IR Assignment - 1

Akhil Mahajan | MT20107

Anjali | MT20082

Prabal Jain | MT20115

Shradha Sabhlok | MT20069

Dataset :

The dataset contains archives of few stories, and this dataset has lots of documents in different formats, which contains 467 files. All the documents contain text data, which are words in English. The dataset folder contains two folders, and the rest of the data is present as files. Each document has a different name. Documents though containing text data, have different extensions like .txt, .hum, .fic, .dos, and many more. There is an index.html file in each folder listing the names & titles of all the documents present in that particular folder. So, we do not have to open each document to extract its title. Some titles are not center-aligned, while most are, but that is not a hindrance in extracting titles as the index files provide that.

Methodology :

The following steps were performed to retrieve documents for the given query:

1. The **stories** dataset was loaded using the index files that were present in each folder.
2. Preprocessing steps, as mentioned in the preprocessing section, were performed on the dataset.
3. After preprocessing, postings were created for all the tokens in the entire dataset. To create postings, tokens were made for each file, and for each token, a column was created in the postings dataframe. Then the respective document id was added under that token.
4. Queries were taken from the user in the format mentioned in the assignment.
5. Each query was preprocessed. After preprocessing, for each command word like OR, AND; **binary_operations** function was called inside, which OR operations were performed using **union** operation and AND operations were performed using **intersection** operation.
6. The **gen_not_set** function was called, inside which the **get_difference** function was called to handle the NOT command word. get_difference function uses the set difference operation to return the difference between the document ids of the entire dataset and the document ids of the corresponding query word.

7. To count the number of comparisons, the **count_comparisons** function was created inside which a count variable was declared to count the number of comparisons happening during AND, OR operations.
8. Thus, we could see the retrieved documents and the number of comparisons for a given query.
9. To view postings for a single word, **the print_postings** function was created. In this function, postings for a given word were retrieved from the postings dataset.
10. To read the content of a document/file, the **view_doc** function was created. In this function, the file corresponding to the id given as an input to the function was opened, and its contents were read using the read() function.

Preprocessing Steps :

Below Functions were implemented to perform Preprocessing :

1. def convert_lower_case(text): It takes text as an argument and converts the given text to the lower case.

2. def remove_stop_words(text): Stopwords are the commonly used words such as “the”, “a”, “an”, “in” so we have implemented this function to remove stop words from the text using nltk.corpus stopwords.

3. def remove_punctuation(text): Since we have to retrieve the document according to the given query, our query should not contain any punctuations symbols like !\"#$%&()*+-./:;<=>?@[\]^_`{|}~.

4. def remove_apostrophe(text): This function takes an argument as text and removes apostrophe from the text.

5. def lemmatization(text): Lemmatization is a way to reduce the word to the root synonym of a word, ensuring that the reduced word must be a dictionary word.

Getting final Query :

To convert the input format into an appropriate preprocessed query, we call the above preprocessing steps on the query, which will preprocess our final query will be preprocessed. It also works for the case scenarios in which two different words are joined together with a special character (for example lion@lion).

Now, Vocab Words will be extracted from our input Query. To pre-process the Second input, which is a list of operation sequences, we have appended the operations in our preprocessed input query from left to right. We made sure that the input sequence operations must fit in the required space correctly.

Results :

Query Preprocessing :

```
Enter the Number of Queries: 2
Enter Input Query lion stood thoughtfully for a moment
['lion', 'stood', 'thoughtfully', 'moment']
Enter input Sequence with comma in between operators [ OR, OR , OR ]
Final Query is lion OR stood  OR  thoughtfully OR moment
Enter Input Query telephone,paved, roads
['telephone', 'paved', 'road']
Enter input Sequence with comma in between operators [ OR NOT, AND NOT ]
Final Query is telephone OR NOT paved AND NOT road
```

For Query 1:

```
Query Words: ['lion', 'stood', 'thoughtfully', 'moment']

Commands: ['or', 'or', 'or']

Number of documents matched for query 1 are: 270

Number of comparisons done for query 1 are: 670

Retrieved Document Ids for query 1 are: [1, 2, 3, 5, 6, 7, 8, 9, 11, 13, 14, 16, 18, 19, 20, 23, 24, 26, 27, 29, 30, 31, 34, 36, 37, 39, 40, 4
1, 42, 48, 49, 50, 53, 54, 56, 58, 59, 60, 61, 63, 64, 65, 66, 68, 69, 70, 71, 72, 73, 75, 76, 77, 78, 79, 80, 82, 83, 85, 87, 90, 91, 92, 93,
94, 95, 97, 100, 107, 110, 112, 113, 116, 118, 119, 120, 123, 124, 126, 128, 129, 130, 131, 133, 134, 135, 140, 141, 142, 143, 144, 145, 147, 1
48, 150, 151, 153, 154, 155, 158, 159, 162, 163, 169, 170, 171, 172, 173, 174, 175, 176, 179, 180, 182, 183, 184, 186, 187, 189, 190, 191, 192,
193, 196, 200, 202, 205, 206, 207, 208, 209, 210, 211, 213, 214, 217, 218, 219, 221, 223, 224, 225, 226, 228, 229, 230, 232, 233, 234, 236, 23
8, 239, 240, 241, 242, 244, 245, 246, 247, 249, 252, 256, 257, 259, 260, 261, 263, 265, 266, 267, 269, 270, 272, 273, 275, 277, 278, 279, 284,
286, 288, 289, 290, 296, 297, 298, 300, 303, 304, 310, 312, 318, 319, 320, 321, 322, 323, 327, 328, 329, 332, 333, 335, 336, 337, 341, 345, 34
6, 349, 356, 359, 360, 362, 363, 364, 366, 367, 368, 369, 370, 371, 372, 375, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389,
390, 391, 393, 394, 395, 396, 402, 403, 407, 408, 410, 413, 418, 420, 425, 426, 428, 430, 431, 433, 434, 435, 436, 441, 442, 443, 450, 451, 45
5, 457, 458, 460, 461, 463, 466]

List of Retrieved Documents for query 1 is:
13chil.txt
14.lws
16.lws
18.lws
19.lws
20.lws
3gables.txt
3lpigs.txt
3student.txt
4moons.txt
5orange.txt
6napolen.txt
7voysinb.txt
ab40thv.txt
abbey.txt
adv_alad.txt
advsayed.txt
aesop11.txt
aesopa10.txt
```

For Query 2:

```
After Processing NOT: ['or', 'and', 'not'] ['telephone', 1, 'road']

After Processing NOT: ['or', 'and'] ['telephone', 1, 2]

Query Words: ['telephone', 1, 2]

Commands: ['or', 'and']

Number of documents matched for query 2 are: 345

Number of comparisons done for query 2 are: 809

Retrieved Document Ids for query 2 are: [2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 17, 18, 19, 22, 23, 24, 25, 28, 29, 30, 31, 32, 35, 36, 38, 40, 4
2, 43, 45, 46, 47, 48, 51, 52, 53, 54, 55, 57, 58, 59, 60, 62, 63, 64, 65, 67, 68, 69, 70, 71, 72, 73, 75, 80, 81, 82, 83, 84, 85, 86, 88, 89,
90, 91, 93, 94, 95, 96, 98, 99, 100, 102, 104, 105, 107, 108, 110, 111, 112, 113, 114, 115, 117, 120, 121, 122, 124, 125, 127, 128, 131, 132, 1
34, 136, 137, 138, 139, 140, 142, 144, 145, 146, 147, 148, 151, 152, 156, 157, 158, 159, 160, 161, 164, 165, 166, 167, 168, 170, 171, 172, 175,
176, 177, 178, 180, 181, 182, 184, 185, 186, 187, 188, 190, 191, 192, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 208, 21
0, 211, 212, 213, 215, 216, 217, 218, 219, 220, 222, 227, 228, 229, 231, 232, 233, 234, 235, 237, 238, 239, 240, 242, 243, 245, 246, 247, 248,
249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 261, 262, 263, 264, 266, 267, 268, 271, 272, 273, 274, 275, 276, 278, 281, 282, 283, 28
5, 286, 287, 288, 289, 292, 293, 294, 296, 297, 299, 302, 305, 306, 307, 308, 309, 311, 312, 313, 314, 316, 317, 318, 319, 321, 322, 323, 324,
325, 326, 327, 328, 329, 330, 331, 333, 334, 335, 336, 338, 339, 340, 341, 342, 343, 344, 345, 347, 348, 349, 350, 351, 352, 353, 354, 355, 35
6, 357, 358, 360, 363, 364, 365, 368, 370, 371, 372, 374, 376, 377, 381, 382, 384, 385, 389, 390, 391, 394, 396, 397, 398, 399, 400, 401, 402,
403, 404, 405, 406, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 426, 427, 429, 430, 432, 436, 437, 43
8, 439, 440, 442, 443, 444, 445, 446, 448, 449, 452, 453, 454, 455, 456, 458, 459, 460, 461, 462, 463, 464, 465, 466]

List of Retrieved Documents for query 2 is:
14.lws
17.lws
18.lws
19.lws
20.lws
3gables.txt
3lpigs.txt
3sonnets.vrs
3student.txt
3wishes.txt
4moons.txt
```