Sheet 5

Question 1

Given Reuters collection of 806791 documents the following table includes the document frequency for four terms (car, auto, insurance , best)

Term	car	auto	insurance	best
Df	18,165	6723	19,241	25,235
idf	1.65	2.08	1.62	1.5

Compute **idf** value for each term

Question 2

Consider the following table of term frequencies for 3 documents denoted Doc1, Doc2, and Doc3:

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

a. Compute the tf-idf weights for the terms car, auto, insurance, best, for each document, using the idf values from question 1.

→Doc1

Term	**TF**	**TF WEIGHT**	**TF.IDF**	**Normalized**
car	27	2.43	4.002	0.67
auto	3	1.47	3.04	0.509
Insurance	0	0	0	0
best	14	2.14	3.222	0.53

$|\mathbf{doc1}| = \sqrt{\mathbf{(}(4.002)^2 + (3.04)^2 + (3.22)^2\mathbf{)}} = 5.968$

→Doc2

Term	TF	TF WEIGHT	TF.IDF	Normalized
car	4	1.602	2.63	0.369
auto	33	2.518	5.21	0.732
Insurance	33	2.518	4.08	0.573
best	0	0	0	0

|doc2| = $\sqrt{}$ **(**$(2.63)^2+ (5.21)^2+ (4.08)^2$**)** = 7.11

→Doc3

Term	TF	TF WEIGHT	TF.IDF	Normalized
Car	24	2.38	3.91	0.601
Auto	0	0	0	0
Insurance	29	2.46	3.99	0.613
Best	17	2.230	3.35	0.51

|doc3| = $\sqrt{}$ **(**$(3.19)^2+ (3.99)^2+ (3.35)^2$**)** = 6.5

b. Consider the query = best insurance. Compute the score for each document for this query, then rank the documents according to relevancy to the query for each of the following cases of term weighting in the query:

1. The weight of a term is 1 if present in the query, 0 otherwise.
2. Normalized idf.

.

	Term Weights		
Term	**Doc 1**	**Doc 2**	**Doc 3**
Car	0.67	0.369	0.601
Auto	0.509	0.732	0
Insurance	0	0.573	0.613
Best	0.53	0	0.51

1- The weight of a term is 1 if present in the query, 0 otherwise.

Query			Product		
Term	**tf**	**w(t,q)**	**Doc 1**	**Doc 2**	**Doc 3**
Car	0	0	0	0	0
Auto	0	0	0	0	0
Insurance	1	1	0	0.573	0.613
best	**1**	1	0.53	0	0.51

Score(q,doc1)= 0.53,
Score(q,doc2)= 0.573,
Score(q,doc3)= 1.123

Ranking = d3, d2, d1

2- Normalized idf.

	Query				Product		
Term	**tf**	**tf weight**	**w(t,q)/tf.idf**	**normalized**	**Doc 1**	**Doc 2**	**Doc 3**
Car	0	0	0	0	0	0	0
Auto	0	0	0	0	0	0	0
Insurance	1	1	1.62	0.733	0	0.42	0.45
best	1	1	1.5	0.68	0.36	0	0.35

Length of query $=\sqrt{((1.62)^2+ (1.5)^2)} = 2.21$

Score(q,doc1)=0.36,
Score(q,doc2)= 0.42,
Score(q,doc3)=0.8

Ranking = d3, d2, d1

Question 3

Consider a very small collection C that consists in the following three documents: d1: “new york times” d2: “new york post” d3: “los angeles times”, the total number of documents is N=3, calculate the similarity values for the mentioned documents and the query "new times"

Question 4

Compute the vector space similarity between the query "digital cameras" and the document "digital cameras and video cameras" by filling out the empty columns in **Table 1.** Assume N = 10,000,000, logarithmic term weighting (wf columns) for query and document, idf weighting for the query only and cosine normalization for the document only. Treat **and** as a stop word. Enter term counts in the tf columns. What is the final similarity score?

	query					document			
word	tf	wf	df	idf	q_i = wf-idf	tf	wf	d_i = normalized wf	$q_i \cdot d_i$
digital			10,000						
video			100,000						
cameras			50,000						

Table 1