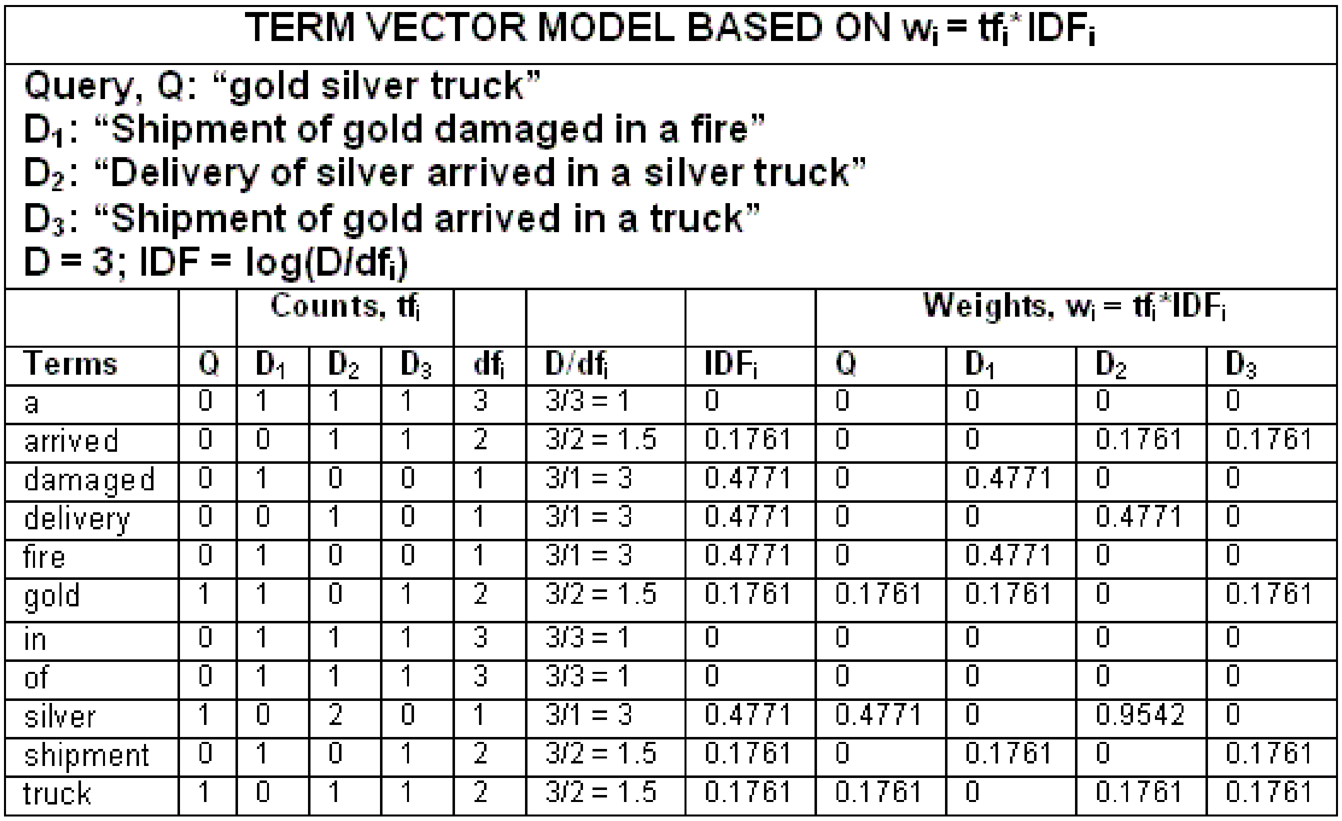
Vector Space Example

Suppose we query an IR system for the query "gold silver truck". The database collection consists of three documents (D = 3) with the following content

D1: "Shipment of gold damaged in a fire"
D2: "Delivery of silver arrived in a silver truck"
D3: "Shipment of gold arrived in a truck"
Retrieval results are summarized in the following table.



TERM VECTOR MODEL BASED ON $w_i = tf_i * IDF_i$

Query, Q: "gold silver truck"
D_1: "Shipment of gold damaged in a fire"
D_2: "Delivery of silver arrived in a silver truck"
D_3: "Shipment of gold arrived in a truck"
D = 3; IDF = log(D/df_i)

		Counts, tf_i						Weights, $w_i = tf_i * IDF_i$		
Terms	**Q**	**D_1**	**D_2**	**D_3**	**df_i**	**D/df_i**	**IDF_i**	**Q**	**D_1**	**D_2**	**D_3**
a	0	1	1	1	3	3/3 = 1	0	0	0	0	0
arrived	0	0	1	1	2	3/2 = 1.5	0.1761	0	0	0.1761	0.1761
damaged	0	1	0	0	1	3/1 = 3	0.4771	0	0.4771	0	0
delivery	0	0	1	0	1	3/1 = 3	0.4771	0	0	0.4771	0
fire	0	1	0	0	1	3/1 = 3	0.4771	0	0.4771	0	0
gold	1	1	0	1	2	3/2 = 1.5	0.1761	0.1761	0.1761	0	0.1761
in	0	1	1	1	3	3/3 = 1	0	0	0	0	0
of	0	1	1	1	3	3/3 = 1	0	0	0	0	0
silver	1	0	2	0	1	3/1 = 3	0.4771	0.4771	0	0.9542	0
shipment	0	1	0	1	2	3/2 = 1.5	0.1761	0	0.1761	0	0.1761
truck	1	0	1	1	2	3/2 = 1.5	0.1761	0.1761	0	0.1761	0.1761

1. Columns 1 - 5: First, we construct an index of terms from the documents and determine the term counts tfi for the query and each document Dj.
2. Columns 6 - 8: Second, we compute the document frequency di for each document. Since IDFi = log(D/dfi) and D = 3, this calculation is straightforward.
3. Columns 9 - 12: Third, we take the tf*IDF products and compute the term weights. These columns can be viewed as a sparse matrix in which most entries are zero.

Now we treat weights as coordinates in the vector space, effectively representing documents and the query as vectors.

Similarity Analysis

First for each document and query, we compute all vector lengths (zero terms ignored)

$$|D_1| = \sqrt{0.4771^2 + 0.4771^2 + 0.1761^2 + 0.1761^2} = \sqrt{0.5173} = 0.7192$$

$$|D_2| = \sqrt{0.1761^2 + 0.4771^2 + 0.9542^2 + 0.1761^2} = \sqrt{1.2001} = 1.0955$$

$$|D_3| = \sqrt{0.1761^2 + 0.1761^2 + 0.1761^2 + 0.1761^2} = \sqrt{0.1240} = 0.3522$$

$$\therefore \ |D_i| = \sqrt{\sum_i w_{i,j}^2}$$

$$|Q| = \sqrt{0.1761^2 + 0.4771^2 + 0.1761^2} = \sqrt{0.2896} = 0.5382$$

$$\therefore \ |Q| = \sqrt{\sum_i w_{Q,j}^2}$$

Next, we compute all dot products (zero products ignored)

$$Q \bullet D_1 = 0.1761 * 0.1761 = 0.0310$$

$$Q \bullet D_2 = 0.4771 * 0.9542 + 0.1761 * 0.1761 = 0.4862$$

$$Q \bullet D_3 = 0.1761 * 0.1761 + 0.1761 * 0.1761 = 0.0620$$

$$\therefore \ Q \bullet D_i = \sum_i w_{Q,j} w_{i,j}$$

Now we calculate the similarity values

$$\text{Cosine } \theta_{D_1} = \frac{Q \bullet D_1}{|Q| * |D_1|} = \frac{0.0310}{0.5382 * 0.7192} = 0.0801$$

$$\text{Cosine } \theta_{D_2} = \frac{Q \bullet D_2}{|Q| * |D_2|} = \frac{0.4862}{0.5382 * 1.0955} = 0.8246$$

$$\text{Cosine } \theta_{D_3} = \frac{Q \bullet D_3}{|Q| * |D_3|} = \frac{0.0620}{0.5382 * 0.3522} = 0.3271$$

$$\therefore\ \text{Cosine } \theta_{D_i} = \text{Sim}(Q, D_i)$$

$$\therefore\ \text{Sim}(Q, D_i) = \frac{\sum_i w_{Q,j} w_{i,j}}{\sqrt{\sum_j w_{Q,j}^2} \sqrt{\sum_i w_{i,j}^2}}$$

Finally we sort and rank the documents in descending order according to the similarity values

Rank 1: Doc 2 = 0.8246
Rank 2: Doc 3 = 0.3271
Rank 3: Doc 1 = 0.0801

Observations

This example illustrates several facts. First, that very frequent terms such as "a", "in", and "of" tend to receive a low weight -a value of zero in this case. Thus, the model correctly predicts that very common terms, occurring in many documents in a collection are not good discriminators of relevancy. Note that this reasoning is based on global information; ie., the IDF term. Precisely, this is why this model is better than the term count model discussed in Part 2. Third, that instead of calculating individual vector lengths and dot products we can save computational time by applying directly the similarity function

Eq 3:

$$\text{Sim}(Q, D_i) = \frac{\sum_i w_{Q,j} w_{i,j}}{\sqrt{\sum_j w_{Q,j}^2} \sqrt{\sum_i w_{i,j}^2}}$$

Of course, we still need to know individual tf and IDF values.