Structured Knowledge Base Q&A System Based on TorchServe Deployment

Xia Ye¹, Ruiheng Liu*¹, Zengying Yue¹

¹Department of Computer Science, Xi’an Research Institute of Hi-Tech, Xi’an, Shan Xi

*Corresponding author’s e-mail: liuruih@foxmail.com

Abstract—Structured tabular data are widely used in various information systems, especially with the development of big data technology, making it more difficult to query on these complex data. SQL facilitates the query on structured tabular, however, the mastery of SQL has a certain threshold for most non-expert users. Therefore, in order to facilitate ordinary users to quickly obtain the required information in complex structured data, we design and implement a Q&A system for structured knowledge. First, we make a detailed distinction between Q&A scenarios for structured data and design different approaches, respectively. Then, we introduce deep learning models in system algorithm layer to enhance the generalization ability. Finally, the TorchServe framework is used to optimize system deployment and improve system performance using batch inference. The experimental results show that the prototype system has certain generalization ability and also has some advantages in performance compared with traditional methods.

1. INTRODUCTION

As a research hotspot in the field of artificial intelligence and natural language processing, Q&A (Question and Answer) systems are widely used in search engines, information retrieval, and intelligent Q&A bots by parsing natural language questions and providing accurate and concise answers. In recent years, with the development of deep learning techniques and the proposal of large-scale corpus, the research on Q&A systems has been expanded from the traditional limited professional domain to open domain, and most of them use retrieval or generation methods to obtain the corresponding answers from the knowledge base, depending on the different ways of knowledge base construction. Although the research on knowledge-based Q&A systems has been fruitful, there are still three limitations in the following aspects. (1) Most of the existing Q&A systems focus on unstructured knowledge represented by conventional text, while there is a lack of research on common structured data such as tables and databases as knowledge bases. One of the key reasons for this is that the relationships between entities are not explicitly specified in the table structure data, thus making the design of the Q&A system difficult. (2) Existing systems mostly ignore the actual Q&A scenarios across domains and multiple languages. Some work develops rules by expert knowledge of a limited domain and uses this to guide the Q&A process, leaving the system model with limited generalization capabilities and therefore unable to adapt to tasks in other domains. In addition to this, the influence of mainstream database management systems such as Oracle and MySQL, which often use English to represent field information, means that there may be language differences between user questions and the knowledge base. (3) In the process of deploying services, most work with circular or multi-threaded architecture of the server, when the client is sending a large number of single discrete or highly concurrent requests,
such methods are prone to linear growth in user waiting time or server down. The contributions of this paper are three main points. (1) A structured knowledge base Q&A prototype system is designed and implemented for common table and database scenarios, capable of supporting tables in common formats in business systems as well as database files as query objects. (2) By using pre-trained models to introduce generic knowledge and existing open datasets as model data drivers, the Q&A system supports open domain and multilingual Q&A. (3) Based on TorchServe framework, a system deployment scheme is designed to support batch processing of user requests, which can reduce system response time with limited computing resources to improve system performance.

2. RELATED WORKS

The knowledge base provides the knowledge source for answering natural language questions and is an important part of the Q&A system, which mainly consists of data with structures such as table structure, graph structure, free text and FAQ (Frequently Asked Questions). Although free text is the most readily available, when used as a knowledge base, there is very little knowledge and relationships that can be directly exploited, which requires a deeper understanding of the knowledge base by the system and is not conducive to Q&A tasks. The creation of the FAQ library greatly simplifies the Q&A process, as the system simply matches user questions in knowledge base and returns the most relevant answers, but the library requires additional costs to create and cannot effectively respond to questions outside the knowledge base. However, table-based data is widely used in various information systems, and its adoption as the data source of knowledge base has the lowest cost. For different types of knowledge bases, many works have proposed corresponding Q&A approaches, which can be broadly classified into two categories: information retrieval-based and semantic parsing approaches.

The literature [1-4] extracts entities and their relational features from questions and later matches the relevant answers in knowledge base, which mostly consists of questions and answer pairs. The literature [5] divides Q&A process into three parts: question processing, information retrieval and answer extraction for Chinese scenarios. Firstly, the key information in question is extracted; then the answer fragments are retrieved and extracted from the knowledge base; finally, the sentence components and relations are analyzed by combining the grammar tree and semantic similarity. The literature [6] extends the online corpus while improving the representation of word vectors, and combines retrieval matching and generative dialogue models to generate results in order to improve the interaction and scene adaptation capabilities of the Q&A system. The literature [7-11] semantically parses the questions so as to generate logically structured representations such as SQL statements and execute them on the knowledge base to get the answers. The literature [12] designs and implements a limited domain Q&A system for report data, generates corresponding SQL statements based on user questions and single domain tables, and implements intelligent report presentation.

We use tabular data and database files, which are widely used in information systems, as data sources to reduce the construction cost of the knowledge base and to facilitate the promotion and use of system. In addition to supporting Chinese Q&A tasks in single table scenarios, we also consider multilingual complex database scenarios in conjunction with the application reality. At the same time, the system supports open domain Q&A since the training and inference phases of the system model are performed on different domains.

3. SYSTEM DESIGN AND IMPLEMENTATION TECHNOLOGY

The overall design of system adopts the MVC-based (Model View Controller) framework model to build system from the bottom up according to three levels, such as information resources, model algorithms and application interfaces, while encapsulating the functions in each level, so as to reduce the coupling of system and improve the flexibility and expandability of different modules. The overall architecture of the system is shown in Figure 1.
3.1. Overall System Framework

3.1.1. Information Resources Layer
This layer is responsible for the integration of different modal data and pre-processing work, to build corresponding knowledge bases for different Q&A scenarios. Structured knowledge serves as an important basis for parsing user queries and directly affects the execution results of the final generated SQL statements. Therefore, it is necessary to standardize and unify the input structured knowledge, and different pre-processing methods are designed to build the corresponding structured knowledge base for Q&A scenarios with tables or databases.

3.1.1.1. Single Table Knowledge Acquisition
Since the table data lacks a fixed format, a table knowledge extraction module has been designed. Based on the source of the table data, it is divided into two categories: general-purpose files and database files. Generic files refer to common form files such as those generated by Excel, which the user needs to populate according to a given 2D form template and following the corresponding form logic. For each table, its template mainly contains table data, scheme data and content. Among them, table data contains the ID and name of the table, which serves as a unique index, so that users can specify table when raising queries; scheme data is the field information, including field names, field types, etc.; content is the specific value of each field. The database files are in the common format of various databases, and each table in them is considered as a knowledge base.

3.1.1.2. Database Knowledge Acquisition
Database tables have more complex links between them, and in Chinese scenarios, the names of tables and fields are often expressed in English. Unlike general files, database have a more uniform file format, such as SQLite files, SQL files, etc., so they can be used directly as data sources. Considering that database files are larger and more complex than tabular data, a database knowledge extraction module is designed in order to realize flexible invocation of data in the model algorithm layer. Input database file $D$ and output its feature class $C_D$, which are processed as follows.
• Create a class $C_D$ for database and add basic information such as database name and storage path to it.

• Extract all table names $T$ in database, unify them to lowercase format, and create a class $C_T$ for the table, add basic information such as table names, number of tables, and connection relationships between tables to them, finally add $C_T$ as a subclass of $C_D$.

• Extract all field names $F$ in database, unify them into the format of “$T.F$”, and create a class $C_F$ for the field, add basic information to it such as field type, whether it is a primary key or a foreign key, finally add $C_F$ as a subclass of $C_T$.

• Export database knowledge feature class $C_D$.

3.1.2. Model Algorithm Layer
We use regular expressions to design a user query rewriting module to correct the query and reduce the impact of formatting errors on subsequent process. Query rewriting operations mainly include: full and half angle symbol unification, number and unit conversion, letter case rewriting, date format unification, meaningless character deletion, user query truncation, etc. Then, the models for different knowledge bases are designed separately, and their structures are shown in Figure 2.

3.1.3. Application Interface Layer
This layer mainly returns the SQL statements generated by model algorithm layer based on user queries and the final results obtained from their execution on knowledge base to users through system interface, and provides an interface for them to manage knowledge base. The front-end is developed with JavaScript as the development language, React.js as the front-end development framework, and Ant Design, a React UI component library, is used for user interface design.

3.2. System Key Function Implementation
According to system functional requirements and the overall design framework, the key functions are designed and implemented in detail. Q&A task is the core function of the system, and three common scenarios are designed according to the type of knowledge base. The main functions are as follows.

• **Single-Table Q&A** is mainly for scenarios that a single table is used as the knowledge base.

• **Database Q&A** is mainly for scenarios that contain multiple tables with linked relationships.

• **Database Single-Table Q&A** is based on the Database Q&A and keeping on Single-table Q&A to further refine the scope.
• **Knowledge Base Management** establishes and maintains the corresponding knowledge base after pre-processing the structured files uploaded by users according to two types of Q&A scenarios: single table and database, it also supports exporting error correction samples generated by users for later model updates.

• **Answer Acquisition** generates SQL statements which reflecting the user's query intent based on knowledge base, and then executes the SQL statements to obtain the final answer.

• **Error Correction** supports users to manually correct the generated SQL statements and export them as new samples for the algorithm model to be updated and trained iteratively.

• **User Management** provides an interface for system administrators to change general user information.

The following is a detailed description of some of the key functions of the system.

3.2.1. Single-Table Q&A
This function answers questions raised by users based on knowledge base constructed from a single table. In addition to supporting simple entity attribute queries and complex entity relationship queries, it also supports operations such as calculating the number, average, maximum and minimum values among entity attributes, and the maximum number of qualifying conditions involved in each query is three.

In this process, users need to select or create a new knowledge base first, they can also preview existed knowledge bases, as shown in interface in Figure 3. After selecting the knowledge base, users input questions in specified location, as shown in the interface in Figure 4. The system will return corresponding SQL statements, and users can obtain final results by executing them. If there are deviations in results, they can also correct SQL statements and save corresponding samples for the update of system model algorithm.

Figure 3. Single-table Q&A interface
3.2.2. **Database Q&A**

The query object of this part is database knowledge, so users can raise queries containing complex intent such as multiple tables linking and nested relations, and the number of qualifying conditions involved in each query is up to three. In addition, users can choose to further parse individual tables in this knowledge base, and the rest of usage process is similar to that of Single-Table Q&A, as shown in Figure 5.

3.2.3. **Knowledge Base Management**

Single table management function can manage existing or import new knowledge bases, as shown in Figure 6. Users can either populate data by downloading table templates or directly import database files of specified formats and build a knowledge base for each table. The use of database management is similar to single table, the difference is that only the specified database format file is supported when importing data, because the database creation process is more complicated.
3.3. System Deployment Method

Since model algorithm requires certain computational resources in inference process, the TorchServe framework is introduced to deploy web services in order to reduce computational costs. TorchServe is a model serving framework developed for the machine learning framework PyTorch to deploy large-scale deep learning models, which can provide low-latency, lightweight services. To make inference process of the model unified on a single server with computational power, while multiple clients only need to send requests and receive computational results from the server, we use the deployment method shown in Figure 7.
In order to cope with the system's ability to handle a large number of highly concurrent or single discrete client requests under the condition of limited computing resources, a batch inference method is also introduced in the process of deploying and building remote services using TorchServe. When the client sends multiple requests, batches are combined first, and after the specified number of batches is reached or specified time has been waited, the batch requests are sent to the model together for inference, finally the results are returned to client in order. Specifically, when registering the model on TorchServe, the entire batch inference process is controlled by defining the configuration file, the main configuration of which is shown in Table 1.

| Variable name   | Descriptions                        | Value     |
|-----------------|-------------------------------------|-----------|
| Batch_size      | Batch size for model input.         | 8         |
| Max_batch_delay | Maximum delay time to wait request. | 300/ms    |
| Min_workers     | Minimum number of worker processes for model input | 1         |
| Max_workers     | Maximum number of worker processes for model input | 4         |
| Response_timeout| Model response timeout              | 300/ms    |
| Default_version | Model default version               | 1.0       |
| Mar_name        | Model file Name                     | TableSQL, DatabaseSQL |

4. System Application and Testing
In order to have a comprehensive evaluation of the performance on Q&A system, this section tests the system from both functional and non-functional points of view. The hardware environment tested was Ubuntu 20.04 operating system; Intel(R) Core(TM) i7-9700 CPU @ 3.00GHz processor; NVIDIA GeForce RTX 3060 graphics card; and 16G of running memory.

4.1. Functional Testing
The system function testing mainly adopts the black-box testing method to input data with specified format to system and check whether the output results can be obtained as expected. The experiments focus on testing the core Q&A functions of our system to check the performance of the system in actual business scenarios.

For training two models in algorithm layer, existing publicly available datasets are used. The TableSQL model uses TableQA dataset, which contains more than 49,000 query and SQL statement pairs and involves more than 5,200 tables; the DataBaseSQL model is trained using CSpider dataset, which contains more than 9,600 query and SQL statement pairs and involves more than 160 databases in total, and each database contains an average of 5.28 tables per database.

In order to test the system's generalization capability under practical application scenarios, experiment collected databases of 5 business information systems to construct test cases and invited relevant business personnel to ask questions under different scenarios, while the results generated by the system were then judged manually. The experiment required business personnel to ask 5 questions for each form and 20 questions for each database, and the logical form accuracy $ACC_{LF}$, execution accuracy $ACC_{EX}$, and average accuracy $ACC_{ME}$ were used as evaluation criteria when assessing the results. Where $ACC_{LF}$ indicates the number of output SQL statements that can exactly match the
correct result in logical form as a proportion of the total test samples; \( ACC_{LF} \) indicates the number of SQL statements whose execution results can match the correct answer as a proportion of the total test samples; and \( ACC_{ME} \) indicates the average of both \( ACC_{LF} \) and \( ACC_{EX} \).

**TABLE 2 EXPERIMENTAL RESULTS ON DIFFERENT DATABASE.**

| Database domain (Number of tables) | Single-Table Q&A | Database Q&A |
|-----------------------------------|------------------|--------------|
|                                   | \( ACC_{LF} \) | \( ACC_{EX} \) | \( ACC_{ME} \) | \( ACC_{LF} \) | \( ACC_{EX} \) | \( ACC_{ME} \) |
| Institutional Staffing (3)        | 80.6            | 85.9         | 83.3         | 41.5          | 43.6          | 42.6          |
| Single project construction (6)   | 69.9            | 72.3         | 71.1         | 33.9          | 37.1          | 35.5          |
| Auxiliary project construction (5)| 77.3            | 78.2         | 77.8         | 36.0          | 39.3          | 37.7          |
| Building area standards (6)       | 75.7            | 79.6         | 77.7         | 25.4          | 26.9          | 26.2          |
| Project Estimated Amount (3)      | 83.4            | 86.6         | 85.0         | 35.4          | 36.2          | 35.8          |

As can be seen from the results in Table 2, the system exhibits some generalization performance on the databases from different domains used for testing. The execution accuracy of both types of tasks is generally higher than the logical accuracy because in some cases the same query intent can be represented by multiple SQL statement forms. It is also found that the system generally outperforms the database Q&A on single-table Q&A, we believe there are two main reasons for this. On the one hand, the knowledge base on the database is more complex and the SQL statements are more difficult to construct. On the other hand, by analyzing the error results, we find that the database Q&A task is less effective in handling queries containing database values, this is because a separate value extraction module is designed in Single-Table Q&A, which fully considers the source of value information in this scenario, and Database Q&A uniformly uses a sequence generation structure based on a pointer network, which lacks some relevance.

**TABLE 3 EXPERIMENTAL RESULTS ON DIFFERENT INSTANCES.**

| Starting Event                  | Ending Event                                           | Q&A Task Module                | Average response time (s) | Server CPU usage (%) | Server GPU usage (%) |
|--------------------------------|--------------------------------------------------------|--------------------------------|---------------------------|----------------------|----------------------|
| Enter a question to start query| Joint knowledge base fed into model encoder            | Single-Table                   | 0.41                      | 20                   | 22                   |
|                                 |                                                        | Database                       | 0.36                      | 21                   | 14                   |
| Encoded vectors are fed to the model decoder | The decoder generates SQL statements and returns to the interface | Single-Table                   | 1.29                      | 23                   | 30                   |
|                                 |                                                        | Database                       | 1.55                      | 23                   | 32                   |
| Execute the generated SQL statements | Return execution results to the interface             | Single-Table                   | 0.22                      | 20                   | 1                    |
|                                 |                                                        | Database                       | 0.25                      | 20                   | 1                    |

4.2. *Performance Testing*

System performance is a description of the operation quality of the system, and is directly related to the user experience. The performance test of this system mainly includes non-functional indicators such as average response time, CPU usage, GPU usage, etc.

In the scenario where a single user sends a request, we test the system by designing different instances for different stages of the Q&A task running process, and the specific results are shown in Table 3. It can be seen that the overall response time for completing a single Q&A task is 1.92s and 2.16s for single table and database Q&A respectively, which is in line with the expected results of the system.
To further explore the performance of system under the stress load conditions of handling multiple Q&A requests, the average response times of two key functions are analyzed with different numbers of requests, respectively, as shown in Figure 8. It can be seen that when the number of requests reaches 8, the average response times of the two Q&A tasks are respectively 10.17s and 9.88s, compared with the one-by-one inference approach, our approach achieves an advantage of 5.19s and 7.40s when using the average response times of 1.92s and 2.16s for two tasks under a single request as the reference benchmark. This also illustrates that the approach of batch inference of multiple requests with limited computational resources (single GPU) helps to reduce system response time and improve system performance.

4.3. Testing Summary
The functional and performance test results of the structured knowledge base Q&A prototype system show that system basically fulfills the task requirements and meets the expected results. The test results further show that the use of batch inference can effectively alleviate the problem of long response times of traditional deep learning model-based systems for processing multiple requests under the condition of limited computing resources. However, it can be seen from the test results that the functions and performance of the system still need to be further optimized, especially in terms of functional details design and performance stability, so the prototype system still has a large gap with the actual application level.

5. Conclusion
In this paper, a prototype system based on structured knowledge base Q&A is designed using the common table structure data in major business systems as the source of knowledge. Using neural network models trained on large-scale datasets, the natural language queries proposed by users are mapped into SQL statements that can be executed on database table structures of different levels of complexity, and the results are returned to users in an interactive question-and-answer format, thus simplifying the query process, increasing query efficiency, and improving user experience.

REFERENCES
[1] H Sun, T Bedrax-Weiss, and WW Cohen. “PullNet: Open Domain Question Answering with Iterative Retrieval on Knowledge Bases and Text,” In EMNLP-IJCNLP, Hong Kong, China, pp. 2380–2390, November 2019.
[2] W Xiong, M Yu, S Chang, X Guo, and WY Wang. “Improving question answering over incomplete kbs with knowledge-aware reader” In ACL, Florence, Italy, PP. 4258–4264, July 2019.
[3] J Han, B Cheng, and X Wang. “Open domain question answering based on text enhanced knowledge graph with hyperedge infusion,” In EMNLP, Online, pp. 1475-1481, November 2020.

[4] A Saxena, A Tripathi, P Talukdar, et al. “Improving multi-hop question answering over knowledge graphs using knowledge base embeddings,” In ACL, Online, pp. 4498-4507, July 2020.

[5] H Wang. “Research on Q&A system based natural language processing,” University Of Electronic Science and Technology of China, 2006.

[6] J Zhang. “Performance optimization of intelligent Q&A system based on deep neural network,” Nanjing University of Posts and Telecommunications, 2020.

[7] A Abujabal, M Yahya, M Riedewald, et al. “Automated template generation for question answering over knowledge graphs.” In WWW, pp. 1191-1200, April 2017.

[8] K Luo, F Lin, X Luo and K Zhu. “Knowledge base question answering via encoding of complex query graphs,” In EMNLP, Brussels, Belgium, PP. 2185–2194, October-November 2018.

[9] Y Sun, L Zhang, G Cheng, Y Qu, et al. “SPARQA: skeleton-based semantic parsing for complex questions over knowledge bases,” In AAAI, Vol. 34, No. 05, pp. 8952-8959, 2020.

[10] Y Chen, H Li, Y Hua, G Qi. “Formal Query Building with Query Structure Prediction for Complex Question Answering over Knowledge Base,” In IJCAI, pp. 3751-3758, 2020.

[11] S Hu, L Zou and X Zhang, “A state-transition framework to answer complex questions over knowledge base,” In EMNLP, Brussels, Belgium, pp. 2098–2108, October-November 2018.

[12] M Huang. “Design and implementation of a Q&A-based reporting system,” Beijing University of Posts and Telecommunications, 2020.