Unstructured Information Processing 4

Divij Singh

30/10/19

1 Q1

Since there is no way to know the total number of words in each document, the following calculations assume that the documents may only contain the words car, auto, insurance, and best.

Thus Document 1 has 44 words, Document 2 has 70 and Document 3 has 70.

The values in the table are attained by the formula:
$\frac{(frequency\ of\ word\ in\ document)}{(number\ of\ words\ in\ document)} \times (corresponding\ idf\ value)$

As such, the following tf-idf table is constructed:

Word	Document 1	Document 2	Document 3
Car	1.0125	0.0943	0.5657
Auto	0.1418	0.9806	0
Insurance	0	0.7637	0.6711
Best	0.4773	0	0.3643

2 Q2

First, let's look at the two methods we are comparing and how they operate.

The Cosine Similarity is a metric which measures the cosine of the angle made by two vectors, which in this context are arrays containing the word counts of chosen words of two documents.

The euclidean distance, on the other hand, looks at the actual distance between the two vectors.

If the two vectors are taken as x and y, the formulae would be as follows:

Cosine similarity: $\frac{\sum_{i=1}^{n} x_i.y_i}{\sqrt{\sum_{i=1}^{n} x_i.x_i}\sqrt{\sum_{i=1}^{n} y_i.y_i}}$ (dot products)

Euclidean distance: $\sqrt{\sum_{i=1}^{n}(x_i - y_i) \times (x_i - y_i)}$ (cross product)

If we normalise the vectors using the L_2 method, we get : $\sqrt{\sum_i x_i^2} = \sqrt{\sum_i y_i^2} = 1$
$\sqrt{(x-y)^2} = (x-y)^T(x-y) = 2 - 2z$ where z is the cosine of x and y.

Now, when we look at ranking documents, we rank them against the query q.
So for cosine similarity, we look at cosine of qd_1, cosine of qd_2 and so on.
Thus for L_2 normalised euclidean distance, we look at $2 - 2$cosine of qd_1, $2 - 2$cosine of qd_2 and so on.

As a result, while the values will differ, the ranking will remain the same.