Carol Dcunha BDA 171046001

# Querying Shakespeare Plays

**Problem Statement:**

To query the collection of Shakespeare plays based on the combination of multiple keywords using:

- Linear Search
- Grep Command
- Term-Document Incidence Matrix

**Approach:**

Linear Search:

- The given query is broken down into words.
- Each word except boolean operators(AND and OR) is linearly searched in each document one by one. A list of documents is generated that contains the particular word. A NOT operation is performed during the search itself where the documents that don't contain the word is returned as a list.
- The lists obtained from each specific word is then combined based on the AND or OR keyword to get the final output.

Grep Command:

- Same approach as in Linear Search. However, searching is done using Linux grep command.
- grep -rl <word> <directory> -- returns the filenames in directory that contains word
- grep -rL <word> <directory> -- returns the filenames in directory that doesn't contains word

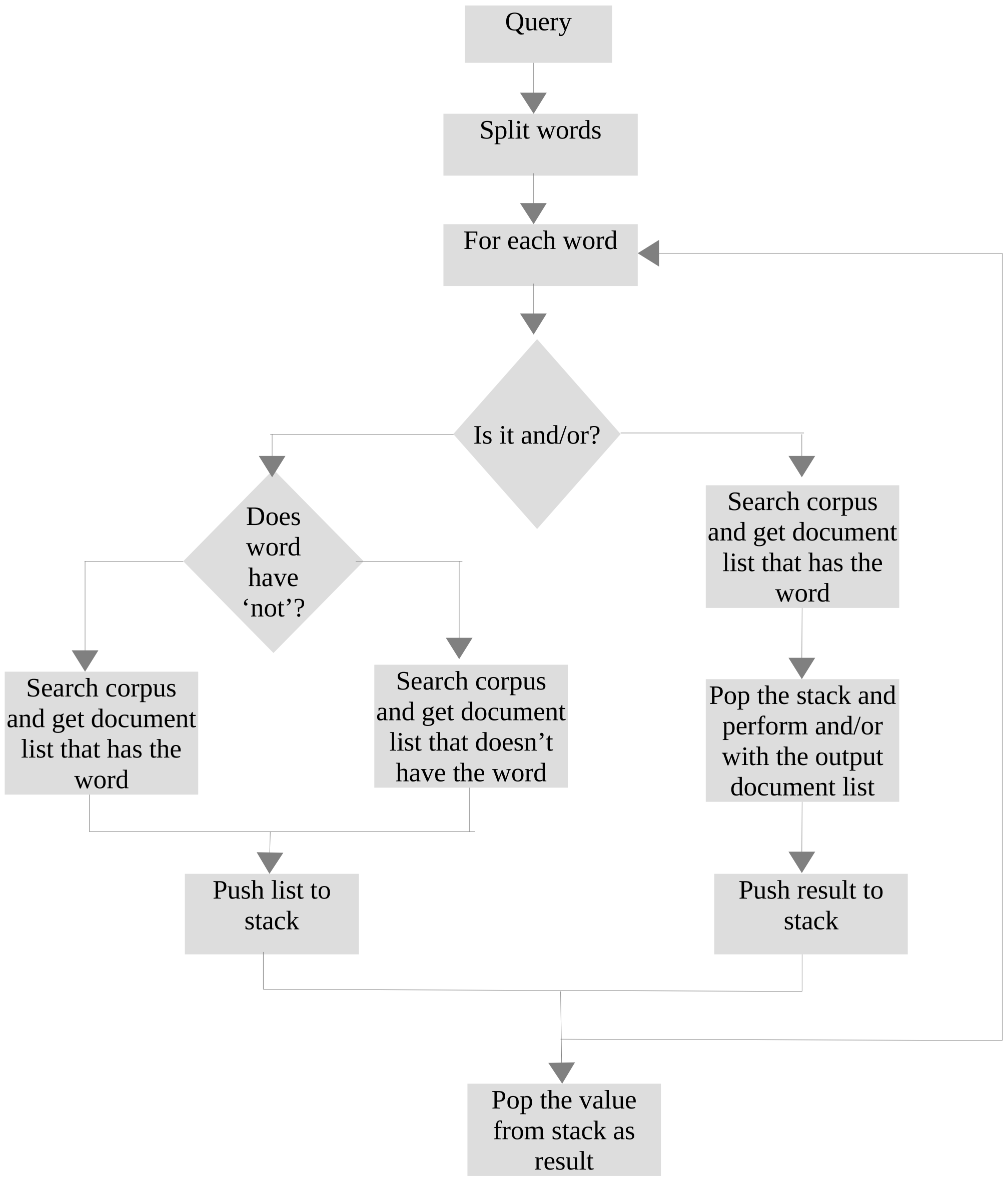
Term-Document Incidence Matrix:

- Pandas DataFrame is used where keywords are the indices and document names are the columns.
- Each time a word is encountered for search, it checks if it exist in the DataFrame. If yes, there is no need to search. Else, the word is searched in the directory and a new entry is made in the DataFrame with word as index and values True/False if it is present/absent in each document respectively.

**Findings:**

- In Linear Search, everytime a repeated word is searched in the documents. This increases the computation time.
- For Term-Document Incidence Matrix, a word is searched just once and its entry is made. Subsequent search for the same word can be taken from the matrix. However, there are many False values in the matrix that suggest wastage of space.

## Algorithm:


Query
Split words
For each word
Is it and/or?
Does word have ‘not’?
Search corpus and get document list that has the word
Search corpus and get document list that doesn’t have the word
Search corpus and get document list that has the word
Pop the stack and perform and/or with the output document list
Push list to stack
Push result to stack
Pop the value from stack as result