Question Answering System Based on Knowledge Graph in Air Defense Field

Liuyang Yu¹, a, Zhigang Guo²*, Gang Chen³, Xingxin Zhang⁴, Yongwang Tang⁵, Han Wei⁶
University of Information Engineering, Zhengzhou 450000, China
*Email: keep_er@163.com
abcg20140801@stu.xjtu.edu.cn

Abstract. With the development of artificial intelligence technology, question answering systems based on knowledge graphs (QAS-KGs) have become a prevalent information retrieval approach. However, current QAS-KGs which usually adopt general knowledge graphs have complex algorithms and high requirements for hardware. Moreover, in some specific fields, such as air defense field, the QAS-KGs usually cannot respond accurately and they can only handle questions composed of one language. In this paper, we introduce an efficient QAS-KG approach which can handle questions composed of Chinese, English, or both languages. First, we construct the knowledge graph in air defense field according to the domain characteristics and user needs. Then we design and implement a question answering system based on the knowledge graph. Finally, we test the system on the self-built data set. The results show that the system can understand the user's intention quickly and make accurate responses without high requirements for hardware.

1. Introduction
With the development of technology, the amount of data on the Internet has grown rapidly. How to quickly and accurately locate the knowledge required by users in massive data has become one of the focuses of people's attention. As an advanced form of information retrieval, question answering system is favored by human beings through its friendly and accurate interactive way[1]. Recently, question answering systems (QASs) have become the basic form of new generation information retrieval technology.

QASs can be generally divided into three categories. One is retrieval-based QASs. Its main technology is keyword-based matching and information extraction, such as IBM Waston. The second is community-based QASs. The main data comes from netizens, such as Baidu Know. The third is the QAS-KGs. As the support for intelligent applications, the concept of knowledge graph (KG) was first proposed by Google in 2012[2]. KG is a graph-based knowledge representation and organization method, which uses a set of "subject-predicate-object" triples to represent the various entities and their relationships[3]. A lot of mature products have been formed currently, but for some special fields, they have bad transplantable characters due to the different ontology models of KG. So, in different fields, users need to design QASs based on their own requirements.

This paper mainly studies the question answering system based on knowledge graph in air defense field (QAS-KG-AD). First, we construct the domain knowledge graph ontology model according to domain characteristics and user needs. Second, we use crawler technology to obtain relevant data on
the Internet and complete the construction of the KG. Then we design and implement the QAS-KG-AD. Finally, we test the system on self-built data set. The result shows that the system can understand users’ intention and response accurately after querying and reasoning. The system has a good development prospect in air defense field.

2. The Construction of Domain Knowledge Graph

A KG is generally composed of two parts: the pattern layer and the data layer. The pattern layer is the logical foundation and conceptual model of a KG. Ontology is usually used as the pattern layer of a KG. The data layer is the instantiation of the ontology. In terms of coverage, KGS can be divided into general KGS and domain KGS. A general KG usually describes a wide range of knowledge, focusing on the integration of more entities. A domain KG is usually used to describe the knowledge in a specific field, focusing on specific applications. In this paper, the KG is the air-defense domain knowledge graph.

2.1. Pattern Layer

According to the application direction, the ontology of KG in this paper mainly includes Flights, Weapons, Electronic Systems, Communication Systems, Countries, Units, Missions, Persons, Places. Different properties and relationships are constructed for different ontology types. Because there are many concepts in the ontology model, we take the Flights as an example to describe the basic framework of the ontology model and the corresponding relationship with the data layer, as shown in Figure 1.

2.2. Data Layer

In order to make the QAS respond correctly, we need a large amount of information. For this purpose, we extract and process two kinds of websites, getting appropriate structure data. The two kinds of websites are: Baike (https://baike.baidu.com) and IAIRForce (http://www.iairforce.com). The
processed data is stored in Neo4j, a graph database. We operate on the data through Cypher, a structured query language.

3. QAS-KG Workflow
In this paper, the QAS-KG-AD has its own characteristics: The focuses of the users' attention are relatively concentrated and the purposes of searching are also relatively clear, which determine the input forms of the system are limited and clear. Therefore, what we focus is no longer complex syntactic structure analysis or semantic analysis, but how to accurately understand a user's intention and map the question to the triples of KG.

The workflow of this system is as follows:
1. User input. The user uses natural language to ask a question about air defense security.
2. Question analysis. The system analyzes the question by using technologies, including word segmentation, part-of-speech tagging, named entity recognition, entity disambiguation and relationship extraction to extract the topic information contained in the question. Then the extracted information is mapped to triples.
3. Question classification. The system uses multi-label naive Bayes classification algorithm[4] to classify questions. If success, go to step 4. Otherwise, go to step 5.
4. Question conversion. After obtaining the question entities and intention of the question through steps 2 and 3, the system will generate at least one Cypher query statement through a manually constructed template library. Then go to step 6.
5. Question summary. The system will summarize all the classification-failure questions, and gives the answers manually in a way similar to the community-based question answering system. All the questions and answers collected by the system will be used to improve the system performance in the future.
6. Answer generation. The system uses Cypher language to query the KG and then gives the answer to the question.

4. Methodology
4.1. Question Analysis
In order to make readers understand this paper more clearly, we explain some concepts as follows:
1. User dictionary: a collection of all entity names, property names, and entity "alias" property value (if there is "alias" property) in the KG.
2. Synonym library: a collection of synonyms, abbreviations and common expressions of all elements in the user dictionary.
3. Named entity library: a collection of all entity names, property names, relationship names, and entity "alias" property values (if there is "alias" property) in the KG.

When the system receives a question, it first pre-processes the question by using word segmentation and part-of-speech tagging technology. We use the THULAC[5](developed by Tsinghua University) toolkit to achieve the above process. The entity names in this paper are usually composed of Chinese and English. If we directly use the word segmentation tools in the general field for word segmentation, it will result in large errors, for example, "F-22" is labeled as "F[x]-[w]22[m]". By creating a user dictionary, "F-22" can be labeled as a whole, "F-22[uw]".

The next task is to extract named entities and relationship from questions. There are many mature named entity recognition methods[6], such as rule/dictionary-based methods, machine learning methods, deep learning methods, hybrid methods and so on. Among them, the rule/dictionary-based method is one of the most effective methods in the early recognition of named entities. When the data set is small and the field is highly professional, the advantages of this method are more obvious. We use this method in this paper. First, we build a named entity library. Then we combine the characteristics of domain knowledge and expert knowledge to manually construct recognition rules and assign weights to each rule. Finally, the extraction is carried out by matching the entities with the rules. By adding
relationship names to the named entity library, we can treat relationships as named entities. So, we adopt a similar method to extract the relationships.

The entities and relationships involved in a question may be abbreviations, aliases, etc., which leads that they cannot be directly mapped into the KG. In this case, we need to perform entity disambiguation. Because the entity names involved in this domain are usually composed of two languages, we use the synonym library and the Levenshtein distance algorithm to disambiguate. The Levenshtein distance algorithm is mainly used for similarity matching of English strings. For Chinese or a mixture of two languages, we use the synonym library.

The Levenshtein distance algorithm is generally used to measure the similarity between two strings, assuming that the two strings are O (original string) and T (target string). The Levenshtein distance is defined as "The number of delete, insertion and replacement operations required to transform the string O into the string T." The formula is as follows:

\[
\text{lev}_{O,T}(s,t) = \begin{cases} 
\max(s,t) & \text{if } \min(s,t) = 0 \\
\text{lev}_{O,T}(s-1,t)+1 & \\
\text{lev}_{O,T}(s,t-1)+1 & \text{otherwise} \\
\text{lev}_{O,T}(s-1,t-1)+x & \end{cases}
\]

The \(\text{lev}_{O,T}(s,t)\) is the edit distance between the first \(s\) characters of \(O\) and the first \(t\) characters of \(T\). The similarity between \(O\) and \(T\) is

\[
\text{sim}_{O,T} = 1 - \left(\text{lev}_{O,T}(O \mid T) / \max(|O|,|T|)\right)
\]

Through the above operations, we successfully extract the valid information from the questions and map them to the KG. Taking the question "Which weapon systems are equipped with F-22" as an example, the mapped triple is "Flights-Load-Weapons".

4.2. Question Classify

This paper adopts the multi-label naive Bayes classification algorithm to classify the question by calculating the probability that a question belongs to a certain category. This paper divides the question into eight categories: Fact type, Enumerate type, Judge type, Compare type, Associate type, Count type, Value type and Other type. Questions that failed to be classified are identified as Other type. The system cannot obtain answers of Other type questions by matching question template. But it will summarize all the classification-failure questions and gives the answers manually in a way similar to the community-based question answering system. All the questions and answers collected by the system will be used to improve the system performance in the future.

The classification steps are as follows:

Step 1: We first constructed 2,980 natural language questions by manually creating and collecting Zhihu data (Zhihu is a community question and answer website). We select 1,980 questions training data (the others are test data), determining its feature properties (such as special interrogative words, etc.) and classify each question. These questions evenly cover the eight categories of questions. The training data and feature properties will be formed by step 1.

Step 2: We calculate the frequency of each question category in the training data. Then calculate the conditional probability of question categories divided by each type of feature properties. Finally we calculate the posterior probability of each category by the multi-label naive Bayes classification algorithm. The category with the highest probability and not less than the threshold \(T\) is the question’s classification result. If the highest probability is less than \(T\), the question will be divided into Other type. The classifier model will be formed by step 2.

Step 3: We use the classifier model trained in step 2 to classify the question and get the result.

4.3. Question Conversion and Answer Generation

After processing the question by the above methods, the system has determined the user's intention and the entities contained in the question. Then the system generates Cypher query statements that fit
the user's intention by matching the question template library. Finally, the system obtains the answer by querying the KG and returns the result to the user.

5. Results and Discussion
In this part, we perform a series of operations to test the system. As mentioned above, we manually constructed 2,980 questions and selected 1,000 of these questions as test data. For these 1,000 questions, we first answer each question manually to complete the test data. Then we test the system on the test data and the F1 value of the system is 94.2%. At the same time, in order to facilitate the reader's understanding of this paper, we give examples of each type of question in Table 1 to illustrate the working process of the system.

Table 1. Examples of different types of questions and process

| Type   | Example                                                                 | Graph |
|--------|-------------------------------------------------------------------------|-------|
| Fact   | What is the maximum speed of A-10?                                      |       |
| Cypher | match (n:Plane{name:'A-10'}) return n.max_speed                          |       |
| Answer | The maximum speed of A-10 is 706km / h.                                  |       |
| Enumerate | What weapons are the F-22 equipped with?                               |       |
| Cypher | match (n:Plane{name:'F-22'})-[[:load]->(p)] return p                   |       |
| Answer | Weapons loaded on the F-22 include AIM-120, AIM-9, GBU-32, M61A2, GBU-39, wind deviation correction ammunition dispenser. |       |
| Judge  | Is the 94th Fighter Squadron of the US Air Force affiliated with Langley Air Force Base? |       |
| Cypher | match (n:Unit{name:'USAF94'})-[[:Belongto]->(p)] return p               |       |
| Answer | Yes                                                                     |       |
| Compare | Which of A-10 and F-22 costs more?                                      |       |
| Cypher | match(n1:Plane{name:'A-10'}) return n1.price                           |       |
|        | match(n2:Plane{name:'F-22'}) return n2.price                           |       |
| Answer | F-22                                                                    |       |
| Associate | What is the relationship between the 94th Fighter Squadron and the 27th Squadron? |       |
| Cypher | match (n1:Unit{name:'USAF94'}) return n1                               |       |
|        | match (n2:Unit{name:'USAF27'}) return n2                               |       |
| Answer | They are all affiliated with Langley Air Force Base.                    |       |
| Count  | How many weapons are equipped with F-22?                               |       |
| Cypher | match (n:Plane{name:'F-22'})-[[:Com_system]->(p)] return p             |       |
| Answer | 6                                                                       |       |
| Value  | Which type of US flight has the largest combat radius?                 |       |
| Cypher | match (n:Plane) where n.country="America" return n.Combat_radius       |       |
| Answer | F-22                                                                    |       |

6. Conclusion
QASs offer a fast and accurate way to obtain information. In this paper, we first construct the KG in air defense field to make it possible to develop intelligent applications in related fields. Then we design and implement the Question Answering System Based on Knowledge Graph in Air Defense Field. The system can quickly and accurately understand natural language questions whether they are Chinese, English or both, and give exact answers. Furthermore, the system has low requirements of hardware. At last, we test the system on the self-built test data, the F1 value of the system is 94.2%, that means the system is easy to be transplanted and has a good development prospect.
Acknowledgments
This research was supported by the National Social Science Fund of China (19CXW027).

References
[1] Feng, G, Du, Z, Wu, X. A Chinese Question Answering System in Medical Domain. Journal of Shanghai Jiaotong University (Science). 2018;23(5): 678–683.
[2] Zhao Z, Han S, So I. Architecture of knowledge graph construction techniques. International Journal of Pure and Applied Mathematics. 2018; 118(19):1869-1883.
[3] Yu T, Li J, Yu Q, et al. Knowledge graph for TCM health preservation: Design, construction, and applications[J]. Artificial Intelligence in Medicine, 2017, 77(3):48-52.
[4] Zhang, M, José M, Peña, Robles, V. Feature selection for multi-label naive Bayes classification[J]. Information Sciences, 2009, 179(19).
[5] Happyprince(CSDN), 2018. NPL---THULAC: An efficient Chinese lexical analysis toolkit. https://blog.csdn.net/ld326/article/details/81328165.
[6] Goyal A, Gupta V, Kumar M. Recent Named Entity Recognition and Classification techniques: A systematic review[J]. Computer science review, 2018, 29(8):21-43.