BOOLEAN IR SYSTEM - DESIGN DOCUMENT

Project Specifications:

The project uses the NLTK, re and os libraries. Re is used to enable wildcard query handling through regex. OS is necessary for loading the dataset into memory from any directory/file present on the system. NLTK is the most important library in the project, which is essential for stop word removal and stemming processes used.

The entire project is written in a Jupyter notebook. This was chosen to easily make modular pieces of code in 'cells' that could be run independently. Jupyter notebooks can also be easily exported into a python script.

As of now, all queries can be run in the same notebook by creating a new cell (format for sending a query has been specified in the README). If the notebook is exported to python, the same functionality can be provided by a menu-driven interface.

The word_doc and bigram_index (refer below) indexes must be built before any queries can be processed. Since this requires loading of NLTK libraries and the entire pre-processing steps of stop word removal and lemmatization, it is the lengthiest process and takes about **25 seconds** overall. This can be performed in a single step by using the "Execute all above cells" command in Jupyter notebook.

Beyond this, query processing is very efficient and completes in **0.1 - 0.3 seconds per query.**

Data Structures:

The application uses two inverted indexes:

word_doc stores all the documents where a particular word occurs, for each word. This uses a string:set of integers as key-value pairs.

bigram_index stores all the words where a particular bigram is found, for each bigram. Hence, this index is of the form bigram:set of words in the key-value organisation.

The data structures used in the system are:

a) Dictionaries - Standard python dictionaries for easy implementation of key-value pairs.

b) Sets - The posting lists are stored as sets to utilize the efficiency of hash sets. Python's inbuilt powerful hashing capabilities ensure collisions are kept to a minimum. Using sets also is greatly helpful to simplify boolean queries into set operations (also easily available in python).
c) Lists - Miscellaneous; general-purpose, temporary storage structures due to convenience of usage