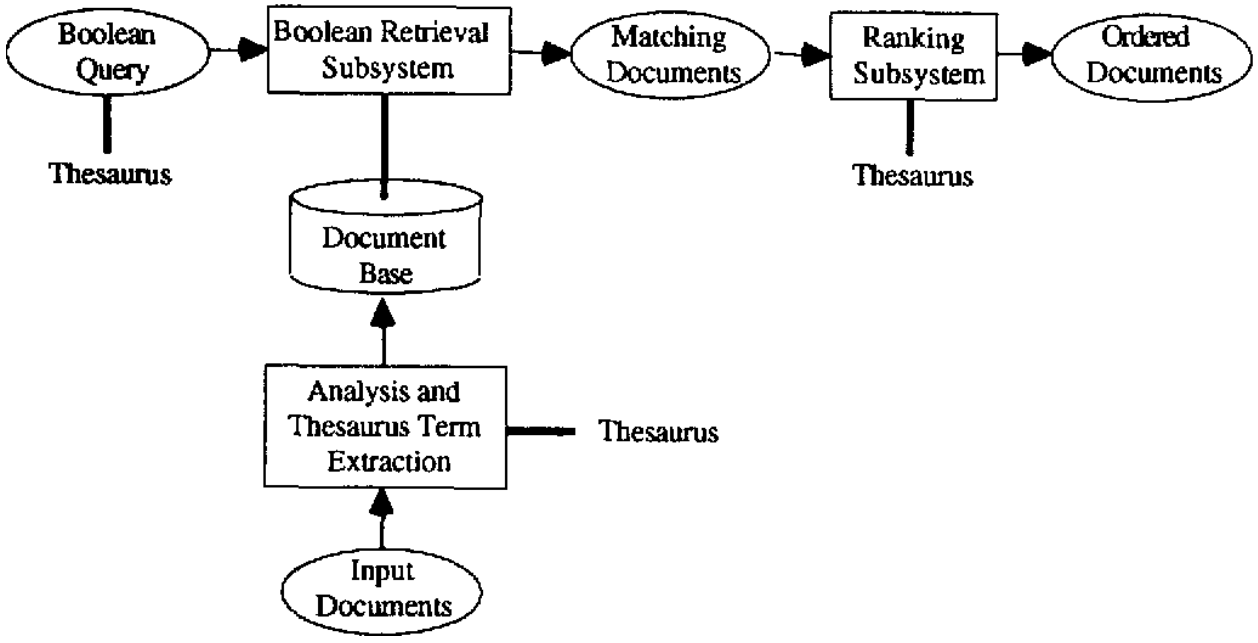
CS F469 Information Retrieval
Assignment-I
Boolean Retrieval System

Rohith Kumar Gattu - 2019A7PS0049H
Rohan Rao N - 2019A7PS0048H
Srikar Sashank M - 2019A7PS0160H

Boolean Retrieval System:

Boolean Retrieval System is one of the models of retrieving data in Information Retrieval. It is a simple retrieval model which is based on set theory and boolean algebra.
Queries are designed as boolean expressions which have very precise semantics and the retrieval strategy is based on binary decision criteria. It just considers if the index terms are present or absent in a document.

The following is the architecture of any boolean retrieval system.



The important data structures used in the application are Inverted index, permuterm index and linked lists.

Inverted Index:
An inverted index is an index data structure storing a mapping from content such as words, numbers from their corresponding documents. They are also called posting lists.
It is a hashmap that directs from a word to its corresponding document.

Permuterm index:
Permuterm index is a special type of index data structure that is generally used for wild card queries where characters in the query are permuted and mapped to the augmented word.
The key is to rotate the wildcard query such that it enables us to find the original vocabulary terms that match the given wildcard query.

Skip lists:
A skip list is a probabilistic data structure that is used to store a sorted list of data using linked lists.It allows the process of elements to view efficiently. It skips several elements in one single step.

Trie:
Trie is a tree data structure which is used to locate a specific key word within a set.These keys are often strings. In order to access the keys, the tree is completely traversed in dfs fashion following the links between the nodes.

Hashtable:
A hash table is a data structure that implements an associative array abstract data type, a structure that can map keys to values. A hash table uses a hash function to compute an index, also called a hash code, into an array of buckets or slots, from which the desired value can be found.

Linked Lists:
A linked list is a linear data structure where elements are not stored in a contiguous memory while they are connected with pointers.

Pre-Processing steps in a Boolean retrieval system.

Text preprocessing is the process of preparing the text data ready for the machine to use that data to perform tasks like analysis, predictions, etc.
Following are some of the text pre-processing techniques used in this application.

1. **Removal of Stop words:**
   The words which are generally filtered out before processing any natural language are called stop words. These generally are more frequently occurring words like prepositions, articles, conjunctions, etc.
   Not all the times stop words are removed while sometimes these are retained to preserve the meaning of a sentence.
   The following is the code to remove the stop words using nltk library.

```
import nltk
from nltk.corpus import stopwords
sw_nltk = stopwords.words('english')
print(sw_nltk)
```

2. **Stemming.**
   Stemming is a process of producing morphological variants of root/base word. The input for the stemmer is tokenized words, where different words are stemmed to the root word.
   The following is the code snippet to perform stemming using nltk library.

```
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize

ps = PorterStemmer()
```

3. **Lemmatization.**
   Lemmatization is the process of grouping together different inflected forms of words so as to consider them as a single word.
   The following is the code snippet to perform lemmatization using nltk library.

```
from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()
```

Spelling correction: Edit Distance method.

Edit Distance is the method to quantify how similar or dissimilar two strings are to one another by counting the minimum number of operations to make one string to another.

Application running time and output is as follows:

```
Enter boolean query: QUEEN OR KING
Processing time: 0.0002751 secs

Doc IDS:
[2, 3, 4, 5]

Enter boolean query: KING
Processing time: 0.0001718 secs

Doc IDS:
[2, 3, 4, 5]

Enter boolean query: QUEEN
Processing time: 0.0001167 secs

Doc IDS:
[2, 3, 4, 5]

Enter boolean query: HONEY
Processing time: 0.0001383 secs

Doc IDS:
[5]

Enter boolean query: KING AND HONEY
Processing time: 0.0002494 secs

Doc IDS:
[5]
```

```
urchinsnout: [43]
looketh: [43]
tusk: [43]
cofferlid: [43]
oerstraw: [43]
combusti: [43]
newsprung: [43]
greendrop: [43]
sweetsmel: [43]
fini: [43]

Enter boolean query: KING AND QUEEN
KING AND QUEEN
KING AND QUEEN
ENTER THE CORRECTED INPUT: KING AND QUEEN
Processing time: 2.243 secs

Doc IDS:
[2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17, 18, 19, 20, 21, 25, 26, 27, 28, 29, 31, 32, 33, 34,
35, 36, 37, 38, 40, 41, 42, 43]
Enter boolean query: KING AND QUEN
KING AND QUEN
KING AND QUEEN
ENTER THE CORRECTED INPUT: KING AND QUEEN
Processing time: 7.818 secs

Doc IDS:
[2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17, 18, 19, 20, 21, 25, 26, 27, 28, 29, 31, 32, 33, 34,
35, 36, 37, 38, 40, 41, 42, 43]
Enter boolean query:
```