mathematics

MDPI

Article

Towards Knowledge-Based Tourism Chinese Question Answering System

Jiahui Li [1], Zhiyi Luo [1], Hongyun Huang [2,*] and Zuohua Ding [1]

[1] School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China; 201920603004@mails.zstu.edu.cn (J.L.); luozhiyi@zstu.edu.cn (Z.L.); zuohuading@zstu.edu.cn (Z.D.)
[2] Library, Center of Multimedia Data Analysis, Zhejiang Sci-Tech University, Hangzhou 310018, China
* Correspondence: zs0040@zstu.edu.cn

Abstract: With the rapid development of the tourism industry, various travel websites are emerging. The tourism question answering system explores a large amount of information from these travel websites to answer tourism questions, which is critical for providing a competitive travel experience. In this paper, we propose a framework that automatically constructs a tourism knowledge graph from a series of travel websites with regard to tourist attractions in Zhejiang province, China. Backed by this domain-specific knowledge base, we developed a tourism question answering system that also incorporates the underlying knowledge from a large-scale language model such as BERT. Experiments on real-world datasets demonstrate that the proposed method outperforms the baseline on various metrics. We also show the effectiveness of each of the question answering components in detail, including the query intent recognition and the answer generation.

Keywords: knowledge graph; tourism question answering system; intent recognition; language model

Citation: Li, J.; Luo, Z.; Huang, H.; Ding, Z. Towards Knowledge-Based Tourism Chinese Question Answering System. *Mathematics* **2022**, *10*, 664. https://doi.org/10.3390/math10040664

Academic Editors: Pasquale De Meo and Oliviu Matei

Received: 5 January 2022
Accepted: 17 February 2022
Published: 20 February 2022

1. Introduction

Question answering (QA) systems [1] are a popular research direction within natural language processing and artificial intelligence. A QA system is a form of information retrieval system. The natural language problem is transformed into a query statement that can be recognized by a machine. The answer is then returned in the form of natural language by querying a knowledge graph [2]. The general question answering systems include three parts: question analysis, information retrieval, and answer generation. The domain question answering system needs to be constructed in detail.

At present, there are many Chinese question answering systems, most of which establish datasets for a certain field. For example, Lin [3] proposed a QA system based on e-commerce. Das [4] examined case-based knowledge base natural language query reasoning, and Huang [5] studied a knowledge graph based on a question answering method for the medical domain. Sheng [6] proposed a QA system, DSQA, based on a knowledge graph for answering domain-specific medical questions. The above systems are based on a knowledge base, and require the design of algorithms for the retrieval of the knowledge or the processing of problems.

In view of the development of tourism in recent years, tourism websites have steadily emerged. However, searching for the desired information on tourism websites requires users to judge the scenic spots, the province or region, and the information they need. Thus, the process is cumbersome and time consuming. Users need to search several websites to obtain the information they need, and the result of the query is often unnecessary for what they want. The establishment of a QA system based on tourism will result in great convenience for users. In the past, people had to obtain information through manual consultation when they visited scenic spots, e.g., by asking the staff at the scenic spot for help, which may lead to inaccurate answers. The emergence of question answering systems centered on natural language processing technology have reduced unnecessary time consumption

when people visit scenic spots. Moreover, the process of natural language can enable the machine to judge people's intentions and query the scenic spots as needed. Furthermore, it can provide more accurate results. Compared with the traditional information retrieval method based on search engines, question answering systems can return more concise and accurate answers to users, which improves the efficiency of information retrieval to a certain extent.

Although there are currently many tourism QA systems for specific regions, no such system exists for Zhejiang tourism. As a major tourism province, Zhejiang is known as Hangzhou, and is referred to as "heaven above and Suzhou and Hangzhou below". Zhejiang is the location of many tourist attractions, so it is necessary to establish a tourism QA system for Zhejiang tourism.

Therefore, in this study, we designed and implemented a question answering system for Zhejiang tourism. Our system can answer people's questions about Zhejiang's scenic spots. We used Baidu keywords to determine the attributes of scenic spots that are often used for queries and searches. We also asked about 100 people, comprising college students, parents, and teachers, to identify the attributes of scenic spots that are often queried. The attributes we used are common to these tourism websites. We obtained the data of the common attributes of tourism websites and the data for the Baidu keywords for some spots. We then designed necessary nodes and relationships, and constructed the knowledge graph of tourist attractions in Zhejiang. The knowledge graph is stored through neo4j, and a set of QA processes was established.

At present, many knowledge bases are represented by a knowledge atlas. When people ask a question, such as "what is the opening time of West Lake?", some systems will extract the name of the scenic spot, "West Lake", and the "opening time" of the content to be queried based on rules or templates. The templates and rules need to be sufficiently comprehensive. However, a name may be changed, such as changing "West Lake" into "Hangzhou West Lake", which will not be recognized by some systems. Similarly, the question may be replaced with "when does the West Lake open to the outside world?"; in this case, the system may not be able to accurately judge the content to be queried, and thus not use the words "time" or "opening". It may determine the correct results and may also query who the attraction is open to, or the times when there are a large number of people at West Lake. Thus, the accuracy of the system's answers is very limited.

In order to solve this problem, our system uses a classifier to recognize users' intention (see Figure 1).

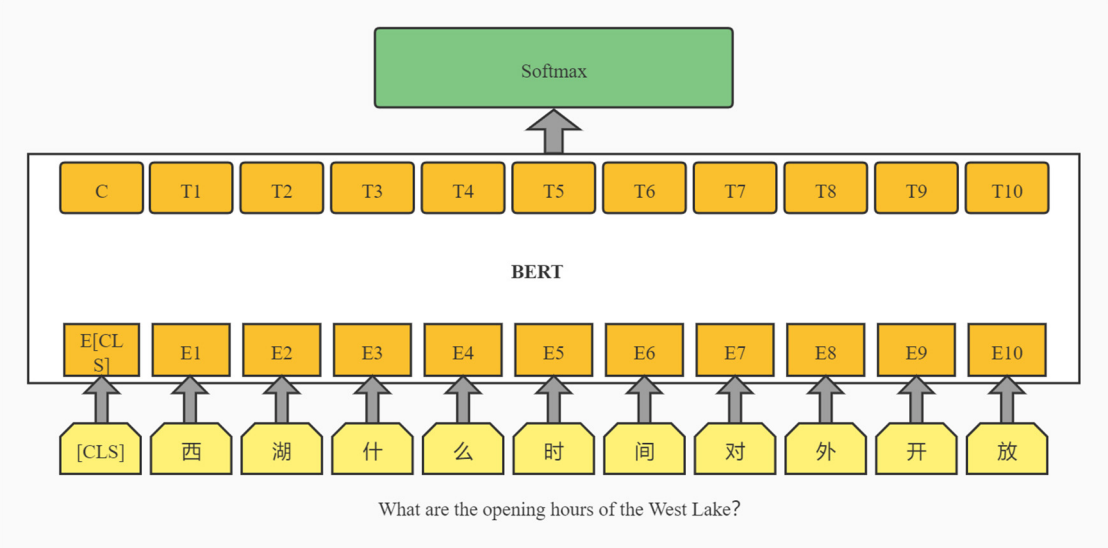

Figure 1. BERT model to perform single sentence classification tasks.

To ensure the knowledge graph is large enough, we adopted a semi-automatic approach by combining both machine learning and manual efforts. We extract information based on various tourism websites and conduct secondary processing. Scenic spots that are closed or have few visitors may be ignored, and missing attributes are searched for and completed using the search engine. If these attributes cannot be supplemented, they are filled with a null entry.

Through the training of downstream task classification using the BERT model [7], the intention of questions can be determined. For example, we designed 10 categories with numbers from 1 to 10, assuming that the number of the opening time intention is 7 (Label 7). After the above questions pass through the classifier, we can obtain the number 7 of the corresponding opening time as the category.

We can then obtain the name of the scenic spot, which is queried, and the result is obtained by searching the knowledge graph by extracting keywords from the original question. The result is organized into the form of natural language by combining the results with the name of the scenic spot and the identified intention. The results show that we can obtain accurate answers and feedback in the field of tourism in Zhejiang (see Figure 2).

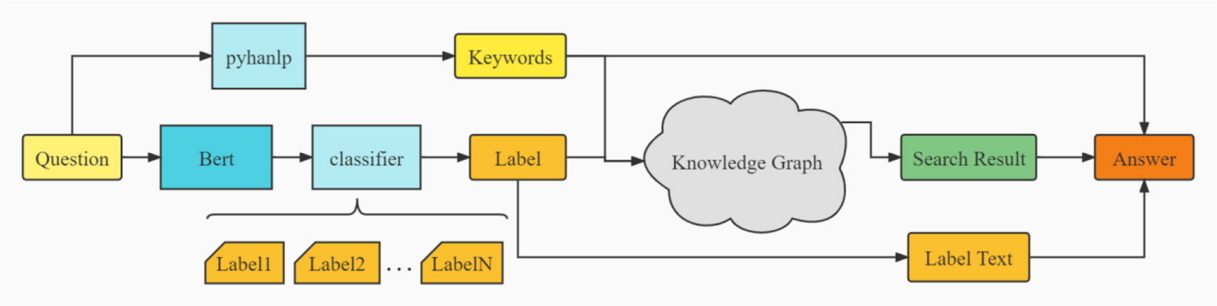

Figure 2. Overall system flow.

The organization of this paper is as follows. First, we introduce related works (Section 2). We then present the process and method of map establishment, the design of the classifier for intention recognition, and the approach for combining key information and retrieving it from the knowledge graph (Section 3). Section 4 presents the overall effect and result evaluation. Section 5 provides conclusions and describes possible future work.

2. Related Works

The main tasks for building a question answering system are the construction of the knowledge base, intention recognition, and the design of the question answering method.

Construction of the domain knowledge graph. Open domains already have a comprehensive knowledge base, such as freebase [8] and DBpedia [9]. Wang et al. [10] devised a question answering system based on templates on freebase. Templates are learned from millions of corpora to answer questions. Many domain knowledge bases also exist. For example, Liu et al. [11] constricted a knowledge map on COVID-19, and Goel et al. [12] built a personalized food knowledge map. Most of the knowledge bases in tourism are commercial and are applied on major tourism websites, such as Ctrip and Qunar. They are generally not open to the outside world. Moreover, many people collect information according to their needs and build their own knowledge base. Our knowledge map was built according to our needs.

Realization of intention recognition. Intention recognition is the key to the results and efficiency of question answering systems. For example, Wu et al. [13] proposed a semantic parsing method based on seq2seq to learn a type of problem by extracting the relationship between classes using the seq2seq model. This generates templates to match more complex problems with more problem types. Zhang et al. [14] were the first to consider the uncertainty estimation problem of KBQA tasks using BNN. They proposed an end-to-end model, which integrates entity detection and relationship prediction into

a unified framework. Their model uses BNN to model entities and relationships under the given problem semantics, and converts the network weight into distribution. The uncertainty can be further quantified by calculating the variance of the prediction distribution. The model showed good performance on a simple questions dataset. Wu et al. [15] proposed a neural union model with a shared coding layer to jointly learn topic matching and relationship matching tasks. A symmetrical two-way complementary attention module based on attention and gate mechanisms was designed to model the relationship between the two sub-tasks. Although the methods are different, this intention recognition method has high accuracy in our dataset. Tan et al. [16] developed a simple but effective attention mechanism for the purpose of constructing better answer representations according to the input question, which is imperative for better modeling of long answer sequences.

Design of question answering system. Question answering systems have become a popular research direction. For example, Li [3] and others designed a QA system based on structured knowledge of e-commerce customer service. They extended the classical "subject predicate object" structure through the attribute structure, key value structure, and composite value type (CVT), and enhanced the traditional KBQA through constraint recognition and reasoning ability. A QA system based on an intelligent maintenance knowledge base of a power plant was devised by Li [17]. They used the BiLSTM-CRF model and the fine-tuned BERT model to capture named entities and relationships in queries. Combined with the BM25 algorithm and the fine-tuned BERT model, a better information retrieval scheme was developed. Huang [5] proposed a medical domain question answering (KGQA) method based on a knowledge graph. Firstly, the relationship between entities and entities was extracted from medical documents to construct a medical knowledge graph. Then, a reasoning method based on weighted path ranking on the knowledge graph was proposed, which scores the relevant entities according to the key information and intention of the given problem. Finally, the answer is constructed according to the inferred candidate entities. Lai et al. [18] constructed a question answering system that can automatically find the commitment entities and predicates of single relationship problems. After the feature-based entity link component and the word vector-based candidate predicate generation component, a deep convolution neural network is used to reorder the entity predicate pairs, and all intermediate scores are used to select the final prediction answer. Zhou et al. [19] used the ranking method based on an alias dictionary to identify the named entities contained in the problem, and used the attention mechanism of BiLSTM to map the problem attributes. Then, the answer was selected from the knowledge base according to the results of the first two steps. Our method differs in the selection of keywords, and has certain advantages.

3. Approach

3.1. Tourism Knowledge Graph Construction

The basis of QA systems is whether the construction of the knowledge base is reasonable and appropriate. We considered that it would be too unilateral and incomplete if we crawled tourism information from only one tourism website, and thus crawled tourism information about Zhejiang from Ctrip (https://www.ctrip.com/, accessed on 20 November 2020), Tuniu (https://www.tuniu.com/, accessed on 24 November 2020), and Cncn (https://www.cncn.com/, accessed on 26 November 2020). We selected all of the scenic spots in Zhejiang, and collected all of the information relating to Zhejiang's scenic spots from the websites. Then, we filtered, integrated, and standardized the crawled information.

We designed a Java distributed crawler framework that can crawl the information of multiple tourism websites through multiple threads, and summarized the information in .xls files. In this way, the framework can also be applied to other systems to collect website information. Our framework structure is relatively simple (see Figure 3). We analyzed the page source code of each website, extracted the information under different labels, and then summarized it into a file. For use of the framework to collect information from other

websites, the labels in the code must be changed, but the overall framework does not need to be changed.

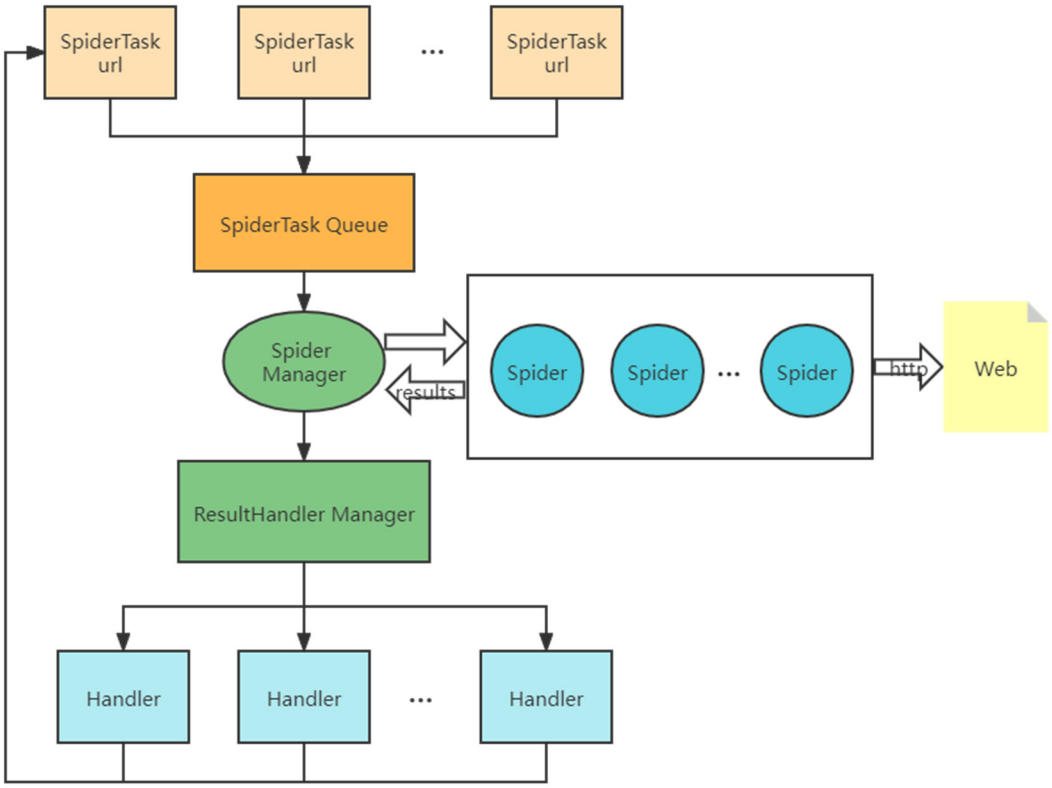

Figure 3. The framework of the crawler.

The structure of the framework is as follows:

- The crawler initiator creates a crawler task and adds it to the task queue.
- The crawler manager intermittently attempts to obtain all tasks from the crawler task queue and assigns them to the crawler to crawl with a new thread.
- The crawler sends an HTTP request to the URL described in the task and returns the result to the result processor connector. It then waits for the crawler manager to assign a task again.
- After receiving the result, the connector of the crawler result processor starts the processing thread and gives it to the corresponding result processor for processing.
- The crawler result processor processes the returned data (matches the desired content and saves it to the desired file). To crawl a deeper URL, a new crawler task can be created during the processing and added to the crawler queue.
- After we collected all the information relating to Zhejiang tourism, the crawled information must be screened, integrated, and standardize using the following measures:
- Screening: Screening is carried out based on two aspects. One is the selection of attribute entries. Based on our research on the information of various tourism websites and the popularity of user queries, the results show that the attributes of scenic spots are the scenic spot name, scenic spot category, grade, ticket price, opening time, the season that is suitable for playing, location, transportation, description, evaluation, contact number, strategy, and the relationship between scenic spots, including the small scenic spots. The second aspect is the choice of scenic spots. We ignore scenic spots if they have too little information or insufficient heat; are abandoned; have valuable information about fewer than three of the above attributes; do not have clear enough attributes; or whose attributes are too sparse.

- Integration: We first identify the names of the scenic spots we need and crawl the information of scenic spots by browsing various travel websites. Then, we use the intersection of each piece of information as a temporary result to view the description of each website and add necessary and valuable information. For some missing information, we conduct a secondary retrieval and fill in the information if it is a popular scenic spot. If the number of searches for the scenic spot is less than 200, we ignore it. In this way, the information integration is basically completed. If a scenic spot's name is not reasonable or the name is orally defined, the spot is directly delete such scenic spots. This process is artificially filtered and spots are deleted individually.
- Regularization: The collected scenic spot attribute information is organized into a table form. Firstly, regular expressions are used for normalized information processing. We then manually browse the description of each item to judge whether it is reasonable and smooth. We make slight modifications for unreasonable or non-smooth information, which are generally completed at the same time as integration. In most cases, the information provided by the websites is relatively standardized, so most of the information does not need to be modified. However, the transportation, description, and evaluation information of each website are different, and the grammar may be colloquial in some places. Thus, it is necessary to replace and delete this information.

The construction of the knowledge base was completed with a total of 2812 scenic spots. The next step is to store the information. Because of the relatively low number of attributes of the scenic spots, we used the lightweight graph database, neo4j. This is a high-performance graph engine with all the characteristics of a mature and robust database. It stores structured data on the network rather than in tables, so it can apply a more agile and rapid development model. More importantly, neo4j solves the issue of performance degradation in a traditional RDBMS by using a large number of connections when queries are made [20]. In summary, neo4j allows simple and diverse management of data. The addition and definition of data are relatively flexible and are not limited by the data type or quantity. In addition, neo4j uses the new query language, cypher, which simplifies the query statement.

Finally, our knowledge map contained 2812 nodes and 30,724 edges (see Figure 4).

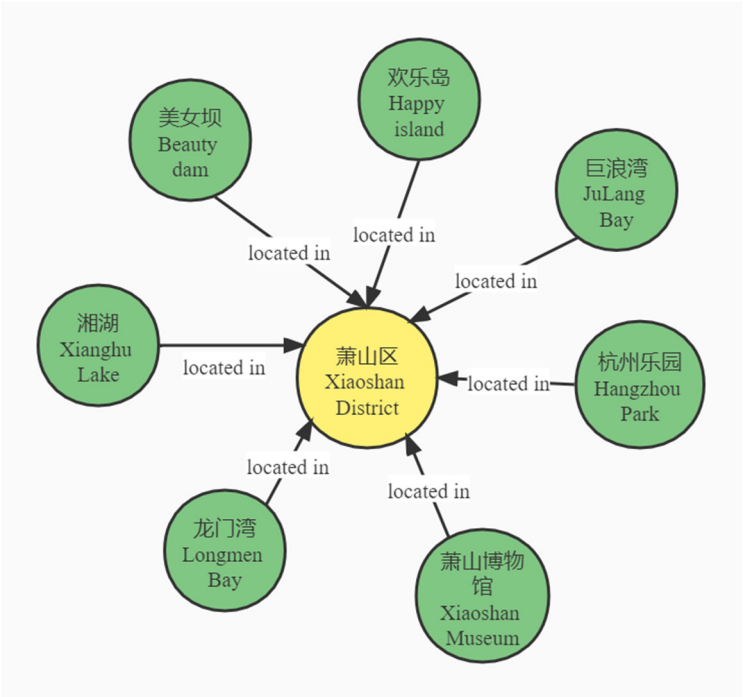

Figure 4. Knowledge graph of Zhejiang tourism.

3.2. Intention Recognition

The core of the question answering system is the accurate judgment of the essentials of the questions raised by users. At present, many question answering systems are based on a template matching method or regular matching. This approach is time and labor consuming, and requires experts to design the templates [21]. In addition, regular matching can only match the intention of specific statements and it is difficult to obtain accurate matching results for slightly changed intentions. Although the effect is good to a certain extent, this is due to the diversity of natural language and the ambiguity of a word. When new intentions are added, the templates need to be designed and a large number of manual interventions are required each time. Therefore, the method based on templates is not sufficient. By comparison, the classifier has a number of advantages. For example, the classifier does not need a large amount of manual intervention during either the initial creation or subsequent maintenance. Furthermore, the classifier can ensure more attention is paid to some characteristics. The classifier can solve the issues of the variability and polysemy of natural language and has high reliability.

We combined the popular BERT model to perform the downstream task classification. BERT was chosen for a number of reasons. First, unlike other language representation models, BERT aims to pre-train deep bidirectional representation by jointly adjusting the context in all layers. Therefore, the pre-trained BERT representation can be fine-tuned through an additional output layer [22], which can be more suitable for QA tasks. Second, although largely derived from the CBOW model, BERT uses a deeper model and a large number of corpora to become embedded. This means it has high accuracy and good generalization ability when performing downstream tasks.

The architecture and principles of the BERT model are discussed in the following.

The BERT model adopts the encoder structure of a transformer, but the model structure is deeper than that of transformer. The core of the model is composed of multiple BERT layers. In fact, the BERT layer of each layer is the encoder block in the transformer. Because BERT uses the encoder block in the transformer, it is effectively still a feature extractor. However, due to the deep network and the use of a self-attention mechanism, it has stronger feature extraction/model learning ability. Because it is only one encoder, BERT does not have the generation ability, and cannot solve the NLG problem alone but can solve the NLU problem. The input of the model is the vector of sentences. The encoder structures of BERT and the transformer have more segment embedding. BERT needs three vectors, token embedding, position embedding, and segment embedding. The output is the vector representation of each word in the text after integrating the full-text semantic information. We use Google's BERT base (https://github.com/google-research/BERT/, accessed on 20 March 2020), for which L = 12, H = 768, A = 12, and total-parameters = 110 m (see Figure 5).

The pre-training task has two aspects, masked LM and next sentence prediction.

Masked task is described as: given a sentence, randomly erase one or more words in the sentence and predict the erased words according to the remaining words, in the same manner as cloze tests. Specifically, 15% of the words in a sentence are randomly selected for prediction. For the words erased in the original sentence, 80% of the cases are replaced by special symbols [mask], 10% of the cases are replaced by arbitrary words, and the remaining 10% of the cases remain unchanged. In this way, the model can rely more on context information to predict vocabulary.

The task description of next sentence prediction is: given two sentences in an article, judge whether the second sentence follows the first sentence in the text, in a manner similar to paragraph reordering. Specifically, 50% correct sentence pairs and 50% incorrect sentence pairs are randomly selected from the text corpus for training. Combined with the masked LM task, the model can describe the semantic information more accurately at the sentence and even the text level.

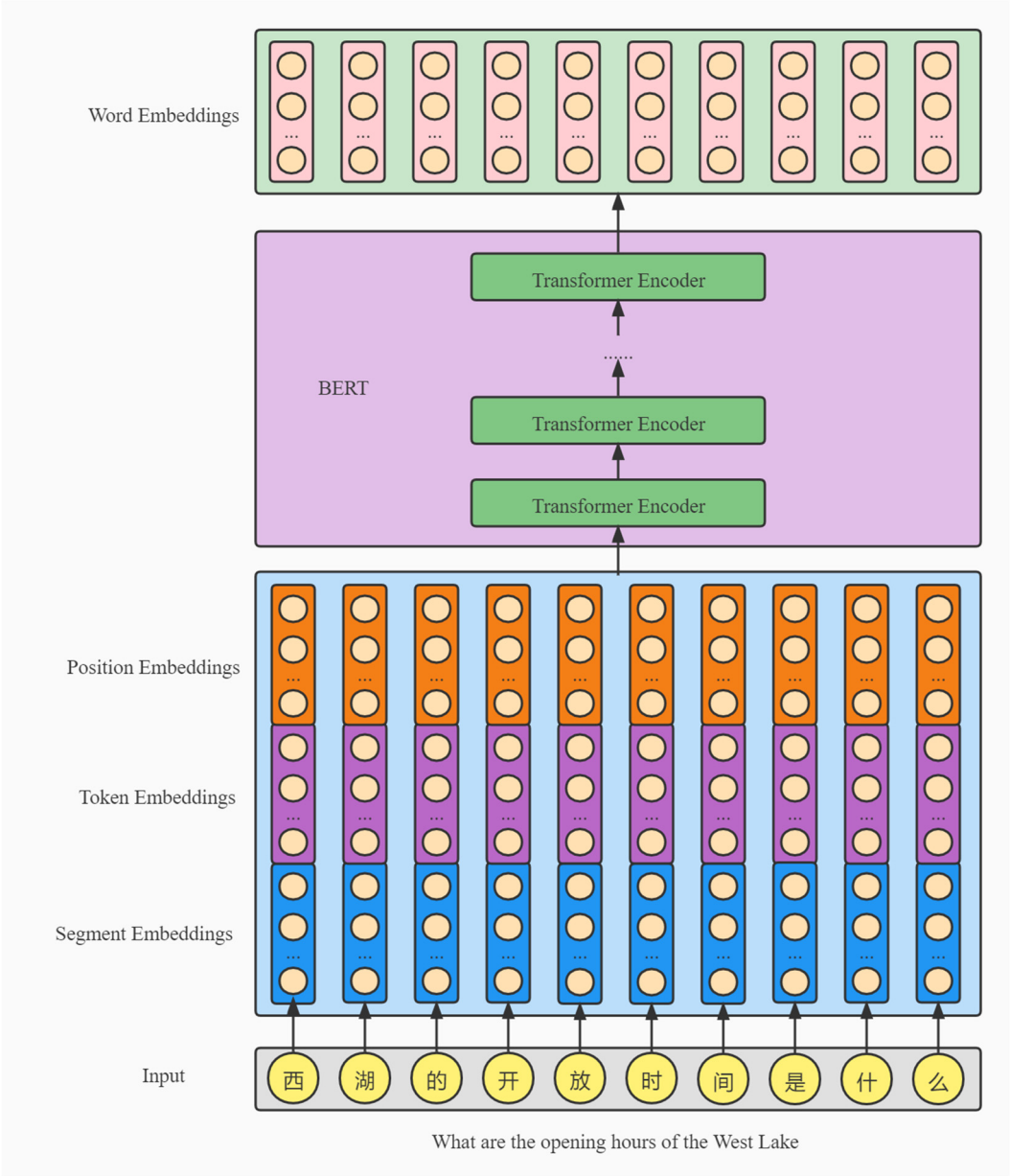

Figure 5. BERT model structure.

BERT's loss function consists of two parts. The first part is the "word level classification task" from mask LM, and the second part is the "sentence level classification task". Through the joint learning of the two tasks, the representation learned by BERT can contain both token level information and sentence level semantic information. The specific loss function is as follows:

$$L(\theta, \theta_1, \theta_2) = L(\theta, \theta_1) + L(\theta, \theta_2) \tag{1}$$

Parameters are present in the encoder part of BERT and the output layer connected to the encoder in the mask LM task, and the classifier parameters are connected to the encoder in the sentence prediction task. In the loss function of the first part, if the word set is marked M, and it is a multi-classification problem on the dictionary size $|V|$, then specifically:

$$L_1(\theta, \theta_1) = -\sum_{i=1}^{M} \log p(m = m_i | \theta, \theta_1), m_i \in [1, 2, \ldots, |V|] \tag{2}$$

In the sentence prediction task, it is also a loss function of a classification problem:

$$L_2(\theta, \theta_2) = -\sum_{j=1}^{N} \log p(n = n_i|\theta, \theta_2), n_i \in [IsNext, NotNext] \quad (3)$$

Therefore, the loss function of joint learning of the two tasks is:

$$L(\theta, \theta_1, \theta_2) = -\sum_{i=1}^{M} \log p(m = m_i|\theta, \theta_1) - \sum_{j=1}^{N} \log p(n = n_i|\theta, \theta_2) \quad (4)$$

Through the above two tasks, the BERT model introduces a two-way language model. The word embedding obtained by the BERT model better integrates the context information and can deal with the problem of polysemy to a certain extent. In this way, the understanding of input problems will be more comprehensive and accurate.

Secondly, we fine-tune the model on our own corpus (see Figure 1). A classification layer (full connection layer + soft-max layer) is added after the output corresponding to the [SEP] position, which is similar to the naive Bayesian classification algorithm [23]. The soft-max layer can expand the score gap. Even if the score results of the score function are not different, the score gap can be further widened through the soft-max classifier, thus making the classification effect more obvious [24]. In this way, the output is the label of the probabilities that we defined to prepare for the subsequent search of the knowledge graph and answer generation. Bayesian classification is based on the Bayesian principle, which is closely related to conditional probability. Bayesian decision theory uses misjudgment loss to select the optimal category classification when the correlation probability is known (https://blog.titanwolf.in/a?ID=00650-dbcda825-0970-4946-b94e-c3f1e40b7aaf, accessed on 27 July 2021). The naive Bayesian algorithm was designed because the attributes are independent of each other and the correlation between the attributes is small. In addition, the space overhead is also small, which is more suitable for our intention recognition. The Bayesian Formula (5) describes the probability of the occurrence of event B under the condition of the occurrence of event A. If event A is replaced with a feature and event B is replaced with a category, the formula is converted into the probability of belonging to category B under the condition of the occurrence of feature A. In this way, we extend the features and categories to multiple dimensions, and define them as categories according to the attributes and edge relations. Then we judge the features on the basis of attributes. Formula (6) can then be obtained.

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} \quad (5)$$

$$P(y_k|x) = \frac{P(x|y_k)P(y_k)}{P(x)} \quad (6)$$

We represent x as the data object with D as attributes, and the training set has K categories. According to the Bayesian theorem, we can obtain the probability of x belonging to the category y_k. For example, we can design 10 categories, namely, opening hours, category, time suitable for playing, scenic spot level, score, scenic spot introduction, location, scenic spot ticket price, playing strategy, and contact information. When we respond to the question "what is the score of Hangzhou West Lake?", we put the output of BERT model into the full connection layer for classification calculation. The classification function we use is:

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}} \quad (7)$$

The process we use is shown in Figure 6.

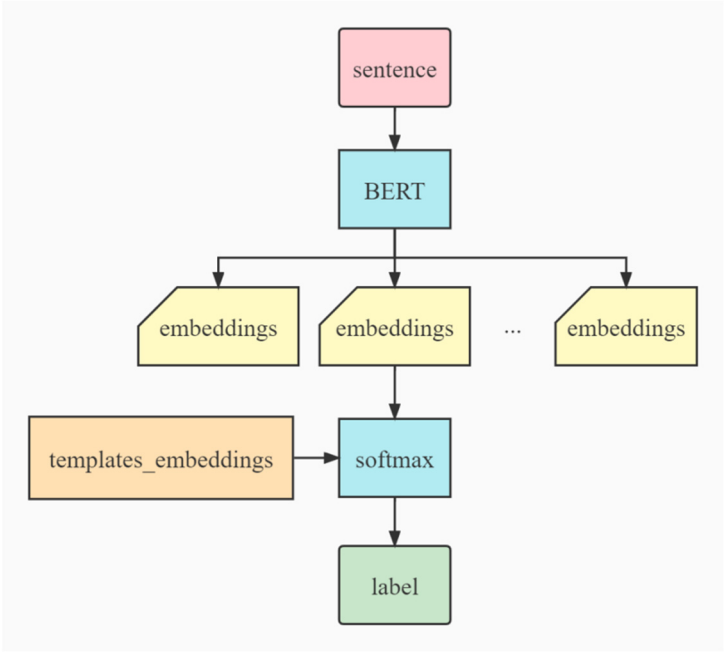

Figure 6. Soft-max Layer.

The model parameter θ is trained to minimize the cost function:

$$J(\theta) = -\frac{1}{\mathrm{m}}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})\right] \quad (8)$$

For a given test input x, the probability value $\mathrm{p}(\mathrm{y} = j|x)$ is estimated for each category j using the hypothesis function. The probability of each classification result of x is estimated. Suppose that the function outputs a k-dimensional vector to represent the k estimated probability values. The probability formula for classifying x into category j is:

$$p(y^{(\mathrm{i})} = j \Big| x^{(i)}; \theta) = \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}} \quad (9)$$

Thus, we obtain ten probabilities, which are 0.12, 0.07, 0.05, 0.04, 0.67, 0.02, 0.01, 0.01, 0.006 and 0.004, respectively, which indicate that the fifth category is the identified intention; therefore, the intention of this sentence is to ask for the score.

3.3. Answer Generation

The category can be obtained through intention recognition. After determining the recognized intention, we can define it as a predicate and recognize the nouns in the question through named entity recognition. At this time, multiple nouns may appear. We define these as the subject candidate set. If there is only one noun, we directly take it as the subject. If more than one exists, we should choose all of the recognized nouns. The named entity recognition tool we use is the open-source natural language processing tool, hanlp (https://github.com/hankcs/HanLP/, accessed on 16 March 2021). Hanlp is familiar to researchers of natural language processing, and is the open-source Chinese natural language processing tool with the largest number of users on GitHub. Hanlp has powerful functions and support for Chinese word segmentation and named entity recognition. It is a very easy-to-use tool. At present, the provided hanlp is a Java version, so we used the Python-wrapped pyhanlp. Using the named entity recognition function of the tool, we can segment a sentence and identify its type, such as person name, place name, and organization name. Through named entity recognition, we can obtain place names.

Multiple place names may appear in a question. At this time, we calculate the distance between each candidate word and the category word according to the Manhattan distance formula, and take it as a factor to distinguish the subject noun. In Formula (10), x_{category} is the word granularity number of the category word, x_i is the word granularity number of the word in the candidate set, and the Manhattan distance between the two words is then the absolute value of the number difference between the two words.

$$D\text{istance}_i = \left| x_i - x_{category} \right| \tag{10}$$

In some recommendation systems, it is assumed that a user's favorite items are i_1, i_2, and i_3. The candidate set can be generated from these three items. The reason why we take the three favorite items is explained later. The generation rule is: calculate the similarity matrix between all items, find the three items most similar to i_1 in this matrix (the value can be set, expressed in N), which are recorded as i_{11}, i_{12}, i_{13}. Similarly, the items most similar to i_2, i_3 can be found, which are recorded as i_{21}, i_{22}, i_{23}, i_{31}, i_{32}, i_{33}. Then the user's preference score is calculated for each item in the candidate set, and, finally, these scores are sorted to generate recommendation results. U represents the user. Then the user's preference score for an item is expressed in P (U, item). Let i_m represent the mth item that the user likes, and i_{mn} represent the nth item in the candidate set generated by the mth item that the user likes. Then, the user's preference score for the item in the candidate set can be expressed as:

$$P(U, i_{mn}) = \frac{\sum_{m=1}^{M} P(U, i_m) sim(i_m, i_{mn})}{\sum_{m=1}^{M} sim(i_m, i_{mn})} \tag{11}$$

M indicates the number of items the user likes. In this way, the design formula can better balance the component between similarity and historical favorite items.

Based on the above, in our system, we record the subject and predicate of each query in the log. Generally, the initial situation of this log is the result record obtained after a simple sentence query. This indicates the words and categories that are often used as the subject and predicate of the query. Then, for the subject candidate set in the new question, the collocation score of each word and category predicate is calculated, and the final score is then calculated according to Formula (10) combined with the Manhattan distance. After sorting according to the final score, the noun with the highest score is taken as the subject noun. The first three items are chosen because a greater or lesser number of choices cannot achieve a good effect. In addition, there are only about 10 attributes designed for each scenic spot in our system, and the attributes of some scenic spots are still empty; thus, taking the first three items is the optimal choice.

$$F_{i,attr} = \frac{N_{i,attr}}{\log(1 + Dis\tan ce_i)} \tag{12}$$

In the formula, attr indicates the type of intention we have identified, x_i is the word in the sentence, $N_{i,attr}$ indicates the number of occurrences of the joint query of these two words in the log. $D\text{istance}_i$ is the Manhattan distance, which is calculated above.

The obtained keyword is taken as the subject noun, and the corresponding category noun is found according to the category number obtained by intention recognition. Through the above operations, the basic information to be queried is obtained, such as the * attribute of * place. The * symbols are then replaced with specific words in the cypher statement to identify the predicate attribute of the subject. The knowledge map is then searched. If there are corresponding results in the knowledge map, the result information will be queried. If there is no result in the knowledge map, it will return null.

According to the results, we reorganize it into natural language. This involves a simple splicing. For example, if the subject noun we obtain is "West Lake", the category number we intend to identify is 2, and the corresponding category name of 2 is "opening time", then the result of searching the knowledge graph is "open all day". The final answer is "the West Lake opens all day". Here, for descriptive answers, such as introduction, strategy,

and traffic, which are long text types or natural language entries stored in the knowledge graph, we return directly without adding subject nouns and intention recognition results.

For simple questions, such as "西湖的开放时间是什么时候?" (What is the opening time of West Lake?), our process is as follows (see Figure 7).

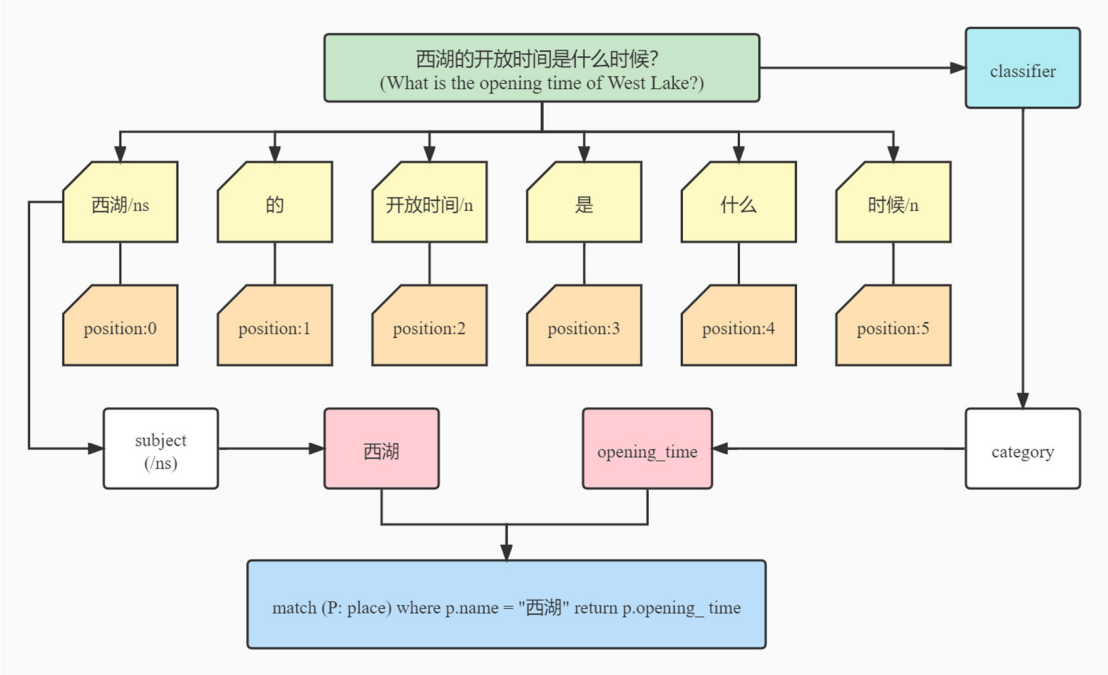

Figure 7. Processing of simple questions.

Through the tool hanlp, we can divide the problem into the following forms: 西湖 /ns 的 开放时间 /n 是 什么 时候 /n. In this way, we can obtain the location of the label /ns and the scenic spot name corresponding to the label of /ns. Through the classifier we designed, we can obtain the class label corresponding to this problem as the opening time. Combining these two things, we can use a cypher statement to retrieve the knowledge map (match (P: place) where p.name = "西湖 " return p.opening_ time) to get the final result.

For complex questions, such as " 杭州西湖区灵隐寺的开放时间是什么时候?" (What is the opening time of Lingyin Temple in West Lake District, Hangzhou), our process is as follows (see Figure 8).

Through the tool hanlp, we can divide the problem into the following forms: 杭州 /n 西湖 /ns 区 灵隐寺 /ns 的 开放时间 /n 是 什么 时候 /n. In this way, we can obtain the location of the label /ns and the scenic spot name corresponding to the label of /ns. At this time, we obtain a list of scenic spot names. Through the classifier we designed, we can obtain the class label corresponding to this problem as the opening time. By calculating the Manhattan distance between each scenic spot in the scenic spot list and the "opening time", we can obtain the closest scenic spot name. Then, a cypher statement is used to retrieve the knowledge map (match (P: place) where p.name = " 灵隐寺" return p.opening_ time) to obtain the final result.

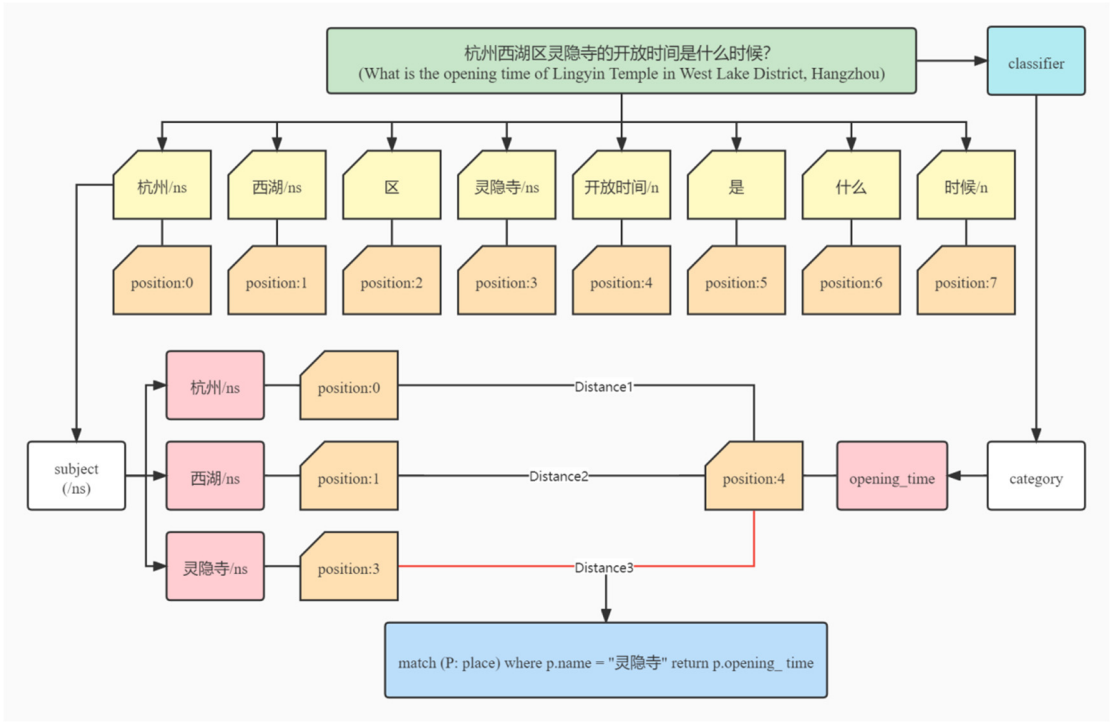

Figure 8. Processing process of complex questions.

Overall, system flow can be changed into a new edition (see Figure 9).

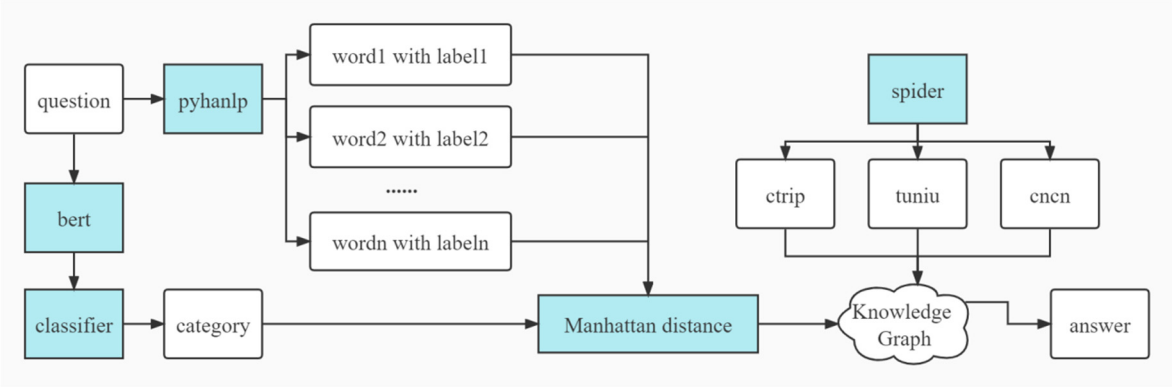

Figure 9. Overall system flow.

4. Experiments

4.1. Dataset and Experiment Setup

4.1.1. Tourism Knowledge Graph

After being collected, the data related to Zhejiang attractions on popular tourist websites, such as Ctrip, Cncn, and Qunar, were analyzed and pre-processed. The knowledge graph of Zhejiang tourist attractions constructed in this study is based on these data (see Table 1).

Table 1. Website description.

Website Name	Scale
Ctrip	2976
Qunar	3010
Cncn	860

After screening, sorting, and integrating the data, we finally obtained 2812 scenic spots. Each scenic spot has a minimum of three attributes and a maximum of 12 attributes (see Table 2).

Table 2. Dataset description.

Spot Name	Type	Level	Ticket	Opening Hours	Suitable Season
西湖 (West Lake)	自然景观 (natural landscape)	5A	免费开放, 景区内一些项目另外收费. (It is free, but some items in the scenic spot are charged separately.)	全天开放 (Open all day)	3–5月, 9–11月 (March to May, September to November)
灵隐寺 (Lingyin Temple)	寺庙 (temple)	5A	75元一位 (75 yuan per person)	全年 07:00–18:15 (07:00–18:15 throughout the year)	四季皆宜 (All seasons)
西溪国家湿地公园 (Xixi National Wetland Park)	自然景观 (natural landscape)	5A	80元一位 (80 yuan per person)	4月1日–10月7日 07:30–18:30 10月8日–次年3月31日 08:00–17:30 (07:30–18:30 from 1 April to 7 October, 08:00–17:30 from 8 October to 31 March of the next year)	春秋最佳 (Best in Spring and Autumn)

We organized the dataset into .csv files, which were stored in neo4j with a graphical form (see Figure 3).

4.1.2. Evaluation Dataset

For intention recognition, we designed a training set, test set, and verification set. In the process of intention identification, we obtained the questions that are often asked by users on the tourism websites, the tourism questions on the Zhihu forum, and Baidu's common search for a scenic spot with automatic filling. Finally, our question set comprised a total of 3635 questions. We defined complex sentences containing multiple attribute information fields or multiple scenic spot names as a complex question, and single attribute problems containing a single scenic spot name as a simple question. There were 1000 complex questions and 2635 simple questions (see Table 3). We organized the question set into .xls files.

After compiling the question set, we needed to design a classifier o deal with the question set. According to the categories we designed, we analyzed the core points of the questions, analyzed the category for each question, and labeled them.

In order to make full use of the sample data, we divided the training set and test set according to the ratio of 4:1 (see Table 4).

4.2. Parameter Setting

The operating system was Windows10, the programming environment was Python 3.6, and the framework was tensorflow 1.14.0. The BERT model adopts Google's official BERT base.

Table 3. Question set description.

Question Type	Scale	Example
Complex question	1000	千岛湖豪华游船和普通游船有什么区别? (What is the difference between a luxury cruise ship and an ordinary cruise ship in Qiandao Lake?) 萧山极乐寺的签文准吗? (Is the signature of Xiaoshan blissful Temple accurate?) 龙门古镇和大奇山国家森林公园附近有什么美食推荐? (What are the food recommendations near Longmen ancient town and daqishan National Forest Park?)
Simple question	2635	西湖的开放时间是什么时候? (When is the West Lake open?) 杭州动物园的门票多少钱一张 ? (How much is the ticket to Hangzhou zoo?) 西溪国家湿地公园在哪里? (Where is Xixi National Wetland Park?)

Table 4. Intention identification set description.

Dataset Name	Simple Question Scale	Complex Question Scale	Total
Classified training data	2108	800	2908
Classified testing data	527	200	727

4.3. Evaluation Criterion

4.3.1. Classified Evaluation Criterion

When evaluating the classification algorithm, we first make manual judgments and number annotations for all of the questions. Then, we take the proportion of the quantity and total quantity whose output label of the model is consistent with the estimated category label as the accuracy of the classification algorithm. Let the number of simple questions in the test set be S and the number of complex questions be C. In simple questions, there are S_{right} in which the classification label is consistent with the estimated label, and in complex questions, there are C_{right} in which the classification label is consistent with the estimated label. Therefore, the accuracy of simple question classification (see Formula (13)) and complex question classification (see Formula (14)) can be obtained.

$$C_S = \frac{S_{\text{right}}}{S} \tag{13}$$

$$C_C = \frac{C_{\text{right}}}{C} \tag{14}$$

Considering the large difference in the number of the two types of questions and a certain contingency, we can obtain the final score F of the overall intention recognition through the proportion of the two types of questions (see Formula (15)).

$$F = \frac{S_{right} + C_{right}}{S + C} \tag{15}$$

4.3.2. QA Evaluation Criterion

The QA tasks are not only to determine the answer to the question that is asked, but also to judge whether the answer is smooth.

The accuracy is measured by whether the answer contains the content of the corresponding attributes of the core scenic spots, and the answer is reasonable. Among all the questions, the number of simple questions is S and the number of complex questions is C. There are S_{currency} simple questions with accurate answers, of which S_{smooth} are smooth and

reasonable, and $C_{currency}$ complex questions with accurate answers, of which C_{smooth} are smooth and reasonable. Finally, the effect of the QA system is measured by the proportion of correct and reasonable answers, and the return is based on the proportion of simple questions and complex questions (see Formula (16)).

$$\text{Score} = \frac{S_{\text{smooth}} + C_{\text{smooth}}}{S + C} \tag{16}$$

We compared our intention recognition model with the recognition results of the naive Bayesian classifier, as shown in Table 5. Through comparison, it can be found that our classification model has a good recognition effect on some related attribute problems. Our system can also have a good recognition effect on the intention recognition of some complex problems.

Table 5. Question classification results of our classification and Bayesian classification.

Classification Name	Simple Question Intention Accuracy	Complex Question Intention Accuracy	Total Intention Accuracy
Our Classification	94%	52.7%	82.4%
Bayesian classification	93%	31.5%	75.8%

4.4. Performance Comparisons and Analysis

In order to prove the effectiveness of the system, it was compared with the existing tourism QA systems using unified test standards. At present, there are few tourism QA systems, so we also undertook a comparison by reproducing other QA systems using our own knowledge graph.

First, the results of intention recognition were compared. We compared them from the perspective of identifying simple questions, identifying complex questions, and the accuracy of overall intention recognition (see Table 6).

Table 6. Main results of intention recognition.

System Name	Simple Question Intention Accuracy	Complex Question Intention Accuracy	Total Intention Accuracy
Our System	96%	56.3%	85.6%
TravelQA	94%	43.7%	79.9%
Movie-QA-System (Reappearance)	88.6%	36.1%	73.9%

It can be seen that our system and other systems have high accuracy in identifying simple questions, but the accuracy is improved in identifying complex questions.

Then, we compared the overall QA effect from the perspective of answering simple questions, answering complex questions, and the overall QA effect (see Table 7). Here, we made a secondary comparison from the perspective of smoothness and rationality.

Table 7. Main results of QA answering.

System Name	Simple Question Accuracy	Simple Question Smoothness	Complex Question Accuracy	Complex Question Smoothness	Total Accuracy
Our System	88.2%	85.5%	49.2%	78.7%	65.3%
TravelQA	82.4%	83.7%	30.1%	76.5%	56.3%
Movie-QA-System (Reappearance)	86.3%	84.2%	32.5%	76.7%	59.5%

It can be seen that our system and other systems have high accuracy in answering simple questions, but the accuracy is improved in answering complex questions. For

example, for the question "How much is the ticket price of * scenic spot?", the answer "The ticket price of * scenic spot is 75 yuan per person" is smooth and reasonable. However, the answer "The ticket price of the scenic spot * is free all day", does not read smoothly. However, this is a common problem and challenge for KBQA, which needs to be further studied and solved.

5. Conclusions

In this study, we implemented a QA system based on Zhejiang tourism that can answer questions relating to scenic spot information. We built a more comprehensive tourism knowledge graph of Zhejiang, and then put the embedding obtained by BERT into the classifier for intention classification. Finally, we searched the knowledge graph by combining the keywords with the results of intention recognition to obtain the final answer. The experimental results show that the effect of simple question answering is similar to that of the existing related systems, but the accuracy of answers to more complex questions was improved. In the future, we will continue to expand and improve our knowledge map and explore better ways to improve the effect of QA. Moreover, the system can be applied in other fields after adjusting the dataset, and the application scope can be expanded in the future.

Author Contributions: Conceptualization, J.L. and Z.L.; methodology, J.L.; software, J.L.; validation, J.L., Z.L. and Z.D.; formal analysis, H.H.; investigation, H.H.; resources, J.L.; data curation, J.L.; writing—original draft preparation, J.L.; writing—review and editing, J.L.; visualization, Z.L.; supervision, Z.D.; project administration, J.L.; funding acquisition, J.L.. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by NSFC grant number 62132014.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Turing, A.M. Computing machinery and intelligence. *Mind* **1950**, *59*, 433–460. [CrossRef]
2. Weizenbaum, J. ELIZA—A computer program for the study of natural language communication between man and machine. *Commun. ACM* **1966**, *9*, 36–45. [CrossRef]
3. Li, F.L.; Chen, W.; Huang, Q.; Guo, Y. Alime kbqa: Question answering over structured knowledge for e-commerce customer service. In Proceedings of the China Conference on Knowledge Graph and Semantic Computing, Hangzhou, China, 24–27 August 2019; Springer: Singapore, 2019; pp. 136–148.
4. Das, S.; Chong, E.I.; Eadon, G.; Srinivasan, J. System for Ontology-Based Semantic Matching in a Relational Database System. U.S. Patent No. 7328209 B2, 5 February 2008.
5. Huang, X.; Zhang, J.; Xu, Z.; Ou, L.; Tong, J. A knowledge graph based question answering method for medical domain. *PeerJ Comput. Sci.* **2021**, *7*, e667. [CrossRef] [PubMed]
6. Sheng, M.; Li, A.; Bu, Y.; Dong, J.; Zhang, Y.; Li, X.; Li, C.; Xing, C. DSQA: A Domain Specific QA System for Smart Health Based on Knowledge graph. In Proceedings of the International Conference on Web Information Systems and Applications, Guangzhou, China, 23–25 September 2020; Springer: Cham, Switzerland, 2020; pp. 215–222.
7. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805.
8. Bast, H.; Haussmann, E. More Accurate Question Answering on Freebase. In Proceedings of the 24th ACM International, ACM, Online, 17 October 2015.
9. Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; Ives, Z. DBpedia: A Nucleus for a Web of Open Data. In Proceedings of the The Semantic Web. ISWC 2007, Busan, Korea, 11–15 November 2007; Lecture Notes in Computer Science. Volume 4825, Chapter 52. pp. 722–735.
10. Wang, W.; Xiao, Y.; Cui, W. KBQA: An Online Template Based Question Answering System over Freebase. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16), New York, NY, USA, 9–15 July 2016.

11. Liu, N.; Chee, M.L.; Niu, C.; Pek, P.P.; Siddiqui, F.J.; Ansah, J.P.; Matchar, D.B.; Lam, S.S.; Abdullah, H.R.; Chan, A.; et al. Coronavirus disease 2019 (COVID-19): An evidence map of medical literature. *BMC Med. Res. Methodol.* **2020**, *20*, 177. [CrossRef] [PubMed]
12. Goel, M.; Agarwal, A.; Thukral, D.; Chakraborty, T. Fiducia: A Personalized Food Recommender System for Zomato. *arXiv* **2019**, arXiv:1903.10117.
13. Zhang, Y.; Yu, M.; Li, N.; Yu, C.; Cui, J.; Yu, D. Seq2Seq Attentional Siamese Neural Networks for Text-dependent Speaker Verification. In Proceedings of the ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; IEEE: Piscataway, NJ, USA, 2019.
14. Zhang, L.; Lin, C.; Zhou, D.; He, Y.; Zhang, M. A Bayesian end-to-end model with estimated uncertainties for simple question answering over knowledge bases. *Comput. Speech Lang.* **2021**, *66*, 101167. [CrossRef]
15. Wu, Y.; He, X. Question Answering over Knowledge Base with Symmetric Complementary Attention. In Proceedings of the International Conference on Database Systems for Advanced Applications, Jeju, Korea, 24–27 September 2020; Springer: Cham, Switzerland, 2020; pp. 17–31.
16. Tan, M.; Dos Santos, C.; Xiang, B.; Zhou, B. Improved representation learning for question answer matching. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; pp. 464–473.
17. Li, Q.; Zhang, Y.; Wang, H. Knowledge Base Question Answering for Intelligent Maintenance of Power Plants. In Proceedings of the 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Dalian, China, 5–7 May 2021; IEEE: Piscataway, NJ, USA, 2021.
18. Lai, Y.; Jia, Y.; Yang, L.; Feng, Y. A Chinese Question Answering System for Single-Relation Factoid Questions. In Proceedings of the National CCF Conference on Natural Language Processing and Chinese Computing, Dalian, China, 8–12 November 2017; Springer: Cham, Switzerland, 2017.
19. Zhou, B.; Sun, C.; Lin, L.; Liu, B. InsunKBQA: A question answering system based on knowledge base. *Intell. Comput. Appl.* **2017**, *7*, 150–154.
20. Partner, J.; Vukotic, A.; Watt, N. *Neo4j in Action*; Pearson Schweiz Ag: Zug, Switzerland, 2014.
21. Lu, Y.; Wang, S. The improvement of question process method in QA system. *Artif. Intell. Res.* **2015**, *5*, 36–42. [CrossRef]
22. Sun, C.; Qiu, X.; Xu, Y.; Huang, X. How to Fine-Tune BERT for Text Classification? In Proceedings of the China National Conference on Chinese Computational Linguistics, Kunming, China, 18–20 October 2019; Springer: Cham, Switzerland, 2019.
23. Peng, X.Y.; Liu, Q.S. Naive Bayesian classification algorithm based on attribute clustering under different classification. *J. Comput. Appl.* **2011**, *31*, 3072–3074.
24. Liu, H.; Zhang, Y.; Li, Y.; Kong, X. Review on Emotion Recognition Based on Electroencephalography. *Front. Comput. Neurosci.* **2021**, *84*, 758212. [CrossRef] [PubMed]