# Latent Semantic Indexing (LSI)
# An Example

(taken from Grossman and Frieder's *Information Retrieval, Algorithms and Heuristics* )

A "collection" consists of the following "documents":

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.*

Suppose that we use the term frequency as term weights and query weights. The following document indexing rules are also used:
- stop words were not ignored
- text was tokenized and lowercased
- no stemming was used
- terms were sorted alphabetically

We wish to use this example to illustrate how LSI works.

**Problem:** Use Latent Semantic Indexing (LSI) to rank these documents for the query *gold silver truck*.

**Step 1:** Set term weights and construct the term-document matrix **A** and query matrix:

| Terms | | d1 | d2 | d3 | | q |
|---|---|---|---|---|---|---|
| a | | 1 | 1 | 1 | | 0 |
| arrived | | 0 | 1 | 1 | | 0 |
| damaged | | 1 | 0 | 0 | | 0 |
| delivery | | 0 | 1 | 0 | | 0 |
| fire | | 1 | 0 | 0 | | 0 |
| gold | **A =** | 1 | 0 | 1 | **q =** | 1 |
| in | | 1 | 1 | 1 | | 0 |
| of | | 1 | 1 | 1 | | 0 |
| shipment | | 1 | 0 | 1 | | 0 |
| silver | | 0 | 2 | 0 | | 1 |
| truck | | 0 | 1 | 1 | | 1 |

**Step 2:** Decompose matrix **A** matrix and find the **U**, **S** and **V** matrices, where

**$A = USV^T$**

$$U = \begin{bmatrix} -0.4201 & 0.0748 & -0.0460 \\ -0.2995 & -0.2001 & 0.4078 \\ -0.1206 & 0.2749 & -0.4538 \\ -0.1576 & -0.3046 & -0.2006 \\ -0.1206 & 0.2749 & -0.4538 \\ -0.2626 & 0.3794 & 0.1547 \\ -0.4201 & 0.0748 & -0.0460 \\ -0.4201 & 0.0748 & -0.0460 \\ -0.2626 & 0.3794 & 0.1547 \\ -0.3151 & -0.6093 & -0.4013 \\ -0.2995 & -0.2001 & 0.4078 \end{bmatrix} \quad S = \begin{bmatrix} 4.0989 & 0.0000 & 0.0000 \\ 0.0000 & 2.3616 & 0.0000 \\ 0.0000 & 0.0000 & 1.2737 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.4945 & 0.6492 & -0.5780 \\ -0.6458 & -0.7194 & -0.2556 \\ -0.5817 & 0.2469 & 0.7750 \end{bmatrix} \quad V^T = \begin{bmatrix} -0.4945 & -0.6458 & -0.5817 \\ 0.6492 & -0.7194 & 0.2469 \\ -0.5780 & -0.2556 & 0.7750 \end{bmatrix}$$

**Step 3:** Implement a Rank 2 Approximation by keeping the first two columns of **U** and **V** and the first two columns and rows of **S**.

$$k = 2$$

$$U \approx U_k = \begin{bmatrix} -0.4201 & 0.0748 \\ -0.2995 & -0.2001 \\ -0.1206 & 0.2749 \\ -0.1576 & -0.3046 \\ -0.1206 & 0.2749 \\ -0.2626 & 0.3794 \\ -0.4201 & 0.0748 \\ -0.4201 & 0.0748 \\ -0.2626 & 0.3794 \\ -0.3151 & -0.6093 \\ -0.2995 & -0.2001 \end{bmatrix} \quad S \approx S_k = \begin{bmatrix} 4.0989 & 0.0000 \\ 0.0000 & 2.3616 \end{bmatrix}$$

$$V \approx V_k = \begin{bmatrix} -0.4945 & 0.6492 \\ -0.6458 & -0.7194 \\ -0.5817 & 0.2469 \end{bmatrix} \quad V^T \approx V_k^T = \begin{bmatrix} -0.4945 & -0.6458 & -0.5817 \\ 0.6492 & -0.7194 & 0.2469 \end{bmatrix}$$

**Step 4:** Find the new document vector coordinates in this reduced 2-dimensional space.

Rows of **V** holds eigenvector values. These are the coordinates of individual document vectors, hence

d1(-0.4945, 0.6492)
d2(-0.6458, -0.7194)
d3(-0.5817, 0.2469)

**Step 5:** Find the new query vector coordinates in the reduced 2-dimensional space.

**$q = q^T U_k S_k^{-1}$**

Note: These are the new coordinate of the query vector in two dimensions. Note how this matrix is now different from the original query matrix **q** given in **Step 1**.

$$q = q^T U_k S_k^{-1} \qquad k = 2$$

$$q = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} -0.4201 & 0.0748 \\ -0.2995 & -0.2001 \\ -0.1206 & 0.2749 \\ -0.1576 & -0.3046 \\ -0.1206 & 0.2749 \\ -0.2626 & 0.3794 \\ -0.4201 & 0.0748 \\ -0.4201 & 0.0748 \\ -0.2626 & 0.3794 \\ -0.3151 & -0.6093 \\ -0.2995 & -0.2001 \end{bmatrix} \begin{bmatrix} \frac{1}{4.0989} & 0.0000 \\ 0.0000 & \frac{1}{2.3616} \end{bmatrix}$$

$$q = \begin{bmatrix} -0.2140 & -0.1821 \end{bmatrix}$$

**Step 6:** Rank documents in decreasing order of query-document cosine similarities.

$$\mathbf{sim(q,\ d) = \frac{q \bullet d}{|\ q\ |\ |\ d\ |}}$$

$$\mathbf{sim(q,\ d_1)} = \frac{(-0.2140)\ (-0.4945)\ +\ (-0.1821)\ (0.6492)}{\sqrt{(-0.2140)^2 + (-0.1821)^2}\ \sqrt{(-0.4945)^2 + (0.6492)^2}} = -0.0541$$

$$\mathbf{sim(q,\ d_2)} = \frac{(-0.2140)\ (-0.6458)\ +\ (-0.1821)\ (-0.7194)}{\sqrt{(-0.2140)^2 + (-0.1821)^2}\ \sqrt{(-0.6458)^2 + (-0.7194)^2}} = 0.9910$$

$$\mathbf{sim(q,\ d_3)} = \frac{(-0.2140)\ (-0.5817)\ +\ (-0.1821)\ (0.2469)}{\sqrt{(-0.2140)^2 + (-0.1821)^2}\ \sqrt{(-0.5817)^2 + (0.2469)^2}} = 0.4478$$

Ranking documents in descending order

$$\mathbf{d_2 > d_3 > d_1}$$

We can see that document d2 scores higher than d3 and d1. Its vector is closer to the query vector than the other vectors.