CS 620/DASC 600 : Introduction to Data Science | Homework 4

Herambeshwar Pendyala | 01130541
hpend001@odu.edu

	Doc1	Doc2	Doc3
car	27	4	24
auto	3	33	0
insurance	0	33	29
best	14	0	17

term	df_t
car	18,165
auto	6723
insurance	19,241
best	25,235

For the Questions (1) and (2), clearly show the intermediate calculations. Tip: Use an Excel sheet for the calculations.

1) Consider the table of term frequencies for 3 documents denoted Doc1, Doc2, and Doc3 and the document frequency of same terms in a document collection of 806,791 documents.

a. Convert the raw term frequencies of car, auto, insurance and best using max frequency normalization (tf of most common term in the document).

```
Normalised Term Frequency - dividing by the frequency of the most common
term in the document:
       tfij =  fij  / maxi{fij}
```

Normalised Term Frequency			
Terms	Doc1	Doc2	Doc3
car	1	0.1212121212	0.8275862069
auto	0.1111111111	1	0
insurance	0	1	1
best	0.5185185185	0	0.5862068966

b. Compute the idf weights for the terms car, auto, insurance, and best using given df in the second table (number of documents, N=806,791). Note: Use base 2 for log scale (idft = log2(N/dft)).

Inverse Document Frequency.	
Terms	IDF
car	5.473283558
auto	6.907267869
insurance	5.390261142
best	4.999018837

c. Calculate the tf-idf weights for the terms car, auto, insurance, best and create document vectors for each of the document where each vector has four components, one for each of the four terms.

	TF-IDF Weights		
Terms	Doc1	Doc2	Doc3
car	5.473283558	0.66342831	4.529613979
auto	0.7674742076	6.907267869	0
insurance	0	5.390261142	5.390261142
best	2.592083841	0	2.930459318

Document Vectors				
Documents	car	auto	insurance	best
doc1	5.473283558	0.7674742076	0	2.592083841
doc2	0.66342831	6.907267869	5.390261142	0
doc3	4.529613979	0	5.390261142	2.930459318

Documents	Vector
Doc1	5.47 * car + 0.767 * auto + 0 * insurance + 2.59 * best
Doc2	0.66 * car + 6.90 * auto + 5.39 * insurance + 0 * best
Doc3	4.52 * car + 0 * auto + 5.39 * insurance + 2.93 * best

2) Consider the query “best car insurance”.

a. Transform the query into vector space using the same df values in the above table and calculate the tf-idf weights for the query without any normalization.

Vector Space	Terms - Term Frequency			
Query	best	car	insurance	auto
best car insurance	1	1	1	0

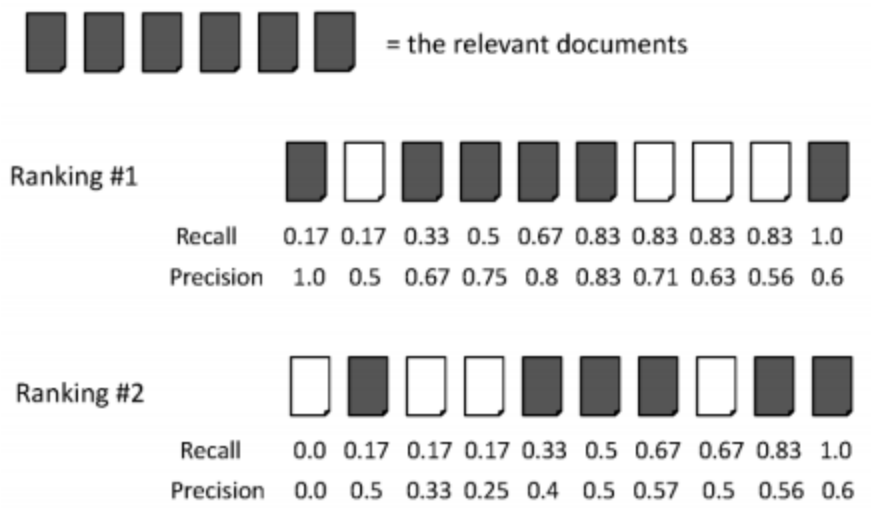
Vector Space	Terms			
Query	car	auto	insurance	best
best car insurance	5.473283558	0	5.390261142	4.999018837

b. Based on the document vectors calculated in question 1, rank the 3 documents for the given query using cosine similarity.

$$\text{CosSim}(d_j, q) = \frac{\vec{d}_j \cdot \vec{q}}{\left|\vec{d}_j\right| \cdot \left|\vec{q}\right|} = \frac{\sum_{i=1}^{t} (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^{t} w_{ij}^2 \cdot \sum_{i=1}^{t} w_{iq}^2}}$$

	sum of squares of each terms in each document	Square root of (sum of squares of document * query	Numerator = A.B	Denominator = sqrt(A^2 * B^2)	Cosine Similarity	Rank
doc1	37.2647482	55.94918271	42.91470885	55.94918271	0.7670301293	2
doc2	77.20540171	80.53200185	32.68604644	80.53200185	0.4058764924	3
doc3	58.15990979	69.89667447	68.49619822	69.89667447	0.9799636212	1
	Sum of squares of each document					
q	84.00193741					

3) Consider the 2 ranking algorithms in the figure below.



a. Calculate the confusion matrix values (tp, fp, tn, fn) for the position 7 in each ranking method.

Ranking 1 For position 7	Confusion Matrix	relevant	NonRelavant
	retrieved	5	2
	not retrieved	1	2

Ranking 2 For position 7	Confusion Matrix	relevant	NonRelavant
	retrieved	4	3
	not retrieved	2	1

b. Using the confusion matrix calculated above, compute the Accuracy and Harmonic Mean at position 7 for both ranking methods.

Accuracy	(tp + tn)/ (tp + tn + fp + fn)	0.7
Precision (P)	tp / (tp + fp)	0.7142857143
Recall (R)	tp / (tp + fn)	0.8333333333
F1-measure (Harmonic Mean)	(2 * r * p)/(r + p)	0.7692307692

Accuracy	(tp + tn)/ (tp + tn + fp + fn)	0.5
Precision (P)	tp / (tp + fp)	0.5714285714
Recall (R)	tp / (tp + fn)	0.6666666667
F1-measure (Harmonic Mean)	(2 * r * p)/(r + p)	0.6153846154

c. Calculate the Average Precision for each ranking algorithms and the Mean Average Precision (MAP) for both ranking methods.

	Average Precision	Mean Average Precision
Ranking 1	0.7716666667	0.6466666667
Ranking 2	0.5216666667	

d. Draw Precision-Recall Curves (using interpolation) for the Ranking #1 and #2 in a same graph and explain which ranking algorithm is better in terms of the Precision-Recall curves.

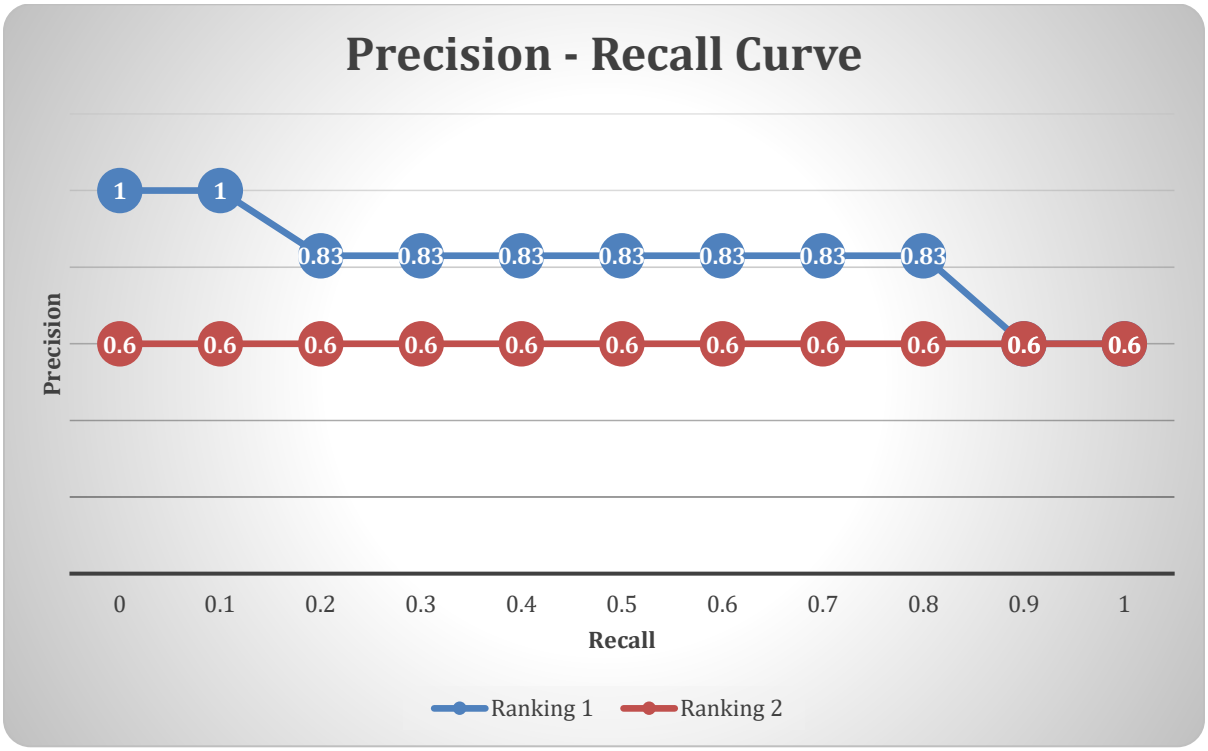
Calculating Interpolated precision

Ranking 1	recall	0.17	0.17	0.33	0.5	0.67	0.83	0.83	0.83	0.83	1	
	Precision	1	0.5	0.67	0.75	0.8	0.83	0.71	0.63	0.56	0.6	
	Recall range	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
	Interpolated Precision #1	1	1	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.6	0.6

Ranking 2	recall	0	0.17	0.17	0.17	0.33	0.5	0.67	0.67	0.83	1	
	Precision	0	0.5	0.33	0.25	0.4	0.5	0.57	0.5	0.56	0.6	
	Recall range	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1

	Interpolated Precision #2	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6

Recall Range	Ranking 1	Ranking 2
0	1	0.6
0.1	1	0.6
0.2	0.83	0.6
0.3	0.83	0.6
0.4	0.83	0.6
0.5	0.83	0.6
0.6	0.83	0.6
0.7	0.83	0.6
0.8	0.83	0.6
0.9	0.6	0.6
1	0.6	0.6



Based on the precision recall curve above we can say that ranking #1 clearly outperforms Ranking #2.

Below attached is the excel sheet used for calculations, please go through it for extra calculations.

TF-IDF.xlsx