Preprocessing :

The following preprocessing steps were carried out:

- Decoding to latin1 format
- Tokenizing the input text
- Removing stop words, punctuations and
- Lemmatization of words

Methodology:

After performing the above preprocessing on the input corpus, an inverted index was created for the terms with their corresponding document names in sorted order, (we have taken the comparison count while evaluating NOT)

This inverted index was then used for developing the 4 functions: OR, AND, OR NOT, AND NOT.

AND:

- A simple merge algorithm was followed with 2 pointers pointing to the posting lists of each of the 2 terms as arguments to the function. Whenever a matching document is found, both the pointers are incremented, else depending upon whichever pointer has a lower value, the corresponding pointer is incremented.

OR:

- A similar approach was followed with 2 pointers pointing to the posting lists of each of the 2 terms as arguments to the function. Since this is an OR functionality, hence documents corresponding to either of the query terms, is declared a match.

AND NOT:

- In this, first the result of the AND query is processed, and then the documents corresponding to query 1 are found out, following which the common documents(that of the result of AND) are removed from the documents belonging to query term1.

OR NOT:

- In this, first the NOT query is processed and comparisons are stored. Now OR query is processed on the result received from NOT query i.e. the documents not containing word **b** and word **a. (a OR (NOT b)).**