Experiment 2
Creation of Block Sort Based Indexer and Term Document Matrix

I. EXPERIMENTATION DONE

Create Block Sort Based Indexer and Term Document Matrix for a large corpus.

II. APPROACH

A. Data pre-processing

1) We downloaded the large Gutenberg Corpus as csv file for creation of our index.
2) Analyzed the corpus to identify the text fields to be indexed.
3) Determined the minimum block size that can be processed in memory before writing to disk.

B. Posting list creation for each block

1) We started reading the documents line by line in chunks of blocks we determined in the last step, thus processing the whole corpus in chunks of block.
2) Then in each block we tokenized our text and removed stopwords followed by stemming of the tokens using PorterStemmer of the nltk python library.
3) Then for each word a dictionary was created in which keys are our word and its value is again a dictionary. In the inner dictionary the keys are the documents and the values are the frequency the word which is the key of the outer dictionary.
4) When we reached our calculated chunk size we sorted our Posting list and saved it to individual files and cleared the dictionary to create space to process next chunk. We created nearly 900 files which we will merge in the next step to create final posting list.

C. Creation of final posting list

1) We merged all the posting lists created earlier into one long sorted posting list using the external merge algorithm.
2) To merge and sort the sorted posting lists created earlier in the file, we used the merge function of the heapq library of Python
3) We iterated through posting list to create a single large posting list by merging all the posting lists of the same token, in the end of the posting list we also store the count of number of documents a token appears in.
4) We create the TF-IDF matrix using the posting lists where we store the frequency of the terms in all the documents(stored at the end of list) and the frequency of the terms in each individual document.

III. DISCUSSION ON THE RESULT

We were able to create an index from the large corpus of 4.5 GB size without running out of memory using block sort based technique. The posting list of each single term stores the documents id along with the frequency of that term in the document, moreover it also store the number of documents in which the given term occurs. This posting also helps us to create Term-Document Matrix.

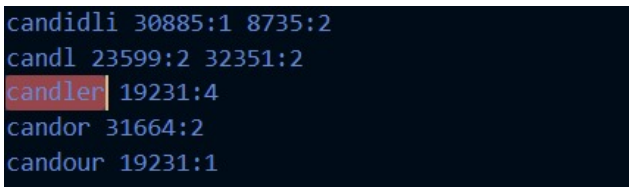

fig 2.1 Individual posting list for word candler.

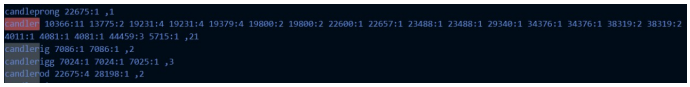

fig 2.2 Final merged posting list for word candler.

IV. CONCLUSION

We were able to build an index for a very large corpus using the block sort based technique without reaching the out of memory state. The final posting list we created provided us substantial information which we also used to create Term-Document matrix. The created posting lists and the Term-Document Matrix can now further be used to create retrieval systems.

V. GITHUB LINK

Click here to visit the Lab2 GitHub repository.