Here is a simplified example of the vector space retrieval model. Consider a very small collection C that consists in the following three documents:

- d1: “new york times”
- d2: “new york post”
- d3: “los angeles times”

Some terms appear in two documents, some appear only in one document. The total number of documents is $N=3$. Therefore, the $idf$ values for the terms are:

- angles: $\log_2(3/1)=1.584$
- los: $\log_2(3/1)=1.584$
- new: $\log_2(3/2)=0.584$
- post: $\log_2(3/1)=1.584$
- times: $\log_2(3/2)=0.584$
- york: $\log_2(3/2)=0.584$

For all the documents, we calculate the $tf$ scores for all the terms in C. We assume the words in the vectors are ordered alphabetically.

<table>
<thead>
<tr>
<th></th>
<th>angeles</th>
<th>los</th>
<th>new</th>
<th>post</th>
<th>times</th>
<th>york</th>
</tr>
</thead>
<tbody>
<tr>
<td>d1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>d2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>d3</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Now we multiply the $tf$ scores by the $idf$ values of each term, obtaining the following matrix of documents-by-terms: (All the terms appeared only once in each document in our small collection, so the maximum value for normalization is 1.)

<table>
<thead>
<tr>
<th></th>
<th>angeles</th>
<th>los</th>
<th>new</th>
<th>post</th>
<th>times</th>
<th>york</th>
</tr>
</thead>
<tbody>
<tr>
<td>d1</td>
<td>0</td>
<td>0</td>
<td>0.584</td>
<td>0</td>
<td>0.584</td>
<td>0.584</td>
</tr>
<tr>
<td>d2</td>
<td>0</td>
<td>0</td>
<td>0.584</td>
<td>1.584</td>
<td>0</td>
<td>0.584</td>
</tr>
<tr>
<td>d3</td>
<td>1.584</td>
<td>1.584</td>
<td>0</td>
<td>0</td>
<td>0.584</td>
<td>0</td>
</tr>
</tbody>
</table>

Given the following query: “new new times”, we calculate the $tf-idf$ vector for the query, and compute the score of each document in C relative to this query, using the cosine similarity measure. When computing the $tf-idf$ values for the query terms we divide the frequency by the maximum frequency (2) and multiply with the $idf$ values.

| q   | 0     | 0     | (2/2)*0.584=0.584 | 0     | (1/2)*0.584=0.292 | 0    |
We calculate the length of each document and of the query:

Length of \( d_1 \) = \( \sqrt{0.584^2 + 0.584^2 + 0.584^2} = 1.011 \)
Length of \( d_2 \) = \( \sqrt{0.584^2 + 1.584^2 + 0.584^2} = 1.786 \)
Length of \( d_3 \) = \( \sqrt{1.584^2 + 1.584^2 + 0.584^2} = 2.316 \)
Length of \( q \) = \( \sqrt{0.584^2 + 0.292^2} = 0.652 \)

Then the similarity values are:

\[
\text{cosSim}(d_1, q) = \frac{(0*0+0*0+0.584*0.584+0*0+0.584*0.292+0.584*0)}{(1.011*0.652)} = 0.776
\]
\[
\text{cosSim}(d_2, q) = \frac{(0*0+0*0+0.584*0.584+1.584*0+0*0.292+0.584*0)}{(1.786*0.652)} = 0.292
\]
\[
\text{cosSim}(d_3, q) = \frac{(1.584*0+1.584*0+0*0.584+0*0+0.584*0.292+0*0)}{(2.316*0.652)} = 0.112
\]

According to the similarity values, the final order in which the documents are presented as result to the query will be: \( d_1, d_2, d_3 \).