**GENRE BASED MOVIE RETRIEVAL SYSTEM**

Abhay Iyer<150953094>

Pawan Kurada<150953023>

Krishna Sameer<150911050>

Vineeth Chitturi<150911252>

**PROBLEM STATEMENT:**

Implementing a Boolean retrieval model with the functionalities mentioned below:

* Reading the content of a document- Dataset of movie documents of .csv format is taken into consideration. This dataset comprises approximately 5000 movies.
* Creating an inverted index-Inverted Index contains Terms and Postings where terms are the genres and postings are the name of the movies.
* Using postings list intersection algorithm without using skip pointer- To map different query terms posting list algorithm is used.
* Operators in the query- consists of ‘AND’ or ‘OR’ or both Boolean operators

* Displaying the query results as well as inverted index

**IMPLEMENTATION DETAILS:**

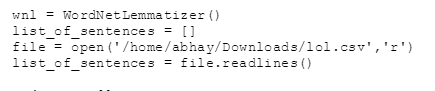
1. **To read the documents** we have implemented the file handling concept in python. File Handling in python is simple to use and easy to operate on.

Functions used are Open() and Readline()

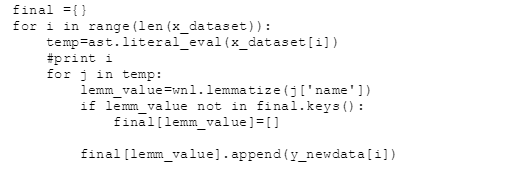
Open(): Is used to open an existing file.

Readline(): Is used to read lines from the file and store them into a dataset.

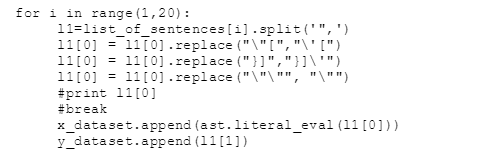
The stored data is then operated upon to extract required Information.



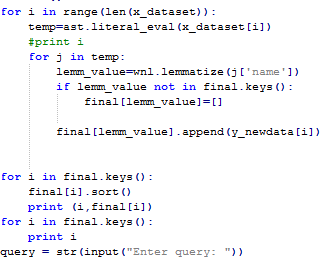
1. **Generating an inverted index**:-
   1. Tokenization of the documents is done by considering each comma separated genre from the .csv file
   2. Linguistic processing is done by using NLTK library of python. NLTK (Natural language toolkit) provides easy to use interfaces along with text processing libraries for classification, tokenization, stemming etc. We lemmatize the input so that every term is converted into it’s base form. That is, Romans gets converted to Roman, Friends gets converted into Friend etc.



* 1. In this step we index the documents. Instead of assigning a document ID for each document we use the name of a movie as it’s ID



1. We next sort the posting lists i.e. the genres in the lexicographical order.

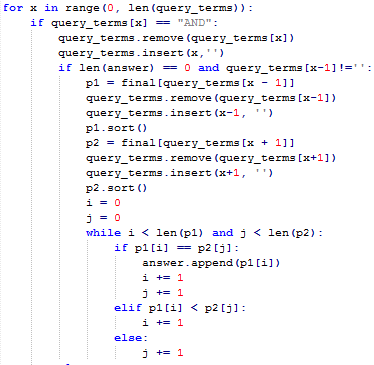


**INTERSECTION OF POSTINGS LIST:**

To intersect the posting lists of various genres we use posting lists intersection algorithm for more than 2 terms.(without using skip pointers)

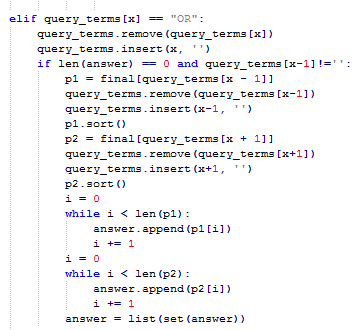
When the query has “AND” operator:

The intersection of the first two terms is found, if it is null then we don’t have to consider the next term but if it is not null then the result of the first is intersected with the last term and the results are displayed.



Implementing the query of “OR” operator:

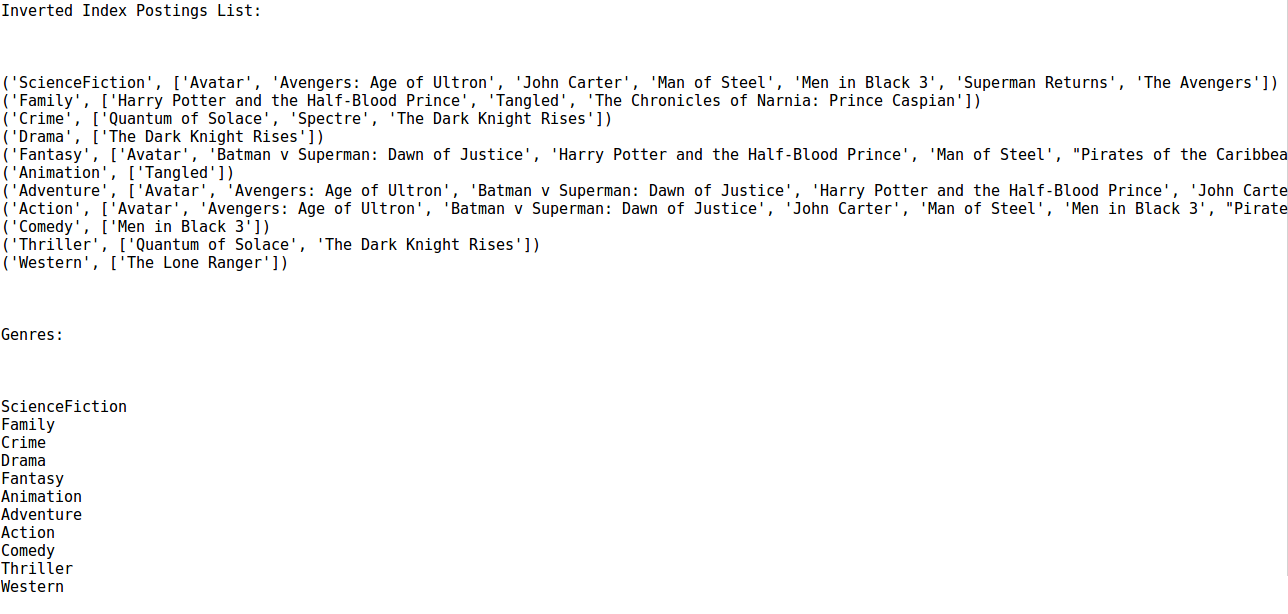
The union of the first two terms is found and the result is again union with the last term and appropriate results are displayed.

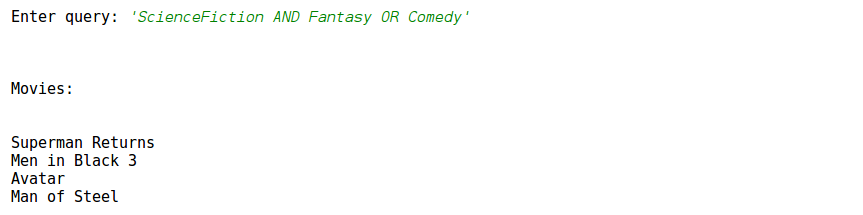


**CONCLUSION:**

This is the output obtained from the execution of our Boolean retrieval system.

**Interface snippets:**

****

****

In conclusion, our code successfully reads the content of the document to extract the necessary information and creates an inverted index out of it. Using the postings list intersection algorithm we obtain the correct query outputs which in this case is the name of the movies which falls in the respective genres as desired by the user. Our code implementation works for a set of 5000 movies and can be scaled even further. This Boolean retrieval technique is applied in a very basic manner in real world search engines along with a lot more functionality. The project applies concepts of linguistic preprocessing and creation of other real world objects which essentially gives us an insight into the functioning of a search engine.

**REFERENCES:**

1. The Quick Python Book, Second Edition – Vernon L. Ceder
2. An Introduction to Information Retrieval – Christopher D.Manning , Prabhakar Raghavan , Hinrich Schutze
3. [www.nltk.org](http://www.nltk.org)
4. <https://media.readthedocs.org/pdf/nltk/latest/nltk.pdf>