Information Retrieval

Exercises week 4

solutions

Exercise 1

What is the idf of a term that occurs in every document? Compare this with the use of stop word lists.

Solution Exercise 1

It is 0. For a word that occurs in every document, putting it on the stop list has the same effect as idf weighting: the word is ignored.

Exercise 2

How does the base of the logarithm in the idf formula below affect the score calculation? How does the base of the logarithm affect the relative scores of two documents on a given query?

$$\text{idf}_t = \log \frac{N}{\text{df}_t} \qquad \text{Score}(q,d) = \sum_{t \in q} \text{tf-idf}_{t,d}$$

Solution Exercise 2

SOLUTION.

6.5 For any base b>0, $\text{idf}_t = \log_b(N/df_t) = (\log_b 10)*(\log_{10}(N/df_t)) = c*(\log(N/df_t))$

where c is a constant.

$\text{tf-idf}_{t,d,b} = \text{tf}_{t,d} * idf_t = tf_{t,d} * c*(\log(N/df_t)) = c*\text{tf-idf}_{t,d}$

$\text{Score}(q,d,b) = \sum_{t \in q} \text{tf-idf}_{t,d,b} = c* \sum_{t \in q} \text{tf-idf}_{t,d}$

So changing the base changes the score by a factor $c=(\log_b 10)$

The relative scoring of documents remains unaffected by changing the base.

Exercise 3

If we were to stem jealous and jealousy to a common stem before setting up the vector space, detail how the definitions of tf and idf should be modified.

Solution Exercise 3

If jealousy and jealous are stemmed to a common term t, then their tf's and their df's would be added together, in computing the tf-idf weights.

Exercise 4

Calculate the tf-idf weights for the terms car, auto, insurance, best, for each document, using the idf and tf values from the figures below. Using these term weights, rank the three documents by computed score for the query "car insurance", for each of the following cases of term weighting in the query:

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

ii. Euclidean normalized idf.

term	df_t	idf_t
car	18,165	1.65
auto	6723	2.08
insurance	19,241	1.62
best	25,235	1.5

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

Solution Exercise 4

i. Term weights:

	car	auto	insurance	best
Doc1	0.897	0.125	0	0.423
Doc2	0.076	0.786	0.613	0
Doc3	0.595	0	0.706	0.383

Term	Query		Product		
	tf	W(t,q)	Doc1	Doc2	Doc3
car	1	1	0.897	0.076	0.595
auto	0	0	0	0	0
insurance	1	1	0	0.613	0.706
best	0	0	0	0	0

Score(q,doc1) = 0.897

Score(q,doc2)= 0.689

Score(q,doc3)= 1.301, so Doc1, Doc2, Doc3

Solution Exercise 4

ii.

Term	Query	
	idf	W(t,q)
car	1.65	0.478
auto	2.08	0.602
insurance	1.62	0.47
best	1.5	0.43

Score(q,doc1) = 0.686

Score(q,doc2)= 0.797

Score(q,doc3)= 0.781, so Doc2, Doc3, Doc2

Exercise 5

- 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 the table below. 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						

Solution Exercise 5

	query					document			
word	tf	wf	df	idf	q_i = wfidf	tf	wf	d_i = normalized wf	q_i x d_i
digital	1	1	10,000	3	3	1	1	$\frac{1}{\sqrt{1^2 + 1^2 + 1.30^2}} = 0.52$	1.56
video	0	0	100,000	2	0	1	1	0.52	0
cameras	1	1	50,000	2.30	2.30	2	1.30	1.3 x 0.52 = 0.68	1.56

Similarity score = 1.56+1.56 = 3.12. Normalized similarity score is also correct: 3.12/len(query) = 3.12/3.78 = 0.825