Part A - Task 1

Group No.: 13

Group Members:

Avijit Mandal	18CS30010
Rajdeep Das	18CS30034
Ashish Gour	18CS30008
Sumit Kumar Yadav	18CS30042

Running the Code:-

Task1A: python PAT1_13_indexer.py ./Data/en_BDNews24

Task1B: python PAT1_13_parser.py ./Data/raw_query.txt

Task1C: python PAT1_13_bool.py ./model_queries_13.pth ./queries_13.txt

Task1A(Building the Index):-

Steps:

1. Parsed the whole corpus stored in the 'en_BDNews24' folder using OS python library and split function to get the document content.

2. Removed stop words, punctuation marks and then performed lemmatization using nltk library to generate tokens from the corpus.

3. Read all the documents and store them in a dictionary with the token as keys and document name as posting.

4. Then sorted all the posting lists. Now, If any term appears multiple times in any document then that document name appears multiple times in that term posting list so to make the unique list we call a function modify_list() in which we run a simple linear time algorithm to calculate the no. of occurrence of any document corresponding to each term.

5. This created the inverted index as a dictionary with tokens as keys, and a pair of the document name and its frequency as postings and stored the inverted index result in the model_queries_13.pth file

Task1B(Query Preprocessing):-

Steps:

1. Used BeautifulSoup to parse the data

2. Used parse function to remove punctuations, lemmatize and make it list

3. Stored the in query_dic : dictionary where key query_id and value is list of parsed words

4. Saved the data in queries13.txt file

Task1C(Boolean Retrieval):-

Steps:

1. Parsed queries13.txt file and made a dictionary parsed_query_dic similar to query_dic in part 1.b

2. Implemented get_result function take take parsed_query_dic and inverted_index created in part 1.a as input

3. Sorted the query words for each query as per their length in inverted_index in ascending order

4. Fetched lists for each query word and merged them using merge_list function one by one