INFORMATION RETRIEVAL
ASSIGNMENT 1
README

Methodology -

1. Firstly we loaded the 'stories' dataset, consisting of 452 files and two directories, and we included 15 files from the 'SRE' directory. The number of documents in the dataset became 467.

2. Then we applied various pre-processing steps such as removing stopwords, tokenization, etc. Then we created a vocabulary of words. Then we built a Unigram Inverted Index, which consists of terms and a posting list containing the document id's, i.e., the documents in which the term appeared. The data structure is as follows {'term1' : ['docid1', 'docid2'], 'term2' : ['docid3']}

3. For query pre-processing, we applied the same pre-processing steps which are applied to the documents.

4. For developing the merge function for all boolean functions (OR, AND, OR NOT, AND NOT), we first switch the operator entered by the user and preprocess the user's query and accordingly perform the operations. We completed the following operations on query **Left to Right**.

5. For OR operation: We take two words at a time, find their postings list, add those to the final list and remove the intersection.
   For AND operation: Here again, we take two words at a time, find their postings list and find their intersection and add it to the final list.
   For AND NOT operation: After finding the postings of two words, we add the posting list of word1 into the final list and then remove the intersection part of word1 and word2 postings.
   For OR NOT operation: Here, for the first word, we find the postings, and for the second word, we find those postings in which that word is not present, basically **NOT** operation. After getting these two postings, we perform normal OR operation as stated above.

6. For finding the number of comparisons, we counted the number of times we performed merge-operations.

Preprocessing Steps -

1. **Data Cleaning -** It includes cleaning junk symbols and non-alphabetic characters. Words, like I'm, are expanded to I am .
2. **Tokenization -** Then tokens are generated
3. **Removing Stop Words -** Stop words like is, the, etc are removed.
4. **Lemmatization**

Assumptions -

- For query preprocessing, we remove the stop words and lemmatize the query such that the length of query after preprocessing is equal to the size of operations given + 1.
  For example - Input query: lion stood thoughtfully for a moment
  Input operation sequence: [OR, OR, OR] (Length 3)
  After preprocessing - lion stood thoughtfully moment (Length 4)

- We also assumed that the user enters a valid query including operations such as AND, OR, AND NOT, OR NOT.

- Every character other than letters is considered invalid .That is code will not be able to handle queries like "120 300" digits will be eliminated and the query becomes null .