Ashutosh Reddy Vanteru

ID: 11652658

Information Retrieval and Web Search

Project Report

1. System Design

   a. *Background:* This project is written entirely in python and python libraries are used. For ease of execution, the entire code is also provided within the python notebook to execute the entire code at once.

   b. *Input Provided:* A folder called "ft911" that has all the files containing the documents and the information is provided along with a stopwords list file that needs to be used to while performing tokenization on the word content.

   c. *File Reading and Text Processing:*

      i)   The first step of the process is to extract the relevant information from the given files. To do this, TextParser's fetchDocs() method is called upon which takes in the given ft911 folder and reads all the documents and loads the name of the documents and the information to memory.

      ii)   The next step consists of creating dictionaries, where the file names are stored in a file dictionary along with a unique identifier. For the word dictionary, words are first tokenized (by taking only non-alphanumeric words and removing stop words) and then stemmed using the NLTK library's PorterStemmer and the unique words are stored in Word dictionary along with a unique identifier.

   d. *Indexing:* There are two indexes that are generated in this step. Forst a forward_index, which has FileIds with all the stemmed words that it contains along with their frequency and an inverted_index that has all the words and the ft911_ file in which it occurs along with the frequency in that file.

   e. *Output Generation:*

      i) parser_output.txt has the word dictionary and file dictionary that are ordered by their IDs.

ii) Two index files, forward_index.txt and inverted_index.txt are generated by the code to store the forward and inverted indexes.

2. Execution Time Analysis and Index Size:

Time taken to Parse and create IDs for all documents is <1second.

Time taken to generate Forward and Inverted Indexes is <1second.

Total Number of Documents = 5368

Total Number of Words = 32606

```python
start_parser = time.time()
parser_output(parser_file, WordDict.fetch_d(), w)#once all the word data is feteched, we write the data to the output file using the 'w' mod
parser_output(parser_file, FileDict.getAllFiles(), a)#once all the filenames are feteched, we append the data to the output file using the '
end_parser = time.time()

total_parser_time = end_parser - start_parser
print(f"Time taken for Parsing the documents = {total_parser_time} seconds")

start_time = time.time()
new_forwardIndex = Indexer.create_forward_index(forwardIndex)
Indexer.write_forward_index(new_forwardIndex, forward_index_file)

inverted_index = Indexer.create_inverted_index(new_forwardIndex)
Indexer.write_inverted_index(inverted_index, inverted_index_file)
end_time = time.time()

execution_time = end_time - start_time
print(f"Time taken to generate the indexes is {execution_time} seconds.")
print(f"Total number of words = {len(inverted_index)}")
print(f"Total number of documents = {len(new_forwardIndex)}")
```

```
[18]   ✓ 31.6s                                                                                              Python

...   Running code at: 2024-03-28 13:11:50.422704
      Parser output- Success: Writing data COmpleted!
      Parser output- Success: Writing data COmpleted!
      Time taken for Parsing the documents = 0.038260459899902344 seconds
      index success in forward_index.txt
      index success in inverted_index.txt
      Time taken to generate the indexes is 0.7784864902496338 seconds.
      Total number of words = 32606
      Total number of documents = 5368
```