, vol. 10, no. 1, DEC 6 2023, Art no. 871, doi: 10.1038/s41597-023-02757-0.

[6] W. F. Hendria, V. Velda, B. H. H. Putra, F. Adzaka, and C. Jeong, "Action knowledge for video captioning with graph neural networks," JOURNAL OF KING SAUD UNIVERSITY-COMPUTER AND INFORMATION SCIENCES, vol. 35, no. 4, pp. 50-62, APR 2023, doi: 10.1016/j.jksuci.2023.03.006.

[7] B. Albreiki, T. Habuza, N. Palakkal, and N. Zaki, "Clustering-based knowledge graphs and entity-relation representation improves the detection of at risk students," EDUCATION AND INFORMATION TECHNOLOGIES, 2023 AUG 9 2023, doi: 10.1007/s10639-023-11938-8.

[8] S. Das, M. Bagchi, and P. Hussey, "How To Teach Domain Ontology-based Knowledge Graph Construction? An Irish Experiment," KNOWLEDGE ORGANIZATION, vol. 50, no. 3, pp. 182-201, 2023, doi: 10.5771/0943-7444-2023-3-182.

[9] C. Ramonell, R. Chacón, and H. Posada, "Knowledge graph-based data integration system for digital twins of built assets," AUTOMATION IN CONSTRUCTION, vol. 156, DEC 2023, Art no. 105109, doi: 10.1016/j.autcon.2023.105109.

[10] J. Li, B. Wang, and Y. Chen, "Construction of knowledge base of middle school Chinese textbook text based on knowledge graph," Digital Teaching in Primary and Secondary, no. 06, pp. 83-87, 2022.

[11] Z. Yang, Y. Wang, J. Gan, H. Li, and N. Lei, "Design and Research of Intelligent Question-Answering(Q&A) System Based on High School Course Knowledge Graph," Mobile Networks and Applications, vol. 26, no. 5, pp. 1884-1890, 2021/10/01 2021, doi: 10.1007/s11036-020-01726-w.

[12] F. Alam, H. B. Giglou, and K. M. Malik, "Automated clinical knowledge graph generation framework for evidence based medicine," EXPERT SYSTEMS WITH APPLICATIONS, vol. 233, DEC 15 2023, Art no. 120964, doi: 10.1016/j.eswa.2023.120964.

[13] S. Gupta, S. Tiwari, F. Ortiz-Rodriguez, and R. Panchal, "KG4ASTRA: question answering over Indian Missiles Knowledge Graph," SOFT COMPUTING, vol. 25, no. 22, pp. 13841-13855, NOV 2021, doi: 10.1007/s00500-021-06233-y.

[14] J. Huang et al., "Trustworthy Knowledge Graph Completion Based on Multi-sourced Noisy Data," Proceedings of the ACM Web Conference 2022, pp. 956–965, 2022, doi: 10.48550/arXiv.2201.08580.

[15] Y. Zhou, Z. Liu, X. Ni, and T. Jin, "Construction of ontology model of Q&A knowledge graph by integrating multi-level data," Library And Information Service, vol. 66, no. 05, pp. 125-132, 2022, doi: 10.13266/j.issn.0252-3116.2022.05.013.

[16] Z. Ma, S. Wang, H. Zhang, and R. Li, "Joint Model of Intelligent Q&A Intent Recognition Based on Knowledge Graph," Computer Engineering and Applications, vol. 59, no. 6, pp. 171-178, 2023, doi: 10.3778/j.issn.1002-8331.2206-0419.

[17] S. Shin and K.-H. Lee, "Processing knowledge graph-based complex questions through question decomposition and recomposition," Information Sciences, vol. 523, no. 07, pp. 234-244, 2020/06/01/ 2020, doi: https://doi.org/10.1016/j.ins.2020.02.065.

[18] J. C. G. Jeynes, M. Corney, and T. James, "A large-scale evaluation of NLP-derived chemical-gene/protein relationships from the scientific literature: Implications for knowledge graph construction," PLoS One, vol. 18, no. 9, SEP 8 2023, Art no. e0291142, doi: 10.1371/journal.pone.0291142.

[19] S. Jaryani, I. Yitmen, H. Sadri, and S. Alizadehsalehi, "Exploring the Fusion of Knowledge Graphs into Cognitive Modular Production," BUILDINGS, vol. 13, no. 9, SEP 2023, Art no. 2306, doi: 10.3390/buildings13092306.

[20] Y. Zhang, S. Wang, B. He, P. Ye, and K. Li, "Named entity recognition method of elementary mathematical text based on BERT," Journal of Computer Applications, vol. 42, no. 2, pp. 433-439, 2022, doi: 10.11772/j.issn.1001-9081.2021020334.

[21] S. Mulla and N. F. Shaikh, "Effective Elytron Vespid-B rank BiLSTM classifier for Multi-Document Summarization," MULTIMEDIA TOOLS AND APPLICATIONS, 2023 NOV 30 2023, doi: 10.1007/s11042-023-17544-7.

[22] H. M. Sahin, T. Miftahushudur, B. Grieve, and H. J. Yin, "Segmentation of weeds and crops using multispectral imaging and CRF-enhanced U-Net," COMPUTERS AND ELECTRONICS IN AGRICULTURE, vol. 211, AUG 2023, Art no. 107956, doi: 10.1016/j.compag.2023.107956.