Research and Practice of Sample Data Set Collection Platform Based on Deep Learning Campus Question Answering System

Wu Zhixia(✉)

School of Mathematics and Information Science, Nanjing Normal University of Special Education, Nanjing, China
250054@njts.edu.cn, 529249027@qq.com

Abstract. This document expounds the design and implementation scheme of a question-and-answer sample dataset collection platform using Spring+ SpringMVC+MyBatis framework and SQL data storage technology. The system mainly provides four functional modules: the text system file import module, the question and answer sample data set collection module, Question and answer sample dataset management module, the question and answer sample dataset output module. This research provides services for domain-specific collection and organization of question answering datasets.

Keywords: SSM · question answering system · BERT

1 Introduction

Intelligent question answering system is a human machine dialogue service that integrates technologies such as knowledge base, information retrieval, ma-chine learning, and natural language understanding. According to different ap-plication fields, intelligent question answering systems are usually divided into open domain question answering systems and limited domain question answering systems. In an open domain Q&A system, it is necessary to have a rich knowledge base, which provides a certain foundation for answering questions in multiple fields [1]. For example, using DeepPavlov's chat robot. DeepPavlov is a powerful open source AI library for chat robots and dialogue systems developed by the Moscow Institute of Physics and Technology (MIPT). It uses a large amount of article data from Wikipedia as its source of knowledge. However, when answering questions in professional fields, open domain Q&A systems are difficult to accurately locate answers. Currently, Q&A systems for limited fields such as law,healthcare, and finance are relatively mature, but Q&A systems in the field of universities are in their early stages [2]. If we can analyze the characteris-tics of the University, take the campus as a guide, and help universities establish a unified and reliable information acquisition platform to automatically answer students' questions, such as enrollment consultation,campus rules and regula-tions, campus library opening hours, and leave approval process, we can reduce labor costs and achieve automation and convenience in campus service work, It can also provide services for dynamic monitoring of students' school situa-tion.

B. Wang et al. (Eds.): ICMTEL 2023, LNICST 535, pp. 3–10, 2024.
https://doi.org/10.1007/978-3-031-50580-5_1

To build a campus intelligent question and answer system, it is necessary to re-duce the occurrence of "wrong answers" during the intelligent question and answer process. Usually, it involves the repeated steps of collecting question and answer datasets, selecting pretrained models, using training sets to participate in training, verifying model effectiveness using test datasets, improving training models, to retrain and to revalidate. Finally, the optimal model was determined and applied to the question answering system to achieve intelligent question an-swering [3]. To build a limited domain campus Q&A system, building a Q&A data collection platform to collect and organize data is a necessary step. This article elaborates on the design and implementation of a sample dataset collection plat-form for a question answering system using the SSM (Spring+SpringMVC+MyBatis) framework and SQL data storage technology.

2 SSM Framework

The SSM framework is composed of three open source frameworks, Spring, Spring MVC, and MyBatis. Spring MVC corresponds to the Controller layer in the foreground and is responsible for the separation of MVC. The MyBatis framework provides support for data persistence. As a lightweight IOC (Inversion of Control) container, Spring is responsible for finding, locating, creating, and managing objects and dependencies between objects, making Spring MVC and MyBatis work better. This topic uses Spring+ Spring MVC+ MyBatis, a lightweight development framework for implementation.

3 BERT for Question Answering System

BERT is Bidirectional Encoder Representation from Transformers, a pre-trained model proposed by Google in 2018. Pretraining is a concept of transfer learning [4]. For example, if we have a large amount of Wikipedia data, we can use this large amount of data to train a model with strong generalization ability. When we need to use it in specific scenarios, we only need to modify some out-put layers, then use our own data for incremental training, and make slight ad-justments to the weights. There are many pretrained language models, and BERT is one of them [5]. It emphasizes the use of a new Maked Language Model (MLM) to generate deep bidirectional language representations, rather than using tradi-tional one-way language models or shallow cascades of two one-way language models for pretraining.

When referring to the "question answering system" as an application of BERT, it usually refers to using BERT for SQUAD (Stanford Question Answering Dataset) [6]. The Stanford Question and Answer Dataset (SQUAD) is a new read-ing comprehension dataset that is based on questions posed in Wikipedia articles. There are over 100000 question and answer pairs in over 500 articles, with each answer being a paragraph or span of the corresponding reading article [7]. In BERT applications, it is necessary to input both the question and the text frag-ment containing the answer, separated by a special symbol [SEP]. BERT can out-put the answer at the beginning and end of the text fragment, which means it can highlight the text range of the corresponding answer [8]. In the later stages of this project, we will conduct incremental training on the collected Chinese sample question answering dataset to build an intelligent question answering system that is in line with the campus.

4 Analysis and Design of Sample Data Set Collection Platform for Question Answering System

4.1 Main Functions of the System

From the perspective of users, the platform is divided into two roles: ordinary users and administrators. From the function module, it is divided into two modules: foreground and background management. The front end is open to ordinary users, providing a Q&A sample data set collection module. The background is open to the administrator role, providing a text system file import module, a Q&A sample data set management module, and a background Q&A sample data set output module.

(1) Text system file import module
This module provides dynamic reading of imported text, and the content of the file is divided into file descriptions, chapters, and regulations, which are stored separately in relevant data tables.
(2) Question and answer sample dataset collection module
This module allows users to select documents, chapters, and regulations to view the content of the regulations. The user enters the question and answer, and marks the reference information block of the answer in the regulatory content.
(3) Question and answer sample dataset management module
For administrators to randomly check the correctness of the collected Q&A pairs and standard reference information segments. If there are errors in the collected Q&A information, the administrator can directly correct or send a reminder to prompt the collector to correct the sample data.
(4) Question and answer sample dataset output module
Generate training and test sets in JSON format for subsequent deep learning.

4.2 Data Flow Diagram

The Data Flow Chart (DFD) is the main tool used to describe the data flow of a system. It uses a set of symbols to describe the flow, processing, and storage of data in the system. The administrator visits the text system file import module and imports a specific domain file for which to collect Q&A pairs. The content of this document will be split and stored separately in the file description table named “Subject”, the chapter description table named “Chapter Title”, and the regulatory table named “Rules”. Ordinary users visit the Q&A sample dataset collection module, select rules, input possible Q&A pairs, and mark reference information segments. The collected question and answer pairing data is stored in a question and answer table called ‘Qanswer’. The administrator visits the Q&A sample dataset management module and randomly checks the correctness of the Q&A and marked reference information segments. They can access and use the data in the Q&A table. After completing the collection of Q&A pairs, administrators can use the Q&A sample dataset output module to generate JSON formatted training and testing sets for further indepth learning. The data flow of the entire project is shown in Fig. 1.

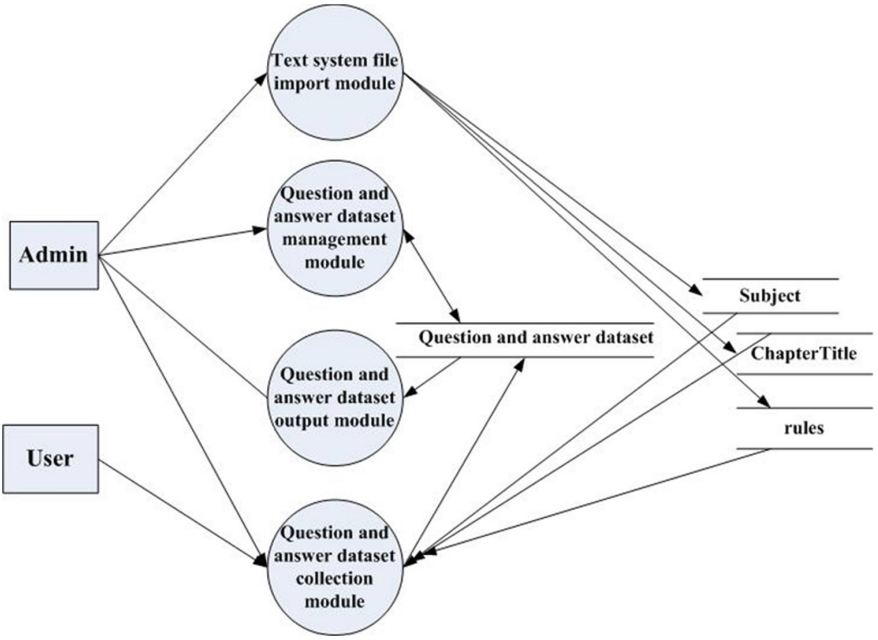

Fig. 1. Data Flow Diagram

4.3 Database Design

Based on the principle of designing a data structure with high efficiency and low redundancy, four tables are mainly designed. Mainly include the following Tables:

(1) subject. A table of subject about the file. Mainly used to record the subject name of the imported file. The fields included are id, subject name, sequence number and so on.
(2) chapterTitle. A table of chapters about the document. Mainly used to record the chapter name of the file. The fields included are id, chapter name, sort number, subject_id, create time.
(3) rules. A table of related regulations. Mainly used to record the rules name. The fields includes are id, title, content, sequence number, subject_id, chapterTitle_id, create time.
(4) qanswer. A table of question and answer. Mainly used to record the question and answer. The fields includes are id, question, answer, referencesTxt, rule_id, subject_id, chapterTitle_id, username, createtime, reviewer1, review time 1, audit level 1, reviewer 2, review time 2, audit level 2, reviewer 3, review time 3, audit level 3, approved level, status.
(5) user. A table of user. Mainly used to record the information about user. The fields includes are id, username, password, sex, tel, address, registerTime, role and so on.

4.4 System Architecture Design

According to the different functions and the layered architecture idea of the SSM framework, the system is strictly divided into five hierarchical structures, namely the entity layer, the data access layer (DAO Layer), the business logic layer, the controller layer, and view layer. Each level complements each other and completes the framework of the system together. The structure of the system is shown in Fig. 2.

(1) Entity layer (Domain Layer): This layer consists of several entity classes.
(2) Data access layer (DAO Layer): This layer consists of DAO interfaces and Mybatis mapping files. The name of the interface uniformly ends with Mapper, and the name of the mapping file of MyBatis is the same as the name of the interface. This layer is mainly used to define the operations of adding, deleting, modifying, and querying database tables as abstract methods in the DAO interface, and provide specific implementations of DAO interface abstract methods in Mybatis mapping files.
(3) Business logic layer (Service Layer): This layer consists of service interface and implementation classes. In this system, the interfaces of the business logic layer are uniformly ended with Service, and Impl is added after the interface name to achieve uniform class names. This layer is mainly used to implement the business logic of the system.
(4) Controller layer (Controller Layer): This layer mainly includes Controller classes in SpringMVC. The Controller class is mainly responsible for intercepting user requests, instantiating corresponding components in the business logic layer, and then calling the corresponding methods provided by the instantiation object to process user requests, and then returning the processing results to the JSP page.
(5) View layer (View Layer): This layer mainly includes JSP and HTML pages. The JSP and HTML pages are mainly responsible for providing an interface for users to input data or display data.

4.5 Sequence Diagram

(1) user login sequence diagram

The sequence diagram of user login access is shown in Fig. 3. It can be observed that the user visits a webpage called login.jsp, enters a username and password on the page, submits the form, and sends the request to the controller layer. A class named UserController in the control layer is responsible for processing requests. It encapsulates the received form information into a User entity, instantiates a class named UserService in the business logic layer into an object, and then calls the checkUser() method provided by the object. The checkUser() method in UserService is implemented by calling the checkUser() method provided by UserDao. UserDao belongs to the class under the data persistence layer, which provides interaction with the database.

(2) Question and answer dataset collection sequence diagram

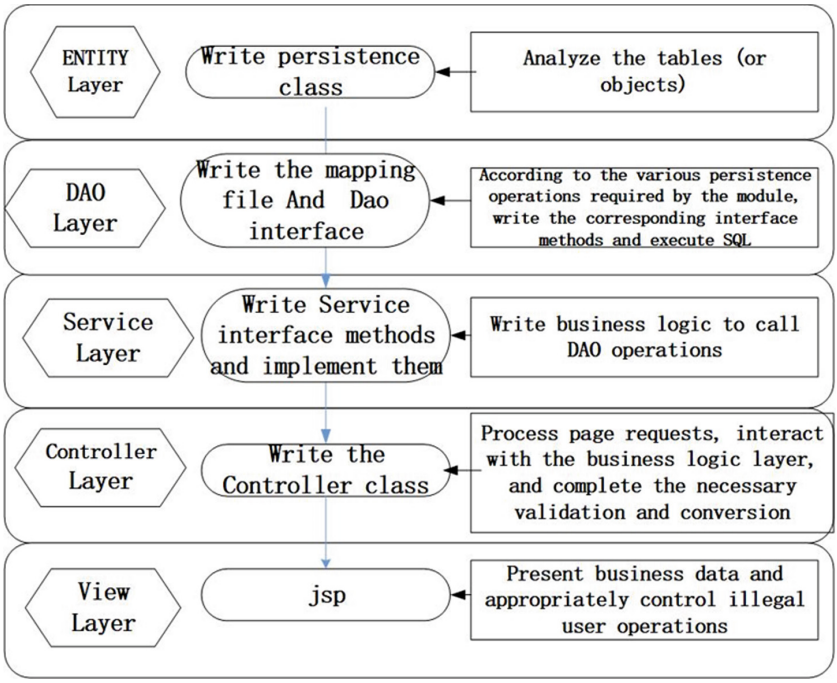

Fig. 2. System hierarchy diagram

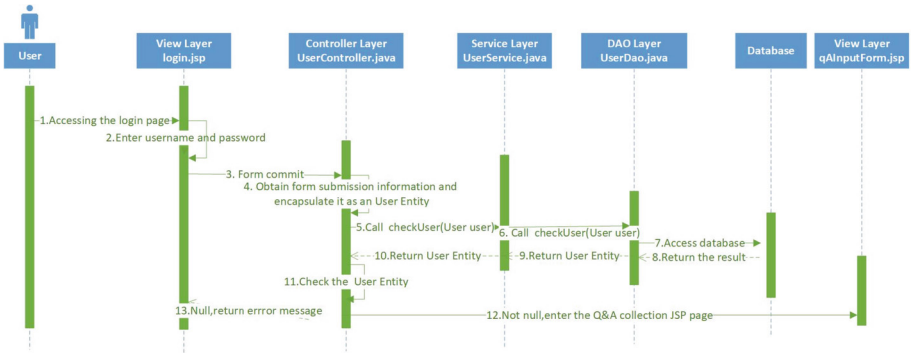

Fig. 3. User login sequence diagram

The sequence diagram of the Q&A dataset collection is shown in Fig. 4. It can be observed that the user visit the Q&A sample dataset collection page, input Q&A pairs on the page, and marked reference information segments, submits the form, and sends the request to the controller layer. A class named QAnswerController in the control layer is responsible for processing requests. It encapsulates the received form information into a QAnswer entity, instantiates a class named QAnswerService in the business logic layer into an object, and then calls the add() method provided by the object. The add () method

in QAnswerService implements the storage of Q&A data in database tables by calling the add () method provided by QAnswerDao.

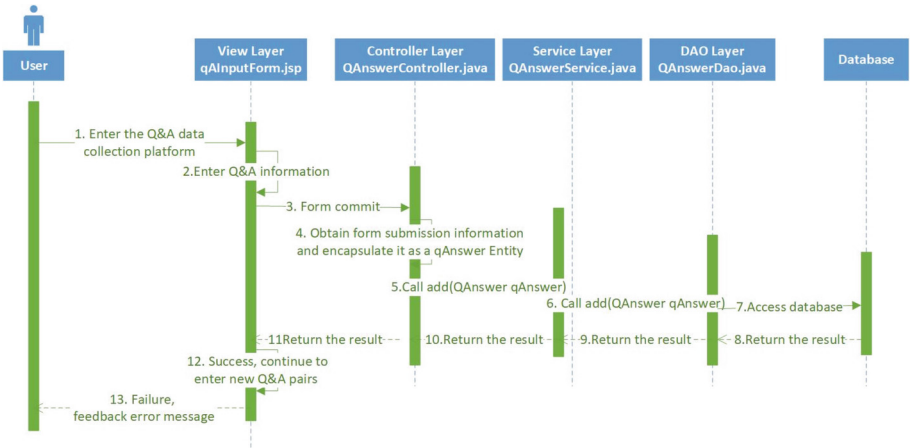

Fig. 4. Question and answer dataset collection sequence diagram

4.6 The Output Can Be Used for the Sample Dataset Based on the BERT Chinese Question Answering System

This platform imported 48 campus rules and regulations documents from a vocational college in Anhui. New students, full-time teachers, and counselors complete the collection and input of Q&A pairs based on familiar rules and regulations, and verify the correctness of the answers on the platform. The platform collected a total of 4500 sample data, including 3100 available data with a certification score of 7.0 or higher. Use the output module provided by the platform to convert the obtained sample dataset into JSON format for pretraining.

4.7 BERT for Question Answering

Divide the collected data set into training set and verification set, apply BERT to the sampled Chinese sample data set, split Chinese words according to spaces, rewrite the code in run_squad_chinese.py, and conduct training and verification experiments. The format of the sample data set generated by the platform can meet the experimental requirements.

5 Conclusions

This paper elaborates on the design and implementation of a Q&A data collection platform using the SSM (Spring+SpringMVC+MyBatis) framework and data storage technology. We collected and organized 4500 sample data related to school rules and regulations from a vocational college in Anhui. The completion of this work provides a good

foundation for the subsequent construction of campus intelligent question answering systems.

Acknowledgment. We thank the students from Ma'anshan Teacher College for assisting in the collection of the question and answer. This work was supported by the Natural Science Research Project of Anhui Universities "Research on Campus FAQ Based on Deep Learning", with the grant number KJ2020A0884.

References

1. Lee, J., Seo, M., Hajishirzi, H., Kang, J.: Contextualized sparse representations for real-time open-domain question answering. In: Procedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 912–919. Association for Computational Linguistics (2020)
2. Li, C., Choi, J.D.: Transformers to learn hierarchical contexts in multiparty dialogue for span-based question answering. arXiv:2004.03561v2 (2020)
3. Baheti, A., Ritter, A., Small, K.: Fluent response generation for conversational question answering. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 191–207 (2020)
4. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M.: RobERTa: a robustly optimized BERT pretraining approach. arXiv:1907.1169zv1 (2019)
5. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805v2 (2019)
6. Liu, Y., Ott, M.: RoBERTa: a robustly optimized BERT pretraining approach. arxiv (2019)
7. Dibia, V.: NeuralQA: a usable library for question answering (contextual query expansion+BERT) on large datasets. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 15–22. Association for Computational Linguistics (2020)
8. Choi, E., et al.: QuAC: question answering in context. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, pp. 2174–2184. Association for Computational Linguistics (2018)