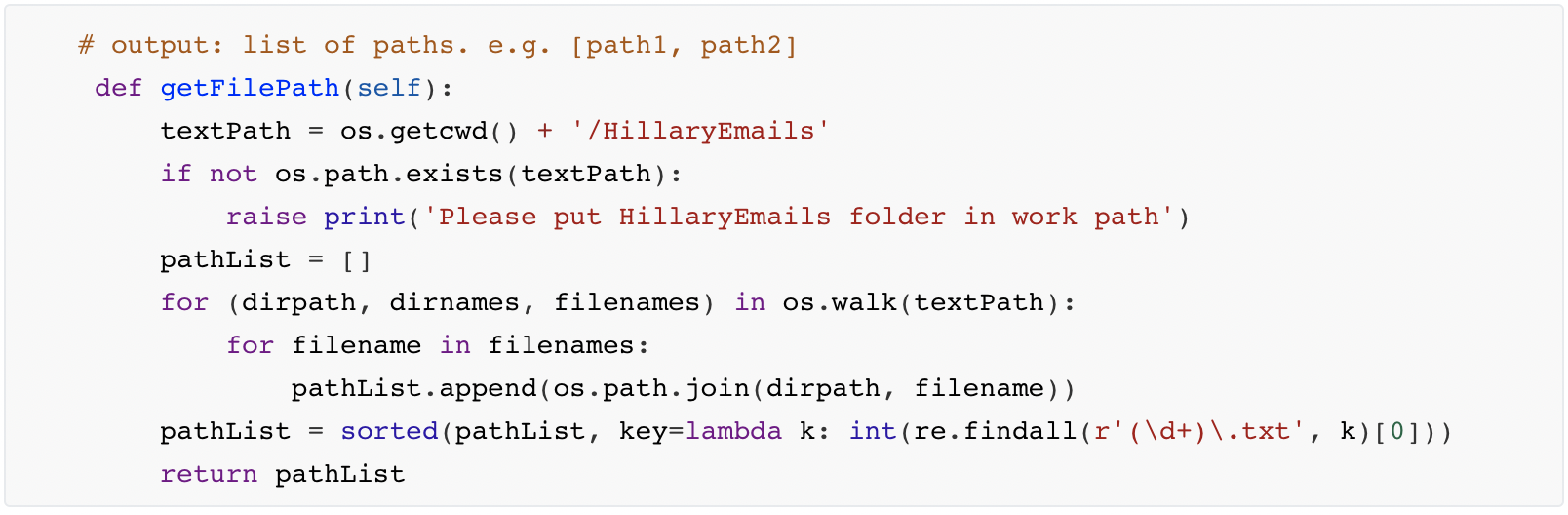
**Optimization**

1. **Optimization Step**

We currently store the full document path in the posting list. Each document path is stored multiple times, which we believe would affects the memory and time requirements for creating and querying indexes. Thus, in the optimization process, we try to store all the file paths in the list to improve the efficiency. Below is our optimization step:

1. We defined a function named getFilePath() which will return a list of paths (eg. [path1, path2]) shown in below:



1. In sortTokens part, we use the getFilePath() to get the pathList, then we use the index of the pathList to obtain the content of the .txt file in corresponding path after which we used the tokenization() function to get the tokenlist which is followed by stemmer() function for stemming. Finally, we push those newTokenList to the allTokenList. Then we sort the allTokenList and return the sortedToken.



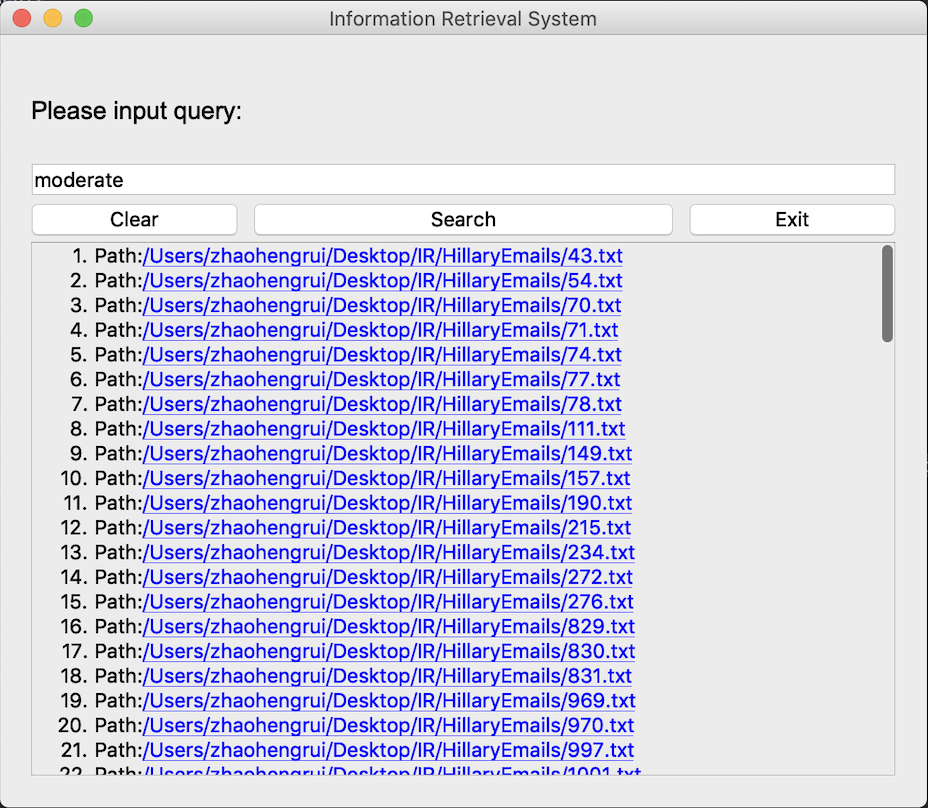
1. In getResult() function, We traverse all the items in result and output all the results as pathList[index]. To enhance our user experience and enrich our functionality, we used a string containing HTML tags to present the final result.

（1）Use ordered list tag <ol><li>to give the answer sort id.

（2）Use the <a> tag to make the hyperlink to each result item to make it clickable. When the user clicks on the result item, the .txt file of the corresponding path will be automatically opened.



An example of the result:

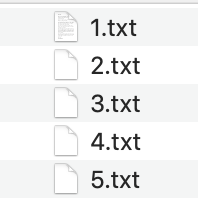


1. **Testing**

In this section, we will start with a small document set of 5 files to do a set of demo functionality test, after which we do black-box test to the whole document to check the accuracy of the query results which includes multiple test cases, such as single word query, multi-word query and OR mixed query. After that, we will test the time and size requirement of Information Retrieval System before and after optimization to evaluate our optimization. It mainly consists of two key steps of testing, creating indexes and querying search results.

1. Demo-test(Small document set of 5 files)

We have retained 5 documents for demo testing.



Case 1 “father”:



Case 2 “mother”

No Result.

Case 3 “wednesday”



Case 4 “father wednesday”

Expected Answer: “no result”

No Result

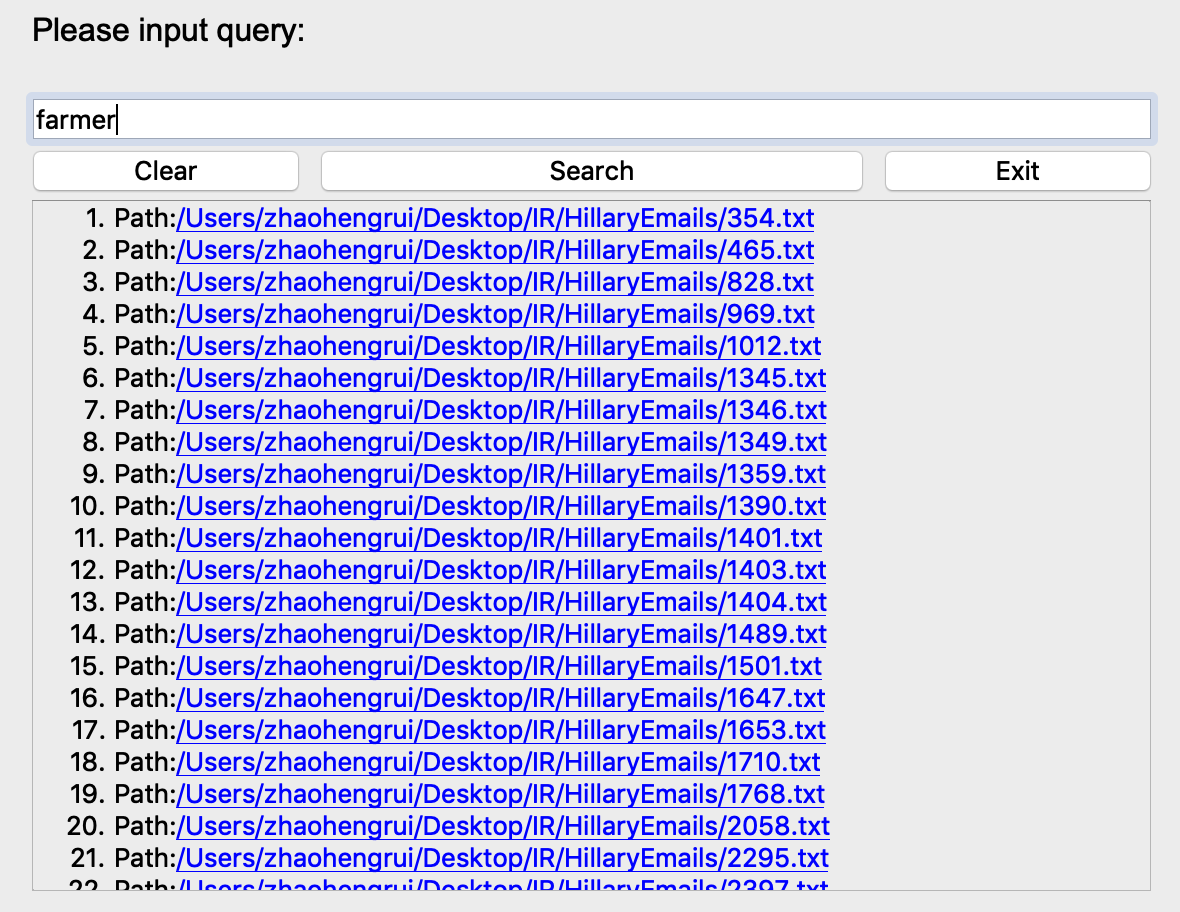
Case 5 “father OR wednesday”

Expected Answer: “1.txt, 2.txt, 3.txt, 4.txt, 5.txt”



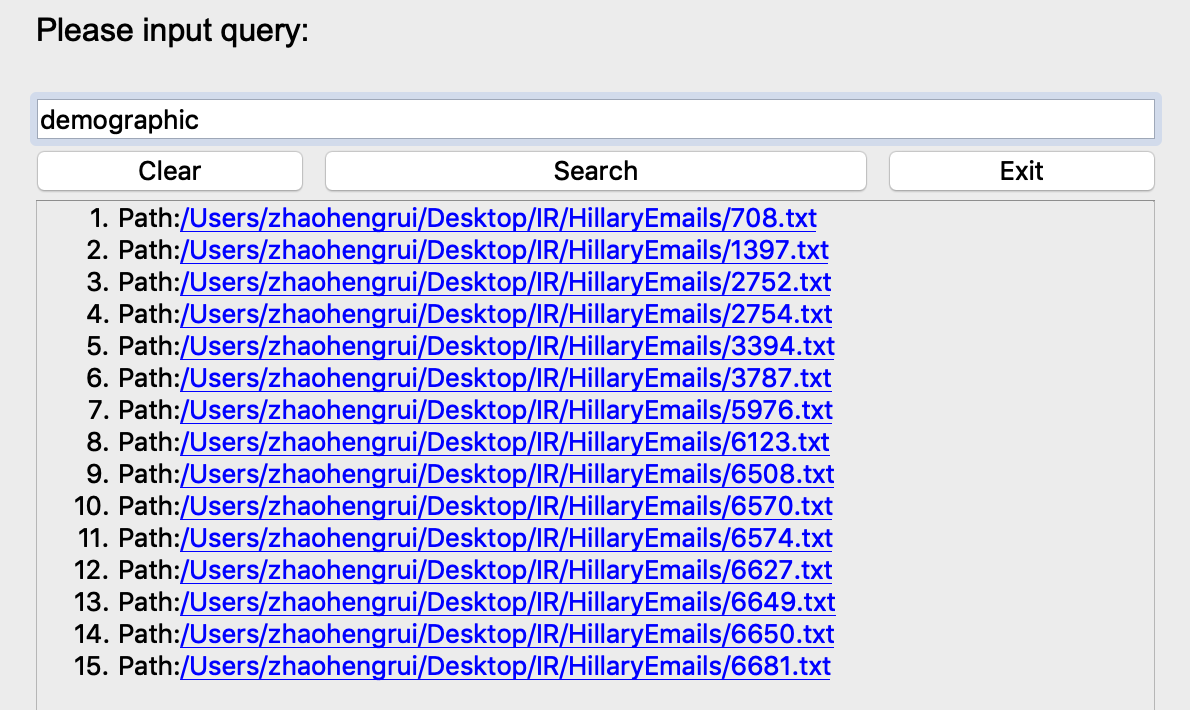
1. Black-box testing for whole document set
2. Single-word query

Test Case 1: ‘farmer’:



The result has 62 items. After check, the result is accurate.

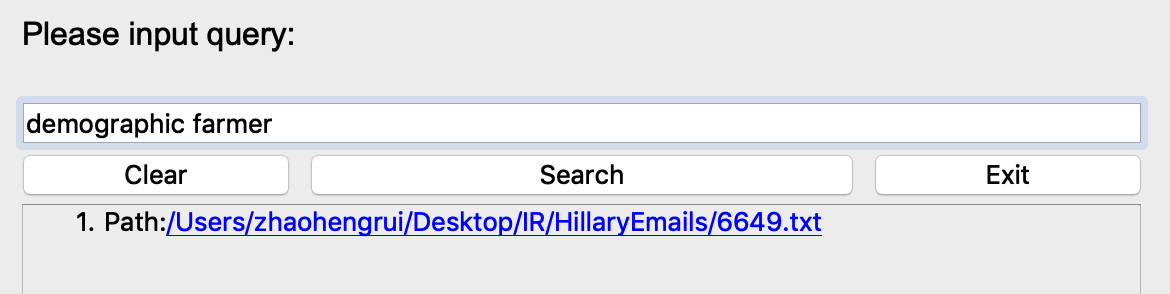
Test Case 2: ‘demographic’



The result has 15 items. After check, the result is accurate.

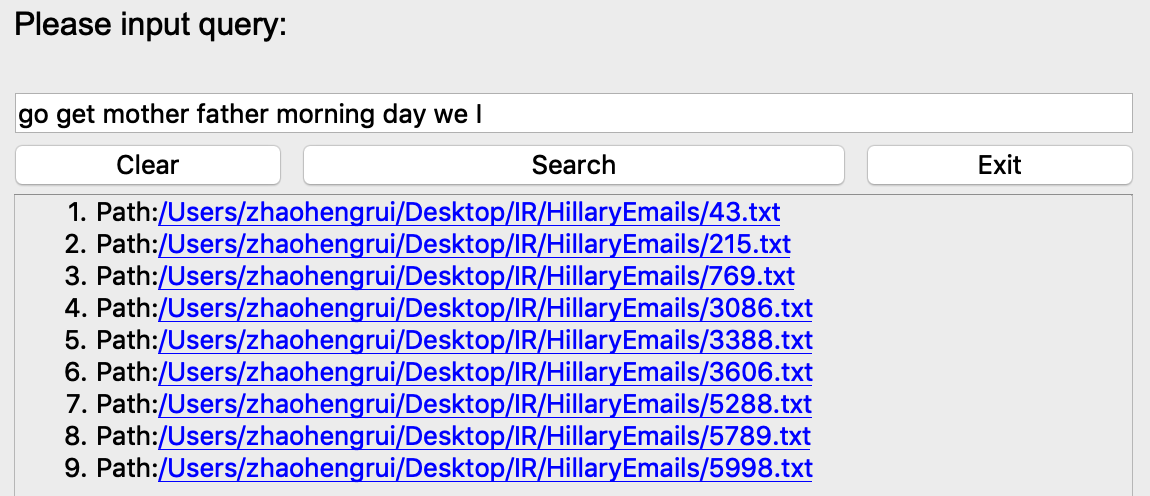
1. Multi-words query

Test Case 1: “farmer demographic” (Shorter Query)



Test Case 2: “go get mother father morning day we I” (Longer query)

The result has 1 item. After check, the result is accurate.



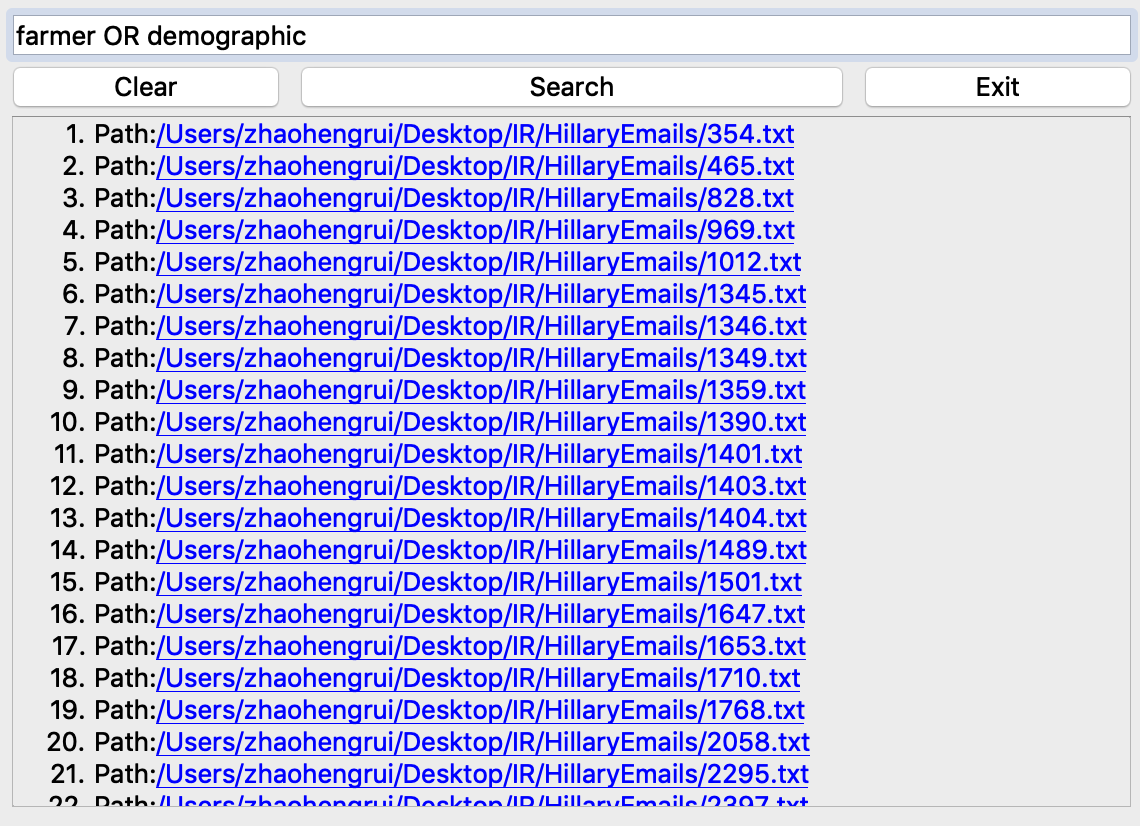
The result has 9 items. After check, the result is accurate.

Test Case 3:” ﻿alphabet dangerous”

The result shows no hits which is correct.

1. ‘OR’ Mixed query

Test Case 1: “farmer OR demographic”



The result has 76 items. After check, the result is accurate.

Test case 2:

1. Time&Size requirement before the optimization
2. Creating index
3. Retrieve specific words
4. Time&Size requirement before the optimization
5. Creating index
6. Retrieve specific words
7. Optimization Analysis