Information Retrieval and Extraction - M22CS4.406

Mini-Project - Search Engine for Wikipedia

Sashank S - 2021701038

1 Introduction

Built Search Engine Platform by creating Inverted Index on the Wikipedia Data Dump (2022) of size 90 GB.

In this project three types of queries are handled :

- Plain Queries Ex. - Mahatma Gandhi
- Field queries Ex - b Mahatma i xyz
- Code Mixed Queries Ex - Hindi and English together i:defend b:नागरिक

The search results are ordered in ranking using a weighted TF-IDF ranking based on occurance of word in Title, Body, InfoBox and so on...

2 Directory Structure of Code

To start the indexer,

```
python3 indexer.py <wikipedia_dump_path> <index_path> <stats file path>
```

If needed for code mixed indexing, then add the flag 1.

```
python3 indexer.py <wikipedia_dump_path> <index_path> <stats file path> 1
```

The SAX Document Parser is found below. It also keeps track of the occurence counts for each unique word, section and document (Index Creation).

```
parserDoc.py
```

The Tokenization, Case Folding, Stop Word Removal and Stemming for each section (Title, Body, Infobox, Category, References and Links) is performed at,

```
pageProcessor.py
```

The writing of the Inverted index and k-way merge sort of smaller files are performed at

```
writer.py
```

To start querying,

```
python3 search_index.py <index_path> <queries txt file>
```

To query a code mixed index add the flag 1.

```
python3 search_index.py <index_path> <queries txt file> 1
```

Query processing including checking if it is a basic or field query and processing the query terms is performed in

```
query.py
```

Finally, retrieving the necessary posting list for the query term by using binary search on the offset files and ranking using Tf-IDF scores (page rank) is done using

```
search_utils.py
```

3 Directory structure of Index

Figure 1 shows how the index is structure. The *intermediate directory* comprises of each small index file that has the following structure:

```
term_id doc_id title frequency : body frequency :
infobox frequency : category frequency : link frequency |
doc_id_2 title frequency : body frequency :
infobox frequency : category frequency : link frequency
```

These intermediate index files are combined by a k-way merge sort to create the index with the posting lists.

The *temp offsets directory* comprise of offsets of each term in the index file. This helps in quick retrieval of posting lists based on the term.

The *DocID_Title_mapping.txt* file has each document id mapped to a title. The index file uses the document id and hence this file is used to retrieve the corresponding title.

The *index_file.txt* has the complete index with each term and its corresponding posting list.

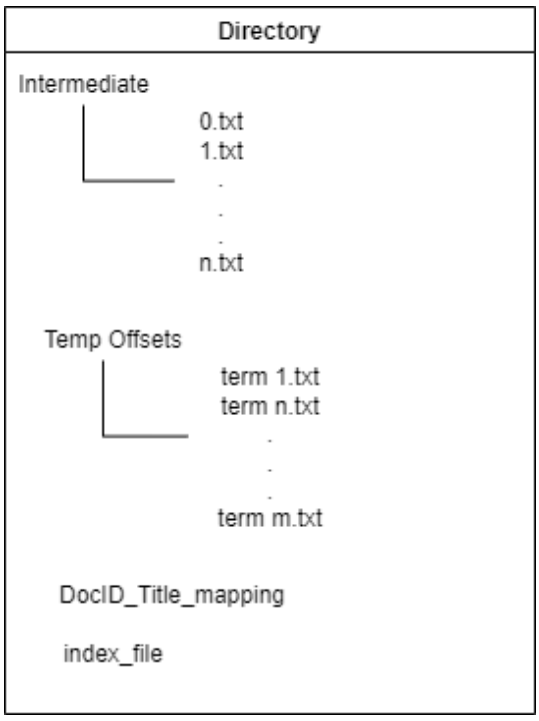

Figure 1: Directory Structure of Index

4 Space and Time Optimization

The main challenge to create an Inverted Index for a huge file has a tradeoff between the size of Inverted Index and the search time. The main Inverted Index created was around 27 GB 3 but 3 levels of offset files is created to make sure the index file loaded in the main memory at a time does not exceed 800 MB.

For each query process word by word. For each word, find offset using secondary index in O(logn) where n is the size of secondary index dictionary. Read the posting list from the primary index file in O(1), and calculate the Tf-IDF score to perform ranking.

5 Index Creation Metrics

Index Creation metrics for 90GB dump:

- Time : 13.31 hours 2
- Size : 22.05 GB 3

```
(base) [sashank.sridhar@ada ~]$ tail -f slurm-750973.out
Time taken to process  22170000 docs  47763.718899965286
/ssd_scratch/cvit/sashank.sridhar/full_index_path/intermediate/738.txt
Time taken to process  22200000 docs  47814.92832970619
/ssd_scratch/cvit/sashank.sridhar/full_index_path/intermediate/739.txt
Time taken to process  22230000 docs  47857.46015548706
/ssd_scratch/cvit/sashank.sridhar/full_index_path/intermediate/740.txt
Time taken to process  22260000 docs  47903.89548063278
/ssd_scratch/cvit/sashank.sridhar/full_index_path/intermediate/741.txt
/ssd_scratch/cvit/sashank.sridhar/full_index_path/intermediate/742.txt
Time taken - 47944.75867938995 s
```

Figure 2: Time taken to create index for 90GB file

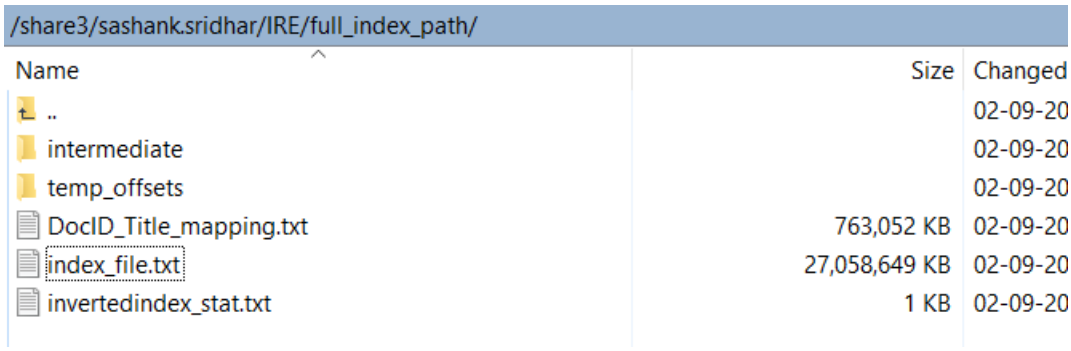

Figure 3: Size of index for 90GB file

6 Format of Index Used

- Primary Index Word : doc id field type frequency. Eg. sachin:d1-t1c2b7|d5-t1
- Secondary Index Word : Offset in the primary index