IEICE TRANSACTIONS

on Fundamentals of Electronics, Communications and Computer Sciences

DOI:10.1587/transfun.2024EAP1090

Publicized:2024/08/23

A PUBLICATION OF THE ENGINEERING SCIENCES SOCIETY

The Institute of Electronics, Information and Communication Engineers

Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER

Intelligent question answering system design with domain-specific knowledge graphs

Beining ZHANG[†a)], **Xile ZHANG**[††b)], **Qin WANG**[††c)], ***Nonmembers***, **Guan GUI**[††d)], ***and*** **Lin SHAN**[†††e)], ***Members***

SUMMARY With the fast development of the mobile internet, various data has been generated explosively in recent years. To solve the data redundancy problem caused by IoE, knowledge graph construction is considered one of the indispensable techniques for performing systematic and accurate representation of data, especially in some specific domains. In this paper, we propose a construction method of panoramic domain-specific knowledge graphs in various domains, which treat sensitive (secret) data, e.g., medical and industrial domains. This method mainly uses the web crawler to obtain data from relevant web pages by category and saves the obtained data as a structured JSON file in the form of dictionaries. This paper takes the military field as an example to construct the domain-specific knowledge graph based on our proposed method. Specifically, the specific domain knowledge graph is stored in Neo4j and MongoDB to provide an intuitive and applied representation of knowledge, respectively. Based on the knowledge graph stored in MongoDB, we develop an intelligent question-answering system in a specific domain, which can better satisfy the information retrieval and knowledge learning of related personnel. Moreover, the template-based question-answering system is designed to effectively solve the problem of semantic repetition of questions. Finally, the constructed knowledge graph and question-answering system are evaluated and analyzed.

key words: *Knowledge graph construction, question-answering system, specific domain, semantic repetition.*

1. Introduction

The era of big data brings both convenience and challenges, such as data clutter, processing difficulties, and distinguishing true from false information, hindering data usage efficiency. While research has addressed these issues in various fields, some domains lag due to their secrecy and sensitivity. Information in these areas often faces overload, authenticity verification challenges, varied formats, and ununified processing methods, complicating data retrieval and learning. Thus, developing a system to organize and uniformly represent massive, multi-source heterogeneous data is urgently needed. This system should establish associations and fusion of information to aid processing, retrieval, and application. Google introduced the concept of the KG (knowledge graph) in 2012, defining it as a database of entities and relationships for information retrieval [1–3]. Recently, KGs have been widely adopted in fields like natural language processing, intelligent retrieval systems, intelligent question-answering systems, and intelligent recommendation systems [4, 5]. A KG systematically describes facts, including entities, relationships, and properties. Entities can be tangible or abstract, relationships link entities, and properties describe entity characteristics [3]. As a typical graph structure, KGs offer advantages: (1) Systematic data representation; (2) Enhanced data retrieval and utilization; (3) Improved knowledge cognition and reasoning.

Due to the specific domain's unique nature, related data development is limited. The data is multi-sourced, complex, and poorly visible, hindering processing and use by developers. Thus, mining and application of specific domain data remain rare. Knowledge graphs, a popular information processing technology, can use graph structures, mathematical theories, and related methods for unified knowledge representation and visualization, mapping potential links and development laws [6]. For example, applying knowledge graphs in the military field enables the mining and accumulation of domain-specific knowledge and establishes associations and fusion of multi-source heterogeneous information at the semantic level. This not only utilizes the value of military data but also aids in the search and learning for military enthusiasts and related personnel.

Constructing a knowledge graph is merely the systematic storage of data; its true value lies in its application. Recently, knowledge graph applications have been widely focused on question-answering systems, recommendation systems, and retrieval systems [7, 8]. The emergence of large-scale knowledge bases like Freebase, Wikipedia, and Yago2 has significantly advanced system design based on knowledge graphs [9]. Specific domain data often have fewer entity types but more entity attributes, making KG-based intelligent question-answering systems a major application direction. The main task of such a system is to find the correct answer from the knowledge base based on the question [10, 11]. This task can be divided into entity recognition and relationship matching. These sub-tasks analyze the problem, extract the surface form of entities and relationships, and map them to the knowledge graph to link the entities and relationships, providing the answer [12]. KG-based intelligent question-answering

[†]The author is with the Portland Institute, Nanjing University of Posts and Telecommunications, Nanjing 210023, China.

[††]The authors are with the College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China.

[†††]The author is with the Center for Information Infrastructure, Shinshu University, Nagano 390-8621, Japan

a) E-mail: p21000104@njupt.edu.cn
b) E-mail: 1222014322@njupt.edu.cn
c) E-mail: wangqin@njupt.edu.cn
d) E-mail: guiguan@njupt.edu.cn
e) E-mail: shanlin@shinshu-u.ac.jp

systems effectively retrieve and utilize domain information, facilitate information authenticity verification, and enhance information mining.

Although knowledge graph construction technology is relatively mature, it remains challenging to combine intuitiveness and applicability for data in fields like military and medical. Additionally, KG-based intelligent question-answering systems rely on the parsing effect of questions, making it difficult to detect semantically identical questions with different phrasing. To address these issues, this paper constructs a panoramic domain-specific knowledge graph stored in MongoDB and Neo4j databases, utilizing text and graph structures to reflect intuitiveness and applicability. A template-based question-answering system construction method is then proposed to realize intelligent question-answering for military information, using text data stored in MongoDB. Finally, the knowledge graph and question-answering system are evaluated and tested, identifying existing problems and suggesting future improvements. The contributions of this paper are as follows:

- Based on the characteristics of domain data, we store the knowledge in Neo4j and MongoDB with graph and document structures to reflect its intuitiveness and applicability, respectively, and analyze the adaptation characteristics of data and knowledge base.
- For the applicability of knowledge graph, we further construct a KG-based intelligent question-answering system and propose a template matching-based approach to avoid the problem of difficult parsing caused by the same semantics but different types of questions.
- For the constructed knowledge graph and question-answering system, we analyze the current shortcomings and propose future improvement solutions in combination with the current mainstream solutions.

2. Related works

Recently, knowledge graph has been widely used in various fields and the application scenarios are getting richer and richer. Scenarios such as public security lead research, question-answering system, drug discovery [13], recommendation system and stock prediction all benefit from the development of knowledge graph.

As an important application direction of knowledge graph, specific domain question-answering systems can well retrieve and utilize databases to facilitate knowledge cognition and knowledge reasoning [14]. Intelligent question-answering system is almost at the same time as the birth of artificial intelligence. Alan Turing proposed a method to verify the degree of machine intelligence in 1950, which mainly determines the intelligence of a machine by judging whether it has the ability to correctly answer questions. Soon after, MIT Weizenbaum designed the name in 1966 [15]. In recent years, researchers have conducted a lot of research in the construction of intelligent question answering systems based on knowledge graphs. X. Dai *et al.* [7] proposed a question-answering system based on the military knowledge graph. Z. Li *et al.* [8] investigated and designed a KG-based question-answering system for COVID-19 cases imported from abroad. C. Zhou *et al.* [16] built a question-answering system based on Chinese medical knowledge graph. Generally, there are three main methods for constructing intelligent question-answering systems based on knowledge graphs: template-based methods, information retrieval-based methods and semantic parsing-based methods [17]. We briefly introduce each of these three methods as follows.

2.1 Template-based methods

Template-based question-answering system construction methods mainly use text parsing technology to map questions to predefined question-answering templates and retrieve answers to questions by matching. S. Ou *et al.* [18] proposed an ontology-based method for automatic question-answering pattern generation, and applies text implication to the construction of template-based question-answering system. H. Bast *et al.* [19] presented a model, namely Aqqu, which built three templates to respond to complex questions. Overall, the template-based question-answering method is a more traditional approach [15]. The designer needs to set up the template in advance based on the data content, which is the main difficulty of this method. The advantage of this method is that it has high accuracy, and the method is simple and intuitive. The disadvantage is that the effectiveness of the quiz depends on the template setting, and the template is difficult to generate and not transferable.

2.2 Information retrieval-based methods

Information retrieval-based question-answering system construction methods focus on retrieving all candidate answers and do not parse the questions into a standard semantic interpretation, after which the retrieved candidate answers are ranked to obtain the best answer [17, 20]. A. Bordes *et al.* [21] presented a question-answering system which utilizes subgraph embedding to predict the credibility of candidate answers. H. Sun *et al.* [22] proposed a distributed representation method for generating questions and candidate answers using neural networks. At present, most information retrieval methods use neural networks to retrieve and rank candidate answers. Although this method has low requirements in question parsing, the accuracy of the answers depends on the retrieval and ranking of the answers.

2.3 Semantic parsing-based methods

Semantic parsing-based question-answering system construction methods focus on semantically analyzing problems into a structured logical form that can be understood by the knowledge base, so that knowledge reasoning and retrieval can be performed through the knowledge base. In a word,

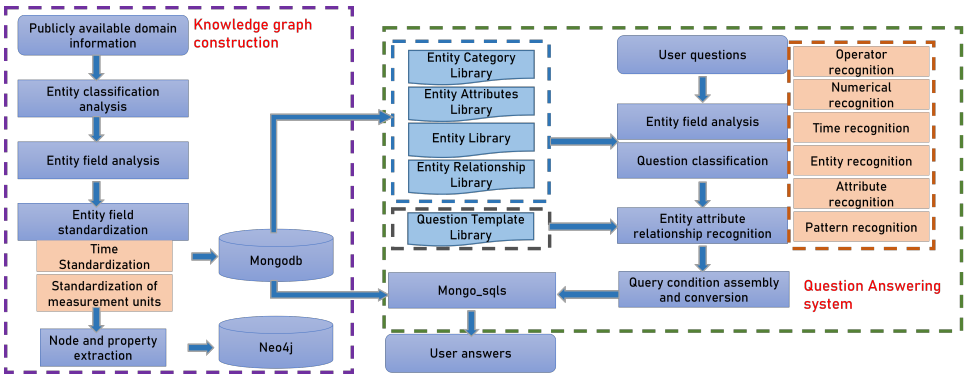

Fig. 1 The system model of knowledge graph construction and intelligent question-answering system design.

the key of semantic parsing-based methods is transforming the problem into a structured form that is easy to handle [17]. Z. Jiang *et al.* [23] proposed a semantic analysis-based question-answering system of medical field, which defines a large number of feature words and question keywords for problem analysis. R. Athreya *et al.* [24] presented a method of learning natural language problems automatically by using recurrent neural networks and classifying them into corresponding templates. The method has good transferability and accuracy, but there are few well-formed logical forms and it is difficult to train a semantic parser.

3. System Model

The system construction in this paper mainly comprises two parts: knowledge graph construction and question-answering system design. Data from different fields have distinct characteristics. Specific domain data are often in text form, with fewer entity types but more entity attributes, making it challenging to balance applicability and intuitiveness. Additionally, the military field's specific characteristics necessitate high data accuracy. Based on these characteristics, the system model of this paper is shown in Fig. 1.

The construction of domain-specific knowledge graphs is based on the life cycle of a knowledge graph and the characteristics of military data. First, data is acquired from public sources according to relevant categories, and this data is analyzed to identify entities and attributes. Next, the data is standardized and unified for knowledge storage. Considering the difficulty of balancing intuitiveness and application, MongoDB, a text database, and Neo4j, a graph database, are used for knowledge storage.

MongoDB is a distributed file storage database that uses BSON (similar to JSON) as the data storage format, with data structures consisting of key-value pairs. Neo4j, on the other hand, is a powerful graph database that stores data as nodes and edges, with nodes capable of holding attributes [25]. Since Neo4j stores structured data (mathematically called graphs) on the network (also mathematically called graphs) rather than in tables [26], it effectively visualizes knowledge, facilitating intuitive understanding and perception.

Given the specific domain's requirement for high data accuracy, this paper employs a template-based construction method known for its high accuracy. The core idea of the proposed method is to build templates of relevant entities, attributes, relationships, and question types based on the database's data content. These templates are then matched with the query conditions summarized after question parsing to retrieve and output the answers. This approach also addresses the issue of incorrect answers resulting from different types of questions with the same semantics, a common problem in question-answering systems.

The process of the knowledge graph construction and the question-answering system will be described in detail in Sections IV and Sections V, respectively.

4. Knowledge graph Construction

This paper uses the military field as an example to construct a specific domain knowledge graph by adopting a bottom-up method [6]. As shown in Fig. 2, the first step involves obtaining data from open-linked military information sources and processing the data to extract the corresponding entities, attributes, and relationships. These extracted elements are added to the data layer of the knowledge graph. Next, the military information contents are summarized to form a pattern layer, completing the construction of the knowledge graph. Finally, the specific domain knowledge graph is stored and visualized.

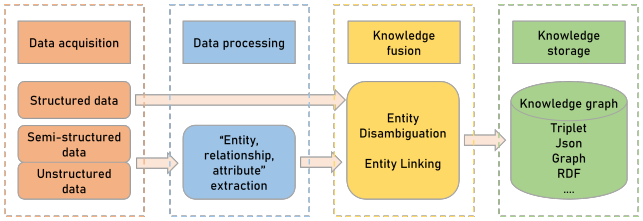

Fig. 2 Flow chart of knowledge graph construction.

4.1 Data Collection and Processing

The data used to build the knowledge graph includes three main types: structured data, semi-structured data, and unstructured data. Structured data primarily comes from collected relational databases or specialized information collection websites, and can be directly used to build knowledge graphs after simple processing. In contrast, semi-structured and unstructured data are more complex to process due to their more confusing content, and often contain knowledge redundancy and missing information.

In order to facilitate the construction of the knowledge graph more easily, the data for this paper is obtained from the Global Military Web Equipments Database, which has highly structured data. The equipment in this database originates from 88 countries, with a total of 5,800 equipment entities containing 8 categories (e.g., aircraft, ships, etc.), 148 specific types (e.g., fighters, destroyers, etc.) and 184 entity attributes.

Due to the highly structured nature and ease of obtaining and processing military data from the Global Military Web, this data is collected using a web crawler method, extracting content separately according to equipment categories. The data used in this paper are publicly available and saved as JSON files after web crawling.

Given the highly structured characteristics of the acquired military data, and the absence of issues such as ambiguity from entities with the same name or multiple referents corresponding to the same entity object, there is no need for knowledge fusion techniques like entity disambiguation. Thus, the collected data only need to be standardized for knowledge storage.

According to the analysis of the data content, there are issues with missing information and inconsistent units in the acquired data. The standardization of data primarily focuses on filling in missing data and standardizing units of measurement. Since the military information on the Global Military Web is relatively complete, any missing information is likely due to confidentiality factors. Therefore, missing information is filled in as "No data" to complete the records.

For the standardization of measurement units, this includes time standardization, unification of distance units, unification of speed units, and unification of weight units. The units of distance and speed are standardized to meters, and the units of weight are standardized to kilograms. Time normalization involves both detailed and fuzzy time. Detailed time is accurate to the month, day, and year, while fuzzy time remains unchanged.

Since the data are saved as JSON files, the "json.loads" function in Python is used to read the files and save them in dictionary format. The key-value pair feature of the dictionary format is convenient for both knowledge storage and retrieval. The "replace" function is then used to replace the corresponding data in a standardized manner.

4.2 Storage and Visualization of Knowledge Graph

Since the acquired data has fewer entity types but many entity attributes and is composed in the form of dictionaries, it is well-suited for knowledge storage in document form. MongoDB, a text database, uses collections as its key storage component. These collections consist of JSON documents, BSON documents, or sub-documents [27]. MongoDB's flexibility, ease of use, and rich query language make it particularly well-suited for text data storage.

Therefore, the standardized data is imported into the MongoDB database using the pymongo library in Python. Additionally, the "id" attribute of the standardized data is removed to facilitate the utilization of the knowledge graph, while the unified units of measure are added to the attribute list through key-value pairs.

The storage of knowledge should consider both applicability for easy retrieval and intuitiveness for cognition and reasoning. Neo4j, a powerful graph database, stores data as graphs and represents objects with nodes, edges, and attributes, making it suitable for storing knowledge graphs and facilitating cognition and reasoning. Using a bottom-up construction method, after building the data layer, knowledge is abstracted into concepts to form the schema layer. The Neo4j database then stores knowledge using the node-relationship-attribute schema to complete the knowledge graph's schema layer. Thus, the visualization steps for a military knowledge graph include extracting nodes and attributes, extracting relationships, and importing them into Neo4j.

Based on the acquired data, there are relatively few types of military information entities, but numerous entity attributes. Different types of equipment have distinct attributes. Therefore, in visualizing the knowledge graph, equipment entities, country entities, types, and specific categories are extracted as nodes, with only pictures and introductions retained as attributes for equipment entities. Since the data layer is saved in JSON format, nodes and attributes can be extracted by reading the data keys to obtain their values and save them as a CSV file, forming an attribute table for importing into Neo4j.

Since the extracted nodes include equipment entity, country, category and specific type, three relationships are constructed based on these nodes: G1=("Equipment","_is", "Type"), G2=("Equipment","belong_to", "Category"), G3=("Equipment","come_from", "Country"). The three relationships are each stored as a triplet table in CSV format for import into Neo4j.

Importing CSV files into Neo4j usually uses the officially provided "neo4j-admin import" method and Cypher statements. The system imports attribute and triplet tables in CSV format into Neo4j through the "Merge" and "Match" statement of Cypher, where the "Merge" statement is used to construct nodes and attributes, and the "Match" statement is used to join relationships. Fig. 3 shows the nodes, attributes and relationships that were successfully

imported into Neo4j.

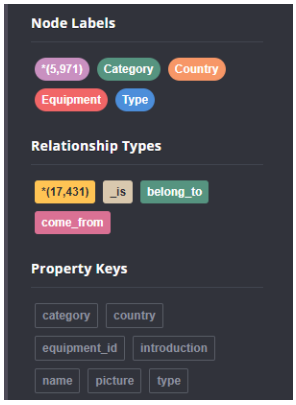

Fig. 3 The nodes, attributes and relationships of Neo4j.

5. Intelligent Question-Answering System Design

Based on the characteristics of the military field, the question-answering system should focus more on the accuracy of the data. Besides, natural language questions are minimal within a specific range. Hence, the question-answering system for that specific domain is more suitable for a template-based construction method. Since the dataset used in this paper is a Chinese dataset, there are multiple words corresponding to one meaning. Hence we first define the keyword library and the corresponding labels according to the data content and define the question template library and the answer template library according to the frequently asked questions. After that, we extract keywords and labels by parsing the questions, obtain the search criteria and aggregate all the query conditions, match them with the question template library. Finally, the answers are output according to the pre-defined answer templates.

5.1 Question Classification

The key of the template-based question-answering system construction method is to build relevant templates based on the data content to determine the type of question, so that the answer can be retrieved and output from the database. We first set up the keyword library and its corresponding labels according to the data content, which is stored in the form of a dictionary, as shown in the Table 1. Based on the frequent question types and data content, we set up seven keyword libraries: equipment library, country library, category library, specific type library, attribute library, comparison library, and most value library. Also, our data involves time and units, and we use regular expressions to detect units and time and add them to the corresponding lexicons "Unit_lib" and "Time_lib". The constructed keyword library is used for the construction of question templates, and the method can effectively solve the problem of detection difficulties caused by different types of questions with the same semantics.

Table 1 The keywords templates.

Keyword labels	Keyword examples
Equipment_lib	{'PWS-19 bomber': 'PWS-19 bomber', 'Scharnhorst': 'Scharnhorst'...}
Country_lib	{'China': 'China', 'United States' : 'United States', 'France': 'France'...}
Category_lib	{'Aircraft': 'Aircraft', 'Ship': 'Ship'...}
Type_lib	{'Aircraft Carrier': 'Aircraft Carrier', 'Bomber': 'Bomber'...}
Attribute_lib	{'Service time': 'Service time', 'Length': 'Length', 'Full length': 'Length', 'How long': 'Length'...}
Compare_lib	{'Higher than': 'More', 'Less than': 'Low'...}
Most_lib	{'Largest': 'Max', 'Slowest': 'Min'...}

After determining the keyword library, the labels of the keyword library are combined with the frequent question types to form the question templates. For example, when the question is "What is the length of Swordfish?", after parsing, the question generates the template "[Attribute_lib, Equipment_lib]". The answer is then retrieved based on this template. The question templates constructed in this paper are shown in the Table 2, which mainly include Q&A on attribute values, Q&A on attribute interval value screening and Q&A on attribute best values. After that, Q&A on attribute values is further subdivided into single-entity single-attribute Q&A, single-entity multi-attribute Q&A, multi-entity single-attribute Q&A and multi-entity multi-attribute Q&A, and the Q&A on attribute interval value screening is subdivided into single-attribute single-interval Q&A and single-attribute multi-interval Q&A. Different question templates are constructed by combining labels of keyword libraries, and different combinations may correspond to the same question type. For example, "What is the length of PWS-19 bomber?" and "PWS-19 bomber' s length" both belong to single-entity single-attribute questions, but their templates are different for ['Attribute_lib', 'Equipment_lib'] and ['Equipment_lib', 'Attribute_lib']. Therefore, we considered various combinations when generating the question templates in order to facilitate accurate classification of the questions.

After determining the question template, we set up the corresponding answer templates for retrieving and outputting the answers according to different question types. As shown in the Table 3, we use the most straightforward method to establish the answer templates, for example, when the template type after question parsing is "[Attribute_lib, Equipment_lib]", we set the answer template as "['Attribute_lib' + 'Equipment_lib' + answer]". This method not only facilitates the establishment of answer templates, but also facilitates knowledge retrieval and output, and is more intuitive.

At this point, all the templates used for question classification in the system have been constructed, including the keyword libraries, question templates and answer templates, with which the subsequent question and sentence analysis and knowledge retrieval will be easier.

Table 2 The question templates

	Type of question	Examples of question	Examples of template
Q&A on attribute values	Single-entity single-attribute Q&A	What is the length of PWS-19 bomber?	['Attribute_lib', 'Equipment_lib']
	Single-entity multi-attribute Q&A	What is the length and height of PWS-19 bomber?	['Attribute_lib', 'Attribute_lib' , 'Equipment_lib']
	Multi-entity single-attribute Q&A	What is the length of PWS-19 bomber and Swordfish respectively?	['Attribute_lib', 'Equipment_lib', 'Equipment_lib']
	Multi-entity multi-attribute Q&A	What is the length and height of the PWS-19 bomber and the length of the Scharnhorst respectively?	['Attribute_lib', 'Attribute_lib', 'Equipment_lib', 'Attribute_lib', 'Equipment_lib']
Q&A on attribute interval value screening	Single-attribute single-interval Q&A	What are the fighter jets with a maximum flight speed greater than 600 km/h?	['Type_lib ', 'Attribute_lib', 'Compare_lib ', 'Unit_lib']
	Single-attribute multi-interval Q&A	What are the fighters with maximum flight speed greater than 500 km/h and less than 600km/h?	['Type_lib ', 'Attribute_lib', 'Compare_lib ', 'Unit_lib', 'Compare_lib', 'Unit_lib']
Q&A on attribute best values	Single entity attribute best value Q&A	Which is the longest ship in length?	['Most_lib ', 'Type_lib ', 'Attribute_lib']

Table 3 The answer templates

Type of question	Template of answer
Single-entity single-attributes Q&A	['Equipment_lib'+ 'Attribute_lib' + answer]
Single-entity multi-attribute Q&A	['Equipment_lib'+ 'Attribute_lib' + answer, 'Equipment_lib'+ 'Attribute_lib' + answer, ...]
Multi-entity single-attribute Q&A	['Equipment_lib'+ 'Attribute_lib' + answer] ['Equipment_lib'+ 'Attribute_lib' + answer] ...
Multi-entity Multi-attribute Q&A	['Equipment_lib'+ 'Attribute_lib' + answer, 'Equipment_lib'+ 'Attribute_lib' + answer, ...] ['Equipment_lib'+ 'Attribute_lib' + answer, 'Equipment_lib'+ 'Attribute_lib' + answer, ...] ...
Single-attribute single-interval Q&A	['Equipment_lib'+ 'Attribute_lib' + answer] ['Equipment_lib'+ 'Attribute_lib' + answer] ...
Single-attribute multi-interval Q&A	['Equipment_lib'+ 'Attribute_lib' + answer] ['Equipment_lib'+ 'Attribute_lib' + answer] ...
Single-entity single-attribute best value Q&A	['Equipment_lib' + answer] ['Equipment_lib'+ 'Attribute_lib' + answer]

5.2 Question Analysis

The most difficult part of a Chinese question-answering system is the processing of Chinese phrases. Compared to English sentences, which can be segmented by spaces between words, segmentation of Chinese sentences is particularly difficult. In this paper, the successful segmentation of question sentences is as important as template building in the template-based construction method. The question can be segmented to extract keywords and corresponding labels, which is the basis of template matching and answer retrieval.

Jieba is a popular open source Chinese word segmentation tool in recent years, which has a good performance in both long sentence segmentation and text analysis. According to the internal dictionary, it scans all words to generate a directed acyclic graph (DAG) consisting of all possible Chinese word construction cases in a sentence, finds the maximum probability path using dynamic programming, and finds the maximum segmentation combination based on word frequency. For words not in the thesaurus, the HMM model based on Chinese character word formation ability is adopted, and the Viterbi algorithm is used to find the most likely hidden state sequence. At the same time, jieba supports user-defined dictionaries and dynamic addition and deletion of dictionaries, which further reduces the difficulty of our development.

In order to segment the question more accurately, we first use the characteristics of jieba dynamic add and delete dictionaries, using the "jieba.add_word" function to add our pre-defined keyword library and its corresponding labels to the jieba lexicon, and then through the "jieba.posseg.cut" function to split the question and get its corresponding labels, as shown in the Fig. 4, where "uj" represents the meaning of "of", "m" represents Numeral and "v" represents Verb. The keywords are extracted according to the labels after question segmentation, and the labels are retained as subsequent processing flags to generate the corresponding templates.

(Chinese) 问题: "沙恩霍斯特号的舰长是多少？ "
(English) Question: " What is the length of Scharnhorst? "

(Chinese) 解析: [(" 沙恩霍斯特号 ", " Equipment_lib "), (" 的 ", " uj "), (" 舰长 ", " Attribute_lib "), (" 是 ", " v "), (" 多少 ", " m ")]
(English) Parsing: [(" Scharnhorst ", " Equipment_lib "), (" of ", " uj "), (" the length ", " Attribute_lib "), (" is ", " v "), (" what ", " m ")]

Fig. 4 Segmentation results of a question.

5.3 Query Processing

The knowledge graph constructed in this paper is mainly stored in MongoDB and Neo4j database with text structure

and graph structure, and the more convenient knowledge retrieval method is based on text structure. The BSON knowledge storage method of MongoDB stores data in the form of key-value pairs, which is convenient for knowledge retrieval, utilization and update. At the same time, because the data has few entity types but many entity attributes, the storage of graph structure only extracts some key attributes and does not make full use of the knowledge. Therefore, this paper constructs the KG-based question-answering system stored in MongoDB.

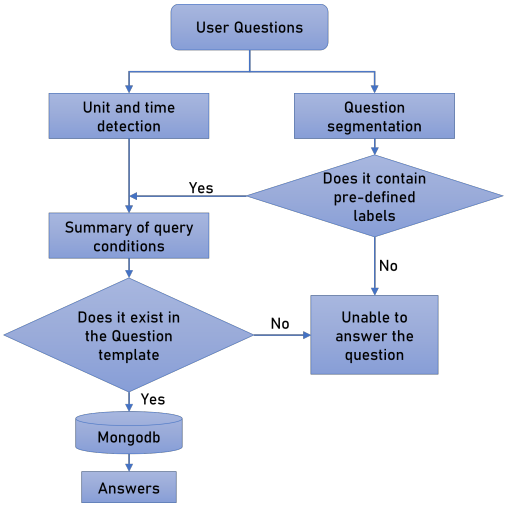

Fig. 5 Implementation flow chart of intelligent question-answering system.

The implementation process of the question-answering system is shown in Fig. 5. When the user asks a question, the question is first analyzed and the time and measurement units are detected by regular expressions, while the question is segmented and the lexical labels of each item after segmentation are extracted to determine whether there are labels in the pre-defined keyword libraries. After that, the keywords and the corresponding tags are extracted to form a complete template of query conditions. Finally, the answers are retrieved and output by matching with the question template libraries according to the answer templates pre-defined for each template. The greatest advantage of this question-answering system is that it is easy to construct and the accuracy of data is high, which fully reflects the systematic characterization of data by knowledge graph. At the same time, the method of extracting keywords and their corresponding labels for query condition construction can well deal with the problem of ambiguity caused by different types of questions and improve the stability of the question-answering system.

6. Results and Analysis

6.1 Results and Analysis of Specific Domain Knowledge Graph

This paper focuses on the military domain, uses crawler technology to obtain data from the Global Military Web as data sources, takes various equipment as core entities, and constructs a panoramic-specific domain knowledge graph to facilitate the knowledge perception of military enthusiasts. At the same time, due to the characteristics of more attributes of data entities, which are not conducive to graph structure storage for visualization, the key attributes are extracted and stored in Neo4j for visual display in this paper, and the results are shown in Fig. 6.

Fig. 6 Specific domain knowledge graph.

In order to balance the intuitiveness and applicability of knowledge graphs, this paper not only extracts key attributes to store in Neo4j database with graph structure for visualization but also uses text structure to store all attribute values to MongoDB for application of knowledge. After comprehensive analysis, for knowledge bases with more entity types and more types of entity relationships, using graph structure for knowledge storage is a better choice, while for data with fewer entity types, fewer entity relationships and more entity attributes, a document-based database is more suitable.

6.2 Results and Analysis of Question-Answering System

In fact, information retrieval in people's daily lives is also a form of question-answering, and question-answering

systems are even more convenient for human-computer interaction. On the one hand, question-answering system can avoid the problem of miscellaneous information in the retrieval process, and output the corresponding answers more directly. On the other hand, because of the specificity of question-answering system in specific domains and high-quality knowledge graph data, the accuracy of answers is relatively high. At present, the performance testing of question-answering systems is mainly based on the accuracy rate. We conducted six experiments on question parsing, keyword detection and template matching to test this intelligent question-answering system, and the experimental results are shown in Fig. 7. Fig. 8 shows the English-translated version of the experimental results.

According to the experimental results, the performance of the template-based question-answering system depends on the degree of template construction. The first two experiments are mainly based on keyword matching, and the experimental results demonstrate that when words that are not available in the keyword library appear, although they can be correctly categorized by the Hidden Markov Model, they still cannot yield the correct answer. The next two experiments test the performance of question parsing, and the experimental results show that although the established keyword library has been added to the jieba library in advance, when there are special characters in the question sentence, the sentence cannot be successfully segmented, which leads to the question parsing failure. The last two experiments are the verification of template matching, which show that the correct answer can be retrieved as long as the template generated after question parsing can be matched with the established template library.

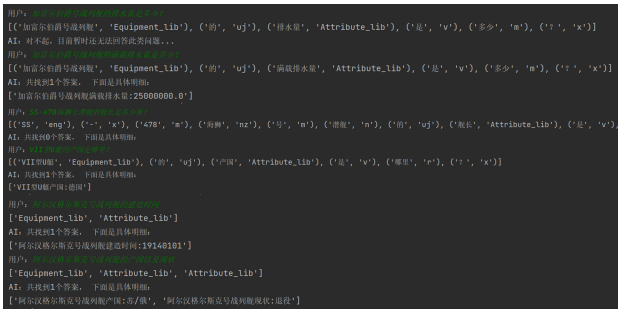

Fig. 7 Experimental results of the question-answering system (in Chinese).

The experimental results show the common disadvantages of the template-based question-answering system construction method. The performance of the question-answering system depends entirely on the established templates, but the templates are generally built manually, which is time-consuming and labor-intensive. How to solve the existing problems is also the direction of our future work, which will be described in the next section.

User: What was the displacement of Conte Di Cavour Class? AI: Unable to answer the question
User: What was the full load displacement of Conte Di Cavour Class? AI: ['The full load displacement of Conte Di Cavour Class': 25000000.0]
User: What is the length of the submarine SS-478 Sea Lion? Parsing: [('what', 'm'), ('is', 'v'), ('the length', 'Attribute_lib'), ('of', 'uj'), ('ss', 'eng'), ('-', 'x'), ('478', 'm'), ('sea Lion', 'nz')] AI: Unable to answer the question
User: What is the country of production of UVII Submarine? Parsing: [('what', 'm'), ('is', 'v'), ('the country of production ', 'Attribute_lib'), ('of', 'uj'), ('UVII Submarine', 'Equipment_lib')] AI: [The country of production of UVII Submarine: Germany]
User: When was the Arkhangelsk battleship constructed? Template: ['Equipment_lib', 'Attribute_lib'] AI: [The construction time of the Arkhangelsk battleship: 19140101]
User: What are the country of production and current status of Arkhangelsk battleship? Template: ['Attribute_lib', 'Attribute_lib', 'Equipment_lib'] AI: [The country of production of the Arkhangelsk battleship: 19140101, The current status of Arkhangelsk battleship: Retirement]

Fig. 8 Experimental results of the question-answering system (in English).

6.3 Future Works

Through the experiments and analysis of the domain-specific knowledge graph and intelligent question-answering system, the existing problems are summarized:

- The data used to build the knowledge graph in this paper is structured data, which does not require too much processing. So the construction process cannot fully reflect the life cycle of knowledge graph.
- For the domain-specific knowledge graph based question-answering system, limited by the high accuracy of military data, this paper adopts a template based construction method. Although the accuracy rate is guaranteed, the performance of the question-answering system is completely dependent on the established templates, and the templates are built manually, which is time consuming and labor intensive.
- For the parsing of interrogative sentences, this paper directly uses the jieba library for interrogative segmentation, which cannot successfully extract keywords containing special characters.

For the above problems, we propose some potential solutions, which are also our future work direction:

- For the structured data problem, we will replace other specific fields to build the knowledge map, and focus on selecting multi-modal material sources.
- To address the shortcomings of the template-based question-answering system construction method, we try other construction methods to conduct comparative experiments, such as using the information retrieval-based method and the semantic analysis-based method, and compare the performance by analyzing the accuracy and complexity of the three solutions.
- For the question segmentation, we plan to try a deep learning approach by training the established keyword library with data enhancement as a dataset to generate a keyword extraction classifier to parse and segment the questions.

7. Conclusion

This paper proposed a construction method of a panoramic

domain-specific knowledge graph, which further facilitates the retrieval of domain information for related personnel. At the same time, to balance the intuitiveness and applicability of the knowledge graph, the knowledge graph is stored in MongoDB and Neo4j in text structure and graph structure, respectively. Then, we developed a specific domain KG-based question-answering system by using a template-based approach to further exploit the systematic characterization of the knowledge graph for data. Finally, experiments and analyses were conducted on the constructed knowledge graph and question-answering system, and the experimental results demonstrate the effectiveness of the proposed method.

References

[1] H. Han, J. Wang, X. Wang, and S. Chen, "Construction and Evolution of Fault Diagnosis Knowledge Graph in Industrial Process," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, no. 3522212, pp. 1–12, 2022.

[2] Z. Chen, Y. Wang, B. Zhao, J. Cheng, X. Zhao, and Z. Duan, "Knowledge Graph Completion: A Review," *IEEE Access*, vol. 8, pp. 192435–192456, 2020.

[3] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A Survey on Knowledge Graphs: Representation, Acquisition, and Applications," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 2, pp. 494–514, Feb. 2022.

[4] G. Liu, and L. Li, "Knowledge Fragment Cleaning in a Genealogy Knowledge Graph," in *2020 IEEE International Conference on Knowledge Graph (ICKG)*, 2020, pp. 521–528.

[5] M. Kejriwal, and P. Szekely, "Knowledge Graphs for Social Good: An Entity-Centric Search Engine for the Human Trafficking Domain," *IEEE Transactions on Big Data*, vol. 8, no. 3, pp. 592–606, Jun. 2022.

[6] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, "A Review of Relational Machine Learning for Knowledge Graphs," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 11–33, Jan. 2016.

[7] X. Dai, J. Ge, H. Zhong, D. Chen, and J. Peng, "QAM: Question Answering System Based on Knowledge Graph in the Military," in *2020 IEEE 19th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC)*, 2020, pp. 100–104.

[8] Z. Li, Q. Xu, W. Zhang, and T. Zhang, "An Approach and Implementation for Knowledge Graph Construction and Q&A System," in *2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*, 2021, pp. 425–429.

[9] M. R. A. H. Rony, D. Chaudhuri, R. Usbeck, and J. Lehmann, "Tree-KGQA: An Unsupervised Approach for Question Answering Over Knowledge Graphs," *IEEE Access*, vol. 10, pp. 50467–50478, 2022.

[10] N. Jayaram, A. Khan, C. Li, X. Yan, and R. Elmasri, "Querying Knowledge Graphs by Example Entity Tuples," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 10, pp. 2797-2811, Oct. 2015.

[11] H. Jin, Y. Luo, C. Gao, X. Tang, and P. Yuan, "ComQA: Question Answering Over Knowledge Base via Semantic Matching," *IEEE Access*, vol. 7, pp. 75235–75246, 2019.

[12] S. Vakulenko, J. D. F. Garcia, A. Polleres, M. de Rijke, and M. Cochez, "Message Passing for Complex Question Answering over Knowledge Graphs," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 1431–1440.

[13] T. Jiang, Q. Zeng, T. Zhao, B. Qin, T. Liu, N. V. Chawla, and M. Jiang, "Biomedical Knowledge Graphs Construction From Conditional Statements," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 3, pp. 823–835, May-June 2021.

[14] L. S. Nair, and S. M. K, "Knowledge Graph based Question Answering System for Remote School Education," in *2022 International Conference on Connected Systems & Intelligence (CSI)*, 2022, pp. 1–5.

[15] Q. Yang, "Research on The Intelligent Question Answering Based on Knowledge Graph," in *2021 International Conference on Big Data and Intelligent Decision Making (BDIDM)*, 2021, pp. 226–229.

[16] C. Zhou, R. Guan, C. Zhao, G. Chai, L. Wang, and X. Han, "A Chinese Medical Question Answering System Based on Knowledge Graph," in *2021 IEEE 15th International Conference on Big Data Science and Engineering (BigDataSE)*, 2021, pp. 28–33.

[17] S. Aghaei, E. Raad, and A. Fensel, "Question Answering Over Knowledge Graphs: A Case Study in Tourism," *IEEE Access*, vol. 10, pp. 69788–69801, 2022.

[18] S. Ou, C. Orasan, D. Mekhaldi, and L. Hasler, "Automatic Question Pattern Generation for Ontology-based Question Answering," in *Flairs Conference*, 2008, pp. 183–188.

[19] H. Bast, and E. Haussmann, "More Accurate Question Answering on Freebase," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 2015, pp. 1431–1440.

[20] S. Hu, L. Zou, J. X. Yu, H. Wang, and D. Zhao, "Answering Natural Language Questions by Subgraph Matching over Knowledge Graphs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 5, pp. 824–837, 2018.

[21] A. Bordes, S. Chopra, and J. Weston, "Question Answering with Subgraph Embeddings," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 615-–620.

[22] H. Sun, B. Dhingra, M. Zaheer, K. Mazaitis, R. Salakhutdinov, and W. W. Cohen, "Open Domain Question Answering Using Early Fusion of Knowledge Bases and Text," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 4231–4242.

[23] Z. Jiang, C. Chi, and Y. Zhan, "Research on Medical Question Answering System Based on Knowledge Graph," *IEEE Access*, vol. 9, pp. 21094–21101, 2021.

[24] R. G. Athreya, S. K. Bansal, A. C. N. Ngomo, and R. Usbeck, "Template-based Question Answering using Recursive Neural Networks," in *2021 IEEE 15th International Conference on Semantic Computing (ICSC)*, 2021, pp. 195–198.

[25] H. Lu, Z. Hong, and M. Shi, "Analysis of Film Data Based on Neo4j," in *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, 2017, pp. 675–677.

[26] Y. Zou, and Y. Liu, "The Implementation Knowledge Graph of Air Crash Data Based on Neo4j," in *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 2020, pp. 1699–1702.

[27] B. Jose, and S. Abraham, "Exploring The Merits of Nosql: A Study Based on Mongodb," in *2017 International Conference on Networks & Advances in Computational Technologies (NetACT)*, 2017, pp. 266–271.

Beining Zhang is currently pursuing B.S. degree in Communications Engineering at Nanjing University of Posts and Telecommunications. Her research interests include specific emitter identification, deep learning, and the corresponding application in wireless communications.

Xile Zhang is currently pursuing B.S. degree in Communications Engineering at Nanjing University of Posts and Telecommunications. Her research interests include specific emitter identification, deep learning, and the corresponding application in wireless communications.

Qin Wang received the Ph.D. degree in communication and information system from Nanjing University of Posts and Telecommunications (NJUPT), Nanjing, China, in 2016. She is currently an Associate Professor with NJUPT. Her research interests include multimedia communications, smart data pricing, resource allocation in 5G/6G, and Internet of Things.

Guan Gui received a Ph.D. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2012. Since 2015, he has been a professor at Nanjing University of Posts and Telecommunications, Nanjing, China. Dr. Gui has published more than 200 IEEE Journal/Conference papers and won several best paper awards, e.g., ICC 2017, ICC 2014 and VTC 2014-Spring. He received the IEEE Communications Society Heinrich Hertz Award in 2021, His research interests include specific emitter identification, deep learning, and the corresponding application in wireless communications.

Lin Shan received the M.E. and Ph.D. degrees from the Graduate School of Informatics, Kyoto University, in 2008 and 2012, respectively. He is a professor at the Center for Information Infrastructure, Shishu University, Nagano, Japan. His research interests include network coding, multiuser-MIMO scheduling, cooperative relaying, and resource allocation in ad hoc and cellular networks. He received the Kyoto University President Prize in 2010. He received the IEICE RCS Active Research Award and the IEEE VTS Japan Young Researcher's Encouragement Award in 2011, and the IEEE Kansai Section Student Paper Award and the IEICE Best Paper Award in 2012 and 2013, respectively.