IR Assignment-3

- SUBHANI SHAIK (MT18117)

1. Tf-Idf based vector space document retrieval to get top 10 documents based on a cosine similarity between query and document vector:

Dataset: 20newsgroups dataset

In this dataset, used **comp.graphics, rec.motorcycles** which have 1000 documents each.

Pre-Processing the dataset:

For pre-processing the text, used NLTK library and done following steps:

- Conversion to lowercase
- Contractions
- Removed unnecessary characters and punctuations and tokenized using RegexpTokenizer.
- Lemmatization

Procedure:

- Initially done the Vector space model using TF-IDF based vector representation.

$$\textbf{Tf-Idf=(tf/N)*log(}N_d\textbf{/df)}$$

Where

tf= frequency of a term in the document

N= number of words in the document

N_d = number of documents in the collection

df = frequency of the document

- By using this formula, done vector of features for our corpus.
- Given a query and done the pre-processed it.
- From output of pre-processed query terms, retrieved the postings of all query terms and took 'union' of them.
- Built the Query vector for query like vector representation of our corpus.
- Now done the cosine similarity for each document and stored their similarity scores.
- Sorted the similarity scores in decreasing order and retrieved the 'top k' documents from the retrieved documents.
- Later asked user to give feedback on resulted output.
- Took feedback from user as marking relevant documents and non-relevant documents.
- Now, updated the query vector using the Rochhio's algorithm by taking parameter values:

updatedQuery = alpha*Q + beta * (relevant docs vectors mean) – gamma * (non-relevant docs vectors mean)

$$\overrightarrow{Q_m} = \left(a \cdot \overrightarrow{Q_o}\right) + \left(b \cdot \frac{1}{|D_r|} \cdot \sum_{\overrightarrow{D_j} \in D_r} \overrightarrow{D_j}\right) - \left(c \cdot \frac{1}{|D_{nr}|} \cdot \sum_{\overrightarrow{D_k} \in D_{nr}} \overrightarrow{D_k}\right)$$

where,

alpha=a=1,beta=b=0.75,gamma=c=0.15

- Plotted the 2D TSNE plot for relevant and non-relevant points and Query vector.
- Repeated the process until user quit the program.

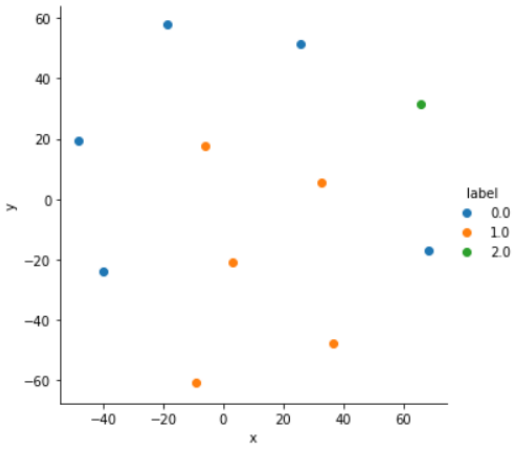
Example:

Input: Motor cycle

Output:

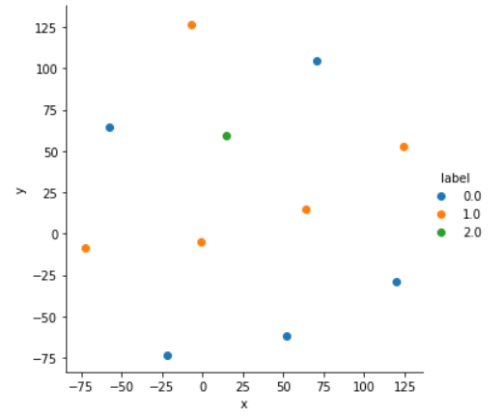
```
total_postings: 66
0 --> ['20_newsgroups/rec.motorcycles/104674', 0.15502359515452596]
1 --> ['20_newsgroups/rec.motorcycles/104754', 0.1347805735413942]
2 --> ['20_newsgroups/rec.motorcycles/105236', 0.12071042956736978]
3 --> ['20_newsgroups/rec.motorcycles/104502', 0.10912162114759702]
4 --> ['20_newsgroups/rec.motorcycles/104966', 0.10396224326967393]
5 --> ['20_newsgroups/rec.motorcycles/103226', 0.090658499829332]
6 --> ['20_newsgroups/rec.motorcycles/103184', 0.08463324655142221]
7 --> ['20_newsgroups/comp.graphics/39005', 0.07724735793560815]
8 --> ['20_newsgroups/rec.motorcycles/104630', 0.07666162730318983]
9 --> ['20_newsgroups/rec.motorcycles/105251', 0.07546043666008086]

---------------------
choice:
1.Give feedback
2.exit
enter choice:1
enter index(comma seperated) of relevant doc:1,2,5,6,7
['1', '2', '5', '6', '7']
[0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 2]
```




```
total_postings: 66
0 --> ['20_newsgroups/rec.motorcycles/104754', 0.1635686974295711]
1 --> ['20_newsgroups/rec.motorcycles/104674', 0.1567563431798681]
2 --> ['20_newsgroups/rec.motorcycles/105236', 0.14505081345205403]
3 --> ['20_newsgroups/rec.motorcycles/104502', 0.12687211645763852]
4 --> ['20_newsgroups/rec.motorcycles/103184', 0.11640850550982908]
5 --> ['20_newsgroups/rec.motorcycles/103226', 0.11544680609457932]
6 --> ['20_newsgroups/comp.graphics/39005', 0.11336374488286062]
7 --> ['20_newsgroups/rec.motorcycles/104966', 0.10466506103388978]
8 --> ['20_newsgroups/rec.motorcycles/104630', 0.0840598046844764]
9 --> ['20_newsgroups/rec.motorcycles/105251', 0.07825682865288765]

---------------------
choice:
1.Give feedback
2.exit
enter choice:1
enter index(comma seperated) of relevant doc:3,4,6,7,8
['3', '4', '6', '7', '8']
[0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 2]
```



Here

label=0 means non-relevant;

label=1 means relevant;

label=3 means query vector.

These retrieved documents are drawn according to this query vector.

2. Precision-Recall Curve:

Dataset: IR-assignment-3-data.txt file which is Microsoft bing dataset.

Pre-Processing the dataset:

- Extracted only ‘qid:4’ data points from the dataset which are 103 in number.
- Extracted tf-idf features which are 75 dimensions in number and extracted corresponding labels.
- Removed unnecessary characters from the extracted data points.

Procedure:

- Initially, sorted the extracted data points according to tf-idf values.
- Now taken ‘top k’ documents each times (where k=1 to 103) and considered each times ‘top k’ points as relevant and remaining as non-relevant.
- From them calculated the precision and recall values and plotted the graph.
- It is shown that recall is reached to value ‘1’ but precision is fluctuating each time.
- Later, plotted the interpolated precision-recall graph using the precision-recall values.

Output: