CN-DBPedia Report

Chunhou Liu, Yang Li, Naixuan Wang

Introduction

The main purpose of this experiment is to build a database by comparing the large data sets, and then design efficient queries to query in the established database to quickly gain the answers we want.

Our database is Data1 - CN-DBpedia, as described in the document, which is a large encyclopedia of general structure, containing millions of entries. The statements we want to query are entered in Chinese. There are two main types of sentences: 1. "中国的官方语言是什么？" The aim is to query another entity based on one entity and one relationship. 2. "周杰伦和昆凌的关系？" The aim is to query the relationship between two entities. In practice, we have made some innovations based on the characteristics of the given data set to improve the query efficiency and the query function.

Design

The query process is as follows: First, we input a question that needs to be queried in the input box on the user interface, such as: "中国和人民代表大会制度的关系", and then click the query button to call the sentence analysis function, through which we can extract the entities in the sentence and some nouns. In a simple input statement, the extracted nouns will not be too many, we can select at most two nouns to pass to the database for query. In the database, if you enter one noun, it must be an entity. For example, when the question is "谁是周杰伦？", the noun entered is "周杰伦", we can return its BaiduCARD content as the introduction information of the entity. If you enter two nouns, such as "中国" and "人民代表大会制

度” in the example above, the first one must be an entity, but whether the second noun is an entity or a relationship cannot be judged, so we will regard it as a relationship at the first time and entities at the second time. After we search it in the database twice, we will have two results, but one of they two is usually null. we then merge the results together. The information obtained by the query will be stored in a list, and because the information of the data set is not neat, and there is much redundant information such as ”$< a >$” in the entity words and related words, so we will first clean the returned results, remove the irrelevant information, and then print the elements in the list orderly so that it looks cleaner.

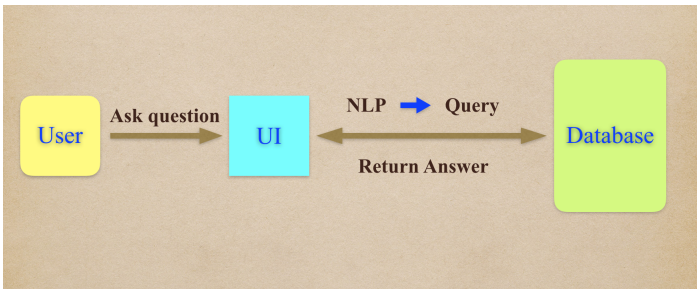

图 1: the flow chart of the searching process

Natural Language Processing

After receiving the user’s question, we analyze the user’s problem showed in natural language form. For the processing of questions, we focus on 2 items: the nouns of the questions and the relations between nouns.

Nouns

Classification and Effect

We divide nouns into two categories: entity nouns and common nouns. Entity nouns such as ”周杰伦”, ”美国” and so on, common nouns include ”工作”, ”作品” and so on. The subject noun is not only the main object of the user’s inquiry, but also the core theme of the question. The recognition

and extraction of subject nouns is the key to understanding the problem. Obviously, compared with ordinary nouns, entity nouns have more information and can represent the inquiry intention of users most of the time. For example: “中国和北京的关系是什么？”. In this question, ”中国” and ”北京”, as two entity nouns, clearly express the purpose of the user's query, and the query in the database will also focus on the relationship between the two nouns. In addition, apart from entity nouns, the role of common nouns is also very important. Common nouns often point to the query direction of entity nouns. For example, “中国的首都是什么？”. In this question, ”中国” is an entity and ”首都” is a common noun. The word tells us that we are looking in the direction of which word ”capital” has a relationship “首都” with “中国”.

Processing

StanfordCoreNLP is a useful tool to process the sentence. For example: “中国和北京的关系是什么？” can be parsed as: 中国-NR, 和-CC, 北京-NR, 的-DEG, 关系-NN, 是-VC,什么-PN, ？-PU and 中国-COUNTRY, 和-O , 北京-STATE-OR-PROVINCE, 的-O, 关系−O, 是−O, 什么−O, ?−O. We extracted entity nouns and common nouns as the result of preliminary processing. They are ‘中国’-‘北京’ and ‘关系’. After that, we would make corresponding choices for the different types of sentences.

Relationship

Generally speaking, according to the number of entity nouns and common nouns in a sentence, we divide the questions into three categories: the first is to find the relationship between the two entities, such as ”中国和北京的关系是什么？” The second category is to find another associated entity given an entity and a relationship. For example, ”中国的首都是什么？” In the third category of sentences, there is only one substantive noun, not a general noun. This kind of question usually asks about the nature of the entity, such as “周杰伦是谁？” And for each type of sentence, the words we extract are different.

For the first, we directly extract two entity nouns as query objects to

query their relationships. In the second kind of sentence, we select the only entity noun and a generic noun as the query object. There are often many general nouns, and we prefer the Special noun, "NR", because the proper noun is more informative than other nouns, "NN". As for the third, we only extract the only entity noun, and think that the question is asking about the nature of the noun.

Query and Experiment

Database design

Our database consists of three tables, the first table is the entity-relationship-entity table, the entity-relationship triplet is recorded, the second table is the name-entity table, the relationship between the record name and the entity, and the third table Is a relational table, mainly used to quickly query the relationship name. Among them, the first column of the first table has an index, the name of the second table is indexed, the third table is not very large, and the index does not need to be indexed.

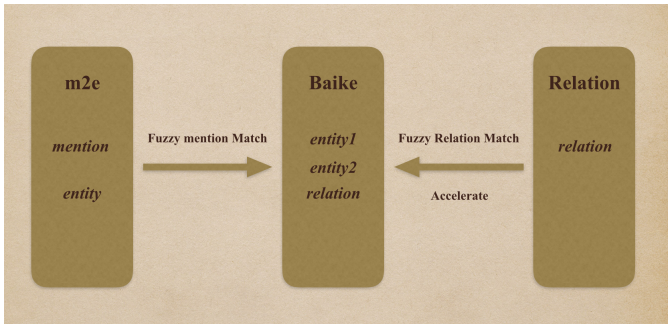

图 2: the design of the three tables of our database

Query design

We mainly use union and with clause to update the speed. For example, querying "中国和人民代表大会制度的关系", if we query the database using the 'AND' and 'OR' to find the relationship, in our experiment, it costs 55 seconds, and if we using 'WITH' and 'UNION', it costs within 0.02

second. For the first query method of querying their relationship through two entities, we first use natural language processing to extract the nouns in the question, and then convert these nouns into entity names through fuzzy matching, and then pass this again. All related entities of the entity are extracted, and then the second entity and all the associated entities of the first entity are fuzzy matched, and the corresponding relationship of the matching entity is returned, because the number of all associated entities of the first entity is not much So you can find it quickly. For the second type to query the corresponding attribute value through an entity and a relationship, similarly, we first use natural language processing to extract the noun in the question, and then convert the first noun into a entity name through fuzzy matching, and then Extract all the relationships and entities associated with this entity name, and then use another noun to make a fuzzy match with all the relations of the first noun, and return the matching result directly. Besides the two basic query, We also designed another question, such as "周杰伦是谁", this question only gives a single entity, we will return its description as a return value to the user.

Experiment

In the process of improving the query statement, we found that using AND and OR as the query interpretation time takes much more time than using UNION and AND. By doing comparative experiments, we find that using AND and OR as the query judgment. As shown in figure time cost, the time spent on the method is 100 times the time spent on UNION and AND. It can be seen that using UNION and AND as the query can greatly reduce the query time.

UI and Results

The graphical user interface is written using a tool of Python called tkinter. The function is relatively simple. It mainly implements multi-line input and output, clicking the button to query or exit and some other functions. The input box supports multiple input queries and Chinese sentences.

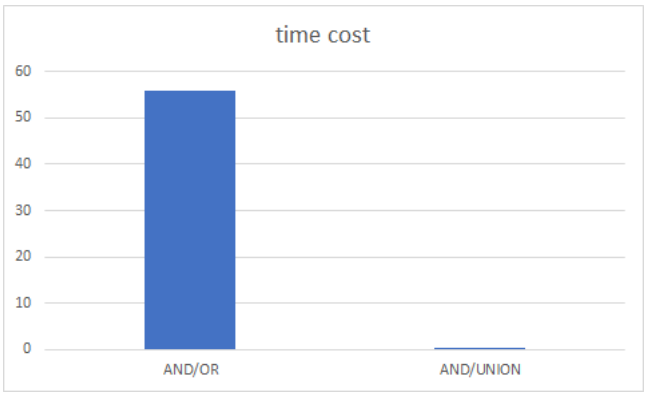

图 3: time-cost comparsion

Also, we have modified the overall layout to make the window look more beautiful.

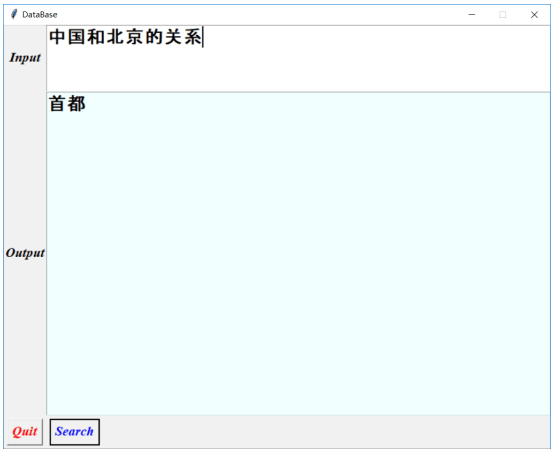

图 4: search for the relation between two entities.

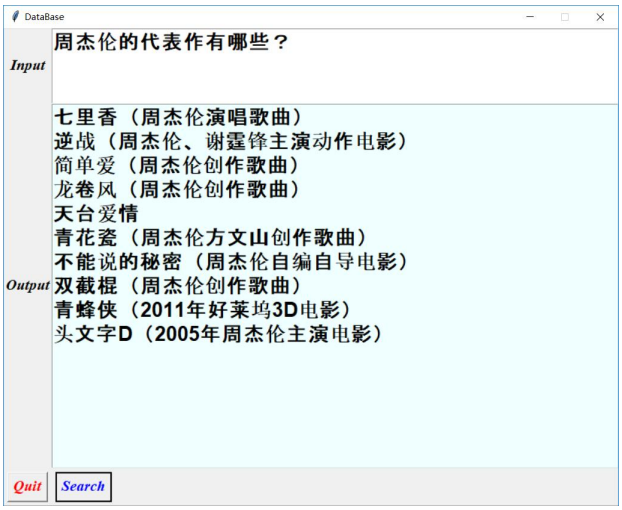

图 5: find another entity based on an entity and a relation.

图 6: if there is only one entity, return the introduction information.

Discussion and Conclusion

In the project, we process the user's questions and select the key information for query. According to the three different questions, We use the

mention2entity query entity, to find the specific nouns that the user's information might refer to, and then use the exact match query. This approach ensures the comprehensiveness of the query and avoids the information redundancy caused by fuzzy matching. And the fuzzy matching method on the relation attribute makes the important relation information not to be omitted.

One more thing that needs to be improved in our project is the query for compound questions. For example, "中国的首都和北京大学是什么关系？" In this case, we can not accurately extract the key information:" 中国的首都 "=" 北京 ". We can only extract the keywords of "中国" and "北京大学". This requires us to make further grammatical analysis and relational dependency analysis of the text.