Assignment 2

Submitted by

Swati Verma

MT19073

Question 1.

1. Document retrieval using the Jaccard Coefficient.

```
input query :
“50,000 variety of flowers”
enter the value of k:
10
Top 10 documents using Jaccard coefficient are:  [('narciss.txt', 0.009900990099009901), ('quarter.c9', 0.00943396226415094 3), ('wall.art', 0.007518796992481203), ('lgoldbrd.txt', 0.006944444444444444), ('ccm.txt', 0.006097560975609756), ('tin', 0.005181 3471502590676), ('bulolli2.txt', 0.004424778761061947), ('stainles.ana', 0.0039840637450199 2), ('bram', 0.0037313432835820895), ('lpeargrl.txt', 0.0036101083032490976)]
```

2. Document retrieval using tf-idf.

```
input query :
“50,000 variety of flowers”
enter the value of k:
10
which variation of idf do you want:
2
which variation of tf do you want:
1
Top 10 documents using tf-idf are:  [('ghost', 0.009417618359298769), ('narciss.txt', 0.006171300219697968), ('mcdonaldl.txt', 0.0056896923985600145), ('day.in.mcdonald', 0.005679384984794507), ('quarter.c9', 0.005368185807545493), ('tin', 0.004790071128 906554), ('wall.art', 0.004721416433142422), ('lgoldbrd.txt', 0.004259538738595881), ('bulnoopt.txt', 0.00413227659702096), ('c cm.txt', 0.003978452425896 6595)]
```

3. Document retrieval using tf-idf considering title weightage.

```
input query :
“50,000 variety of flowers”
enter the value of k:
10
which variation of idf do you want:
2
which variation of tf do you want:
1
Top 10 documents using tf-idf with title are:  [('ghost', 0.006592332851509138), ('narciss.txt', 0.004319910153788578), ('mcdon aldl.txt', 0.0039827846789920 1), ('day.in.mcdonald', 0.0039755694893561545), ('quarter.c9', 0.003757730065281845), ('tin', 0.00 3353049790234587 3), ('wall.art', 0.003304991503199695), ('lgoldbrd.txt', 0.0029816771170171163), ('bulnoopt.txt', 0.00289259361 7914672), ('ccm.txt', 0.0027849166981276613)]
```

4. Document retrieval using Cosine Similarity.

```
input query :
“50,000 variety of flowers”
enter the value of k:
10
which variation of idf do you want:
2
top  10 documents using cosine similarity are :  [('non4', 0.02369306580117141), ('ghost', 0.017179711894426436), ('tin', 0.016
69870314208227), ('bulolli2.txt', 0.015147833759114038), ('stainles.ana', 0.014745250642026414), ('fic4', 0.01419161081012072
8), ('narciss.txt', 0.01390700689726716), ('bram', 0.013727081298052996), ('mcdonaldl.txt', 0.013353334456296374), ('day.in.mcd
onald', 0.013341233561689194)]
```

5. Document retrieval using Cosine Similarity considering title weightage.

```
print("input query :")
query=input()
print("enter the value of k:")
k=int(input())
final_idf={}
print("which variation of idf do you want: ")
var=int(input())
final_idf=idf_variation(var)
output=cosine_similarity_with_title(query,k)
print("top", k, "documents using cosine similarity considering title are: ",output)
```

```
input query :
“50,000 variety of flowers”
enter the value of k:
10
which variation of idf do you want:
2
top 10 documents using cosine similarity considering title are:  [('non4', 0.016585146060819986), ('ghost', 0.01202579832609850
5), ('tin', 0.011689092199457587), ('bulolli2.txt', 0.010603483631379826), ('stainles.ana', 0.01032167544941849), ('fic4', 0.00
9934127567084509), ('narciss.txt', 0.009734904828087011), ('bram', 0.009608956908637096), ('mcdonaldl.txt', 0.009347334119407
46), ('day.in.mcdonald', 0.009338863493182435)]
```

Question 2

```
Enter the input string

swati verma hello
input the value of k
10
['swati', 'verma']
{'swati': [('swat', 1), ('sat', 2), ('at', 3), ('satin', 3), ('st', 3), ('sw', 3), ('swain', 3), ('swath', 3), ('swats', 3),
('sweat', 3)], 'verma': [('vera', 1), ('era', 2), ('em', 3), ('er', 3), ('ma', 3), ('ra', 3), ('versa', 3), ('a', 4), ('aver',
4), ('e', 4)]}
```

Analysis:

1. Jaccard coefficient is easy to compute so it is the easiest of all the three techniques.
2. In Jaccard coefficient based document retrieval we are only taking union and intersection of query and document and we are not considering how my times each word is coming in the document. So, it may give us the documents in which words occur less frequently than the documents in which a particular word occurs very frequently. For example, the word “flower” is coming only one time in the document narciss.txt then also it gives narciss.txt first rank, while in ghost.txt the word “flower” is coming many times then also it is given less importance. This is one of the major cons of using Jaccard coefficient based information retrieval.
3. In Jaccard coefficient based information retrieval, rare values are more informative than frequent values, hence gives more false-positive values.
4. Tf-idf based search gives weightage to term frequency, but if a word occurs more frequently in a document does not imply that the particular document is relevant. For example, stops words occurs more frequently in any document but that document might not be relevant. That is why we also consider document frequency that is that word is occuring in how many documents. So calculating tf-idf gives better result than Jaccard.
5. Tf-idf does not consider the ordering of query terms, so this method also my not give the most relevant document.
6. Cosine similarity-based document retrieval computes the similarity between document and query and determines how relevant a document is. That is why it performs better than Jaccard. For example, the query “The rigors of life in the bush are told in tales of man eating mosquitoes, of murderous hordes of black flies, of the lumps of flesh carried away by the giant bull dog flies” is present in document 100west.txt but when we are calculating tf-idf then 100west.txt got 6th rank while in cosine similarity 100west.txt got 2nd rank. So we can say cosine similarity is performing better.