CS567

Information Retrieval & Text Mining

Week 05

Muhammad Rafi

March 08, 2021

Problems

Chapter No. 6

Different Weighting Schemes

Term frequency		Document frequency		Normalization	
n (natural)	$tf_{t,d}$	n (no)	1	n (none)	1
l (logarithm)	$1+\log(tf_{t,d})$	t (idf)	$\log \frac{N}{df_t}$	c (cosine)	$\frac{1}{\sqrt{w_1^2+w_2^2+\ldots+w_M^2}}$
a (augmented)	$0.5+\frac{0.5\times tf_{t,d}}{\max_t(tf_{t,d})}$	p (prob idf)	$\max\{0, \log \frac{N-df_t}{df_t}\}$	u (pivoted unique)	$1/u$ (Section 6.4.4)
b (boolean)	$\begin{cases}1 & \text{if } tf_{t,d}>0\\ 0 & \text{otherwise}\end{cases}$			b (byte size)	$1/CharLength^{\alpha}$, $\alpha<1$
L (log ave)	$\frac{1+\log(tf_{t,d})}{1+\log(ave_{t\in d}(tf_{t,d}))}$				

▶ **Figure 6.15** SMART notation for tf-idf variants. Here *CharLength* is the number of characters in the document.

Exercise# 1

Exercise 6.10

Consider the table of term frequencies for 3 documents denoted Doc1, Doc2, Doc3 in Figure 6.9. Compute the tf-idf weights for the terms car, auto, insurance, best, for each document, using the idf values from Figure 6.8.

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

Reuters collection of 806,791 documents.

Exercise# 1

Exercise 6.10

Consider the table of term frequencies for 3 documents denoted Doc1, Doc2, Doc3 in Figure 6.9. Compute the tf-idf weights for the terms car, auto, insurance, best, for each document, using the idf values from Figure 6.8.

terms	Doc1	Doc2	Doc3
Car	44.55	6.6	39.6
Auto	6.24	68.64	0
Insurance	0	53.46	46.98
Best	21	0	25.5

Exercise# 2

Example 6.4: We now consider the query best car insurance on a fictitious collection with $N = 1{,}000{,}000$ documents where the document frequencies of auto, best, car and insurance are respectively 5000, 50000, 10000 and 1000.

term	query				document			product
	tf	df	idf	$w_{t,q}$	tf	wf	$w_{t,d}$	
auto	0	5000	2.3	0	1	1	0.41	0
best	1	50000	1.3	1.3	0	0	0	0
car	1	10000	2.0	2.0	1	1	0.41	0.82
insurance	1	1000	3.0	3.0	2	2	0.82	2.46

In this example the weight of a term in the query is simply the idf (and zero for a term not in the query, such as auto); this is reflected in the column header $w_{t,q}$ (the entry for auto is zero because the query does not contain the termauto). For documents, we use tf weighting with no use of idf but with Euclidean normalization. The former is shown under the column headed wf, while the latter is shown under the column headed $w_{t,d}$. Invoking (6.9) now gives a net score of $0 + 0 + 0.82 + 2.46 = 3.28$.

Exercise # 3

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 6.1. 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 6.1** Cosine computation for Exercise 6.19.

Exercise # 3

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 6.1. 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?

Word	Query					document			qi*di
	tf	wf	df	idf	qi=wf-idf	tf	wf	di=normalized wf	
digital	1	1	10,000	3	3	1	1	0.52	1.56
video	0	0	100,000	2	0	1	1	0.52	0
Cameras	1	1	50,000	2.3	2.3	2	1.3	0.68	1.56

Similarity score = 1.56+1.56 = 3.12

Exercise # 4

Consider the following document collection consist of the four documents:

D={d1,d2,d3,d4}
d1= mary had a little lamb
d2= little lamb mary had
d3= mary went with lamb
d4= lamb went with mary

The collection vocabulary is given by V ={ a, had, lamb, little, mary, went, with}
Assume that we use TF= 1+ log($tf_{t,d}$+1) for computing the term frequency of term t in document d. and IDF is define as log (N/df_t). we can use TF*IDF to represent every term in vector representation of document. **Give the document vectors for all four documents. Given a query vector q=<0,0,2,2,2,0,0> which document is most relevant and why?**

Exercise # 4

	d1	d2	d3	d4	q
a	1	0	0	0	0
had	1	1	0	0	0
lamb	1	1	1	1	2
little	1	1	0	0	2
mary	1	1	1	1	2
went	0	0	1	1	0
with	0	0	1	1	0

Exercise # 4

	TF					IDF	TF*IDF				
	d1	d2	d3	d4	q		d1	d2	d3	d4	q
a	0.301	0	0	0	0	0.602	0.181				
had	0.301	0.301	0	0	0	0.301	0.090	0.090			
lamb	0.301	0.301	0.301	0.301	0.47	0					
little	0.301	0.301	0	0	0.47	0.301	0.090	0.090			0.141
mary	0.301	0.301	0.301	0.301	0.47	0					
went	0	0	0.301	0.301	0	0.301			0.090	0.090	
with	0	0	0.301	0.301	0	0.301			0.090	0.090	

Sim(d1,q)=0.406
Sim(d2,q)=0.707
Sim(d3,q)=0
Sim(d4,q)=0

d2 is highly relevant.

Example # 5

D1: dil dil Pakistan jan jan Pakistan
D2: Pakistan hum sub ki jan
D3: dil aur jan Pakistan Pakistan
Q: dil jan Pakistan

V={ aur, dil, hum, jan, ki, Pakistan, sub}

Term vector in V- dimension space

D1: < 0, 1, 0, 1, 0, 1, 0>
D2: < 0, 0, 1, 1, 1, 1, 1>
D3: < 1, 1, 0, 1, 0, 1, 0>
Q: < 0, 1, 0, 1, 0, 1, 0>

Example # 5

Sim (D1, Q) = (D1 . Q) / |D1| * |Q|

= (< 0, 1, 0, 1, 0, 1, 0> . < 0, 1, 0, 1, 0, 1, 0>) / |D1| * |Q|

= 3/3 = 1

Sim (D2, Q) = (D2 . Q) / |D2| * |Q|

= (< 0, 0, 1, 1, 1, 1, 1>. < 0, 1, 0, 1, 0, 1, 0>) / |D2| * |Q|

= 2/V5 * V3 = 0.516

Sim (D3, Q) = (D3 . Q) / |D3| * |Q|

= (< 1, 1, 0, 1, 0, 1, 0> . < 0, 1, 0, 1, 0, 1, 0>) / |D3| * |Q|

= 3/V4 * V3 = 0.866

Example # 6

D1: dil dil Pakistan jan jan Pakistan
D2: Pakistan hum sub ki jan
D3: dil aur jan Pakistan Pakistan
Q: dil jan Pakistan

V={ aur, dil, hum, jan, ki, Pakistan, sub}

Term Frequency vector in V- dimension space

D1: < 0, 2, 0, 2, 0, 2, 0>
D2: < 0, 0, 1, 1, 1, 1, 1>
D3: < 1, 1, 0, 1, 0, 2, 0>
Q: < 0, 1, 0, 1, 0, 1, 0>

Example # 6

Sim (D1, Q) = (D1 . Q) / |D1| * |Q|

= (< 0, 2, 0, 2, 0, 2, 0> . < 0, 1, 0, 1, 0, 1, 0>) / |D1| * |Q|

= 6/ (V12 * V3) = 1

Sim (D2, Q) = (D2 . Q) / |D2| * |Q|

= (< 0, 0, 1, 1, 1, 1, 1>. < 0, 1, 0, 1, 0, 1, 0>) / |D2| * |Q|

= 2/V5 * V3 = 0.516

Sim (D3, Q) = (D3 . Q) / |D3| * |Q|

= (< 1, 1, 0, 1, 0, 2, 0> . < 0, 1, 0, 1, 0, 1, 0>) / |D3| * |Q|

= 4/V7 * V3 = 0.872