Vector Space Model with TF-IDF Term Weighting

October 4, 2016

Task

In this exercise we'll have a look at how the TF-IDF ranking works.
There are 5 different documents in the collection:

D1 "If it walks like a duck and quacks like a duck, it must be a duck."

D2 "Beijing Duck is mostly prized for the thin, crispy duck skin with authentic versions of the dish serving mostly the skin."

D3 "Bugs' ascension to stardom also prompted the Warner animators to recast Daffy Duck as the rabbit's rival, intensely jealous and determined to steal back the spotlight while Bugs remained indifferent to the duck's jealousy, or used it to his advantage. This turned out to be the recipe for the success of the duo."

D4 "6:25 PM 1/7/2007 blog entry: I found this great recipe for Rabbit Braised in Wine on cookingforengineers.com."

D5 "Last week Li has shown you how to make the Sechuan duck. Today we'll be making Chinese dumplings (Jiaozi), a popular dish that I had a chance to try last summer in Beijing. There are many recipies for Jiaozi."

For the query Q = "Beijing duck recipe", find the two top ranked documents according to the TF-IDF score. Assume the cosine similarity measure and the vocabulary V = {beijing, dish, duck, rabbit, recipe, roast}.

Solution

First we record the count of the occurrences of terms in all the documents (f_{ij}). Counting is case insensitive (Duck = duck) and we need to perform stemming (recipes = recipe).

term	D1	D2	D3	D4	D5
beijing		1			1
dish		1			1
duck	3	2	2		1
rabbit			1	1	
recipe			1	1	1
roast					

Table 1: Term-document matrix.

Use normalized frequencies for TF weight, i.e.:

$$tf_{t,d} = \frac{f_{t,d}}{|d|}, \tag{1}$$

$f_{t,d}$ is the number of occurrences of term t in document d and $|d|$ is the document length (=total number of terms); see the values in Table 1.

We also compute the IDF values for each term using this formula:

$$idf_t = \log \frac{N}{n_t}, \tag{2}$$

where N is the total number of document and n_t is the number of documents that contain term t. The base of the logarithm does not matter as long as the same one is used for all the IDF calculations.[1] There is a special case for the term "roast" which does not appear in any document, the IDF value is zero.

[1] Here, $\log_{10}$ is used.

	TF					
term	D1	D2	D3	D4	D5	IDF
beijing		0.25			0.25	0.398
dish		0.25			0.25	0.398
duck	1	0.5	0.5		0.25	0.097
rabbit			0.25	0.5		0.398
recipe			0.25	0.5	0.25	0.222
roast						0

Table 2: TF and IDF values. (Empty cells mean zero values.)

We then compute the TFIDF weights by multiplying each TF cell in Table 2 by the corresponding IDF value.

	TFIDF				
term	D1	D2	D3	D4	D5
beijing		0.099			0.099
dish		0.099			0.099
duck	0.097	0.048	0.048		0.024
rabbit			0.099	0.199	
recipe			0.055	0.111	0.055
roast					

Table 3: Document TFIDF values. (Empty cells mean zero values.)

Next, we compute the weighted query term vector the same way we did for documents.

term	TF	IDF	TFIDF
beijing	0.333	0.398	0.133
dish		0.398	
duck	0.333	0.097	0.032
rabbit		0.398	
recipe	0.333	0.222	0.074
roast		0	

Table 4: Query TFIDF values. (Empty cells mean zero values.)

Finally we use the cosine measure to compute the similarity. We take the cosine between each document vector (in Table 3) and the query TFIDF vector (in Table 4). For example the similarity for D2 is the cosine of the angle between the query vector: $(0.133, 0, 0.032, 0, 0.074, 0)$ and the TFIDF vector for D2: $(0.099, 0.099, 0.048, 0, 0, 0)$ is 0.639.

	D1	D2	D3	D4	D5
$score(Q, D)$	0.208	0.639	0.295	0.232	0.760

Table 5: Document scores.

Based on Table 5 the two most relevant documents are D5 and D2.