A medical Q&A system based on a knowledge graph

GUAN FENG MING

Master's Program in Informatics
Degree Programs in Comprehensive Human Sciences
Graduate School of Comprehensive Human Sciences
University of Tsukuba
September 2022

A medical Q&A system based on a knowledge graph

Name: GUAN FENG MING

As a result of China's current dearth of medical physicians and medical resources, as well as the growth of the Internet, an increasing number of people are holding medical question and answer sessions online. The goal of this research is to create a medical question-answering system based on a knowledge graph that will allow individuals to acquire answers to medical queries rapidly on the computer.

This system starts from creating a medical knowledge graph in Neo4j which is a high-performance NOSQL graph database. When users enter a question, the system will perform intent recognition and named entity recognition on the question. The system will next use Cypher language to search the knowledge graph for the answer based on the extracted query intent and entity, using the specified rule template that was formed during slot filling.

Two models are used in this study to fulfill the tasks of intent recognition and named entity recognition, and the system also enhanced the model's results with information from the knowledge graph using the template matching method. Additionally, the entity linking procedure has been finished in order to better integrate models into a question-and-answer system. The entity linking function can link non-standard words in user questions to common disease names on the knowledge graph, to better obtain answers from the knowledge graph.

The intent recognition used the BERT-TextCNN model, and the accuracy obtained on the CMID public dataset was 0.67. The named entity recognition used the BiLSTM-CRF model, and the accuracy obtained on the cMedQANER public dataset was 0.96. The entity linking process used the ESIM model, and the accuracy obtained on the Yidu-N7K public dataset was 0.96. The system used the above trained models and template matching methods to answer a total of 13 types of questions: 'Definition', 'Cause', 'Prevention', 'Disease manifestations', 'Associated conditions', 'Treatment', 'Department', 'Infectious', 'Cure rate', 'Taboo', 'Physical Examination Program', 'Treatment time' and 'Food recommend'. To compare the effectiveness of this system with the system that only uses the template matching method, the experiment was designed with 200 questions divided into four groups. 'whether the question uses the standard entity names in the knowledge graph' and 'whether the question uses the feature words in the intent template' were controlled separately in different groups. The experimental results demonstrate that the correct answer rate of this system is much higher than that of the system based on template matching only when users enter questions without using standard entity names from the knowledge graph or without using feature words from the intent template.

Main Academic Advisor: Kei Wakabayashi
Secondary Academic Advisor: Haitao Yu

Contents

List of Figures

List of Tables

Chapter 1

Introduction

According to current study, China's medical resource allocation has steadily become inadequate to fulfill the country's expanding medical needs [1]. It is most visible in the severe disparity in medical resource distribution between urban and rural areas, supply shortages in some underdeveloped areas, and the urgent need to strengthen professional medical personnel and medical diagnosis services.

At the same time, with the advancement of the Internet, people are becoming increasingly eager to acquire medical, health, and other related guidance resources over the Internet [2]. Medical and health are the most searched topics, according to the report's findings, with a search ratio of 73.8 percent. People are increasingly turning to the Internet to swiftly and properly receive health and medical services.

Furthermore, medical artificial intelligence is the present research and development priority of China's health informatization. With the advancement of medical digitalization, a great amount of medical knowledge has been generated on the Internet in the medical industry [3]. The focus of study has shifted to how to collect and utilise this information in order to assist people in obtaining answers to medical issues.

In this regard, this study develops a knowledge graph-based medical question-answering system.

In recent years, knowledge graphs have gotten a lot of attention and have developed quickly [4]. Its knowledge acquisition and organizing model is ideal for gathering a significant volume of medical information and retrieving it over the Internet. It keeps triples like <entity, relationship, entity> in the form of a graph, where an entity refers to a thing and an entity is linked to another via a relationship. This knowledge representation of the knowledge graph is ideal for building question-answering systems, because any two elements in a triple can be used to generate another element [5].

There are two sorts of approaches used in question-answering systems based on knowledge graphs. One is semantic parsing, whose fundamental idea is to convert unstructured natural language problems into a series of logical forms, and then query the knowledge base

to obtain logical forms that can represent semantics. The retrieval of information is the second method. The goal is to extract information from the question, then use the knowledge base to generate candidate responses, which are then sorted to get the final answer. Its performance is not as good as semantic parsing, but it is reasonably easy and can be used in a variety of applications.

This study employs the semantic parsing method framework, which includes processes such as intent recognition, named entity recognition, and entity linking, as well as slot filling method and Cypher language to get the answer on the knowledge graph.

Chapter 2

Related Work

2.1 Knowledge graph

Entities, relationships, and semantic descriptions make up knowledge graphs, which are structured representations of facts. Entities can be physical objects or abstract concepts, relations reflect connections between them, and semantic representations of entities and their relationships comprise well-defined kinds and characteristics. When nodes and relations have properties or attributes, property or nature diagrams are commonly employed.

The early concept of a knowledge graph came from Tim Berners-Lee's vision of the Semantic Web [6], which aimed to use graph structures to represent and preserve associative relationships and knowledge among things in the world in order to successfully achieve more accurate object-level search.

The process of building a knowledge graph typically begins with the most basic data (including structured, semi-structured, and unstructured data) and proceeds through a series of automatic or semi-automatic technical means to extract knowledge facts from the original database and third-party databases and deposit them into the data layer and schema layer of the knowledge base. This process is divided into four steps [7]: Information extraction, knowledge representation, knowledge fusion, knowledge reasoning. Knowledge graph technologies have been widely and successfully implemented in a variety of domains, including search engines, intelligent question and answer systems, language understanding, recommendation computing, and big data decision analysis.

Knowledge graphs also have some advantages in the Chinese area for processing massive amounts of data and have a wide range of application situations [8].

2.2 Knowledge graph based question answer

Semantic parsing and information retrieval are the two method usually used in knowledge graph-based QA systems.

Identifying the central entity of the question, creating a number of candidate responses to the question, and then using scoring and ranking to determine the answer that best fits the original question is the typical approach of information retrieval. The method first extracts the features from the Query and the candidate answers, then links to the Topic

Entity in KG and extracts the Subgraph associated with the Topic Entity as the collection of candidate answers. Finally, a ranking model [9] is used to model and anticipate the Query and candidate replies. This approach's basic architecture is quite compact, and it produces superior results for simple situations. The information retrieval strategy typically presupposes that the problem is straightforward, and that there is only one central entity in the problem. It also requires that the solution to the problem is close enough to the central entity in the knowledge graph.

The semantic parsing approach entails first understanding the semantics of natural language questions with semantic parsing, then transforming the questions into logical forms with the same semantics, and finally querying the generated logical forms through a query engine to obtain the final results. Phrase detection, resource mapping, and semantic combination are the three tasks that can be broken down into.

The purpose of phrase detection is to find phrases that contain valid information in the problem, which are referred to as representational phrases.

The purpose of resource mapping is to connect each word to a specific knowledge base element. Resource mapping can be separated into two primary tasks based on the many types of mapping elements: entity linking and relation identification. In the subject of natural language processing, these two tasks have been extensively researched. Entity linking is the process of connecting entities in a problem to entities in the knowledge graph by using entity disambiguation, normalization, and other techniques based on the recognized entities [10]. The basic goal of relationship identification is to determine the relationship between the problem target and the entities in the problem so that the relationship between the entities can be searched in the knowledge graph. Matching based on established templates, recognition based on a developed phraseological relational recapitulation lexicon [11], and similarity calculation using neural network models are among the method to do the relationship identification [12].

Semantic combination refers to combining these elements, which already correspond to entities and relations, into logical forms corresponding to those in the knowledge graph. For simple problems, which generally contain only a single entity and relation, it is sufficient to connect them. For complex problems, which may contain multiple entities and relations, it is necessary to consider how the entities and relations are paired and combined with each other. Berant J et al. [13] pre-defined several templates of Lambda expressions, first generated several candidate Lambda expressions for problem N according to the templates, then generated several paraphrase statements for each expression, and then compared the similarity of these paraphrase statements and N by Paraphrase Model to select the most suitable expression. Bast H et al. [14] designed three query templates for the specific database Freebase and the review set WebQuestions, covering nearly 95 percentage of these questions. When the questions are transformed into a structured logical form, it is sufficient to execute the queries using the corresponding data query engines.

It is sufficient to execute the queries using the respective data query engines once the

questions have been translated into an organized logical form.

2.3 Medical QA based on knowledge graph

Understanding the semantics of questions, or how to turn natural language queries into a form that computers can understand, is at the heart of building a knowledge graph-based QA system.

Abacha et al. [15] proposed a method to convert questions into SPARQL templates, which has been applied to medical QA systems; Zhang et al. [16] proposed an end-to-end character-level multiscale convolutional neural framework, which uses character embedding to extract contextual information of questions or answer sentences from various degrees, and the framework has been applied to Chinese. Abacha et al. [17] also developed MEANS, a medical question-and-answer system that combines natural language processing and semantic network technologies. Feng et al. [18] proposed first converting the user's queries into phrase vectors, then creating a knowledge base, and using a matching algorithm to compare the user's questions with the questions in the knowledge base, selecting the most comparable ones as responses.

Jiang et al. [19] proposed a semantic parsing based approach to complete the medical KBQA system, which performed semantic parsing for the questions, but based only on the template matching approach. The user's natural language problem is classified and analyzed based on the defined feature words, domain Actree, dictionary and question words to get the main entities and relations of the problem. In addition, intent recognition is also done by template matching and finally constructing Cypher language for querying in the knowledge graph. Although the template recognition approach will perform well, it greatly relies on the richness of the template and it is not as accurate as using deep learning models for intent recognition and named entity recognition.

Chapter 3

Proposal

3.1 Structure

This study aims to build a medical knowledge graph based question and answer system, the main process includes building a knowledge graph and semantic parsing based KBQA. in this study, the semantic parsing based KBQA is mainly divided into named entity recognition, entity linking, intent recognition and slot filling process.

Firstly, a knowledge graph is constructed and stored in Neo4j. The knowledge graph includes multiple entities and relations, which are stored in the form of triples.

When the user enters a question, the first step is named entity recognition, which identifies the entities and entity types in the question. For example, if you type 'What if my child has polio', 'polio' will be labeled as an entity and the entity type will be 'disease'.

The second step is entity linking, which means that the identified entities are associated with entities stored in the knowledge graph.

In the third step, the intent is identified. For the question 'What should I do if my child has polio', the intent is identified as 'treatment'.

The fourth step is to fill in the slots. After setting the slots to be filled and entering the corresponding information (disease name and intent) in the slots, a cypher statement will be used to query the knowledge graph according to the set logic template and return the result as the final answer.

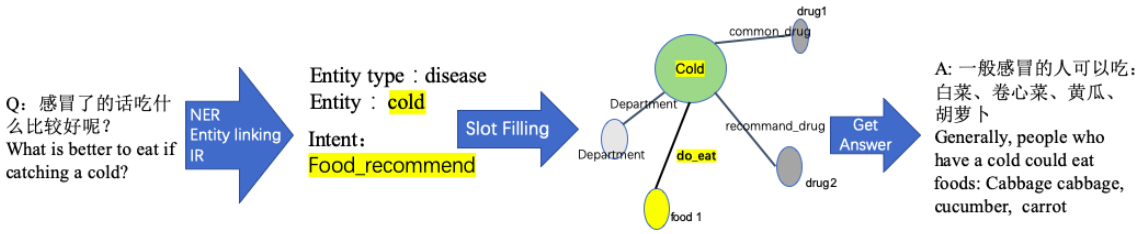

Figure 3.1: System structure

3.2 Build knowledge graph

In this paper, the knowledge graph is constructed mainly through the semi-structured dataset already established by liuhuanyong [1]. The data source comes from the Chinese medical question and answer webpage named 'Seeking Medicine'.
The semi-structured data are transferred to a database, and the medical knowledge graph is constructed through the Neo4j open source graph database platform, and the corresponding visualization interface is shown in Figure 3.2.

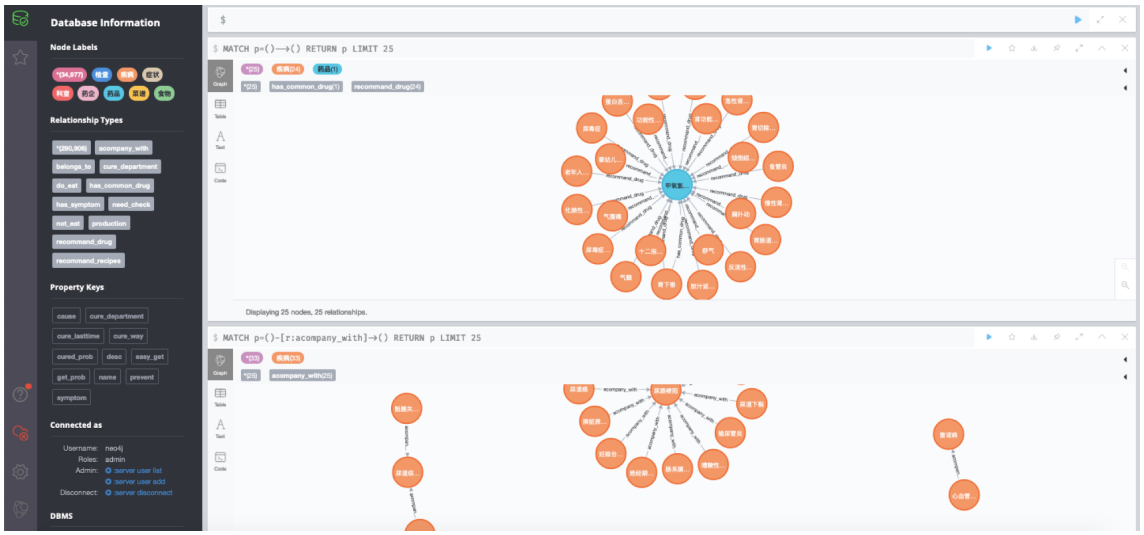

Figure 3.2: Knowledge graph stored in Neo4j

The storage form is a series of Disease, Relationship, Othertriples. The disease node is the center, and the surrounding departments, symptoms, tests, food, drugs and pharmaceutical companies are directly or indirectly associated with it. There are 34,977 entities and 29,906 relationships.

There are 8 types of entity nodes, such as medicine, check, disease, symptoms, departments, pharmaceutical companies, recipe and food. The relationship types are 'acompany_with', 'belongs_to', 'cure_department', 'do_eat', 'has_common_drug', 'has_symptom', 'need_check', 'not_eat', 'production', 'recommand_drug', 'recommand_recipes'.

The disease class entity also has attributes, with 11 types of attributes: 'cause', 'cure_department', 'cure_lasttime', 'cure_way', 'cure_prob', 'name', 'prevent', 'symptom'. Some of the nodes of the medical knowledge graph constructed by Neo4j are shown in Figure 3.2, and the number of each type of nodes is shown in the Figure 3.3.

[1] https://github.com/liuhuanyong/QASystemOnMedicalKG

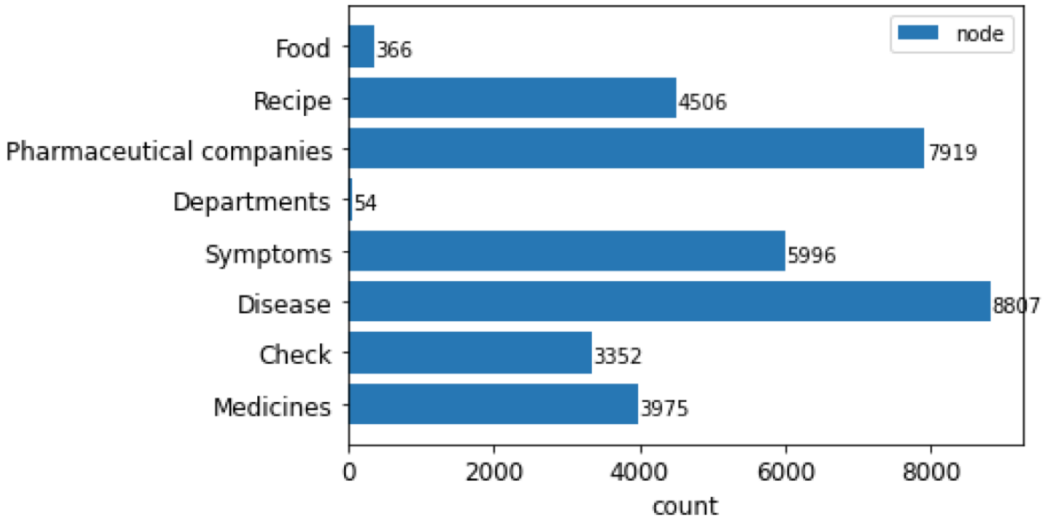

Figure 3.3: Amount of different entity type

3.3 Named entity recognition

With the goal of identifying named entities in text and allocating them to the respective entity types, named entity recognition seeks to locate and categorize named entities in text into pre-defined categories. It can be divided into rule-based, unsupervised, and supervised learning methods. This study uses template matching in combination with the BiLSTM-CRF [20] model for named entity identification.

To better incorporate the features of the knowledge graph, named entity recognition based on template matching [21] is first used. When there is no entity in the user input question that exactly matches the template, model-based named entity recognition is used to complement the results.

Named entity recognition based on template matching: First, all disease entities in the constructed knowledge graph are used as feature words, and the Aho-Corasick automation is used to support fast determination of whether the feature words are within the input question sentence. If it exists, it would be used as the named entity result of the question. The entity type will be quickly matched to by a dictionary with the entity name as key and the entity type as value.

Template matching can well identify entities in the user's question sentence that are consistent with those in the knowledge graph, but many times, template matching also fails to detect valid entities. This is because users often enter non-standard entity words or entity words that do not exactly match the knowledge graph. In this case, it is necessary to use the BiLSTM-CRF model to identify the entities and entity types in the current question and match them with the subsequent entity links.

Many studies have shown that BiLSTM-CRF has reached or surpassed the rich feature-based CRF model to become the most mainstream model among the current deep learning-based NER methods [22]. The model inherits the benefits of deep learning approaches without the need for feature engineering, and it performs well with both word and character

vectors.

The BiLSTM-CRF model applied in NER mainly consists of an Embedding layer (mainly with word vectors, character vectors, and some additional features), a bi-directional LSTM layer, and finally a CRF layer. First, the input of BiLSTM-CRF is the word vector and the output is the BIO sequence annotation of the prediction for each word.The BIO sequence annotation labels each element as'B-disease', 'I-disease' or 'O', where 'B-disease' means that the fragment in which the element is located is of type disease and the element is at the beginning of the fragment, 'I-disease' means that the fragment in which the element is located is of type disease and the element is in the middle of the fragment, and 'O' means that it does not belong to any type.

Step 1: After the text input, it enters the embedding layer in the form of words for embedding, which are mapped into word vectors using the Skip-gram model.

Step 2: The word vector is input into the BiLSTM layer, where the semantic expression vector of each word in its context is learned by BiLSTM, each word is multicategorized, and the score probability of each word for each label is produced.

Step 3: If the output is obtained by using the BiLSTM output directly as the final prediction, there may be cases where the model effect is reduced, such as I as the first word, the presence of two consecutive B's words, B-disease and I-body concatenated together, and so on. The CRF layer can learn the transfer probabilities between labels in the dataset by keeping track of a probability transfer matrix, which can then be used to correct the BiLSTM layer's output and ensure that the predicted labels are reasonable. The output of all BiLSTMs will be used as the input of the CRF layer to obtain the final prediction results by learning the order dependence information between the labels.

3.4 Entity linking

When the named entity is identified, if the result is from the BiLSTM-CRF model, the entity must be a name description that does not exactly match the disease name in the knowledge graph. For example, the relatively colloquial 'senile dementia' and 'Alzheimer's disease', where 'senile dementia' will be labeled by the model as a disease-like entity, but in the knowledge graph, the name of the disease is not exactly the same as the name of the disease. The 'senile dementia' is labeled as a disease-like entity by the model, but it cannot be found directly in the knowledge graph.

Thus, the entities identified by the model in the question but cannot be found in the knowledge graph need to be transformed into entity names stored in the knowledge graph. Therefore, a part of entity linking is added within this system.

For the identified entities, the BM25 algorithm will be used to recall all the entities of the same type within the knowledge graph and find the entities with top-k similarity ranking (the K value is set to 20 in this experiment), and then the trained ESIM model will be used to finely rank them and select the top-3 similarity models as the possible results of the final entities that are used in the slot filling process.

Among them, the ESIM model was trained using other medical dataset alone, the training set was Yidu-N7K competition data [2], which contains mainly spoken/non-standard

[2]http://www.openkg.cn/dataset/99e3fa10-c5f3-4af8-b147-fe689e67e260

medical terms, with the corresponding standard medical terms. In this experiment, the dataset was appropriately processed to fill in the negative data: for the non-standard medical nouns, the corresponding similar entities were recalled in the knowledge graph (the knowledge graph that comes with the competition data, not the knowledge graph in this system) using BM25, and the entities among them that are exactly the same as the standard medical nouns were removed, and the remaining entities were used as negative data.

The dataset with both positive and negative data is put into the ESIM [23] model for training, which eventually makes ESIM learn the matching probability between the non-standard medical text and the standard medical text, so it can accomplish the task of matching and making ranking between the entities recalled by BM25 and the entities in the QUESTION.

For non-standard/spoken entities, after adding the recall of BM25 and the refined ranking data of ESIM, the ranking can match well with the name of the entity that the user originally wanted to ask if the top 3 entities are taken as the result. Example of the effect is shown in Table 3.1

Question	How should blood cancer be treated?
Template matching	unrecognized
BiLSTM-CRF	< entity: 'blood cancer', type: 'disease' >
Entity linking	Leukemia

Table 3.1: Example of entity linking

3.5 Intent recognition

Intent recognition is to determine what the user wants to do. For example, a user asks a question to a robot, so the robot needs to determine whether the user is asking about the weather, a trip or information about a movie? In the end, intent recognition is a text classification problem. In order to complete the intent recognition task of this system, I chose to use BERT to complete the word embedding after using textcnn for the text classification task.

BERT [24] is a pre-trained model proposed by Google AI Institute in October 2018 and the model itself has been pre-trained using a large corpus of sentences. Training is done by masking a number of words (about 15% of words) in a sentence, and then letting the model predict those masked words. As the model is trained to predict, it learns to generate a robust internal representation of the words, i.e., word embeddings. BERT is built on top of transformer and has powerful language representation and feature extraction capabilities, and because it has an inheritable pre-trained model, it can be easily combined with many kinds of classification models to accomplish classification tasks.

About the TextCNN: Convolutional Neural Network (CNN) comes from the field of computer vision. Its main idea is to use many small segments of convolving filters applied to a two-dimensional image, scrolling along the x and y axes on the image to discover local features. In the text domain, a sentence is a one-dimensional structure, but can be turned into a two-dimensional structure by word vectorization. Chen et al. [25] applied CNN to a text classification task by using multiple kernels of different sizes to extract key

information in a sentence (similar to a multi-window size ngram), which can better capture local relevance, hence TextCNN is available.

In this system, a text vector is first obtained by encoding the input text using the BERT model, and then the vector is fed into the TextCNN model. Finally, after the convolutional, pooling, fusion and fully connected layers and then the Softmax layer, the classification probability of the text is obtained and the label with the highest classification probability is taken as the prediction result.

Because the model-based intent recognition results may be biased and may sometimes be directly classified as 'Others' because the actual intent cannot be identified, the system also uses template matching-based intent recognition to complement the model-based intent recognition.

When the intent recognition model cannot determine the intent of the current question (the confidence level of the recognition result is too low or the recognition result is 'Others'), it will retrieve whether the question contains the query words from the set template, such as 'what to eat' 'medication', etc., to classify the intent. Examples of usage are shown in the Table 3.2.

Question	'My wife is diabetic and I usually prepare the breakfast in our house, what should I make for her to eat better?'
BERT-TextCNN	<intent:'Others'>
Template matching	<intent:'Food recommend'>

Table 3.2: Example of template matching of intent recognition

The final result of the intent classification is also used in conjunction with the slot function to confirm the intent of the question with the user when it enters the slot-filiing step. The specific logical sequence is as follows:

- If the result of the model is not 'Others':
  - If the confidence level is greater than 0.8: go to slot filling and queries the answer directly.
  - If the confidence level is greater than 0.6 and less than 0.8: go to slot filling and query the answer after confirming with the user.
  - If the confidence level is less than 0.6: select the result of the template match, enter the slot filling and check the answer after confirming with the user.
- If the result of the model is 'Others':
  - Select the result matched by the template, go to slot filling and check the answer after confirming to the user.

3.6 Slot-filling

The essence of semantic slot filling is to predefine the logic templates corresponding to different entity types and intent types, and to predefine the corresponding operations to be

performed after filling in the corresponding information. Usually, the number of intents in an application scenario requires the prior definition of the corresponding semantic slots.

For example, when the input statement is 'What to do if you have epilepsy', the named entity recognition and intent recognition modules can identify <entity:epilepsy, type:disease > and <intent:treatment> respectively. After filling in the corresponding content, the cypher statement 'MATCH(p:disease) WHERE 'p.name=epilepsy' can be used. RETURN p.cure_way', etc., to query and return the answer within the knowledge graph.

In addition, based on the slots, it is possible to output rhetorical questions to correct the intent results based on the pre-defined rhetorical templates under the slots when the confidence level of intent recognition is low or when the intent recognition is entirely from the rule templates. For example, if the disease entity is known to be 'epilepsy', the user can be asked 'Are you asking about a treatment for epilepsy?' when the user is not sure if the intent is to query for a treatment. After getting an affirmative answer, we can formally query the knowledge graph and return the answer and present it to the user according to a discourse template, e.g., 'The treatment for epilepsy is ' plus the answer from the knowledge graph.

Because the categories of intent recognition are not exactly equal to the types of relationships and attributes in the knowledge graph, multiple relationships or attributes are queried for an intent at the same time in the slot filling process. For example, when the intent is 'Treatment', the pre-defined cypher statement in the slot finds and returns the 'cure_way' attribute for the corresponding disease, as well as the entities that have 'recommand_drug' and 'recommand_recipes' relationships with that disease. This approach takes good care of the many possibilities of the user's questioning intents. An example of one of the slots in the slot fill is as Table 3.3.

```
semantic_slot = {
 ...
 'Definition':{
 'slot_list':['Disease'],
 'slot_values':None,
 'cypher_template': 'MATCH(p:disease) WHERE
 p.name='Disease' RETURN p.desc',
 'reply_template': 'Disease' is ',
 'ask_template': 'Did you ask about the definition of 'Disease',
 'deny_response': 'I'm sorry I didn't understand you.'
 },
 ...
}
```

Table 3.3: Example of slot-filling

Chapter 4

Experiment and evaluation

4.1 Experimental results of models

4.1.1 BiLSTM-CRF

In the named entity recognition section, the BiLSTM-CRF model uses a dataset from the open source dataset cMedQANER [26]. 11 entity types are used: 'drug', 'body', 'treatment', 'physiology', 'disease', 'symptom', 'crowd', 'department', 'test', 'feature'and 'time'.

There are 1,673 question sentences in the training set with about 177,502 Chinese characters, 174 question sentences in the validation set with about 15,199 Chinese characters, and 215 question sentences with about 18,777 Chinese characters in the test set.

The model is built based on Keras, and the CRF model uses the CRF model built by bojone [1], and the Loss function of CRF is also directly followed. The data embedding part is done using Word2vec from the gensim library. The hyperparameters of the model are set as : 'epochs = 80', 'batch_size = 32', 'max_len = 128', and 'vocab_size = 2,410', 'embedding_dim = 200' and 'lstm_units = 128'

Figure 4.1 shows the variation of the accuracy of the model training process plotted by matplotlib, and Table 4.1 shows the Micromean scores of the model on the cMedQANER data set,in which the F1 score is 0.7163, and precision score is 0.7467, and recall score is 0.6884, and accuracy score is 0.9329.

[1] https://github.com/bojone/crf

entity type	precision	recall	F1 score	support
drug	0.5000	0.3443	0.4078	61
body	0.7473	0.6070	0.6699	229
treatment	0.5912	0.6483	0.6184	145
physiology	0.9459	0.8140	0.8750	43
disease	0.7582	0.7859	0.7718	411
symptom	0.8298	0.6903	0.7536	226
crowd	0.8553	0.8442	0.8497	77
department	0.6667	0.7500	0.7059	8
test	0.5952	0.5319	0.5618	47
feature	0.9630	0.9286	0.9455	28
time	0.7500	0.2903	0.4186	31
micro avg	0.7467	0.6884	0.7163	1306
macro avg	0.7478	0.6884	0.7121	1306

Table 4.1: Result of BiLSTM-CRF

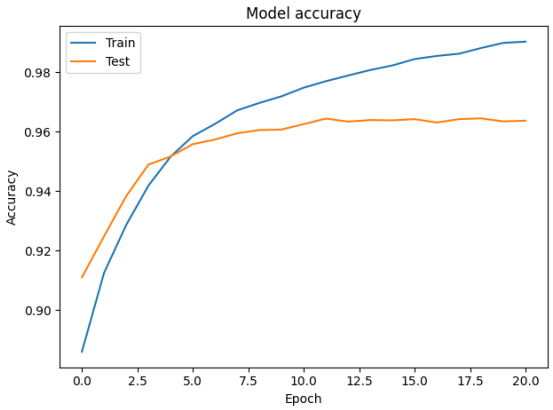

Figure 4.1: BiLSTM-CRF

4.1.2 BERT-TextCNN

The dataset used for intent recognition is the CMID [2] open source dataset, with 13 types of intent recognition types, namely 'Definition', 'Cause', 'Prevention', 'disease manifestations', 'Associated conditions', 'Treatment', 'Department', 'Infectious', 'Cure rate', 'Taboo', 'Physical Examination Program', 'Treatment time', 'Others'.

There are 7,274 question sentences in the training set and 810 question sentences in the test set.

The BERT model is constructed based on the bert4keras library under the Keras framework, using the loss function 'sparse_categorical_crossentropy' and the optimizer is Adam(5e-6) .The hyperparameters in the model are set as follows: 'class_nums = 13', 'maxlen = 128', 'batch_size = 8', 'epoch = 10'.

Figure 4.2 shows the accuracy of the model training process plotted by matplotlib, and Table 4.2 shows the Micromean scores of the model on the CMID data set.The training

[2]https://github.com/liutongyang/CMID

intent type	precision	recall	F1 score	support
Definition	0.70	0.63	0.67	41
Cause	0.58	0.56	0.57	90
Prevention	0.81	0.54	0.65	24
disease manifestations	0.53	0.61	0.57	141
Associated conditions	0.88	0.39	0.54	18
Treatment	0.76	0.78	0.77	169
Department	1.00	1.00	1.00	9
Infectious	0.75	1.00	0.86	9
Cure rate	0.88	0.66	0.75	32
Taboo	0.47	0.58	0.52	31
Physical Examination Program	0.90	0.96	0.93	28
Treatment time	0.93	0.87	0.90	15
Others	0.66	0.66	0.66	202
accuracy			0.67	809
macro avg	0.76	0.71	0.72	809
weighted avg	0.68	0.67	0.67	809

Table 4.2: Result of BERT-TextCNN

early stopped in epoch 5. The val accuracy in test dataset is 0.6947 and the loss is 0.946 and the F1 score is 0.67.

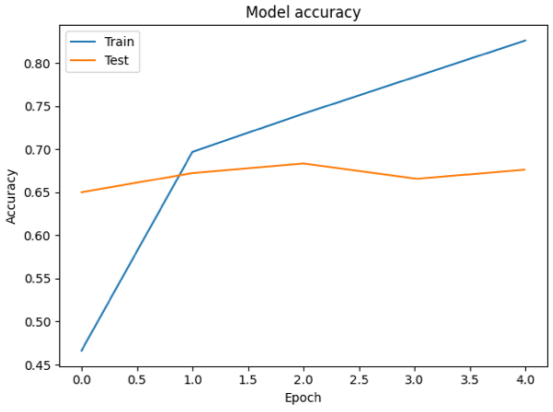

Figure 4.2: BERT-TextCNN

4.1.3 ESIM

The BM25 used for entity linking is from the gensim.summarization library, and the ESIM used for sorting is trained using the open source dataset Yidu-N7K. Because most of the datasets are non-standard terms and standard terms for medical procedures, but the main need in this system is to link to disease-type entities, so I collected and added 300 diseases with corresponding standard term data after searching by myself, and constructed negative data for the whole dataset, and the modified training set has 67,554 data and the test set has 1,013 data.

During the training process, the model is built using Keras, the loss function is 'cat-

egorical_crossentropy', and the optimizer is 'adam'. The hyperparameters are set as: 'batch_size=64', 'epochs=30'.

Figure 3.1 shows the accuracy variation of the model training process plotted by matplotlib,and the accuracy is 0.96 in the test dataset. Accuracy in this experiment represents the accuracy of esim in determining whether two texts (non-standard terms and standard terms) match or not.

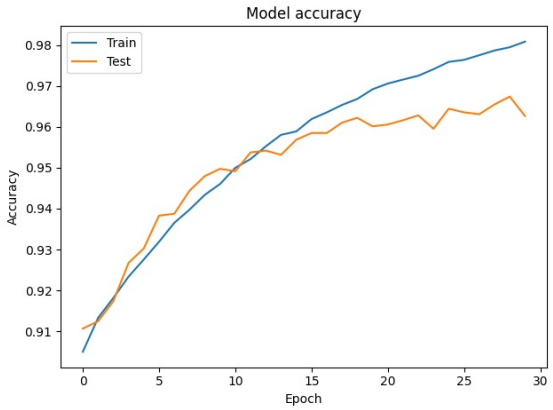

Figure 4.3: ESIM

label	precision	recall	F1 score	support
1	0.93	0.95	0.93	313
0	0.97	0.97	0.97	700
accuracy			0.96	1013
macro avg	0.95	0.96	0.95	1013
weighted avg	0.95	0.96	0.96	1013

Table 4.3: Result of ESIM

4.2 System effectiveness evaluation

The main advantage of using multiple models in this system over the past systems based on template matching is that the models can identify and connect to the knowledge graph for ambiguous entities, and the models can detect the real intent of the user in the process of intent recognition for questions that do not use templates. At the same time, the system still uses template matching as a backing and supplement in each step to better combine multiple models together to accomplish the KBQA task. The system can answer a total of 13 disease-related questions, namely: 'Definition', 'Cause', 'Prevention', and 'Disease manifestations', 'Associated conditions', 'Treatment', 'Department ', 'Infectious', 'Cure rate', 'Taboo', 'Physical Examination Program', 'Treatment time', 'Food recommend '.

Table 4.4 depicts the entity attributes or relationships that need to be used to answer the different categories of questions and the total number of such attributes or relationships in the knowledge graph. It also describes the accuracy of the model in recognizing the

intent of different categories of questions and the number of templates for different intents. These give a rough indication of the ability of the model to answer different categories of questions.

Question categories	Number of corresponding attributes or relationships	Precision of intent recognition model	Number of feature words in the template
Definition	'description':7,906	0.70	19
Cause	'cause':7,905	0.58	11
Prevention	'prevent':7,906	0.81	17
Disease manifestations	'symptom':7,906 'has_symptom':54,710	0.53	10
Associated conditions	'acompany_with':12,023	0.88	12
Treatment	'cure_way': 7,905	0.76	12
Department	'cure_department': 8,806	1.00	10
Infectious	'get_prob':7,906	0.75	7
Cure rate	'cured_prob':7,906	0.88	8
Taboo	'not_eat': 22,239	0.47	11
Physical Examination Program	'need_check':39,416	0.90	16
Treatment time	'cure_lasttime':7,905	0.93	6
Food recommend	'do_eat':22,230 'recommend_recipes':40,221	Null	13

Table 4.4: Evaluation of ability to answer different questions

To confirm the effectiveness of this system while comparing the system based on template matching only, in addition to the experimental results of different partial models, after the system was completed, I constructed 200 questions by myself and entered them into the system to manually judge whether they were correct results.

The 200 questions were created with artificially controlled variables, in which every 50 questions were divided into 4 groups. The first group was constructed by freely combining disease entities from the knowledge graph with some feature words from the intent recognition template. The second group consists of questions constructed from diseases entities in the knowledge graph and feature words outside the intent recognition template. The third group consists of problems constructed from diseases entities outside the knowledge graph with feature words in the intent recognition template. The fourth set of questions is constructed from diseases entities outside the knowledge graph and feature words outside the template (because the entity linking process outputs 3 answers to ensure that the correct answer is covered, and as long as there is a correct result within the 3 answers, the answer is considered correct).

The template matching system in the experiments uses the rule templates constructed by people who created medical knowledge graphs. For each question, the answers are manually marked correct or incorrect and then counted, and the percentage of correct answers in each group are shown in the Table 4.5.

Question group	Use diseases entities in the knowledge graph	Use intent words in the template	Correct answer rate of template matching based system	Correct answer rate of this system
1	Y	Y	100%	100%
2	Y	N	6%	90%
3	N	Y	2%	84%
4	N	N	0%	76%

Table 4.5: Evaluation of the system

The reason why there are still questions that can be answered by template matching when no template or no standard disease entity is used is because of the peculiarities of Chinese characters, where words that are not perfectly consistent are still partially matched by the template by chance. However, it is easy to see that once the range set by the template is exceeded, This system will be better than a template matching based system.

Chapter 5

Conclusion

A medical KBQA system is designed in this work by combining multiple models and using template matching. Knowledge graph building, named entity recognition, entity linking, intent recognition, and slot filling are the most important procedures. Among them, named entity recognition uses template matching and BiLSTM-CRF model, entity linking uses BM25 and ESIM model, and intent recognition uses BERT-TextCNN model and template matching. Multiple open source datasets were used to train different models, and each model produced good results. The focus of this study is on how to use the models to do multiple functions and combine them into a system that can better execute the QA function, as the different models were produced by other researchers. This system performs substantially better in the question and answer function than the system that only uses template matching.

Meanwhile, this research has a lot of potential for improvement. To begin with, the current system can only respond to inquiries about the disease itself and cannot deduce the disease from the user's symptoms. Second, the intent recognition part's accuracy isn't great, and we can try using different models to improve the outcomes. Moreover, named entity recognition and intent recognition in this system are trained using different models separately because there is no same dataset that can be used for both tasks. However, future attempts can be made to jointly train the two tasks with the same model using two datasets, which may better improve the capability of the system. For the entity linking part, additional text matching algorithms might be considered, and it would be better if a large dataset only about 'disease description, disease criteria words could be utilized for training.

Acknowledgements

First and foremost, I am grateful to the University of Tsukuba for providing a conducive research environment as well as numerous courses. I'd also like to convey my gratitude to Mr. Tezuka, as well as my advisor Mr. Wakabayashi and my associate advisor Mr. Yu, for all of their help and advice. In addition, I'd like to acknowledge the scholarly efforts of many researchers working in this field, as well as the authors of several open source datasets and model source codes.

References

[1] Y Lin, L Siyun, Influencing factors and improvement of unbalanced allocation of medical resources in urban and rural areas[J]. Economic Trends, 2016 (9): 57-68.

[2] Report on the Search Behavior of Chinese Netizens for Science Popularization Needs (Q1 2019) [EB/OL]. [2021-06-19]. http://www.kepuchina.cn/notice/ss/.

[3] Ziang Z, Jihaeng L. Research on Development and Challenges of Chinese Medical Artificial Intelligence[C]//2021 International Conference on Public Management and Intelligent Society (PMIS). IEEE, 2021: 371-375.

[4] Wang Q, Mao Z, Wang B, et al. Knowledge graph embedding: A survey of approaches and applications[J]. IEEE Transactions on Knowledge and Data Engineering, 2017, 29(12): 2724-2743.

[5] Ji S, Pan S, Cambria E, et al. A survey on knowledge graphs: Representation, acquisition, and applications[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021.

[6] Berners-Lee T, Cailliau R, Groff J F, et al. World-Wide Web: the information universe[J]. Internet Research, 1992.

[7] Cui Z, Kapanipathi P, Talamadupula K, et al. Type-augmented relation prediction in knowledge graphs[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2021, 35(8): 7151-7159.

[8] Wu T, Qi G, Li C, et al. A survey of techniques for constructing Chinese knowledge graphs and their applications[J]. Sustainability, 2018, 10(9): 3245.

[9] Yang F, Gan L, Li A, et al. Combining deep learning with information retrieval for question answering[M]//Natural Language Understanding and Intelligent Applications. Springer, Cham, 2016: 917-925.

[10] Shen W, Wang J, Han J. Entity linking with a knowledge base: Issues, techniques, and solutions[J]. IEEE Transactions on Knowledge and Data Engineering, 2014, 27(2): 443-460.

[11] Zou L, Huang R, Wang H, et al. Natural language question answering over RDF: a graph data driven approach[C]//Proceedings of the 2014 ACM SIGMOD international conference on Management of data. 2014: 313-324.

[12] Yih S W, Chang M W, He X, et al. Semantic parsing via staged query graph generation: Question answering with knowledge base[J]. 2015.

[13] Berant J, Liang P. Semantic parsing via paraphrasing[C]//Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2014: 1415-1425.

[14] Bast H, Haussmann E. More accurate question answering on freebase[C]//Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. ACM, 2015: 1431-1440.

[15] Ben Abacha A, Zweigenbaum P. Medical question answering: translating medical questions into sparql queries[C]//Proceedings of the 2nd ACM SIGHIT international health informatics symposium. 2012: 41-50.

[16] Zhang S, Zhang X, Wang H, et al. Chinese medical question answer matching using end-to-end character-level multi-scale CNNs[J]. Applied Sciences, 2017, 7(8): 767.

[17] Abacha A B, Zweigenbaum P. MEANS: A medical question-answering system combining NLP techniques and semantic Web technologies[J]. Information processing management, 2015, 51(5): 570-594.

[18] Feng G, Du Z, Wu X. A Chinese question answering system in medical domain[J]. Journal of Shanghai Jiaotong University (Science), 2018, 23(5): 678-683.

[19] Jiang Z, Chi C, Zhan Y. Research on medical question answering system based on knowledge graph[J]. IEEE Access, 2021, 9: 21094-21101.

[20] Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging[J]. arXiv preprint arXiv:1508.01991, 2015.

[21] Ben Abacha A Zweigenbaum P. Automatic extraction of semantic relations between medical entities A rule based approach[J].Journal of Biomedical Semantics20112(5)1-11.

[22] Panchendrarajan R, Amaresan A. Bidirectional LSTM-CRF for named entity recognition[C]//Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation. 2018.

[23] Chen Q, Zhu X, Ling Z, et al. Enhanced LSTM for natural language inference[J]. arXiv preprint arXiv:1609.06038, 2016.

[24] Devlin J, Chang M W, Lee K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[25] Chen Y. Convolutional neural network for sentence classification[D]. University of Waterloo, 2015.

[26] Zhang N, Jia Q, Yin K, et al. Conceptualized representation learning for chinese biomedical text mining[J]. arXiv preprint arXiv:2008.10813, 2020.