CAB431 Tutorial (Week 5): IR models

****

Task 1. TF-IDF is the product of two statistics, term frequency and inverse document frequency, to measure **the weight of a term's appearance in a document**. Various ways for determining the exact values of both statistics exist.

Discuss the following recommended tf*idf weighting schemes and the one we discussed in lecture notes.

weighting scheme	document term weight	query term weight
1	$f_{t,d} \cdot \log \frac{N}{n_t}$	$\left(0.5 + 0.5 \frac{f_{t,q}}{\max_t f_{t,q}}\right) \cdot \log \frac{N}{n_t}$
2	$1 + \log f_{t,d}$	$\log\left(1 + \frac{N}{n_t}\right)$
3	$(1 + \log f_{t,d}) \cdot \log \frac{N}{n_t}$	$(1 + \log f_{t,q}) \cdot \log \frac{N}{n_t}$

Task 2. Manually calculate the df value for each term in the following table.

Document	term1	term2	term3	term4	term5
D1	3	0	0	5	7
D2	5	3	4	6	0
D3	0	0	5	4	6
D4	9	0	0	1	2
D5	0	1	0	3	2
D6	3	0	2	4	4
df					

Task 3. Design a python function *c_df*(*docs*) to calculate *df* value for each term in *docs* to verify if you can get the same result as you did in Task 2. The function returns a {term:df, …} dictionary. In your program, you can represent the above table as follows when you use it to test your python function.

```
docs = {'D1':{'term1':3, 'term4':5, 'term5':7},'D2':{'term1':5,
'term2':3, 'term3':4, 'term4':6}, 'D3':{'term3':5, 'term4':4,
'term5':6}, 'D4':{'term1':9, 'term4':1, 'term5':2}, 'D5':{'term2':1,
'term4':3, 'term5':2},'D6':{'term1':3, 'term3':2, 'term4':4,
'term5':4}}
```

Task 4. Let Q = {US, ECONOM, ESPIONAG} be a query, and
$\boldsymbol{C}$ = {D_1, D_2, D_3, D_4, D_5, D_6, D_7} be a collection of documents, where

D_1 = {GERMAN, VW}
D_2 = {US, US, ECONOM, SPY}
D_3 = {US, BILL, ECONOM, ESPIONAG}
D_4 = {US, ECONOM, ESPIONAG, BILL}
D_5 = {GERMAN, MAN, VW, ESPIONAG}
D_6 = {GERMAN, GERMAN, MAN, VW, SPY}
D_7 = {US, MAN, VW}

Assume relevant and non-relevant documents (user feedback) are labeled as follows:

Document ID	**Terms: d_{ij}**	**Relevance to Q**
D_1	GERMAN, VW	0 no
D_2	US, US, ECONOM, SPY	1 yes
D_3	US, BILL, ECONOM, ESPIONAG	1 yes
D_4	US, ECONOM, ESPIONAG, BILL	1 yes
D_5	GERMAN, MAN, VW, ESPIONAG	0 no
D_6	GERMAN, GERMAN, MAN, VW, SPY	0 no
D_7	US, MAN, VW	0 no

For a given incoming document D = {US, VW, ESPIONAG}, let *term* 1 = 'US', *term* 2 = 'VW' and *term* 3 = 'ESPIONAG'. Based on binary independence model, work out the missing values for the following contingency tables, where d_i = 1 if term i is present in the document, and 0 otherwise.

	Relevant	**Non-relevant**	**Total**
$d_1 = 1$	r_i = 3	$n_i - r_i$ = 1	n_i = 4
$d_1 = 0$	$R - r_i$ = 0	$(N-R)-(n_i - r_i) = N - n_i - R + r_i$ = 3	$N - n_i$ = 3
Total	R = 3	$N-R$ = 4	N = 7

	Relevant	**Non-relevant**	**Total**
$d_2 = 1$	r_i =	$n_i - r_i$ =	n_i =
$d_2 = 0$	$R - r_i$ =	$N - n_i - R + r_i$ =	$N - n_i$ =
Total	R =	$N-R$ =	N =

	Relevant	**Non-relevant**	**Total**
$d_3 = 1$	r_i =	$n_i - r_i$ =	n_i =
$d_3 = 0$	$R - r_i$ =	$N - n_i - R + r_i$ =	$N - n_i$ =
Total	R =	$N-R$ =	N =