

# 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 recipes for Jiaozi.”

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

## 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.:

$$tf_{t,d} = \frac{f_{t,d}}{|d|}, \quad (1)$$

$f_{t,d}$  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:

$$idf_t = \log \frac{N}{n_t}, \quad (2)$$

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. There is a special case for the term “roast” which does not appear in any document, the IDF value is zero.

term	TF					IDF
	D1	D2	D3	D4	D5	
beijing						
dish						
duck						
rabbit						
recipe						
roast						0

Table 2: TF and IDF values.

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

term	TFIDF				
	D1	D2	D3	D4	D5
beijing					
dish					
duck					
rabbit					
recipe					
roast					

Table 3: Document TFIDF values.

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

term	TF	IDF	TFIDF
beijing			
dish			
duck			
rabbit			
recipe			
roast			

Table 4: Query TFIDF 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).

	D1	D2	D3	D4	D5
$score(Q, D)$					

Table 5: Document scores.