Design and Implementation of Intelligent Question Answering Robot for Human Resources and Social Security Based on DCU and ChatGLM6B Model

Lei Hua 1, Mingliang Wang 2, Xingyue Mao 2, and Zilin Fu 3,*

1 Liaoning Provincial Social Insurance Enterprise Service Center, Liaoning, China
2 Shenyang Dingli Software Technology Co., Ltd, Liaoning, China
3 Northeastern University, Liaoning, China

* fuzilin_0917@hotmail.com

Abstract. This paper presents a human resources and social security based on DCU and ChatGLM6B model. This solution combines the high performance computing power of iguang DCU with the natural language processing technology of ChatGLM6B model, aiming to provide users with efficient and convenient voice interaction experience. This paper first introduces the research background and significance of intelligent question answering robot, then expounds the design scheme and implementation process of the system, and finally conducts the test and performance analysis of the system. The experimental results show that the intelligent question-answering robot designed in this paper has high answer accuracy and good user experience, and can provide intelligent services for the field of human resources and social security.

Keywords: Haiguang DCU, ChatGLM6B, intelligent question-answering robot, human resources and social security.

1. Foreword

With the popularization of the Internet and the development of artificial intelligence technology, the intelligent question-answering robot has become an important way of intelligent service. In the field of human resources and social security, intelligent question-answering robots can provide users with policy consultation, business management and other services to improve service efficiency and quality. However, there are some problems in the current intelligent question-answering robot, such as low accuracy and poor user experience. Therefore, this paper presents a design scheme for intelligent question-answering robots for human resources and social security based on the sea-light DCU and ChatGLM6B models, aiming to solve these problems.

2. System Design

The intelligent question-answering robot designed in this paper mainly consists of three parts: speech recognition module, natural language processing module and answer generation module. Among them, the speech recognition module is responsible for converting the user's speech into text, the natural language processing module is responsible for the word segmentation, part of speech annotation and other processing of the text, and the answer generation module generates the corresponding answers according to the user's questions. The design of the three modules is elaborated in detail below.

2.1 Speech recognition module

2.1.1 Overview

The speech recognition module is a key part of the intelligent question answering robot. Its function is to convert the speech signals sent by users into text form and provide input for the subsequent natural language processing. In the field of human resources and social security in
China, the accuracy, real-time, and robustness of speech recognition are particularly important due to the large number of consultation and services involved.

2.1.2 Application of Marine light DCU in speech recognition

The Marine light DCU (processor) plays a central role in the speech recognition module. Its powerful computing power ensures the real-time operation of speech recognition algorithms, and the low-power design enables the robot to maintain stable performance in the continuous working state. For the deep learning algorithms of speech recognition, the Marine Light DCU provides hardware acceleration to ensure the efficient execution of the algorithm[1].

2.1.3 Speech recognition algorithm

The proposed speech recognition algorithm is based on deep learning techniques. Deep learning technology can automatically extract deep features in speech signals, and represent speech data more effectively than traditional methods. Through a large amount of speech data training, the deep learning model can learn the mapping relationship between speech and text, and realize high-precision speech recognition.

During the training phase of the model, we adopted a large-scale speech dataset. These datasets cover different scenarios, accents and speeds, ensuring that the model has strong generalization ability. By using the sea optical DCU, we quickly completed the training of the model.

2.1.4 Voice preprocessing

Before recognition, voice signals need to go through a series of preprocessing operations, such as noise reduction, frame segmentation, and window adding. These operations can remove background noise, segment speech signals into suitable frame length, and reduce spectral leakage. Preprocessing aims to improve the accuracy of speech recognition and to reduce the computational amount of subsequent processing.

2.1.5 Real-time performance of speech recognition

In the application of intelligent question-answering robot, real-time performance is an important indicator. We optimize the model structure and algorithms to reduce the computational complexity and realize real-time processing of speech recognition through the high performance computing power of sea light DCU. This allows the robot to instantly respond to the user's problems and provide a smooth human-computer interaction experience.

2.1.6 Robustness of speech recognition

In practice, the robustness of speech recognition is a challenge. User pronunciation, accent, speed, and background noise may all affect the accuracy of recognition. To improve the robustness, we employ a data augmentation technique to increase the generalization ability of the model by transforming the raw speech data and adding noise. Moreover, we introduce contextual information such as language models to use the pre-and post associations to improve the accuracy of identification.

2.2 Natural language processing module

2.2.1 Overview

Natural language processing (NLP)[2] module is one of the core components of the intelligent question answering robot. It is responsible for the semantic understanding and analysis of the user's questions, and generating the corresponding answers. In the field of human resources and social security, the NLP module plays an important role in interpreting policies, understanding users' needs and giving accurate responses. The design, technology, and application of the NLP module will be described in detail[3].
2.2.2 Application of the ChatGLM6B model

This paper has adopted the ChatGLM6B model as the basic framework for the NLP module. ChatGLM6B is a language model based on the Transformer structure, which is trained on a large amount of text data and learns the rules and patterns of natural language. By fine-tuning the ChatGLM6B model, we can adapt it to specific tasks in the areas of HR and social security.

2.2.3 Text preprocessing

A series of text preprocessing operations is required before entering user problems into the ChatGLM6B model. These operations include word segmentation, part of speech annotation, removal of stop words, etc. Word segmentation is a process of dividing continuous text into meaningful word sequences. For Chinese text, word segmentation is a necessary step. We use an efficient Chinese word segmentation algorithm to process the user problems. Part of speech marking is the process of labeling each word with its part of speech (such as nouns, verbs, adjectives, etc.), which helps us understand the semantics and structure of the problem. The removal of stop words is to remove some commonly used but no practical words to reduce noise and interference.

2.2.4 Semantic understanding and analysis

After pre-processing, we input the user's problems into the ChatGLM6B model for semantic understanding and analysis. The ChatGLM6B model captures the long-term dependencies and contextual information in the problem through the self-attention mechanism and the Transformer structure. It is able to understand the intentions, entities and relationships of the problem and generate the corresponding semantic representation. This semantic representation can capture the deep meaning of the problem, not just the surface.

2.2.5 Answer generation and optimization

According to the semantic representation, we further generate the corresponding answers. Answer generation can be used template matching, generative model and other methods. This paper adopts a generative model to learn the generation rules of answers by training data sets and generate answers that conform to the question semantics. At the same time, we also applied the answer optimization algorithm to sort and screen the generated answers to ensure the accuracy and rationality of the answers. These optimization algorithms can be implemented based on predefined rules, statistical information, or deep learning models. By continuously optimizing and adjusting the answer generation algorithm, we can improve the quality and diversity of the answers.

2.2.6 Context understanding and dialogue management

In the field of human resources and social security, user questions often involve previous dialogue content and contextual information. Therefore, the NLP module also needs to have the ability of context understanding and dialogue management. We designed a conversation state tracking mechanism that records the historical information and current state of conversations to make full use of contextual cues when answering questions. This mechanism can help the robot to better understand the user's intentions and needs, and to give coherent and accurate answers.

2.3 The fine-tuning and optimization of the ChatGLM6B model

The fine-tuning and optimization of the ChatGLM6B model is a key step to improve the performance of the intelligent question-answering robot. These two processes are described in detail below:

2.3.1 Fine-tuning (Fine-tuning)

Fine-tuning is further trained on the pre-trained model to make the model better adapted to a specific task. For the ChatGLM6B model, the fine-tuning process includes the following steps:
Data preparation: First, specific data sets applicable to human resources and social security need to be collected and prepared. These datasets should contain questions and answers from the field for training the model.

Model loading: load the pre-trained ChatGLM6B models as the starting point for fine-tuning.

Set fine tuning parameters: according to the task requirements and calculation resources, set the appropriate fine tuning parameters such as learning rate, batch size, training rounds and so on.

Training: Use the prepared dataset to fine-tune and train the model. During training, the model learns domain-specific knowledge and adjusts its own parameters to better complete the task.

Verification and Testing: Evaluation of the finely tuned model performance on the validation set and test sets. Depending on the evaluation results, you can adjust the fine-tuning parameters or try other optimization strategies.

2.3.2 Optimization

Model optimization is an ongoing process designed to continuously improve model performance and efficiency. For the ChatGLM6B model, the following optimization strategies can be considered:

Model structure adjustment: According to the task requirements, you can try to adjust the structure of the model, such as increasing or reducing the number of layers, changing the size of the hidden layers, etc., to find the best model configuration.

Parameter optimization: By adjusting the learning rate and regularization ability of the model, the training effect and generalization ability of the model can be improved.

Integrated learning: Use integrated learning methods, such as bagging, boosting, etc., to combine the outputs of multiple models to improve the overall performance.

Continuous learning: With the continuous update of data and the emergence of new problems, the model can be regularly retrained and fine-tuned to enable the model to adapt to new data and tasks.

It should be noted that the fine-tuning and optimization process requires full consideration of computational resources and time costs. In practice, experiments and experience can be used to find suitable fine-tuning and optimization strategies to achieve the best balance of model performance and efficiency.

2.4 Establish a local knowledge base scheme based on ChatGLM-6B

To enhance the performance of the ChatGLM-6B model in specific domains [4](such as human resources and social security), it is crucial to build a local knowledge base. Here is a detailed scenario for building a local knowledge base:

2.4.1 Knowledge collection

Field data collection: First, collect information related to human resources and social security from official websites, documents and policy documents.

Expert consultation: consult with experts or practitioners in the field to collect their insights, experiences and suggestions.

Data crawl: crawl domain-related question and answer data from relevant websites and forums, which can provide real question and answer pairs for the knowledge base.

2.4.2 Knowledge sorting and storage

Text cleaning: For the collected original text, clean it to remove irrelevant characters, advertisements, repetitive content, etc.

Structured processing: structure the cleaned text, such as extracting the keywords and topics of the question, labeling the answers, etc.

Indexing: In order to quickly retrieve the content in the knowledge base, you need to index the questions and answers.

Storage scheme: select storage schemes suitable for large-scale text data, such as Elasticsearch, MongoDB, etc., to ensure efficient query and storage.
2.4.3 Combination of the knowledge base with ChatGLM-6B

Knowledge enhancement: When fine-tuning the ChatGLM-6B model, the question and answer pair in the knowledge base is used as additional data and trained together with the original training data, so that the model can fully absorb the domain knowledge.

Combination of retrieval and generation: When the model receives a question, it first searches the knowledge base to find whether there are similar questions and corresponding answers. If found, return the answer directly; if not, rely on the model to generate the answer.

Answer fusion: For some questions, similar answers can be found from the knowledge base and answers can be generated by the model. At this time, the answer fusion strategy can be adopted to combine the advantages of the two to generate the final answer.

2.4.4 Update and maintenance of the knowledge base

Regular updates: As policies and environments change, the knowledge base needs to be updated regularly to ensure that the information is the most up to date and most effective.

User feedback driven update: Encourage users to provide feedback, such as errors or deficiencies in the knowledge base.

Data security: Ensure the data security in the knowledge base to prevent data leakage and illegal access.

Building a local knowledge base combined with the ChatGLM-6B model can significantly improve the performance of the model in specific areas, and provide users with more accurate and intelligent answers.

2.5 The deployment process of ChatGLM-6B model and local knowledge base using Python language can be divided into the following steps:

2.5.1 Environmental preparation

First, make sure that the Python interpreter and the necessary dependency library are already installed in your deployment environment. You can use the following commands to install the desired libraries:

```
python
!pip install grpcio grpcio-tools tensorflow
```

2.5.2 Load the ChatGLM-6B model

In Python, you need to use the appropriate library (such as TensorFlow) to load the pre-trained ChatGLM-6B model. First, the weights and profiles of the model are downloaded and saved under the suitable path. Then, load the model with the following code:

```
python
import tensorflow as tf
# Specifies the model path
model_path = "path/to/chatglm-6b/model"
# Load the model
model = tf.keras.models.load_model(model_path)
```

2.5.3 Load the local Knowledge base

Next, you need to load the local knowledge base to use in conjunction with the model. Select the appropriate library (such as Elasticsearch or MongoDB) based on your knowledge base storage scheme, and use the corresponding Python client for the connection and loading operations. Here is an example of loading a local knowledge base with Elasticsearch:

```
python
from elasticsearch import Elasticsearch
# Connect to the Elasticsearch instance
es = Elasticsearch([{'host': 'localhost', 'port': 9200}]
# Load the knowledge base index
```
2.5.4 Deployment of services

Once the model and knowledge base are loaded, you can write a Python service to receive the user's questions and return the corresponding answers. You can use frameworks like gRPC or Flask to build services. The following is an example code for using the gRPC:

```python
# Define the class for the gRPC service
class ChatService(your_pb2_grpc.ChatServicer):
    def __init__(self, model, knowledge_base):
        self.model = model
        self.knowledge_base = knowledge_base
    def AnswerQuestion(self, request, context):
        question = request.question
        # Retrieve answers in the knowledge base or generate answers using the model
        answer = self.get_answer_from_knowledge_base(question) or self.generate_answer_from_model(question)
        return your_pb2.Answer(answer=answer)
    def get_answer_from_knowledge_base(self, question):
        # Code logic for retrieving the answers in the knowledge base
        pass
    def generate_answer_from_model(self, question):
        # Code logic for generating the answers using the model
        pass
    def serve():
        server = grpc.server(futures.ThreadPoolExecutor(max_workers=10))
        your_pb2_grpc.add_ChatServicer_to_server(ChatService(model, knowledge_base), server)
        server.add_insecure_port('[::]:50051')
        server.start()
        server.wait_for_termination()
        if __name__ == '__main__':
            serve()
```

In the above code, you need to modify it accordingly according to your specific requirements and the gRPC definition file. Make sure to replace `your_pb2` and `your_pb2_grpc` with your own definition file and implement the `get_answer_from_knowledge_base` and `generate_answer_from_model` methods according to your knowledge base and model logic. Finally, run the script to start the gRPC service.

This is an example of a simplified deployment process that may involve more details and configurations. Please adjust it accordingly according to your specific needs and environment.
3. System Implementation and Testing

3.1 System implementation

In the process of realizing the intelligent question answering robot, we follow the principles of software engineering and carry out modular design and development. First, we divided the whole system into speech recognition module, natural language processing module and answer generation module. Then, we adopted the appropriate development tools and framework to implement each module specifically.

In the implementation of the speech recognition module, we utilized the high-performance computing power provided by the Marine light DCU, and combined with the deep learning algorithm to realize the speech-to-text conversion. We used efficient speech preprocessing to remove noise and interference and extract clear speech features. Then, through the trained deep learning model, the speech features are identified and transformed to obtain the corresponding text results.

The implementation of the NLP module is based on the ChatGLM6B model. We fine-tuned and optimized the model to accommodate specific tasks in the areas of HR and social security. We use preprocessing technology to process word segmentation, part-of-word annotation, and then input the processed text into the ChatGLM6B model for semantic understanding and analysis. Through the self-attention mechanism and the Transformer structure of the model, we obtain the semantic representation of the problem, which provides the basis for the subsequent answer generation.

The implementation of the answer generation module adopts the method of the generative model. According to the trained generative model of the problem, we generated the answers according to the semantic representation of the problem. We also applied the answer optimization algorithm to filter and sort the generated answers to ensure the accuracy and rationality of the answers. In addition, we implemented a context understanding and dialogue management mechanism that record historical information and current status of conversations to provide coherent and accurate responses.

3.2 System test

After completing the system implementation, we conducted a comprehensive test work to verify the correctness and performance of the system. We designed multiple test cases covering a variety of issues in the areas of human resources and social security. Test cases include different scenarios, accents, speed, and semantic complexity to fully test system robustness and adaptability.

We evaluated the system performance using metrics such as accuracy, recall and F1 value. By comparing the system output with the standard answer, we calculated the value of each indicator. Meanwhile, we also recorded the time consumption of the system processing each test case to evaluate the real-time performance of the system.

During the testing process, we encountered some challenges and problems. For example, some users' vague pronunciation or too fast speech may lead to errors in the speech recognition module. In addition, some problems have a high semantic complexity, which requires a deeper model understanding and reasoning capabilities. For these problems, we continuously optimize the algorithm and adjust the model parameters to improve the performance and accuracy of the system.

After multiple rounds of testing and performance tuning, our intelligent question-answering robot has achieved satisfactory test results. All the evaluation indicators meet the expected requirements, proving the effectiveness and feasibility of the system.

3.3 Simulation example: question and answer accuracy description

To illustrate the accuracy of the question-answering system, we can simulate a simple example. The following is a simulated dialogue based on the ChatGLM-6B model and the local knowledge base:

User: What is human resources and social security?
System: Human resources and social security refer to the development, management, protection and utilization of human resources through a series of policies and measures, so as to protect the people's rights and interests in employment and social security, and promote the development of economy and society.

In this example, the user raises a definition question about human resources and social security. Based on the trained ChatGLM-6B model and the local knowledge base, the system can correctly understand the user's intention and retrieve the corresponding answers from the knowledge base. Therefore, the system gives a matching and accurate answer to the question.

Accuracy is one of the important indicators to measure the performance of the question-answering system. In practice, the accuracy can be assessed by comparison with the standard answers. For example, manually annotated standard answers can be used to compare systematic responses and calculate metrics such as accuracy, recall and so on. If the system answer matches the standard answer well and can answer the user's question correctly, then the question and answer system can be considered highly accurate.

However, it should be noted that the accuracy of the question-answering system is affected by many factors, including the performance of the model, the quality and coverage of the knowledge base, and the clarity and accuracy of user questions. Therefore, in practical applications, the continuous optimization of the model and knowledge base, as well as providing user-friendly interaction methods, is the key to improve the accuracy of the question-answering system.

References