

Project Proposal

Title

Local Search Engine

Problem Descripton

I will be creating an local search engine for text file. Users are able to input the keywords in file content and this local search engine will provide most related file name. TF-IDF(Term Frequency Inverse Document Frequency of records) will be used as searching algorithm. The idexing result of files will be stored as a *index.json* in local computer, which means this local search engine works without internet.

Data Descripton

The UML class diagrams

Lexer
- content : string
+ nextToken() : string
+ hasNext() : bool

Reader
+ readText(string) : string

IndexModel
- docs : hashmap<string, Doc>
- df : hashmap<string, int>
+ addDocument(string, string) : void
+ removeDocument(string) : void
+ search(string) : vector<(string, float)>

Doc
- tf : hashmap<string, int>
- totalToken : int

Data structure

I will use **vector** and **hashmap** to store Model in my program. Different Model would be stored in vector, and basic information, such as file content and

indexing score, would be stored in hashmap.

File I/O

I will store the indexing result as an external file named **index.json**, and the score in the **index.json** will be used to find the most related file according user input.

Procedural description

```
Model model = Model();  
Reader reader = Reader();  
string content = reader.readFromFile(path);  
model.addDocument(content);  
file::save(model);  
result = model.search(input);  
print(result);
```

SPECIAL NEEDS/CONCERNS

No.