LATENT SEMANTIC ANALYSIS

Latent Semantic Analysis is a technique in Natural Language Processing used to discover **relationships between words** and their meanings by analyzing large amounts of text (corpus). For example it scans all the wikipedia articles (corpus) and tries to keep a track of words that frequently occur together.

It has the following steps

Building the Term Document matrix

Rows represent unique words and columns represent document
Each cell contains the frequency of the word in the document.

Identify Co-occurrence Patterns

This is used to understand the semantic relationship between two words and how often the words appear together.

Dimensionality Reduction

Use Singular Value Decomposition (SVD) to reduce the high-dimensional word space into a smaller “semantic space,” where related words are positioned closer together.

Cosine Similarity:
Measures the cosine of the angle between two vectors.
Focuses on orientation rather than magnitude.
Useful in text analysis because the actual magnitude (word frequency) can vary widely, but the direction (semantic meaning) remains important.

LATENT SEMANTIC ANALYSIS

Imagine you have a huge spreadsheet where:
Rows = every word in English (100,000+ words)
Columns = every Wikipedia article (millions)
Cell value = how many times a word appears in an article
That's too big and too messy.
Many words mean almost the same thing (car, automobile, vehicle), so we don't need to store them separately in full detail.

Word	Doc1	Doc2	Doc3
Car	3	0	2
Automobile	3	0	2
Apple	0	4	0

From the above example , the document term matrix is as follows.
car = (3,0,2) , Automobile = (3,0,2) ,Apple = (0,4,0)

from this it can be observed that the words car and automobile are more similar and can be grouped into one and also their vector values are the same. Apple is different because the the word is very different from car and automobile.

SVD will find that there are basically 2 concepts here:
1. Vehicles (Car & Automobile)
2. Fruit (Apple)

Instead of storing 3 numbers per word, we now store 2 numbers per word — how much that word relates to Concept 1 and Concept 2.
Step 3: Reduced Representation
After reducing from 3D to 2D space:

- Car = (0.95, 0.00) → Mostly "Vehicles"
- Automobile = (0.95, 0.00) → Mostly "Vehicles"
- Apple = (0.00, 0.98) → Mostly "Fruit"

Now:

- "Car" and "Automobile" are close together in this smaller space.
- If a new doc has "Automobile," LSA will also see it's related to "Car" because they share the Vehicles concept.