COMP1011 Presentation Script

Group 8

May 2021

1 Introduction

Hello, we are group 8, and our project is implementing a searching engine for subjects offered by the department of computing. We have finished this project and now it not only fulfills the project requirement but also has more advanced features. For example, in addition to pre-requisite and subject title, we added subject level, credits, and content. We will search in all contents for the users' input and that can match more accurate answers. Moreover, we also have some user interaction such as allowing users to get more detailed information and download the pdf files for their reference. Because well-organized algorithm and careful consideration, we made the searching process fast and easy.

2 Usage

Let me introduce how to use this program. It is very easy to use, anyone who knows how to use google can use it. You just need to type keywords, and our program will automatically find related subjects to you, sorted by relative marks. A detailed use demo will display at the end of this presentation.

3 Algorithm

Now I'll introduce the Implementation of the core algorithm of search. We try to break through important issues in the search field ——speed. In the preview, we embedded Elasticsearch into the commercial project of cloud service. In the actual project, we need to consider scalability and application scenarios. Combined with the usage, we not only created the original search algorithm but also improved ElasticSearch to make it more suitable for our needs. While retaining the JSON format, we also redesigned the reverse index algorithm. We realize that behind the simple sentences, the designer's wisdom is embodied. In terms of reverse index, As you can see, there are two tables, One is the keywords, the other records the weight of keywords and their relationship with documents. We only need to find the field corresponding to each ID. It can index to our target document, and at the same time add the weight. These

tables exist in the form of a C++ Class. In addition, we also tried B-Tree and dichotomy algorithm.

4 Data Structure

We have created two headers that can help us use object-oriented programming skills. the two headers contain two classes, the first class is a dictionary, its instance includes the word name, id, the document id which the word appeared in the title, and the document id which the word appeared in content. By using this dictionary, we can know the word's id and it can shorten the search time afterward.

The second class named course is used to store the information related to the subjects, it includes objects such as subject name, subject id, subject level, subject pre-requisite, subject title, and maps that store how many times the word appeared in the document. This can be useful to calculate the relative score of documents. Another thing we'd like to mention is that in course class, the data objects are set to private because it is beneficial for data security, and we can only access the data by calling public functions.

5 Implementation

We divided our work into some parts for distribution, which including data pre-process, data structure design, user input process, relative index calculation, and result display. We transform pdf files to JSON files aiming at C++ program reading faster and more conveniently. The red-black trees are implemented as sets and map objects in C++, and the proper design of the algorithm with C++ features makes the program faster. Also, the thought of object-oriented programming can make this program more portable and easier to maintain.

6 Conclusion

That's the end of our presentation, we use a lot of knowledge learned and we also learn more advanced skills and headers in C++. At the same time, we gain much learning experience and we are fully convinced that this will be helpful in our future development. Now please enjoy our demo.