Machine Reading Comprehension to Answer COVID-19 Queries Using Bio-Bert and Multi-task Learning

by

Yong Wang

A thesis submitted in partial fulfillment

of the requirements for the degree of

Master of Science (MSc) in Computational Science

The Office of Graduate Studies

Laurentian University

Sudbury, Ontario, Canada

© Yong Wang, 2022

THESIS DEFENCE COMMITTEE/COMITÉ DE SOUTENANCE DE THÈSE
Laurentian Université/Université Laurentienne
Office of Graduate Studies/Bureau des études supérieures

Title of Thesis Titre de la thèse	Machine Reading Comprehension to Answer COVID-19 Queries Using Bio-Bert and Multi-task Learning
Name of Candidate Nom du candidat	Wang, Yong
Degree Diplôme	Maste of Science
Department/Program Département/Programme	Computational Sciences
Date of Defence Date de la soutenance	April 20, 2022

APPROVED/APPROUVÉ

Thesis Examiners/Examinateurs de thèse:

Dr. Kalpdrum Passi
(Supervisor/Directeur(trice) de thèse)

Dr. Ratvinder Grewal
(Committee member/Membre du comité)

Dr. Luckny Zephyr
(Committee member/Membre du comité)

Dr. Pradeep Atray
(External Examiner/Examinateur externe)

Approved for the Office of Graduate Studies
Approuvé pour le Bureau des études supérieures
Tammy Eger, PhD
Vice-President Research (Office of Graduate Studies)
Vice-rectrice à la recherche (Bureau des études supérieures)
Laurentian University / Université Laurentienne

ACCESSIBILITY CLAUSE AND PERMISSION TO USE

I, **Yong Wang**, hereby grant to Laurentian University and/or its agents the non-exclusive license to archive and make accessible my thesis, dissertation, or project report in whole or in part in all forms of media, now or for the duration of my copyright ownership. I retain all other ownership rights to the copyright of the thesis, dissertation or project report. I also reserve the right to use in future works (such as articles or books) all or part of this thesis, dissertation, or project report. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that this copy is being made available in this form by the authority of the copyright owner solely for the purpose of private study and research and may not be copied or reproduced except as permitted by the copyright laws without written authority from the copyright owner.

Abstract

In recent years, with the development of artificial intelligence technology in many fields, the question-answering system has brought significant change to knowledge acquisition mechanism. The question-answering system based on machine reading comprehension can obtain short and accurate answers compared with the traditional retrieval question-answering system. This thesis designs an intelligent question-answering method based on information retrieval and multi-document machine reading comprehension. Firstly, a two-stage information retrieval recall strategy is designed. After the information retrieval by using Bm 25, a structure called Dual Bio-Bert Retrieval is designed, which uses two Bio-Bert to extract semantic features in questions and paragraphs respectively in the training stage. Then, the information between the two Bio-Bert is interacted by utilizing Performer. At the reading comprehension stage, an additional method—Matching Tech is designed to improve the model. Compared with other common methods, the results of the information retrieval and reading comprehension shows that the models designed in this research have good performance.

Keywords: Question-answering system, machine reading comprehension, multi-task leaning, information retrieval

Acknowledgments

First of all, I would like to thank my supervisor Dr. Kalpdrum Passi. The smooth development of my thesis and the acquisition of knowledge during my postgraduate study is inseparable from his help. His profound knowledge, academic rigor, sensitivity to new things and new technologies, and spirit of inquiry are all worthy of my long-term study. His broad teaching vision, rigorous academic attitude, and approachable nature to others have set a positive and optimistic example for me.

Secondly, I would like to thank the leaders, teachers and classmates in the Mathematics and Computer Science Department. Without your tireless organization and dedication behind the scenes, there would not be such a comfortable study and living environment as I am now.

Then, I would like to thank my family. It is your concern, understanding and support that enable me to move forward bravely, and your uncomplaining dedication is the biggest motivation for me to persist in completing my studies.

Finally, I would like to express my heartfelt thanks to all the experts and professors who participated in the review of the thesis and provided valuable comments on this research!

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

In recent years, artificial intelligence technology has been accelerated in almost all domains of life, and the question answering system has brought revolutionary changes to people's current knowledge acquisition and answering questions. At the same time, with the rapid development of deep learning and the improvement of computing power, more and more deep learning techniques are used in question answering systems. With the influx of more and more deep learning technologies into the field of question answering systems, the accuracy of question answering systems has been greatly improved. At the same time, the deployment of human resources of the traditional question answering system is also reduced (Zhu, 2021).

Compared with traditional retrieval-based question answering systems, for a given natural language question, reading comprehension does not return relevant paragraphs based on the similarity to the question, but a short and accurate answer in natural language form corresponding to the question (Balduccini et al., 2008). In recent years, with the development of deep learning, the performance of machine reading comprehension tasks has improved significantly, especially for tasks with low semantic reasoning. The reading comprehension

based on pre-training even exceeds the level of human beings and has emerged in practical applications.

Doctor-patient consultation service is a specific application area of a question answering system. In the doctor-patient service, there is a lot of problem in consultation work, and most hospitals adopt the online and offline consultation mode. This model faces a series of problems such as low consultation efficiency, loss of answering data, untimely responses, and limited answering time. This model leads to patients' consultation needs not being met in a timely manner. Through our investigation, we found that there is a lot of repetition and randomness in the questions that patients ask. Therefore, the starting point of this research is to use the data accumulated in offline consultation and some commonly used problem explanation data to establish an automatic Q&A consultation platform for hospitals to provide patients with real-time answering to patient’s queries.

Question answering systems are widely used; for example, in search system engines, personal intelligent question and answer services and equipment, automatic question and answering robots in service halls, intelligent verification codes, etc.

- Search system engine. It is difficult for traditional search engines to obtain clear and concise answers from a large amount of data, and the question answering system is a more advanced form of information retrieval.

- Personal Smart Question Answering Services and Devices. For example, the voice intelligent assistant Apple Siri, Microsoft Cortana, etc.
- Automated question-and-answer bots in service halls. For example, ASUS' question-answering robot and customer service robots of some products. The use of this type of question-answering system greatly reduces the communication problems between users and merchants.
- Smart verification code. The verification code is also called human-computer interaction inspection, which is generated to prevent the server from being burdened by malicious access by hackers or criminals. However, the traditional image-based recognition method is easy to crack through machine learning methods, or the image is so blurry that the model cannot recognize it, and it is difficult for people to interactively verify it. The captcha based on unstructured automatic text knowledge response is difficult to crack and has better effect.

Machine reading comprehension can solve one of the challenges of traditional retrieval-based question answering-precisely locating answers. Usually, after a user enters a question, several candidate documents are retrieved from a massive document set, and these several candidate documents are segmented and sorted by paragraphs, and finally the answer is directly fed back to the user in units of paragraphs (Bouziane et al., 2015). However, these paragraphs often also contain redundant information. Using the technology of machine reading comprehension, it is possible to get more direct, concise and accurate answers.

1.2 Current Situation

The machine reading comprehension (Machine Reading Comprehension, MRC) task mainly refers to letting the machine answer text-related questions according to the given text, so as to measure the machine's ability to understand natural language. The origins of this mission can be traced back to the last actual 70s. However, due to the limitations of small datasets and traditional rule-based methods, machine reading comprehension systems could not meet the needs of practical applications at that time. This situation changed in 2015, mainly due to the following two points: a deep learning-based machine reading comprehension model was proposed. This type of model is more suitable for mining the contextual semantic information of text, and its performance is significantly improved compared with traditional models; the release of a series of large machine reading comprehension datasets, such as CNN and Daily Mail (Cui et al., 2016), SQuAD (Rajpurkar et al., 2016), MS MARCO (Nguyen et al., 2016). These datasets make it possible to train deep learning reading comprehension models and also serve as a good way to test the effects of the models.

The development track of artificial intelligence technology and question answering system blends with each other. To be precise, the question answering system rises and falls with the rise and fall of artificial intelligence. Reading comprehension systems appeared after the 1970s, but reading comprehension systems at that time had to script to describe the answers to questions. With the emergence of various new technologies, around the 1990s, the retrieval-

based question answering system gradually replaced the expert question answering system (Deng et al., 2020).

However, such retrieval-type question answering systems can only answer simpler questions based on a single fact. To address this problem, websites featuring question-and-answer communities have popped up on the Internet. By building a Q&A community with a large number of real users, users can actively submit questions, and users who follow this community tend to answer these questions, and the answer owners are of high quality (Hu et al., 2018). However, this kind of question answering community cannot be called an intelligent question answering system, because the whole process mainly uses relatively shallow keyword matching technology, and does not use the semantic information contained in the question. With the increase of data scale and the application of deep learning technology, as well as the rapid progress of reading comprehension technology in recent years, question and answer based on knowledge graph and question and answer based on reading comprehension have been used. Especially after 2018, the emergence of various pre-trained language models have further improved the effect of reading comprehension models, providing a strong impetus for the implementation of reading comprehension-based question answering systems.

1.3 Thesis Objective

The main goal of this thesis is to improve multi-document machine reading comprehension algorithms and build a system for automatically and efficiently answering patient concerns

about COVID-19. For the document data in the question answering system, the answers generated by the traditional retrieval question answering system are difficult to locate accurately. To solve the above problems, this thesis designs a multi-document reading comprehension question answering system. Figure 1.1 shows the brief framework. For a question, first, use information retrieval technology to retrieve relevant candidate documents, and then use the methods of fine recall and paragraph division to reorder the obtained document fragments and paragraphs, and then perform machine reading comprehension tasks to recall the final refined answer.

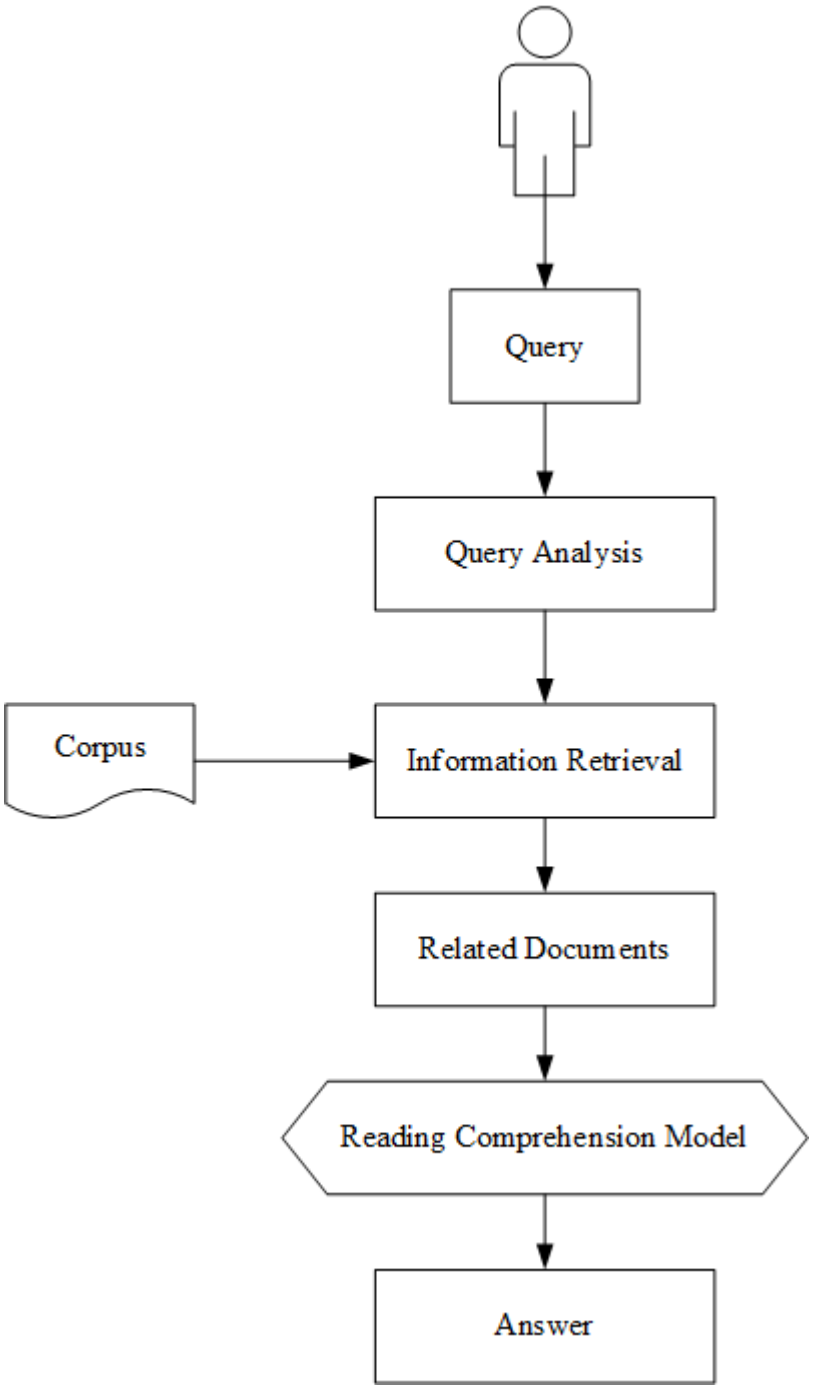

Figure 1.1. Multi-document Machine Reading Comprehension Framework Diagram

The main research methods and contents of this thesis are arranged as follows. First of all, combining the pre-training model and the traditional scheme, this thesis proposes a pipeline-style multi-document machine reading comprehension structure, which can return accurate answer fragments from the candidate document library after inputting relevant questions. Secondly, machine reading comprehension tasks, structures, and loss functions are improved to improve model performance.

The contribution of this thesis can be summarized as follows. First, a two-stage information retrieval strategy is designed to improve the special case that the traditional fine recall method only considers the surface information of the article such as word frequency and inverse document, but ignores the actual central word in the text and propose a fine recall model Dual Bio-Bert Retrieval for information retrieval based on the feature vector of the pre-training model. Secondly, a multi-task joint learning strategy is designed to improve the performance of machine reading comprehension. In the reading comprehension task, the task of matching the question and the paragraph is added to strengthen the interaction ability of the pre-training structure to the question and the candidate paragraph, and the downstream structure can improve the understanding of the actual task through the interaction of the downstream structure between different tasks.

1.4 Thesis Outline

This thesis consists of seven chapters, the main contents are as follows.

Chapter 1 is the introduction. It mainly introduces the research background of this research, and briefly introduces the main content and organization of this thesis.

Chapter 2 is a literature review, which introduces the history and development of machine reading comprehension.

Chapter 3 introduces some work related to this research, including the commonly used structures of question answering systems and machine reading comprehension and the metrics used to evaluate model performance.

Chapter 4 details the two-stage information retrieval strategy designed in this thesis and the experiments performed on information retrieval.

Chapter 5 introduces the multi-document machine reading comprehension structure designed in this thesis and the detailed experimental setup.

Chapter 6 is the summary and outlook, which summarizes the work of the full text, and looks forward to the deficiencies.

Chapter 2

Literature Review

2.1 Question and Answering System

The intelligent question answering system has a history of seventy years. The early intelligent question answering system, for example, Baseball (Green et al., 1961) and Lunar (Woods, 1973), is mostly designed for specific fields and the amount of data is also very limited, which is not easy to expand. These intelligent question-answering systems, which were proposed in the 1950s and 1960s, would only accept specific forms of natural language questions. In addition, very little data can be trained by the intelligent question answering system. Therefore, open-domain question answering based on big data cannot be performed and is not widely used.

After entering the 1990s, due to the development of the Internet, a large number of Q&A pairs for training can be collected from the Internet. Especially the launch of TREC-QA evaluation has greatly promoted the development of intelligent question answering systems. Researchers trained and tested various question answering models on the corpus, and successively proposed methods based on logical reasoning (Kelly & Lin, 2007), template matching (Moldovan & Rus, 2001), machine learning (Soubbotin, 2001), data redundancy (Mendes & Coheur, 2013) and many others. At this stage, people mainly use information retrieval or shallow semantic

understanding technology to find answers from a large number of candidate sets to build an intelligent question answering system, so the search question-answering technology has made great development. However, there is a shortcoming in a search question and answer technology, where the answer must contain at least one character or word contained in the user's question. However, this is often not true in actual situations. Although shallow semantic understanding technology partially solves this problem, user questions are natural language. Natural language has natural complexity. Due to the above-mentioned shortcomings, search-style question and answer technology cannot really well solve the needs of users.

For a long time, the two most important factors hindering the development of intelligent question answering systems are the lack of high-quality data and powerful natural language processing technology. However, with the rise of Internet applications such as Wikipedia, Bing, Sogoupedia, etc. based on user-generated content, more and more high-quality data has been accumulated and obtained. Based on this, a large number of knowledge bases such as Freebase, YAGO, DBpedia (Bizer et al., 2009), etc. that have been carefully designed to be generated automatically or semi-automatically have been established. As for another problem, with the rise of statistical machine learning methods, great progress has been made in various sub-fields of natural language processing technology. It can be said that the two biggest problems hindering intelligent question answering systems are being gradually solved by researchers.

In recent years, the intelligent question answering system has made great development and progress. Many intelligent question answering system products have come out. For example, Watson, an intelligent question answering robot developed by IBM, defeated the contestants in the American quiz show “Jeopardy!”. Its Deep QA system integrates deep technology such as statistical machine learning, information extraction, knowledge base integration, and knowledge reasoning. Another successful example is Apple's Siri system and Microsoft's Cortana that have achieved great results in the iPhone and Windows 10 operating systems, respectively. We can see these robots not only provide emotional chatting functions but also professional functions such as personal secretary and intelligent customer service. The emergence of these intelligent systems indicates that intelligent question-answering technology is becoming mature, and it is expected that more functional robots will come out in the future and solve the various needs of users.

Despite the impressive achievements, the intelligent question answering system is far from being perfect. The intelligent question answering system covers a wide range of fields, among which the main key technologies are the extraction and representation of knowledge, the semantic understanding of user questions and the answering through knowledge inference. These areas require in-depth research before we can get a better intelligent question answering system. However, these fields all exist relatively and independently and adopt very different

methods, and these methods have their own bottlenecks. Therefore, the modern research of intelligent question answering systems basically revolves around these three aspects.

2.1.1 Automatic Information Extraction and Construction of Knowledge Graph

The realization of an intelligent question answering system requires a very strong and comprehensive knowledge as a foundation. Although the Internet currently has a large number of knowledge resources, most of these resources are unstructured knowledge, and there are many different structures, which makes it more difficult to extract knowledge. Finding a way to fuse these massive amounts of heterogeneous data together, understand and extract them, and convert them into a form that computers can process has become a challenge.

In fact, scientists have been committed to building a larger and more complete knowledge resource base. In the early days, most of the knowledge resource bases were professional domain knowledge resource bases constructed by experts in various fields. The advantage of manually constructing a knowledge resource library by experts is that the knowledge is of high quality, but the disadvantage is that it is time-consuming and laborious to build a large-scale knowledge resource library. Besides, when changing domains, it is necessary to manually construct a knowledge resource library in another domain, and the construction method of the knowledge resource library in each field and the structure of the completed knowledge resource library are generally inconsistent. The construction method is not universal and the structure of the knowledge resource library is not consistent, and it cannot produce an intelligent question

answering system in the general field. We can roughly divide the knowledge used in the intelligent question answering system into language knowledge and world knowledge.

Linguistic knowledge refers to the knowledge of semantic units, including word meaning information, upper and lower relations, etc. The vocabulary knowledge base containing knowledge about these languages includes English WordNet (Miller, 1995), FrameNet (Baker et al., 1998), etc. World knowledge is the organization and representation of entities in the real world and the facts that occur in entities. People have established a world knowledge base Cyc artificially. The common-sense knowledge base involves half a million concepts, 30,000 relationships, and millions of facts. It is by far the world's largest common-sense database created entirely by humans. But even so, it is far from meeting the demand for knowledge resources of intelligent question-answering systems in the development field.

In fact, most of the knowledge now exists in unstructured text data. In order to overcome the time-consuming and laborious difficulty of manually organizing the knowledge resource database, researchers hope to use powerful information extraction technology to automatically acquire large-scale knowledge from massive unstructured texts to build a large-scale knowledge resource database. In this process, we need to perform entity recognition on the text first, then classify and disambiguate the entities, and then extract relationships and events, so that we can build a knowledge resource library composed of entities, relationships and events.

At present, scientists generally use Wikipedia as a source of building knowledge resources, because it gathers the wisdom of the group and has a large number of high-quality data resources. Therefore, a lot of work directly or indirectly uses Wikipedia resources for knowledge extraction. The Max Planck Institute in Germany built a large-scale knowledge base category system YAGO by integrating Wikipedia and WordNet. It defines dozens of relationships to describe the relationships between entities. Other representative works include ReVerb (Yates et al., 2007), R2A2 (Sil & Lin, 2021), and OLLIE (Fader et al., 2011) of the Turing Laboratory of the University of Washington; Wanderlust (Etzioni et al., 2011), KrakeN (Wu & Weld, 2010), etc. of the DSIM group of the Technical University of Berlin, Germany. CMU's NELL (Mausam et al., 2012) system builds a massive-scale network knowledge base that can handle a variety of intelligent information needs by continuously extracting and mining knowledge from the Internet. The latest work includes DBpedia and Freebase, which automatically generate a structured knowledge repository based on Wikipedia.

Recently, the knowledge resource library automatically constructed based on Wikipedia is the most popular, because these knowledge resource libraries can continuously update their own resource library with the update of Wikipedia, and thus have received great attention from search engine giants. After Google acquired Freebase in 2010, it has been committed to building a huge Knowledge Graph of interconnected entities and their attributes, and from this it has strengthened Google's semantic search.

2.1.2 Reading Comprehension and End-to-end Intelligent Question Answering System

We not only need to build a strong knowledge resource base, but also need a deep understanding of the issues raised by people. People's problems are presented in natural language. What the question comprehension needs to do is to transform natural language into a formal language that the computer can understand. It is very difficult for computers to understand human language. This is also the core problem that Natural Language Processing must solve. There are two different ways of solving this problem, one is Semantic Parsing, and the other is based on information retrieval. The semantic analysis method is very consistent with people's intuition. It parses a question in the form of a natural language into a semantic expression according to the grammatical rules of the specific language. After obtaining the semantic expression, we can easily convert it into a database query language.

First of all, for semantic analysis methods, researchers have devised many methods to convert natural language question sentences to semantic expressions (Fan et al., 2012). The most commonly used method is to use the combinatorial category grammar CCG (Mausam et al., 2012) and the core of CCG is vocabulary. First, the vocabulary in the natural language question is mapped to the vocabulary in the semantic expression. In addition to vocabulary, CCG also combines vocabulary according to the specific grammatical rules to obtain the final semantic expression (Liang et al., 2011). However, most of the vocabularies that play an important role

in such methods as CCG are artificially generated, which limits the conversion and expansion of the field—if a domain conversion occurs, a batch of specific vocabulary in the new domain must be regenerated. This kind of method also has the inherent shortcoming of artificial generation —it takes a lot of manpower and time to generate a large-scale vocabulary. Therefore, automatic learning of this kind of vocabulary has become the focus of researchers' exploration (Kwiatkowski et al., 2011).

Another way to solve the problem of question comprehension is based on information retrieval. First, use English word segmentation, named entity recognition and other natural language processing tools to find the entities and keywords involved in the question. Then go to the knowledge resource database to search. For example, we only conduct simple fact-based questions and answers for Freebase (Bollacker et al., 2008). First, use the named entity tool to identify the entity in the question, which is easy to process because the expression of an entity is very limited. Then find out the relationship in the question. This step is relatively difficult because there are many ways to describe the same relationship in natural language. In practical applications, we can easily eliminate many irrelevant relationships using various methods. After obtaining the entities and relationships, you can finally find the answers easily in the Freebase Knowledge Resource Library. The method based on information retrieval is relatively mature, simple and practical, and does not require artificial vocabulary generation like CCG, so it avoids the shortcomings of CCG and other analytical methods that need to generate new vocabulary and need to generate artificial vocabulary. But, the disadvantage is that the method

based on information retrieval requires that the answer must contain at least one character or word in the question, so it is not as accurate as the semantic analysis method.

As can be seen from the above, the understanding of natural language questions is the core and most difficult part of the intelligent question answering system. At this stage, the problem to be solved is how to most accurately convert natural language into a form that can be represented and understood by computers. This is not only a problem that the intelligent question answering system needs to solve, but also one of the core problems that need to be solved in the field of artificial intelligence.

The best results can be achieved as deep learning methods continue to be validated in academia and industry. People apply end-to-end thinking in the field of intelligent question answering. (Sukhbaatar et al., 2015) directly match the question with the final answer in order to solve the complex and diverse characteristics of natural language questions. For example, questions formed by replacing synonyms with different forms or changing the order of words still have the same meaning in fact (Bordes et al., 2015). The idea of memory was added to the deep neural network, and knowledge can be read in and written out on an end-to-end basis (Lao et al., 2011). Through the above end-to-end thinking, the most difficult question understanding step has been bypassed. The deep neural network has played an important role in it, and this method also achieved results comparable to traditional methods.

2.1.3 Knowledge Reasoning

In the intelligent question answering system, not all questions can be answered directly by using the existing knowledge base, mainly because the knowledge coverage is still limited after all. However, we can find that there is actually a lot of implicit knowledge that we can use the extracted knowledge to reason and answer. For example, the knowledge base contains the attribute information of a person's "place of birth", but there is no attribute information of the person's "nationality". In fact, we can infer the "nationality" attribute information of this person from the place of birth, because a certain place must belong to a certain country. Therefore, on the computer, we need to express and learn similar reasoning knowledge. The knowledge reasoning task is to obtain implicit knowledge that is not in the knowledge base by reasoning knowledge.

The early knowledge reasoning methods mostly summarized the existing knowledge and learned the reasoning rules of symbolic logic. For example, the inference system PRA developed by CMU (Wong & Mooney, 2006) can use existing knowledge to infer knowledge that does not exist in the knowledge base. However, these inference methods based on the rich have failed to effectively consider the semantics of the rich themselves, and the number of inference rules increases exponentially with the number of relationships, so it is difficult to expand to large-scale knowledge resources.

In addition, knowledge-reasoning technology also has new ideas and methods because of the maturity of deep learning technology. A lot of work focuses on the representation learning of entities and relationships. By encoding the entities and relationships of the knowledge resource library under global conditions, the entities, concepts and relationships are expressed as vectors or matrices in low-dimensional space, and the task of knowledge inference is completed through numerical calculations in low-dimensional space. In addition, the memory-based end-to-end deep neural network technology has also studied the reasoning problems involved in the intelligent question answering system (Lao et al., 2011). Although the effect of this type of reasoning is still far from practical at present, it is worth continuing to study this method in-depth, especially the inference technology that integrates symbolic logic, representation learning and end-to-end deep neural network technology based on memory mechanisms.

2.2 Machine Reading Comprehension

The task of machine reading comprehension (MRC) mainly refers to allowing the machine to answer text-related questions based on the given text in order to measure the machine's ability to understand natural language. The origin of this task can be traced back to the 1970s, but limited by small-scale data sets and traditional rule-based methods, machine reading comprehension systems at that time could not meet the needs of practical applications. This situation changed in 2015, mainly due to the following facts. The first is the proposal of a Machine Reading Comprehension model (neural machine reading comprehension) based on

deep learning. This type of model is better at mining the contextual semantic information of text, and significantly improved its effect compared with traditional models. The second is the publication of a series of large-scale machine reading comprehension data sets, such as CNN & Daily Mail (Bordes, Weston, et al., 2014), SQuAD (Bordes, Chopra, et al., 2014), MS MARCO (Graves et al., 2014), etc. These data sets make it possible to train deep neural models, which can also test the effect of the model very well. Neural machine reading comprehension has gradually received more and more attention in recent years and has become a research hotspot in academia and industry.

A typical machine reading comprehension system generally includes four modules: embedded coding, feature extraction, article-question interaction and answer prediction.

- **Embedded coding:** This module converts the input natural language articles and questions into fixed-dimensional vectors for subsequent processing by the machine. In the early days, the commonly used methods were traditional word representation methods, such as one-hot representation and distributed word vectors. Context-based word representation methods pre-trained by large-scale corpora in the past two years have also been widely used, for instance ELMo, GPT, Bert (Liu et al., 2019), etc. At the same time in order to better represent information such as semantic syntax, the above-mentioned word vectors can sometimes be combined with language features

such as part-of-speech tags, named entities, and question types for more fine-grained representation.

- **Feature extraction:** The word vector representation of the article and question obtained through the embedded coding layer encoding is then passed to the feature extraction module to extract more context information. The commonly used neural network models in this module are Recurrent Neural Network (RNN), Convolutional Neural Network (CNN) and Transformer structure based on a multi-head self-attention mechanism (Vaswani et al., 2017).

- **Article-Question Interaction:** The machine can use the interactive information between the article and the question to figure out which parts of the article are more important for answering the question. In order to achieve this goal, the article-question interaction module often uses one-way or two-way attention mechanisms to emphasize the more relevant parts of the original text. At the same time, in order to dig deeper into the relationship between the article and the question, the interaction process between the two may, sometimes be performed multiple times, in order to simulate the behaviour of humans repeated reading during reading comprehension.

- **Answer prediction:** This module makes the final answer prediction based on the information accumulated by the aforementioned three modules. Since common

machine reading comprehension tasks can be classified, according to answer types, the implementation of this module is highly task-related.

Compared with traditional natural language processing tasks, the task of machine reading comprehension involves many aspects of information such as morphology, syntax, grammar, semantics and pragmatics. Natural language processing and comprehension techniques such as text representation, analysis, comprehension and reasoning need to be integrated. Formally speaking, solving the reading comprehension problem can be expressed as the process of seeking the optimal solution that maximizes the $P(A|Q,D)$

where: A represents a candidate's answer

Q represents the question itself

D represents a given document or contextual information.

In order for the machine to complete the task of reading comprehension, we must solve three important problems—the problem of representation, the problem of retrieval, and the problem of answer generation. The problem of representation is how we can transform natural language into a computer-readable representation, and be able to get the information needed to complete reading comprehension tasks from this representation. The retrieval problem is because the context of reading comprehension is relatively broad, so how to retrieve the part related to the

question and answer from the large-scale context information is an important question. The quality of search results will directly determine the quality of answers to reading comprehension tasks. The answer generation question is how to select or generate the required answer based on the question and the relevant context retrieved. The current solutions to this problem are mostly solved, by treating the reading comprehension problem as a matching problem or a scoring problem. For a given (D, Q, A) triplet

where: D represents the document

Q represents the question

A represents the answer

There are two solutions as follows. The first is to merge each (Q, A) pair into a hypothesis through some heuristic rules, and then calculate the matching situation between hypothesis and D. Comprehensively consider D and Q, then calculate the A that best matches the (D, Q) pair, and calculate the specific representation of A under D. So in general, the problem is transformed into a sorting problem that scores (D, Q, A) triples.

For the problem of machine reading comprehension, there are currently three solutions: methods based on traditional feature engineering, methods based on neural networks, and graph-matching algorithms based on deep-level semantic information.

Method based on traditional feature engineering

The core of the algorithm is to select different features, and then construct and learn a ternary scoring function $F(a, q, d)$ based on the selected features, and use the candidate answer a with the highest score as the answer to question q in document d. The more commonly used scoring functions are linear models and logarithmic models.

The problem with methods based on traditional feature engineering is that most traditional features are based on discrete string matching. Therefore, it is more difficult to solve the problem of expression diversity. Most feature engineering methods are based on window matching, and it is difficult to deal with the long-distance dependence between multiple sentences. Although, some scholars have recently proposed that a model based on a variety of different levels of windows can alleviate this problem, window or n-gram (Cui et al., 2016) is not the most effective semantic unit. There are problems such as lack of semantics (the lack of partial words that make the semantics complete) or noise (the introduction of words that have nothing to do with the main body's semantics).

Method based on Neural Network

In neural networks, various semantic units are represented as vectors in a continuous semantic space, which can effectively solve the problems of semantic sparsity and paraphrase. In this direction, the most representative one is the Memory Network (Cho et al., 2014). It includes four parts, input feature mapping, generalization layer, output feature layer and feedback layer.

Input feature mapping transforms the input layer (text sequence) into the feature space, which is similar to embedding. The generalization layer is to update the old and existing memory according to the new input. The output feature layer is to output one or more feature vectors as feedback based on current memory. The Feedback layer converts the output vector of the feature space into feedback information in the real space, such as a word. This model can be used for QA problems very well, as long as the BOW of the document is used as a memory layer. Use Question as the latest input, and then match the most relevant sentences in the original text through the network to get the corresponding output. Since the model retains the information of each input semantic unit (sentence or fragment), the long-distance dependency problem can be well handled in the output feature layer. At the same time, due to the multi-layer structure used in the output feature layer, the model can handle the reasoning problem between multiple sentences.

However, there is an argmax operation in this model, which causes the model of Weston (Cho et al., 2014) and others to use a fully supervised algorithm. In the training process, not only the answer to each question needs be marked, but also the key sentence that matches the answer. In order to solve this problem, (Clark & Gardner, 2018) proposed an end-to-end model. In this model, the output layer of the original model is modified, and the original causal matching mechanism is replaced with an attention-based soft matching mechanism.

Graph-matching algorithms based on deep-level semantic information

(Devlin et al., 2019) proposed a graph matching algorithm. First, through a method similar to semantic role annotation convert the entire article into a graph structure and then combine the question and answer (called a query). Finally, consider the matching degree between the graph structure of the article and the graph structure of the query.

The construction of question answering systems in specific fields usually uses knowledge graphs. The basic composition of the knowledge graph is the triplet. The tool system of knowledge graph includes knowledge graph editing tools and knowledge graph query tools. The biggest disadvantage of the knowledge graph question and answer is that it needs to build a knowledge graph. The question answering system is not transferable, and different fields need to organize different knowledge, which is the largest part of the workload. Moreover, the compilation of the knowledge graph requires expert knowledge, and this work is a difficult task and is handed over to the programmer or to the business side. The advantage of the knowledge graph question answering lies in its simplicity, directness, strong interpretability, and high-level logic queries.

2.3 Question and Answer System Based on Machine Reading Comprehension

The question answering system based on reading comprehension can be considered to a certain extent to solve the problems of knowledge graph question answering. The knowledge stored in

the knowledge graph is highly structured, which cannot be matched with real life, and many articles are difficult to extract the knowledge structure. The knowledge storage form of a question answering system based on reading comprehension is unstructured text fragments. There are two types of question answering systems based on reading comprehension: extractive and retrieval.

In recent years, when machine reading comprehension has been applied to question answering systems, people have become increasingly interested in generating questions for reading comprehension. (Hu et al., 2018) proposed that neural models based on the encoder-decoder framework can generate better problems than rule-based systems. (Huang et al., 2018) proposed that to generate answer-specific questions, one can simply indicate the location of the answer with an additional function in the context. (Jia & Liang, 2017) introduced latent variables used to capture variability and observed variables used to control problem types.

Data enhancement of machine reading comprehension has been tried many times to enhance training data to improve machine reading comprehension ability. According to the type of augmented data, these tasks can be categorized as an external data source, paragraph or question. (Yates et al., 2007) fine-tuned the BERT on the SQuAD dataset together with another dataset TriviaQA. (Erk & Smith, 2016) suggested generating questions based on unlabeled text for semi-supervised question answering. (Klein et al., 2017) proposed a rule-based system to generate multiple-choice questions with candidate options on paragraphs.

2.4 Conclusion

Machine reading comprehension can solve the last mile problem of traditional search-based question and answer system, which accurately locates the answer. The traditional search question and answer retrieves a number of candidate documents from a large set of documents after the user enters a question. Next, paragraph segmentation and sorting of these candidate documents is done. Finally, the answer is fed directly back to the user in units of paragraphs. However, a series of related redundant information is usually contained in such paragraphs. Using the technology of machine reading comprehension can get more direct, concise and accurate answers. Therefore, this research provides a certain reference basis and theoretical value for solutions to related problems of intelligence.

Chapter 3

Methods of Information Retrieval and Machine Reading Comprehension

This section mainly introduces information retrieval methods, commonly used structures for machine reading comprehension and evaluation methods.

3.1 Information Retrieval

In order to reduce the use of computing power for Machine Reading Comprehension model and the consumption of reasoning time by irrelevant documents, it is necessary to carry out information retrieval technology before the multi-document machine reading comprehension task. At present, commonly used algorithms for information retrieval include TF-IDF (Paik, 2013), Bm25 (Robertson & Zaragoza, 2009) and PageRank (Gleich, 2015), etc. Some researchers also use machine learning to perform information retrieval on the encoded text vector using a sorting method, such as SVMrank.

With the rise of deep learning in recent years, Google has proposed information retrieval technology based on Bert's deep learning model and applied it to actual scenarios. Compared with supervised learning, TF-IDF and Bm25 can be quickly started without manual labeling. It has a relatively good document recall rate, is currently the most commonly used technology in

search engines, and is usually used for coarse text recall in multi-document machine reading and comprehension tasks.

3.1.1 Best Matching 25 Algorithm

The Best Matching 25 Algorithm (Bm25) was released in 1994 and was named the 25th iteration of the adjustment correlation calculation. It calculates the relevance of sentences and documents.

The principle of Bm25 is very simple. First of all, the input *Query* is parsed to get the morpheme qi. Secondly, calculate the weighted sum of the correlation scores of each morpheme qi and the candidate result is added in *D*. Finally, get the correlation score between $Query$ and D. The formula of the Bm25 algorithm is shown in equation (3.1).

$$Score\ (Q, d) = \sum_i^n W_i R(q_i, d) \qquad (3.1)$$

where: Q represents the query question $Query$

q_i means every word after Q participle

d means a document

W_i represents the weight of q_i

$R(q_i, d)$ represents the relevance score of the word q_i and the document d

In fact, Bm25 can be regarded as an improvement of TF-IDF, in which W_i is an improvement to IDF, and $R(q_i, d)$ is an improvement to TF. Thus, the weighted sum of the relevance of each word in the query to the document is the relevance score of the query to the entire document, which is the Bm25 score.

In terms of weight W_i, there are many weighting methods to judge the relevance of a word in a document. The more commonly used method is Inverse Document Frequency (IDF). The formula of IDF is shown in equation (3.2).

$$IDFq_i = log \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} \qquad (3.2)$$

where: N is the total number of documents in the index

$n(q_i)$ is the number of documents containing q_i

From the formula of IDF, we can see that the larger the $n(q_i)$, the larger the denominator and the smaller the numerator, which means the corresponding IDF value is smaller. This is because if a word appears in multiple documents and it exists in any sentence, then it can be explained to a certain extent that the word should be a commonly used word, and it cannot reflect the particularity of the sentence. Therefore, the IDF value is smaller. Substituting the IDF value into the BM25 algorithm as the weight, that is to say, the more a word appears in the document *d*, the correlation score calculated by it and document *d* should be given a smaller weight.

Besides, the score $R(q_i, d)$ between morpheme q_i and document d also influences the Bm25 score. The formula of $R(q_i, d)$ is shown in equation (3.3) and equation (3.4).

$$R(q_i, d) = \frac{f_i \cdot (k_1+1)}{f_i+K} \cdot \frac{qf_i \cdot (k_2+1)}{qf_i+k_2} \qquad (3.3)$$

$$K = k_1 \cdot (1 - b + b \cdot \frac{dl}{avgdl}) \qquad (3.4)$$

where: k_1, k_2, and b are free parameters, and usually $k_1 = 2$, $k_2 = 1$, and $b = 0.75$

qf_i is the frequency of occurrence of q_i in query

f_i is the frequency of occurrence of q_i in d

dl is the length of the document d

$avgdl$ is the average length of all documents

In most cases, q_i will only appear once in Query, where qf_i = 1. Therefore, the formula can be simplified to equation (3.5).

$$R(q_i, d) = \frac{f_i \cdot (k_1+1)}{f_i+K} \qquad (3.5)$$

It can be seen from the formula of K that the parameter b is the effect of adjusting the document length on the relevance. If b is larger, the impact of the document length on the relevance score is greater, and vice versa. The longer the relative length of document d, the larger the value of K, that is, the larger the denominator of R, and the smaller the R

correlation score. It can be understood here that when the document is longer, the more words it contains, the more likely it is that the word q_i is included accordingly. However, although the possibility is greater if the frequency f_i of q_i is the same, the correlation between long documents and q_i should be weaker than the correlation between short documents and q_i. To sum up, the formula for the correlation score of the BM25 algorithm can be summarized as equation (3.6).

$$Score\ (Q,d) = \sum_i^n IDF(q_i) \cdot \frac{f_i \cdot (k_1+1)}{f_i + k_1 \cdot (1-b+b \cdot \frac{dl}{avgdl})} \qquad (3.6)$$

3.1.2 TF-IDF (Term Frequency – Inverse Document Frequency)

TF-IDF (Term Frequency – Inverse document frequency) is a commonly used weighting technique for text mining and information retrieval. The core idea is that the more a word appears in an article and the less it appears in all documents, the more it can represent the article. For example, we suppose that we want to find articles related to "reading comprehension". Those articles with "reading comprehension" only appeared once in the content, and they may be talking about natural language processing. Incidentally, they mentioned "machine reading comprehension" and those articles where it appeared two or three times are probably discussing related technologies of "machine reading comprehension". Through intuition, we can draw a judgment that the more the keyword appears, the higher the matching degree between the document and the keyword.

However, "principle" is a very general term, and "machine reading comprehension" is a specialized term. Intuition tells us that the word "machine reading comprehension" is more important to our search than "principle". Therefore, when considering the word frequency of a document, it is also necessary to consider whether the word is a universal word, that is, the "inverse document frequency" of the current vocabulary. Inverse document frequency reflects that the importance of a word is proportional to the number of times it appears in the document, and at the same time, it is inversely proportional to the frequency of its appearance in the corpus. The proposal of TF-IDF uses this idea and considers more situations to improve it.

TF is the word frequency of the vocabulary word. In order to reduce the word frequency of long and short texts is often low, and the word frequency of long documents is too high, TF_{score} is the word frequency divided by the length of the document to normalize the word frequency. The formula is shown in equation (3.7).

$$TF_{score} = \frac{The\ number\ of\ occurrences\ of\ the\ term\ w\ in\ a\ certain\ category}{The\ number\ of\ entries\ in\ this\ category} \tag{3.7}$$

IDF is inverse document frequency. The core idea is that if there are fewer documents containing the term t, the larger the IDF, which means that the term has a good ability to distinguish categories. The reason for adding 1 to the denominator in the formula is to prevent the denominator from being zero. That is, all documents do not contain the special case of the word. The introduction of log allows the result to take the logarithm of the obtained value to

converge the maximum number of inverse documents to e to prevent IDF_{score} from overly affecting the overall score. The formula is shown in equation (3.8).

$$IDF_{score} = \frac{The\ total\ number\ of\ documents\ in\ the\ corpus}{The\ number\ of\ documents\ containing\ the\ term\ w+1} \quad (3.8)$$

3.2 Machine Reading Comprehension

Before the Bert and other pre-training structures based on Transformer were proposed, commonly used machine reading comprehension structures such as BiDAF and QANET and other classic structures usually included four modules: word vector module, feature extraction module, text interaction module, and answer prediction module. The design of coding layer information and the information interaction between text and questions play a vital role in reading comprehension performance.

3.2.1 BiDAF

BiDAF (Bi-Directional Attention Flow) is a machine reading comprehension structure article published by (Seo et al., 2016) at the ICLR conference in 2016. The contribution of this article in the field of machine reading comprehension is very significant, and the two-way attention mechanism it proposes has become a part of the basic architecture of a general encoder or inference unit. BiDAF does not summarize the text as a fixed-length vector but flows the vector to reduce the loss of the weighted sum of early information. In addition, at each moment, only the query and the context paragraph of the current moment are calculated, and it does not

directly rely on the attention of the previous moment. This makes the subsequent attention calculations not be affected by the previous incorrect attention information, and there is a layer of interaction in the structure. It only has the relevance of the article and the question. The attention information in the query-to-context (Q2C) and context-to-query (C2Q) directions is calculated. The constructed C2Q and Q2C mechanisms can actually complement each other.

The highlight of this model is the proposal of a two-way attention mechanism. This two-way attention mechanism acts as an encoder or a link in the inference unit in QA tasks and has a greater impact on subsequent performance.

3.2.2 QANET

QANet is a machine reading and comprehension structure article published by Google Brain at the ICLR conference in 2018 (Yu et al., 2018). This article proposes a new framework that uses CNN (Convolutional Neural Network) and self-attention instead of traditional RNN (Recurrent Neural Network) to build a machine reading comprehension model, which greatly improves the speed of model training and inference. At the same time, the paper uses the method of turning back to increase the data. At that time, the structure achieved scores close to the highest scores in the SQuAD dataset and riviaQA. It mainly includes Input Embedding Layer, Embedding Encoder Layer, Context Query Attention Layer, Model Encoder Layer and Output Layer. Because the model is trained faster, more data is used for training in the article.

The general idea is to translate the existing corpus into another language and then translate it back.

3.2.3 Bio-Bert

Biomedical text mining is becoming more and more important with the rapid growth of the amount of biomedical literature. With the advancement of natural language processing (NLP), the extraction of valuable information from biomedical literature has become popular among researchers, and deep learning has facilitated the development of effective biomedical text mining models. Bio-Bert (Lee et al., 2019) is the first domain-specific BERT (Devlin et al., 2019) model based on a biomedical corpus. Under the original BERT model, Bio-Bert has done professional data collection and training on the biomedical corpus, so that it can improve the named entity recognition, relation extraction and question and answer content in the professional field.

As a general-purpose language representation model, BERT is pre-trained on English Wikipedia and BooksCorpus. However, biomedical domain texts contain quite a few domain-specific proper nouns and terms. Therefore, Bio-Bert is pre-trained using PubMed abstracts (PubMed) and PubMed Central full-text articles (PMC). Specially, in PubMed, Bio-Bert used 4.5 billion words for pre-training and in PMC, Bio-Bert is pre-trained with 13.5 billion words. With a nearly identical architecture across tasks, Bio-Bert outperforms BERT by a large margin on many biomedical text mining tasks after pre-training on biomedical corpora.

There is no need for complex structural design such as BIDAF and QANET and cumbersome Pipeline-style interactions, as shown in Figure 3.1. Just add a simple Answer Predict module after Bert to extract the starting position of the answer, you can do the reading comprehension task and often achieve particularly good results.

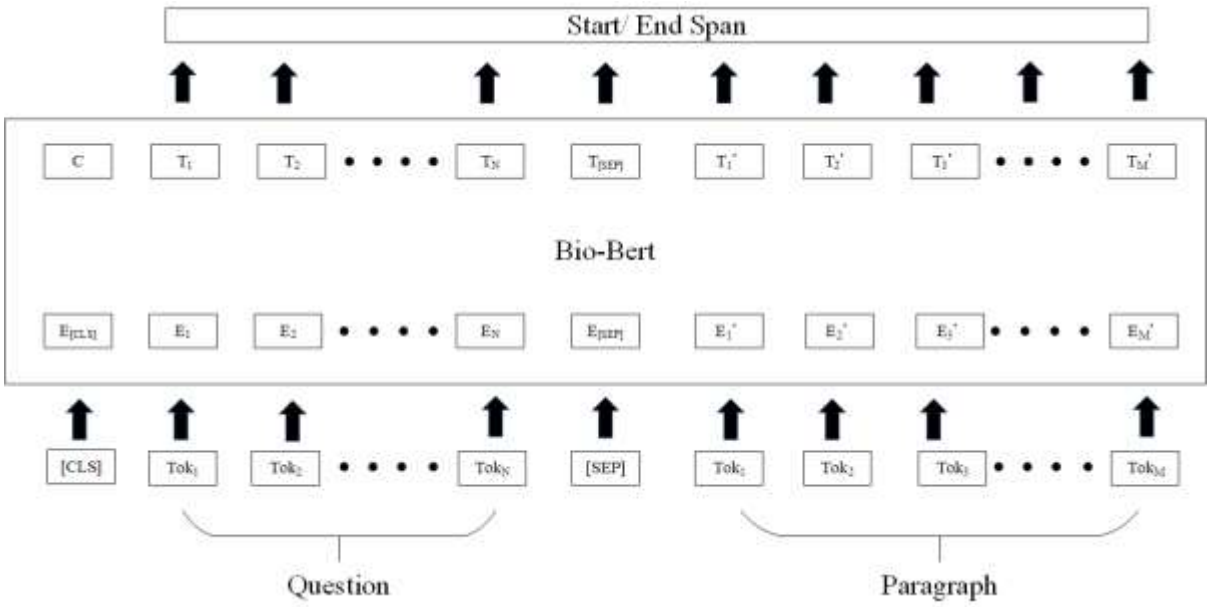

Figure 3.1. The Machine Reading Comprehension Task of Bio-Bert

The principle of extracting the answer is to learn two vectors Start and End. These two vectors correspond to the probability that the token is the beginning of the answer and the probability that the answer is the end. In fact, to put it bluntly, each word judges whether it is the beginning of the answer or the end of the answer. Specifically, it is to learn two vectors, S and E respectively, which correspond to the probability that the word element is the beginning of the answer and the probability of the end of the answer. The formula is shown in equation (3.9) and equation (3.10). Specifically, for each token, the multiplication of each token S and T_i in

the second sentence, and applying SoftMax function, will give the probability that each token in this segment is the beginning of the answer. T_i represents the last hidden vector corresponding to the input token. In the same way, the probability of being the end of the answer is calculated. The formula is shown in equation (3.11).

$$S \in R^H \quad (3.9)$$

$$E \in R^H \quad (3.10)$$

$$P_i = \frac{e^{S \cdot T_i}}{\sum_j e^{S \cdot T_i}} \quad (3.11)$$

Bio-Bert's model structure consists of input characters, position information, and segmentation information after splicing and outputting to a multi-layer Transformer for semantic information interaction. Two pre-training strategies are used namely, *Masked Language Model* and *Next Sentence Predict*. *Masked Language Model* uses random masks for massive unsupervised text data and *Next Sentence Predict* constructs upper and lower sentences to construct labels for training. The multi-layer Transformer structure in Bert is converged through a large amount of data. The structure of this is actually in line with the word encoding, semantic interaction and other information of the common framework for “machine reading comprehension” tasks.

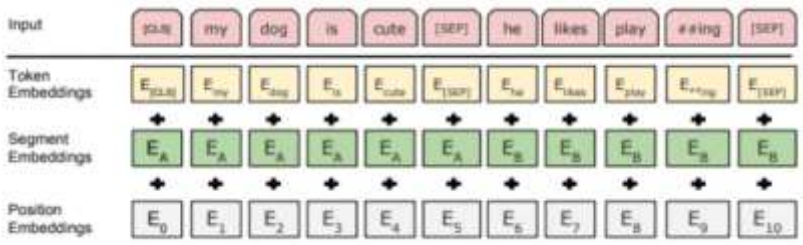

Figure 3.2. The Embedding Structure of Bert (Devlin et al., 2019)

As shown in Figure 3.2, when using Bio-Bert for the reading comprehension task, the questions and paragraphs are input into the model with [CLS] + question + [SEP] + paragraph + [SEP] token as the data stream. The model performs Token Embeddings, Segment Embeddings, and Position Embeddings encoding on all characters to obtain the character vector, the block vector and the position vector of the text. Then, by adding the corresponding positions of its dimensions to fuse the vocabulary and position information, the encoding layer vector $E \in R^{S \times H}$ corresponding to the current input is obtained, where S is the length of the input question, text and special characters and H is the size of the hidden layer. After this, the vectors $E \in R^{S \times H}$ are input into the multi-layer Transformer for semantic interaction, as shown in Figure 3.3.

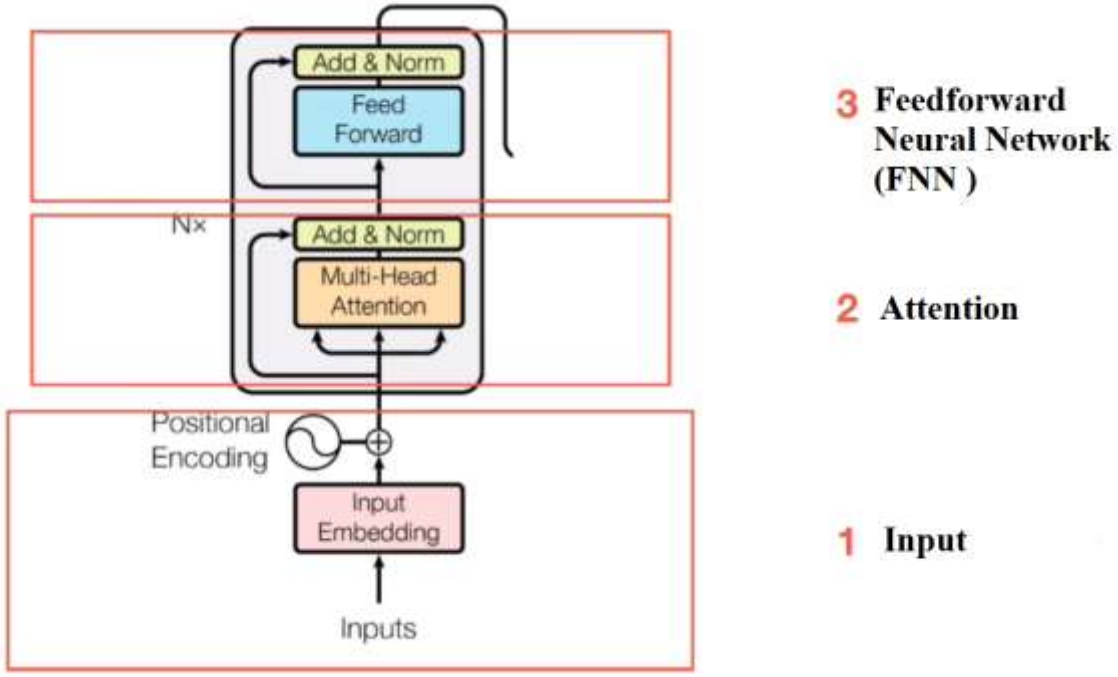

Figure 3.3. The Encoder Structure of Transformer (Vaswani et al., 2017)

The Transformer structure uses the global self-attention mechanism of self-attention to semantically interact with the input question and text. At this time, the input encoding layer vector $E \in R^{S \times H}$ enters the Transformer structure, then performs multi-head attention and then uses the fully connected layer to map the information of the question and text interaction. The Transformer used in Bert reuses the attention mechanism. For the obtained encoding layer vector E, first use three different fully connected layers to map it into Q, K, V namely query, key, value vector, and calculate the context dependency between each input character by the following formula in Figure 3.4. In the formula, dividing by $\sqrt{d_k}$ scales the result of the dot product of Q and K to prevent the value from overflowing.

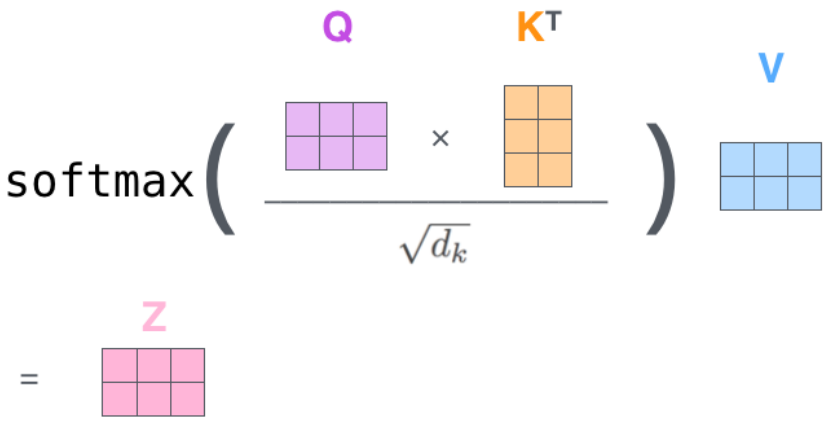

Figure 3.4. Self-attention Calculation in Matrix Form (Alammar, 2018)

In the text, the Attention mechanism is called Scaled Dot-Product Attention, that is, the dot product attention ratio. Similar to the bidirectional attention mechanism proposed by BIDAF, the Attention mechanism learns the attention of each word of the model to the context by

calculating the dot product of the query vector and the key vector, and then applies the SoftMax, and obtains the weight of the median value through the attention of the value.

In addition, during the experiment, Google researchers also found that it is effective to use different learned Q, K, and V for attention to obtain multi-dimensional information. To this end, a Multi-Head Attention mechanism is proposed. Specifically, multiple Q, K, V are set for the same input encoding vector E to perform Scaled Dot-Product Attention calculation, and then the results obtained by multi-head attention are spliced and fused in the last dimension. The design of the multi-head experiment is similar to the more comprehensive intuition of viewing something from multiple dimensions in daily life. Through the design of the multi-head mechanism, the model can pay more attention to the context and deep semantic information between the input texts. The overall steps are shown in Figure 3.5. The Transformer structure is extremely difficult to converge during the training process due to the huge number of parameter designs. Bert uses massive amounts of data for unsupervised pre-training tasks to converge the multi-layered Transformer, allowing Bert's attention mechanism with a large number of parameters to express deeper questions and interact with textual information, which also promotes the success of reading comprehension tasks.

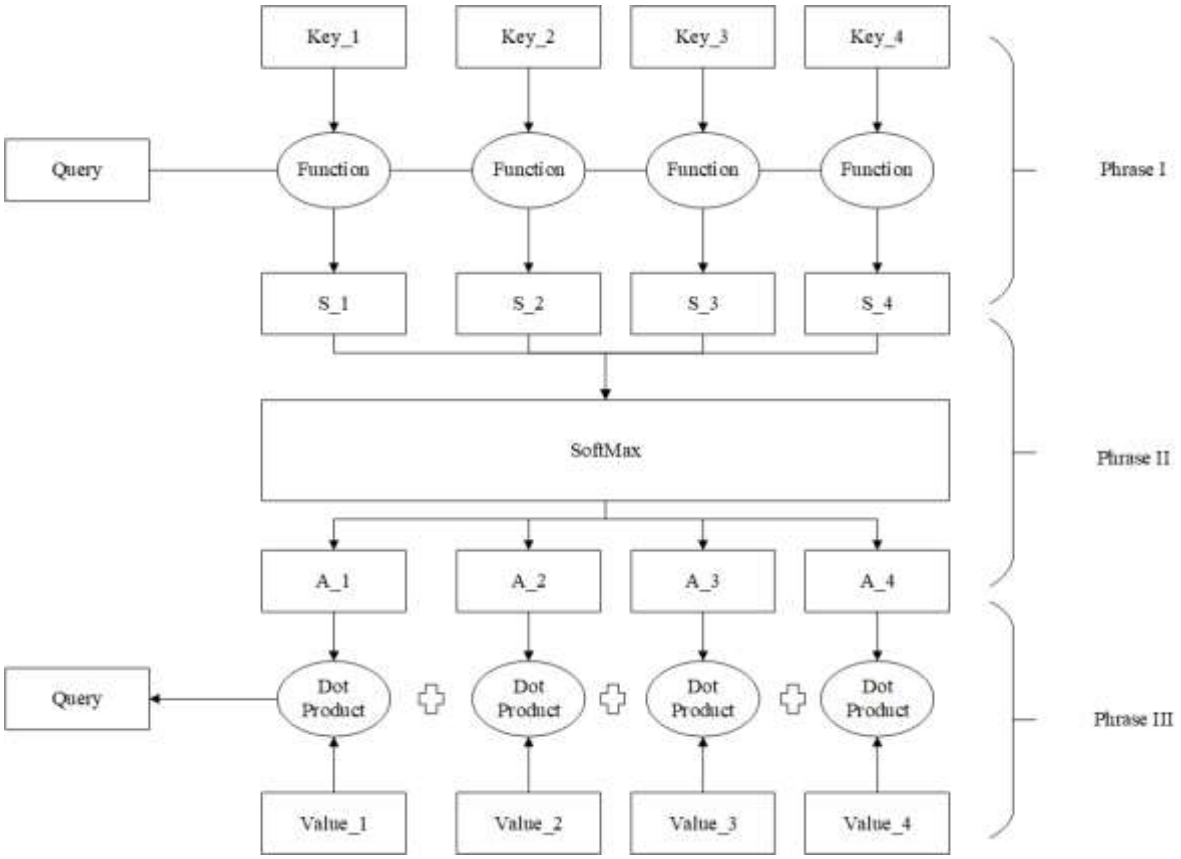

Figure 3.5. The Steps of Self-Attention

Although Bert uses multiple layers of Transformers to interact with contextual information, the inherent weakness of Transformer for position information, direction information and local information makes it possible to directly fine-tune Bert for reading comprehension tasks. There is still a certain room for performance improvement.

3.3 Downstream Structure

3.3.1 Performer

The O(n^2) complexity of the Attention mechanism in the Transformer structure affects the calculation speed of the model and the length of the input sequence under the constraints of

computing power. These constraints make it difficult for the model to be implemented in real application scenarios, and lead to a large number of improvements to this work. As shown in Figure 3.4 below, Tay et al., 2020 and others summarized the improved methods of Transformer. Among them, Performer (Choromanski et al., 2020) proposed by Google in 2020 has slightly reduced performance while increasing the speed, and takes up less memory.

Figure 3.6. Transformer improves the structure speed performance video memory comparison (the size of the circle represents the video memory usage) (Ruder, 2021)

The idea to change this complexity is linearized attention computation. Compared with Transformer, Performer can make the model forward calculation faster, while allowing the model to process longer input sequences, and its attention mechanism can be linearly scaled.

Arbitrary attention matrices can be efficiently approximated by random features. The new mechanism to achieve this uses positive random features, which are positive nonlinear functions of the original query and key. This avoids instability during training and enables a more accurate approximation to regular SoftMax attention.

3.4 Evaluation Methods

3.4.1 Exact Match

Exact Matching Score (EM) is often used for fragment extraction tasks, which can evaluate whether the predicted answer segment exactly matches the standard real sequence. Out of m questions, if n questions are answered correctly, then the EM score is shown in equation (3.13).

$$EM = \frac{n}{m} \qquad (3.13)$$

3.4.2 Rouge-L

Rouge is often used to evaluate answers with high degrees of freedom such as automatic summarization and machine translation (Lin, 2004). It measures the "similarity" between the automatically generated answer and the reference answer by comparing the automatically generated answer with the real answer and calculating the corresponding score. The definition in the Rouge-N indicator is shown in equation (3.14).

$$Rouge - N = \frac{\sum S \in \{ReferenceSummaries\} \sum gram_N \in SCount_{match}(gram_N)}{\sum S \in \{ReferenceSummaries\} \sum gram_N \in SCount(gram_N)} \qquad (3.14)$$

In the formula, the denominator is the number of n-grams, and the numerator is the number of n-grams shared by the correct answer and the model inference answer. For example, the Rouge-1 and Rouge-2 scores of the standard answer "I like natural language processing" and the model inference answer "I also like natural language processing" are shown in the Table 3.1. The numerator is the number of 1-grams in which both the model inference answer and the standard answer appear, and the denominator is the number of 1-grams in the standard answer. Between the precision rate and the recall rate, the recall rate is more concerned at this time, so the denominator selects the true label 1-gram instead of the 1-gram of the inference answer. This is also the same as the formula for ROUGN-N above.

Table 3. 1 Rouge Evaluation Example

Standard Answer (SA)	I like natural language processing
Model Answer (MA)	I also like natural language processing
1-gram	SA: I | like | natural | language | processing
	MA: I | also | like | natural | language | processing
2-gram	SA: I like | like natural | natural language | language processing
	MA: I also | also like | like natural | natural language | language processing

ROUGE-1	$\frac{5}{6}$ (I, like, natural, language, processing)
ROUGE-2	$\frac{4}{5}$ (I like, like natural, natural language, language processing)

Rouge-L is also often used for free Q&A evaluations. L in Rouge-L refers to LCS, which is the longest common subsequence. Its calculation method is shown in equations (3.15), (3.16), and (3.17).

$$P_{lcs} = \frac{LCS(X,Y)}{n} \qquad (3.15)$$

$$R_{lcs} = \frac{LCS(X,Y)}{m} \qquad (3.16)$$

$$F_{lcs} = \frac{(1+\beta^2)\times P_{lcs}\times R_{lcs}}{R_{lcs}+\beta^2\times P_{lcs}} \qquad (3.17)$$

where $LCS(X, Y)$ is the length of the longest common subsequence of the input X and Y, representing the length of the standard answer and the predicted answer respectively

P_{lcs} is the precision of the answer

R_{lcs} is the recall of the answer

Generally speaking, β is set to a large number, so Rouge-L almost only considers R_{lcs}, which corresponds to the recall. An advantage of using LCS is that it does not require sequential matching and reflects sentence-level word order for sequential matching. Since it automatically

contains the longest sequential generic n-gram, there is no need for a predefined n-gram length. However, since only one longest subsequence is calculated, the final Rouge-L score also ignores the influence of other alternative longest subsequences and shorter subsequences.

3.5 Conclusion

This chapter introduces the common structures of information retrieval and machine reading comprehension, as well as the work of text encoding and feature extraction. In addition, the commonly used evaluation indicators for machine reading comprehension were listed.

Chapter 4

A Two-stage Information Retrieval

The multi-document reading comprehension structure proposed in this thesis is divided into two parts, namely information retrieval and machine reading comprehension. For a problem, first of all, use the Bm25 method to roughly recall candidate documents. Secondly, design algorithms to finely recall and reorder related paragraphs to obtain answers to paragraphs that are highly relevant to the question. Through these passages, perform reading comprehension tasks, and finally, find concise and short answers. Figure 4.1 shows the steps. This chapter will introduce in detail the information retrieval strategy and structure used in the multi-document machine reading comprehension program proposed in this thesis.

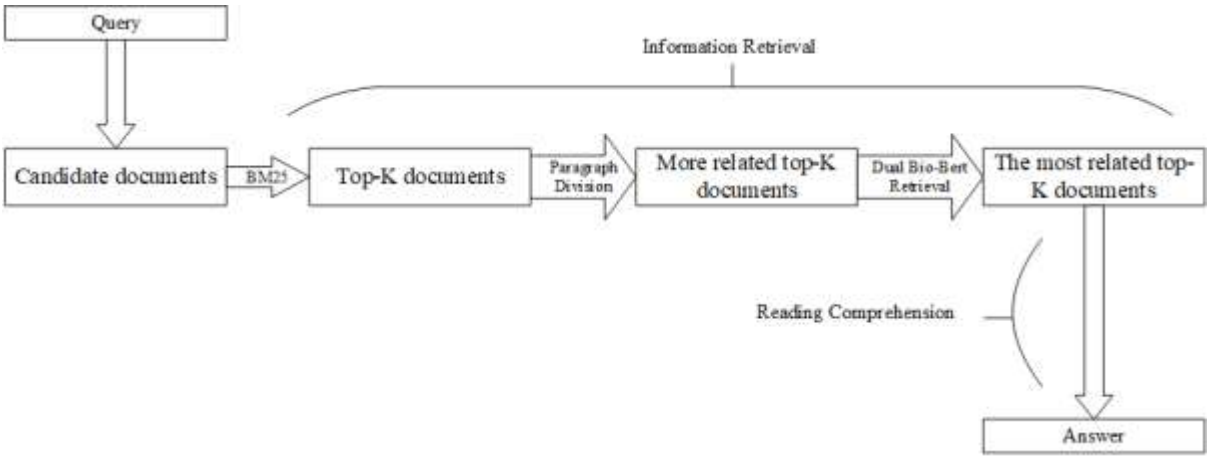

Figure 4.1. Multi-document machine reading comprehension steps

4.1 Dual Bio-Bert Retrieval

The retrieval method proposed in this article is in the form of a pipeline. The first step is to sort the collection of related documents from the massive document collection, and the second step uses a deep network that integrates semantic information to reorder to improve retrieval accuracy.

4.1.1 1st Recall—BM25

Similar to most searches, the first stage order used in this thesis is Bm25. The calculation of the Bm25 score of a document D_i in a question Q and candidate document set D is shown in the pseudo-code of Table 4.1. The inverse document rate (IDF) is used as the weight of the word W_i, and the word frequency information TF is the relevance score of the question and the document $R(q_i, d)$. Then calculate the relevance scores of the word segmentation q_i of the question and the candidate document d respectively. Lastly, sort the scores, and select the first recalled documents as the first-stage related documents D_{stage1}.

Table 4.1 Pseudo-code for the first stage of information retrieval

Input: Candidate document set D, question Q, specify hyperparameters k_1, k_2, b

Output: Related documents D_{stage1}

1. For the question Q, divide the words and remove the stop words, and get set$Q = \{q_0, q_1, \ldots, q_n\}$;

2. Calculate the average length of all documents

$$avgdl = \frac{\sum_{k=0}^{D.length} D_k.length()}{D.length()};$$

3. $score\_array = []$

4. $for\ (k = 0; k < D.length(); k++)do$

5. $score_k = 0;$

6. $for\ (i = 0; i < set_q.length(); i++)do$

7. Count the frequency of occurrence of all words in the document set in set_q

$$n(q_i) = \sum_{k=0}^{D.length} \begin{cases} 1, if\ \ q_i\ \ in\ \ D_k \\ 0, else \end{cases};$$

8. Calculate the weight w_i of each word

$$w_i = IDF(q_i) = log\frac{nums(D) - n(q_i) + 0.5}{n(q_i) + 0.5};$$

9. Calculate the frequency of appearance q_i and in the document D_k

$$f_i = Counts(D_k, q_i);$$

10. Calculate the correlation score between q_i and the document D_k

$$R(q_i, D_k) = \frac{f_i \cdot (k_1 + 1)}{f_i + k_1 \cdot (1 - b + b \cdot \frac{D_k.length()}{avgdl})}$$

11. $score_k = score_k + w_i \cdot R(q_i, D_k)$

12. Add $score_k$ to $score_{array}$;

13. $D_{stage1} = D[Rank_{100}(score_{array}).index()]$

When BM25 is used for retrieval and recall, the word segmentation weight of the current problem w_i and the relevance of the problem to the document (Q, D) are considered and measure the matching degree of the two texts through statistical word frequency information. The word frequency of the document or question indicates the importance of a word in the context, and the inverse document frequency indicates the global importance of the term in the document library.

4.1.2 2nd Recall—Dual Bert Retrieval

The BM25 method recalls more relevant documents by comprehensively measuring the frequency of vocabulary in the question and the overall important performance of the document. However, in the actual scene, the frequency of a word does not necessarily indicate whether a word is important or closer to the expression of the text. If only word frequency information is used as document recall, context semantics will be ignored. It is often difficult to guarantee a reliable recall rate when considering performance and limiting the amount of recalled document data. In this research, the strategy designed is shown in Figure 4.2. Reorder the candidate documents obtained by the BM25 recall of the questions entered by the user and proposed the second stage recall based on the deep semantic information of the pre-training model.

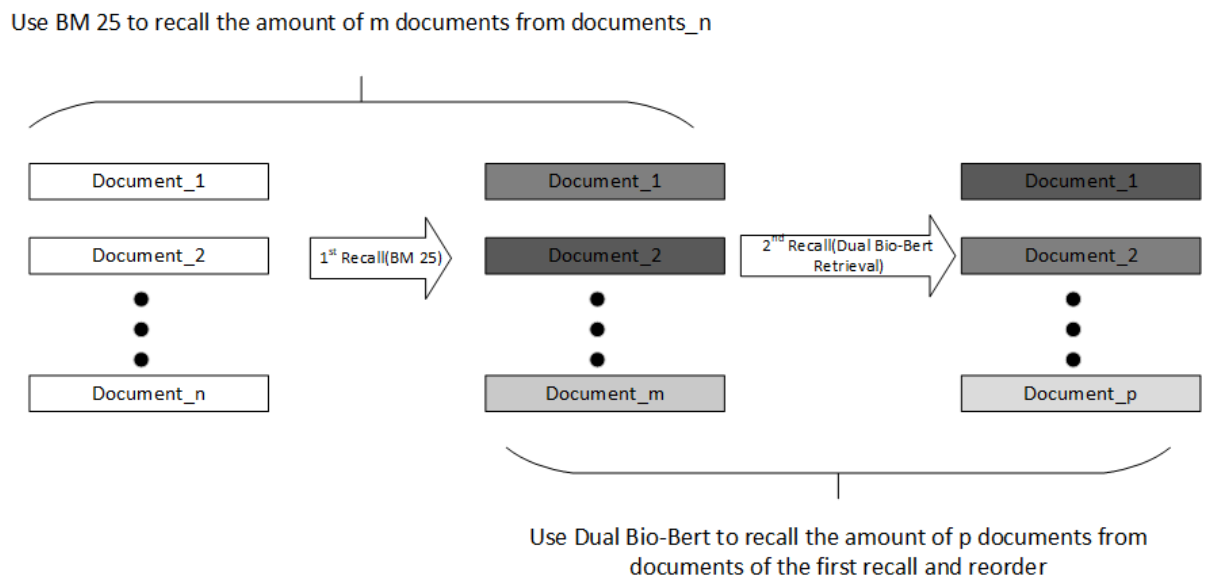

Figure 4.2. The Recall methods

After Bert and other Transformer-based pre-training structures were proposed, some scholars also tried to use Bert for information retrieval. A commonly used solution is to use Bert to do a binary classification task for input questions and paragraphs as to whether they have answers so that the model has the ability to determine the relevance between questions and paragraphs. The relevant paragraphs are determined by sorting the output probability of the two classification results of the question. Although this kind of method introduces real semantic information and has a higher performance, but because the calculation amount of Transformer and the calculation amount of input text is $O(n^2)$, therefore, it has problems such as slow online reasoning speed and high resource consumption.

Besides, some scholars have also tried to independently encode questions and documents in the Bert model in an unsupervised manner. During inference, the candidate paragraphs are vectorized and stored offline, and the document is retrieved by calculating the similarity between the question and the paragraph encoding. However, because the output vector of the Bert pre-training structure is the task of the Next Sentence Predict and Masked Language Model, the CLS vector used to measure the entire sentence information is often an implicit representation of whether the sentence is the upper and lower sentence before being fine-tuned downstream. Therefore, generally speaking, there are many gaps in the effect of the latest methods.

In order to increase the speed of secondary retrieval while ensuring the high recall rate of paragraphs, this thesis proposes the Dual Bio-Bert Retrieval structure shown in Figure 4.3. The structure is divided into two stages. In the training phase, two Bio-Berts are used to extract feature vectors for the question Q and the candidate paragraph P_i. After that, get the deep semantic representation of the question and the paragraph E_Q and the paragraph representation E_P respectively. In order to accelerate the model reasoning at the same time for the relevance of the learning model to the question and candidate paragraphs, the Performer structure mentioned above is used for semantic interaction between E_Q and E_P($E_{interaction} = Performer([E_Q; E_P])$). After this interaction vector is fully connected to a neuron, the Sigmoid activation function is used to obtain the output probability of the model p. At this point, the cross-entropy used by the model output and the real label is optimized through the

gradient descent and backpropagation model to optimize the overall parameters and performance. The loss function design is shown in equation (4.1).

$$Loss = -y \cdot logp - (1 - y) \cdot \log(1 - p) \qquad (4.1)$$

In the inference stage of the model, since the candidate document set D is fixed and known, all candidate documents can be split into paragraphs for offline storage (the structure on the right as shown in Figure 4.3). After real-time reasoning on the user's input question, the semantic interaction with the candidate paragraphs obtained by Bm25 is performed to infer the questions Q and P_i.

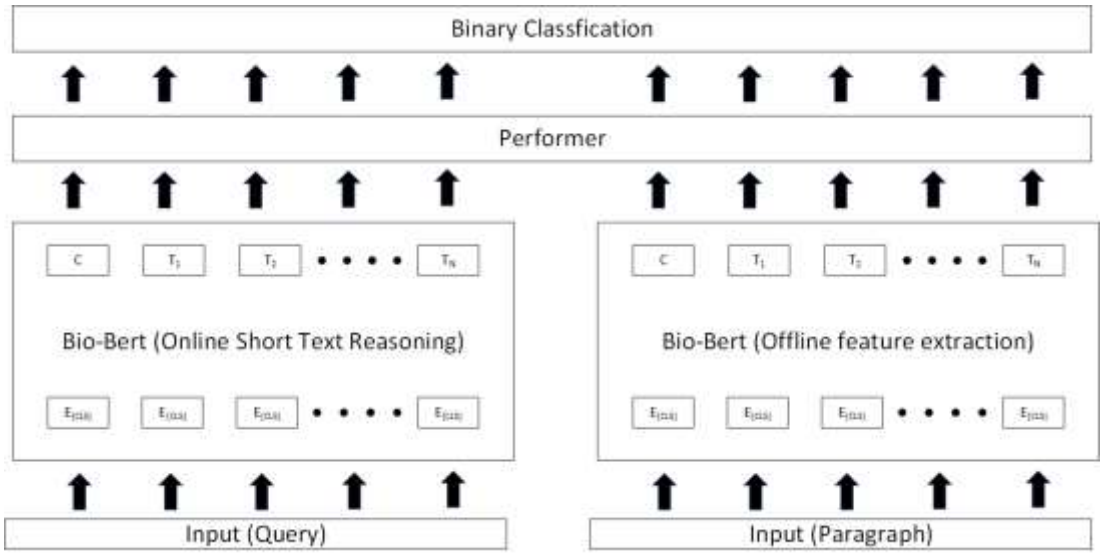

Figure 4.3. The 2nd information retrieval method—Dual Bio-Bert Retrieval

Generally speaking, Bert uses a multi-layer Transformer to encode deep semantic information and interact with questions and answers in machine reading comprehension tasks. In order to increase the speed of model inference, this structure uses the strategy of encoding all

paragraphs offline during inference to improve the timely responsiveness of the model and reuses Bert's deep semantic coding capabilities. In order to allow the question and the candidate paragraph to exchange information, Dual Bio-Bert Retrieval uses Performer's self-attention mechanism, in order to complete the answer decoding process of the question and the paragraph with less memory usage and fewer parameter calculations.

To sum up, for a question raised by the user Q, a subset of D_{stage1} is obtained by using Bm25 based on word frequency information to perform a rough screening from a large number of candidate documents D. At the same time, the obtained candidate documents are divided into sub-paragraphs P. Use the structure proposed above to filter and reorder the candidate paragraphs to obtain paragraph answers with respective score rankings.

4.2 Experiment Evaluation

In order to verify the proposed two recall strategies and the Dual Bio-Bert Retrieval model, the following experiments were carried out on the CORD-19: The COVID-19 Open Research Dataset (Wang et al., 2020) from the recall rate of the retrieved paragraphs and the model's reasoning speed.

4.2.1 Experiment Environment

The experimental environment in this thesis is the Centos7 version Linux system, computing resources include 128G CPU memory and multiple Quadro GP100 16GB HBM2 graphics

cards. Use Python as the development language and Pytorch as the deep learning framework in the experiment. For specific information, see Table 4.2.

Table 4.2. Experimental Environment Configuration Table

Environmental Item	Environmental Parameters
Operating System	Linux (Centos)
Development Language	Python
Deep learning framework	Pytorch
CPU	Core i5-8300H 128G
GPU	Quadro GP100 16GB

4.2.2 Data Description

CORD-19 (Wang et al., 2020) is a corpus of academic papers on COVID-19 and related coronavirus research. It is curated and maintained by the Semantic Scholars team at the Allen Institute for Artificial Intelligence to support text mining and NLP research. The COVID-19 Open Research Dataset (CORD-19) is a growing resource of scientific papers on COVID-19 and related historical coronavirus research. CORD-19 aims to facilitate the development of text mining and information retrieval systems through its rich metadata and structured full-text collections.

4.2.3 Experiment Content

The first method is BM25 Paragraph Recall. The commonly used structure for information retrieval is BM25, which is also the baseline model of the recall paragraph in this thesis. Put all training sets containing questions and answers Q_{set} in accordance with 4:1 for each question in the training set, then respectively verify whether the first Top_1, Top_5, $Top_{10}, Top_{100}, and\ Top_{1000}$ paragraphs of its recall contain the correct answer.

The second method is BM25 Document Recall and Bio-Bert information Retrieval. In order to test the effectiveness of the two-stage retrieval strategy proposed in this thesis and the advantages of Dual Bio-Bert Retrieval, this thesis uses Bert for information retrieval tasks as the comparison structure of Dual Bio-Bert Retrieval. For a question Q_i in the training set, paragraphs are divided by calculating the first Top_{100} documents recalled. Then Dai & Callan proposed a Bert-based deep learning model in the information retrieval (Dai & Callan, 2019) model as a comparative experiment of deep learning.

The third method is BM25 Document Recall + Dual Bio-Bert Retrieval. For a question, after getting the relevant Top_{100} documents, divide the documents into paragraphs and use Dual Bio-Bert Retrieval to construct a binary classification task.

4.2.4 Results

In the aforementioned experimental environment, experiments were carried out according to the hyperparameters shown in Table 4.3.

Table 4.3. Hyperparameters Settings

Parameters	Value
Batch Size	32
Epoch	10
Max Length	512
Performer Hiddensize	768
The Number of Performer Multi-Head	8
Bi-LSTM & Bi-GRU Layer Numbers	2
The Learning Rate of Downstream Structure	1e-4

After statistically storing all Q&A word frequency information, this experiment randomly selects 10,000 questions from all training sets and runs them 10 times, using the three solutions mentioned above. The results are shown in Table 4.4.

Table 4.4. The Results of Information Retrieval

Methods	$\boldsymbol{Top_{10}}$ **Recall Rate**	$\boldsymbol{Top_{100}}$ **Recall Rate**	$\boldsymbol{Top_{1000}}$ **Recall Rate**	**Reasoning Speed/ 100 documents**
BM25 Paragraph Recall	0.165	0.378	0.855	130ms

BM25 Document Recall + Bio-Bert Information Retrieval	0.352	0.481	0.857	180s
BM25 Document Recall + Dual Bio-Bert Retrieval	0.376	0.532	0.891	3s

As can be seen, in terms of the reasoning speed, the result of BM25 Paragraph Recall is impeccable among the three methods. However, its Top_{10} recall rate is not ideal. When the model recalls 100 paragraphs, its recall rate is only 0.378. After introducing Bert's deep semantic information, BM25 Document Recall and Bio-Bert Information Retrieval make the reordering result greatly improve the model's sensitivity to the matching of questions and answers. Its Top_{10} paragraph recall rate reached 0.352. However, it takes the 90s for the model to infer all paragraphs of the 50 documents recalled by Bm25 to complete the reordering, which makes retrieval that requires immediate response impractical.

At the same time, compared with Bio-Bert Information Retrieval, using Dual Bio-Bert Retrieval proposed in this thesis showed almost equivalent performance on this task. Although the Dual Bio-Bert Retrieval introduces an extra layer of Performer calculations and Bert forward inference, in addition to a large number of Transformer parameters and the complex matrix calculation of Word embedding, the model's inference speed is not as fast as in the ideal state. However, the reordering response time only takes 3 s to complete the reordering of the

100 documents recalled by the problem, and the rapid reasoning of the model is still completed while ensuring a high recall rate.

Due to the difficulty of the task, the recall rate of the model in recalling 100 documents is not ideal. In order to consider how many documents need to be recalled, the recall rate can be considerable, so experiments were conducted. The upper limit of the task was tried, and it was found that the recall rate converged after the model recalled 1000 documents, which was around 0.9. However, it is unrealistic to recall 1,000 documents in practical applications, because reading comprehension consumes a lot of resources and time. So, the subsequent experiment recalled 100 documents.

4.3 Conclusion

As can be seen, when recalling A, B, and C, the model proposed in the thesis has a certain improvement over other models. Moreover, while having a good recall rate, it also has a certain feasible inference speed. This chapter studies the design of related algorithms for paragraph retrieval and proposes a two-stage paragraph retrieval strategy. Compared with the classic retrieval structure of traditional BM25, a pre-trained model is introduced to reorder to improve the recall rate of related documents, and a comparative experiment was carried out on the CORD-19: The COVID-19 Open Research Dataset (Wang et al., 2020) dataset.

Chapter 5

Machine Reading Comprehension

This chapter mainly introduces the machine reading comprehension model designed in this research and the experimental verification of the proposed structure in the CORD-19: The COVID-19 Open Research Dataset (Wang et al., 2020).

5.1 The Structure of the Multi-task Machine Reading Comprehension

For a user's question q, we assume that after information retrieval through the strategy proposed above, the question recalls the first paragraph from thousands of documents with a high recall rate. At this time, extracting reading comprehension for the document will greatly reduce the difficulty of the document machine reading comprehension task. However, based on information retrieval technology, there is no guarantee on the recall rate of TOP_1, which means that recalling only one document will cause large error propagation. That is, when the document recalled by information retrieval is wrong, no matter how well the machine reading comprehension model learns, it cannot answer the user's question. Therefore, in order to ensure the recall rate of information retrieval and the number of samples to be processed for reading comprehension, the Bm25 combined with the information retrieval technology of Dual-Bio Bert was selected based on the experimental results in Chapter 4 to recall TOP_1, TOP_5,

TOP_{20}, TOP_{100} candidate paragraphs of user documents for machine reading comprehension task.

At this point, the input question q will get TOP_5 relevant paragraphs after information retrieval. Building a classic machine reading comprehension using Bio-Bert has been mentioned in Section 3.2. As shown in Figure 5-1 below, for the input question q_i and the candidate document D_i, the word embedding in the Bert structure is used to embed the text content, and the Bert multi-layer stacked Transformer structure is used to perform feature extraction on the encoded information of the question and paragraph. The vector is obtained through the self-attention interaction in the Transformer, the index of the start and end positions of the answer is found by the pointer method, and the answer is obtained by extraction.

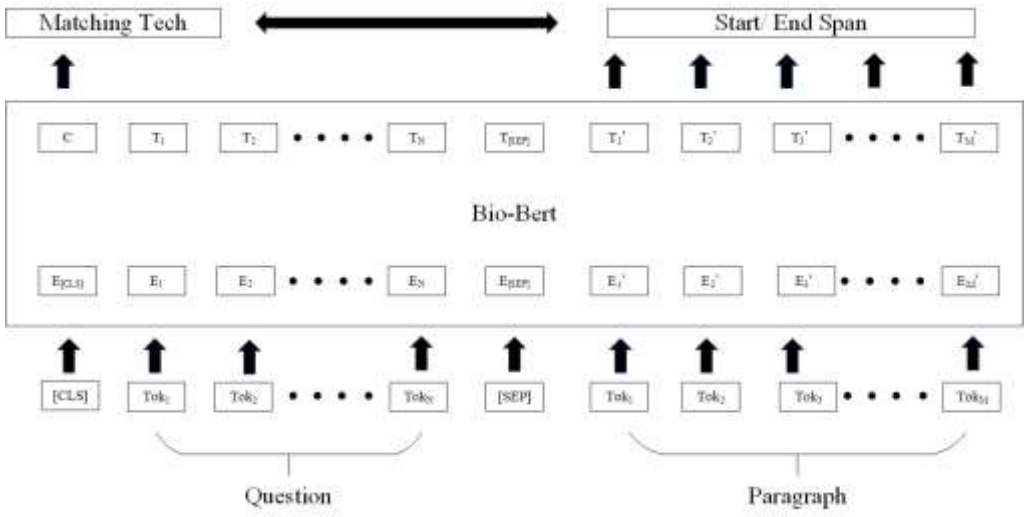

Figure 5.1. The Model of Multi-Task Learning

5.1.1 Multi-Task Learning

Despite the double-check of coarse recall and fine recall, the recall of the model in the document where the correct answer is located is still not fully guaranteed. In order to obtain refined and short answers, the model needs to measure whether the question and the corresponding document is related, so as to prevent the special cases of different answers in these candidate paragraphs. Therefore, the test set often has many documents that need to appear in the form of no answers. For this reason, while constructing the reading comprehension task, this research constructs pairs of input questions and candidate documents to fine-tune the Bert structure to train the question and the binary classification task of whether the text matches, that is, multi-task learning.

What is multi-task learning? It is defined as a machine learning method that learns multiple related tasks together based on a shared representation. In multi-task learning, multiple related tasks are computed forward and back-propagated simultaneously. Multiple tasks help each other learn to improve generalization by sharing underlying representations and task interactions. Simply speaking, multi-task learning learns multiple related tasks together to improve the model's ability to understand deep information from shared representations.

Intuitively, when humans learn to understand a piece of semantics, they can solve problem A through this semantics, and at the same time, they can also solve problem B with this semantic

understanding. At this time, there is often a deeper understanding of this passage. In the analysis of the training process of the deep learning black box, when multiple tasks are learned together, each task has related parts and different places. When learning a certain main task, the unrelated tasks are equivalent to noise, so the introduction of multiple tasks is equivalent to improving the generalization ability of the model. In addition, when learning a single task, the back-propagation of the gradient tends to fall into a local optimum, while the location of the local minima of different tasks is also different, and the shared layer can help the parameters of the hidden layer escape from the local optimum through interaction.

For this purpose, multi-task joint learning is designed as shown in Figure 5.1. For an input question q_i and a question paragraph pair constructed from a candidate paragraph p_i, suppose context is $[x_1, \dots, x_k]$ and query is $[q_1, \dots, q_j]$. The model obtains an output vector $T \in R^{(J+K)\times h}$ by sharing deep Bert to encode questions and documents respectively, feature extraction and semantic interaction, where h is the dimension of the hidden layer of Bert output vector, the value is usually 768. At this time, vector T covers a large amount of deep semantic information and interaction information between questions and paragraphs through the multi-layer Transformer that has been pre-trained and converged on massive data, and an additional task—Matching Tech—is designed for this vector.

5.1.2 The Design of the Loss Function

As shown in Figure 5.1, the Matching Tech method is to predict whether the question is related to the paragraph. In Bert's pre-training task Next Sentence Predict, the output vector of the CLS position is used to judge whether it is the upper and lower sentences, so the vector corresponds to the CLS Token in the pre-training model $E_{cls} \in R^{1\times h}$ is used to fully connect a neuron. Then use the sigmoid activation function to get the matching degree $p_{(q,P)}$ between the question and the paragraph, which is used to distinguish whether the question and the answer match. As shown in equation (5.1), binary classification of cross entropy is used to calculate prediction label $p_{(q,P)}$ and real tag $y_{yes,no}$ for $loss$ transmission error in back propagation.

$$Loss_1 = -(y_{yes,no} \cdot logp_{(q,P)} + (1 - y_{yes,no}) \cdot \log(1 - p_{(q,P)})) \quad (5.1)$$

In order for the model to more accurately recall the answers when decoding the answers, the ability of Matching Tech to distinguish between questions and passages is extremely important.

5.2 Experiment Evaluation

In the CORD-19: The COVID-19 Open Research Dataset(Wang et al., 2020), this research verifies the structure of the multi-task learning model proposed above.

5.2.1 Experiment Description

On the CORD-19: The COVID-19 Open Research Dataset(Wang et al., 2020), the training set of multi-task learning is divided into positive and negative samples. The positive samples are

the training set questions and their answers and the paragraphs where the answers are located, and the negative samples are the Top_{100} non-answer paragraphs that were recalled for information retrieval in the previous structure.

When recalling reading comprehension answers, this structure also considers the output structure of the reading comprehension task and whether the question and answer match the task, as shown in equation (5.2).

$$score_{ans} = \exp(\frac{w_s logP_s + w_e logP_e}{w_s + w_e}) \qquad (5.2)$$

Because the log function is more sensitive to smaller values, if one of the items has a lower probability, it will make the overall score lower. w_s and w_e represent the weight ratio of start and end respectively. Since the magnitude and importance of start and end are often the same, the default setting in the experiment is 1:1. At this time, the overall score of the answer is measured as shown in equation (5.3).

$$score = \exp(\frac{w_{nsp} logP_{nsp} + w_{ans} logP_{ans}}{w_{nsp} + w_{ans}}) \qquad (5.3)$$

where: w_{nsp} and w_{ans} are the score weights for retrieval and reading.

This is a pair of hyperparameters, and when the retrieval range is particularly large, the results are more sensitive to this hyperparameter, and the weight ratio of the question and the answer needs to be adjusted very high. In this experiment, the weight ratio is set to 0.98:0.02. The

reason why part of the weight of ans is reserved is that if all the retrieval scores are used, the results will be reduced.

At the same time, this research also uses the typical structure of reading comprehension QANET and BIDAF for experimental comparison, and the model performance is measured by the Rouge-L and EM evaluation indicators of the model's extracted answers. The hyperparameters of the model are shown in Table 5.1.

Table 5.1. The Hyperparameter Design of the Model

Parameters	**The Dataset**
Batch Size	32
Epoch	20
Max Length	512
TENER Hidden Size	256
The Multi-Head Amount of TENER	4
Bert-finetune-learning Rate	2e-5
The Learning Rate of Downstream	1e-4

5.2.2 The Analysis of the Experimental Results

The experimental results are shown in Table 5.2 and Table 5.3. It can be seen that the multi-task machine reading comprehension structure designed in this research performs better than other methods in the CORD-19: The COVID-19 Open Research Dataset (Wang et al., 2020).

The Top_{100} Rouge-L score of the model reached 85.53, while the Rouge-L scores of BiDAF and QANET are 81.12 and 81.17 respectively. On the Top_1 (43.92), Top_5 (67.15), and Top_{20}(78.21) Rouge-L score, the results are also higher than the other two.

Table 5.2. The Rouge-L Score Comparison among the Three Models

Plans	Top_1 **Rouge L Score**	Top_5 **Rouge L Score**	Top_{20} **Rouge L Score**	Top_{100} **Rouge L Score**
BiDAF	39.58	63.38	73.19	81.12
QANET	37.13	63.16	73.36	81.17
The thesis model	43.92	67.15	78.21	85.53

In addition, in the EM (exact matching) score, the Top_{100} EM score of the model reached 51.97, while the EM scores of BiDAF and QANET are 45.81 and 46.20 respectively. On the Top_1(25.32), Top_5(44.72), and Top_{20}(48.86) EM scores, the results are also higher than the other two.

Table 5.3. The EM Score Comparison among Three Plans

Plans	Top_1 **EM Score**	Top_5 **EM Score**	Top_{20} **EM Score**	Top_{100} **EM Score**
BiDAF	21.05	41.51	45.33	45.81
QANET	20.79	39.97	46.21	46.20
The thesis model	25.32	44.72	48.86	51.97

The following graphs (Figure 5.2 and Figure 5.3) show the Rouge-L score and EM scores comparison among the three plans.

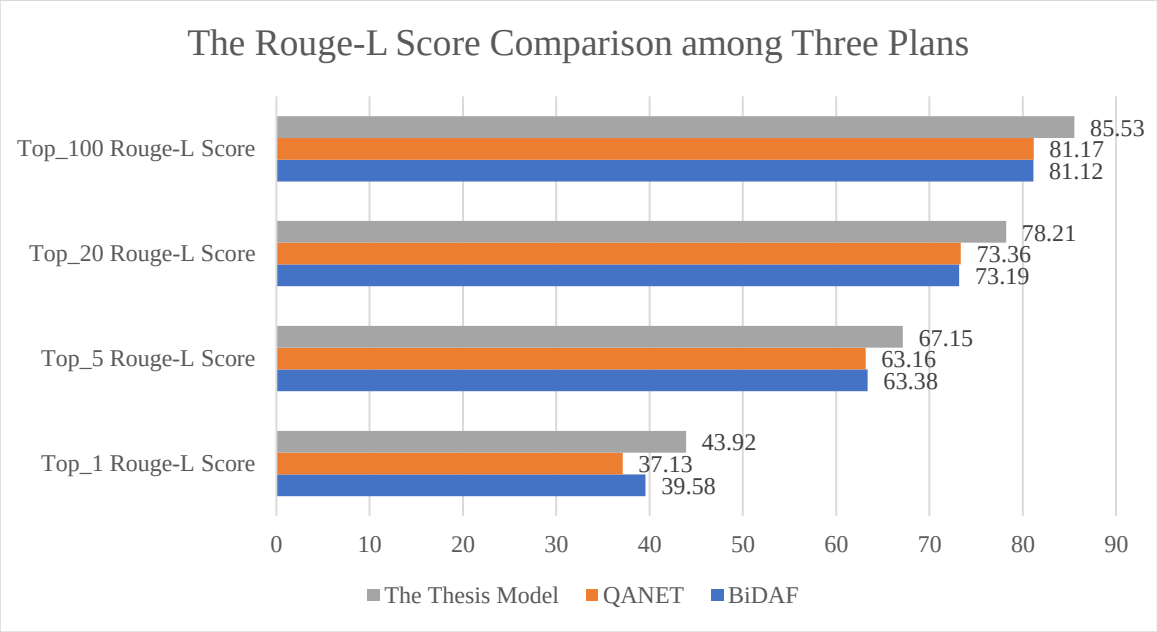

Figure 5.2. The Rouge-L Score Comparison among Three Plans

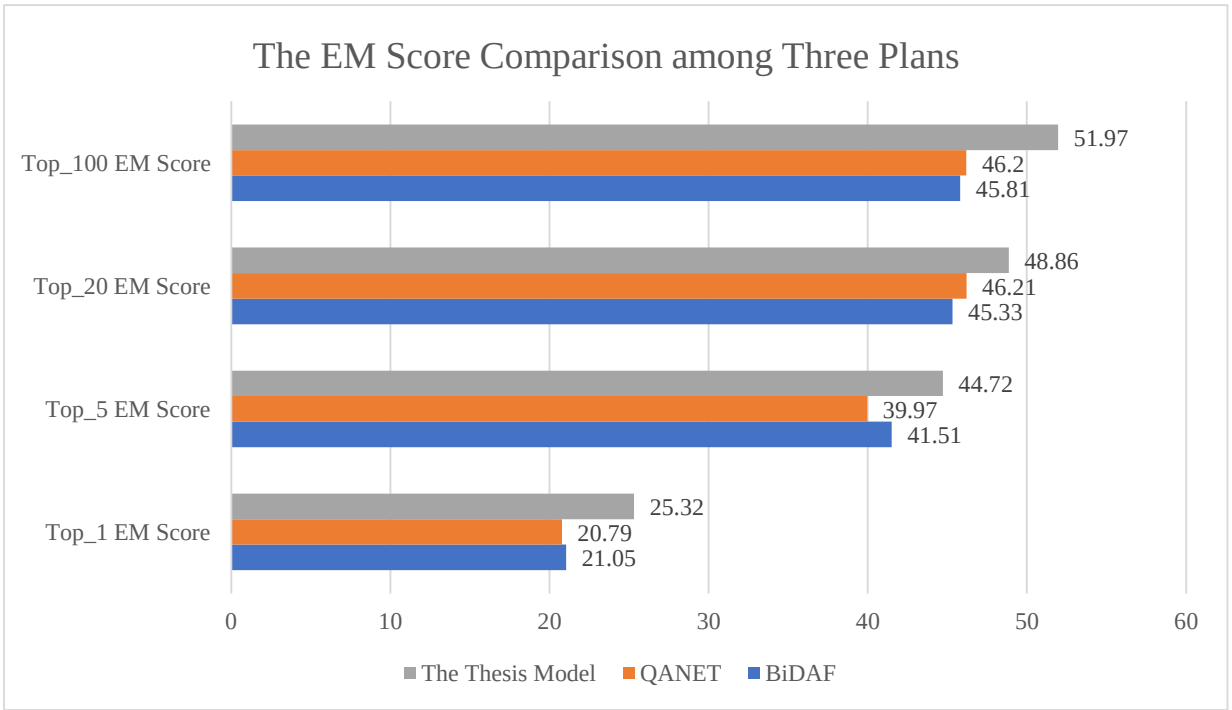

Figure 5.3. The EM Score Comparison among Three Plans

5.3 Conclusion

This chapter studies the algorithms related to machine reading comprehension and designs a multi-task machine reading comprehension structure based on a pre-trained model. Based on Bert's reading comprehension, this structure adds a method called Matching Tech, which assists the model to output a better result by setting different weights for characters labelled 0 and 1.

The structure has been verified on the CORD-19: The COVID-19 Open Research Dataset (Wang et al., 2020), and the experimental results show that the Rouge-L and EM exact matching rates of the multi-task reading comprehension structure proposed in this chapter have improved to varying degrees on this dataset.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Intelligent question answering robots have the ability to quickly deal with simple and repetitive questions, which can help hospitals solve a large number of repetitive tasks. In this way, the staff at the hospital reception desk can undertake a larger workload under the premise of having robots, so that they can receive more patients every day. When using the online intelligent question answering robot, the robot can reply to simple questions more quickly and standardized, and the difficult questions are transferred to the reception service, which can make the overall service quality higher and the patient experience better. Using voice intelligent question answering robots, many hospitals may be able to automatically screen patients through robot outbound calls, transfer serious patients to the reception desk, and improve the conversion rate of medical staff. For hospitals with more mature business forms, when intelligent robots undertake a lot of repetitive work, the hospital's demand for labour will be reduced, which can help hospitals save labour costs.

This thesis conducts in-depth research on multi-document machine reading comprehension tasks and proposes a multi-document machine reading comprehension framework. The framework consists of two-stage information retrieval and machine reading comprehension

structure that focuses on candidate passages from a given set of documents. The corresponding improved models are designed mainly in its retrieval stage and reading comprehension task stage and are verified by the CORD-19: The COVID-19 Open Research Dataset (Wang et al., 2020). Experiments show that the retrieval model has certain efficiency and reliability, and the reading comprehension structure also shows strong performance.

In general, a paragraph recall strategy for two-stage information retrieval is designed. For a problem, first, recall TOP_{100} related documents using Bm25's word frequency feature-based method are determined, and propose a Dual-Bio Bert Retrieval structure to fine-tune the model. In the training phase, two Bert's are used to extract semantic features of paragraphs and questions respectively. By introducing Performer, the model interacts with the information of the two. When the model is offline inference, the vector representations of all paragraphs are saved locally in advance. In real-time prediction, it is only necessary to reason about the text of the word length of the question, which greatly improves the response speed while ensuring high recall. In addition, a multi-task machine reading comprehension structure is designed. Based on Bert's reading comprehension extractive answer task, this structure uses the method of multi-task learning to use an additional task—Matching Tech—to strengthen the semantic information interaction between text and candidate paragraphs and the information expression of high-dimensional feature space.

6.2 Future Work

The experimental results show that the information retrieval strategy and multi-document machine reading comprehension structure studied in this topic have a good improvement in speed and performance. However, the use of pre-trained models has problems of high response speed and resource consumption in practical application scenarios and still has great disadvantages compared to traditional algorithms used in current real scenarios. In addition, because the questioning methods of the annotators are relatively uniform, it is difficult to cover the multi-user and multi-angle questions in the real scene.

In the future work, this paper can make feasible exploration and improvement on the above problems.

- First, knowledge distillation is used to learn the true labels of the dual Bio-Bert and multi-task machine reading comprehension models proposed in the paper, and the soft labels of the model obtained after distilling the temperature T in the smaller model. Let the model learn Dual Bio-Bert Retrieval and multi-task machine reading comprehension structures with less memory and computation. And let the model teach how to learn the current sample by appending smooth labels. The purpose is to reduce the use of video memory and improve performance while the model loses as much effect as possible.

- Secondly, use adversarial training to add perturbations to the text embedding layer in the pre-trained structure, and choose perturbations that make the model more error-prone to make the model learn the current task as wrong as possible. At the same time, in the case of model errors, the model can learn more anti-interference parameters through gradient descent, and improve the robustness and domain adaptive ability of the model to answer various questions of different users in different ways.

- Last but not least, common questions from patients are collected and the answers are annotated with relevant document processing questions, and then the explored and improved structures are used to learn and train and deploy in real-world application scenarios to answer everyday questions from patients. At the same time, experts in related fields can also participate, making the answers more authoritative.

References

Alammar, J. (2018). *The Illustrated Transformer*. https://jalammar.github.io/illustrated-transformer/

Baker, C. F., Fillmore, C. J., & Lowe, J. B. (1998). The Berkeley FrameNet Project. *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*. COLING 1998. https://aclanthology.org/C98-1013

Balduccini, M., Baral, C., & Lierler, Y. (2008). Chapter 20 Knowledge Representation and Question Answering. In F. van Harmelen, V. Lifschitz, & B. Porter (Eds.), *Foundations of Artificial Intelligence* (Vol. 3, pp. 779–819). Elsevier. https://doi.org/10.1016/S1574-6526(07)03020-9

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., & Hellmann, S. (2009). DBpedia—A crystallization point for the Web of Data. *Journal of Web Semantics*, *7*(3), 154–165. https://doi.org/10.1016/j.websem.2009.07.002

Bollacker, K., Evans, C., Paritosh, P. K., Sturge, T., & Taylor, J. (2008). Freebase: A collaboratively created graph database for structuring human knowledge. *SIGMOD Conference*. https://doi.org/10.1145/1376616.1376746

Bordes, A., Chopra, S., & Weston, J. (2014). Question Answering with Subgraph Embeddings. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 615–620. https://doi.org/10.3115/v1/D14-1067

Bordes, A., Usunier, N., Chopra, S., & Weston, J. (2015). Large-scale Simple Question Answering with Memory Networks. *ArXiv:1506.02075 [Cs]*. http://arxiv.org/abs/1506.02075

Bordes, A., Weston, J., & Usunier, N. (2014). Open Question Answering with Weakly Supervised Embedding Models. *ArXiv:1404.4326 [Cs]*. http://arxiv.org/abs/1404.4326

Bouziane, A., Bouchiha, D., Doumi, N., & Malki, M. (2015). Question Answering Systems: Survey and Trends. *Procedia Computer Science*, *73*, 366–375. https://doi.org/10.1016/j.procs.2015.12.005

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734. https://doi.org/10.3115/v1/D14-1179

Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J., Mohiuddin, A., Kaiser, L., Belanger, D., Colwell, L., & Weller, A. (2021). Rethinking Attention with Performers. *ArXiv:2009.14794 [Cs, Stat]*. http://arxiv.org/abs/2009.14794

Clark, C., & Gardner, M. (2018). Simple and Effective Multi-Paragraph Reading Comprehension. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 845–855. https://doi.org/10.18653/v1/P18-1078

Cui, Y., Liu, T., Chen, Z., Wang, S., & Hu, G. (2016). Consensus Attention-based Neural Networks for Chinese Reading Comprehension. *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 1777–1786. https://aclanthology.org/C16-1167

Dai, Z., & Callan, J. (2019). *Deeper Text Understanding for IR with Contextual Neural Language Modeling | Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. https://dl.acm.org/doi/abs/10.1145/3331184.3331303

Deng, C., Zeng, G., Cai, Z., & Xiao, X. (2020). A Survey of Knowledge Based Question Answering with Deep Learning. *Journal of Artificial Intelligence*, *2*(4), 157–166. http://dx.doi.org/10.32604/jai.2020.011541

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. https://doi.org/10.18653/v1/N19-1423

Erk, K., & Smith, N. A. (Eds.). (2016). *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics. https://doi.org/10.18653/v1/P16-1

Etzioni, O., Fader, A., Christensen, J., Soderland, S., & Mausam, M. (2011). *Open Information Extraction: The Second Generation.* 3–10. https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-012

Fader, A., Soderland, S., & Etzioni, O. (2011). Identifying Relations for Open Information Extraction. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 1535–1545. https://aclanthology.org/D11-1142

Fan, J., Hoffman, R., Kalyanpur, A., Riedel, S., Suchanek, F., & Talukdar, P. P. (Eds.). (2012). *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*. Association for Computational Linguistics. https://aclanthology.org/W12-3000

Gleich, D. F. (2015). PageRank Beyond the Web. *SIAM Review*, *57*(3), 321–363. https://doi.org/10.1137/140976649

Graves, A., Wayne, G., & Danihelka, I. (2014). Neural Turing Machines. *ArXiv:1410.5401 [Cs]*. http://arxiv.org/abs/1410.5401

Green, B. F., Wolf, A. K., Chomsky, C., & Laughery, K. (1961). Baseball: An automatic question-answerer. *Papers Presented at the May 9-11, 1961, Western Joint IRE-AIEE-ACM Computer Conference*, 219–224. https://doi.org/10.1145/1460690.1460714

Hu, M., Peng, Y., Huang, Z., Qiu, X., Wei, F., & Zhou, M. (2018). Reinforced Mnemonic Reader for Machine Reading Comprehension. *ArXiv:1705.02798 [Cs]*. http://arxiv.org/abs/1705.02798

Huang, H.-Y., Zhu, C., Shen, Y., & Chen, W. (2018). FusionNet: Fusing via Fully-Aware Attention with Application to Machine Comprehension. *ArXiv:1711.07341 [Cs]*. http://arxiv.org/abs/1711.07341

Jia, R., & Liang, P. (2017). Adversarial Examples for Evaluating Reading Comprehension Systems. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2021–2031. https://doi.org/10.18653/v1/D17-1215

Kelly, D., & Lin, J. (2007). Overview of the TREC 2006 ciQA task. *ACM SIGIR Forum*, *41*(1), 107–116. https://doi.org/10.1145/1273221.1273231

Klein, G., Kim, Y., Deng, Y., Senellart, J., & Rush, A. M. (2017). OpenNMT: Open-Source Toolkit for Neural Machine Translation. *ArXiv:1701.02810 [Cs]*. http://arxiv.org/abs/1701.02810

Kwiatkowski, T., Zettlemoyer, L., Goldwater, S., & Steedman, M. (2011). Lexical Generalization in CCG Grammar Induction for Semantic Parsing. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 1512–1523. https://aclanthology.org/D11-1140

Lao, N., Mitchell, T., & Cohen, W. W. (2011). Random Walk Inference and Learning in A Large Scale Knowledge Base. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 529–539. https://aclanthology.org/D11-1049

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2019). BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, btz682. https://doi.org/10.1093/bioinformatics/btz682

Liang, P., Jordan, M., & Klein, D. (2011). Learning Dependency-Based Compositional Semantics. *Proceedings of the 49th Annual Meeting of the Association for*

Computational Linguistics: Human Language Technologies, 590–599. https://aclanthology.org/P11-1060

Lin, C.-Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. *Text Summarization Branches Out*, 74–81. https://aclanthology.org/W04-1013

Liu, S., Zhang, X., Zhang, S., Wang, H., & Zhang, W. (2019). Neural Machine Reading Comprehension: Methods and Trends. *ArXiv*. https://doi.org/10.1016/j.apsusc.2018.11.214

Mausam, Schmitz, M., Soderland, S., Bart, R., & Etzioni, O. (2012). Open Language Learning for Information Extraction. *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 523–534. https://aclanthology.org/D12-1048

Mendes, A. C., & Coheur, L. (2013). When the answer comes into question in question-answering: Survey and open issues. *Natural Language Engineering*, *19*(1), 1–32. https://doi.org/10.1017/S1351324911000350

Miller, G. A. (1995). WordNet: A Lexical Database for English. *Communications of the Acm*, *38*, 39–41.

Moldovan, D., & Rus, V. (2001). Logic Form Transformation of WordNet and its Applicability to Question Answering. *Proceedings of the 39th Annual Meeting of the*

Association for Computational Linguistics, 402–409. https://doi.org/10.3115/1073012.1073064

Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., & Deng, L. (2016). *MS MARCO: A Human-Generated MAchine Reading COmprehension Dataset*. https://openreview.net/forum?id=Hk1iOLcle

Paik, J. H. (2013). A novel TF-IDF weighting scheme for effective ranking. *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 343–352. https://doi.org/10.1145/2484028.2484070

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuAD: 100,000+ Questions for Machine Comprehension of Text. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2383–2392. https://doi.org/10.18653/v1/D16-1264

Robertson, S., & Zaragoza, H. (2009). *The Probabilistic Relevance Framework: BM25 and Beyond*. Now Publishers Inc.

Ruder, S. (2021). *ML and NLP Research Highlights of 2020*. https://ruder.io/research-highlights-2020/

Sil, A., & Lin, X. V. (Eds.). (2021). *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human*

Language Technologies: Demonstrations. Association for Computational Linguistics. https://aclanthology.org/2021.naacl-demos.0

Soubbotin, M. M. (2001). Patterns of Potential Answer Expressions as Clues to the Right Answers. *In Proceedings of the Tenth Text REtrieval Conference (TREC*, 293–302.

Sukhbaatar, S., Szlam, A., Weston, J., & Fergus, R. (2015). End-To-End Memory Networks. *ArXiv:1503.08895 [Cs]*. http://arxiv.org/abs/1503.08895

Tay, Y., Dehghani, M., Abnar, S., Shen, Y., Bahri, D., Pham, P., Rao, J., Yang, L., Ruder, S., & Metzler, D. (2020, September 28). *Long Range Arena: A Benchmark for Efficient Transformers*. International Conference on Learning Representations. https://openreview.net/forum?id=qVyeW-grC2k

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. *ArXiv:1706.03762 [Cs]*. http://arxiv.org/abs/1706.03762

Wang, L. L., Lo, K., Chandrasekhar, Y., Reas, R., Yang, J., Burdick, D., Eide, D., Funk, K., Katsis, Y., Kinney, R., Li, Y., Liu, Z., Merrill, W., Mooney, P., Murdick, D., Rishi, D., Sheehan, J., Shen, Z., Stilson, B., … Kohlmeier, S. (2020). CORD-19: The COVID-19 Open Research Dataset. *ArXiv:2004.10706 [Cs]*. http://arxiv.org/abs/2004.10706

Wong, Y. W., & Mooney, R. J. (2006). Learning for semantic parsing with statistical machine translation. *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, 439–446. https://doi.org/10.3115/1220835.1220891

Woods, W. A. (1973). Progress in natural language understanding: An application to lunar geology. *Proceedings of the June 4-8, 1973, National Computer Conference and Exposition*, 441–450. https://doi.org/10.1145/1499586.1499695

Wu, F., & Weld, D. S. (2010). Open Information Extraction Using Wikipedia. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 118–127. https://aclanthology.org/P10-1013

Yates, A., Banko, M., Broadhead, M., Cafarella, M., Etzioni, O., & Soderland, S. (2007). TextRunner: Open Information Extraction on the Web. *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, 25–26. https://aclanthology.org/N07-4013

Yu, A. W., Dohan, D., Luong, M.-T., Zhao, R., Chen, K., Norouzi, M., & Le, Q. V. (2018). QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. *ArXiv:1804.09541 [Cs]*. http://arxiv.org/abs/1804.09541

Zhu, C. (2021). Chapter 1—Introduction to machine reading comprehension. In C. Zhu (Ed.), *Machine Reading Comprehension* (pp. 3–26). Elsevier. https://doi.org/10.1016/B978-0-323-90118-5.00001-1