What is Information Retrieval?

Information Retrieval (IR) is the process of finding unstructured material (usually documents) that satisfies an information need from within large collections.

User Input: A query (e.g., "best places to visit in Kerala").

The System: An IR system with an index of a document collection.

The Output: A ranked list of documents, with the most relevant ones at the top.

The Core Problem: Ranking

- We need a way to calculate a "relevance score" for every document in our collection against the user's query.

- Classic Solution: The Vector Space Model.
    - We can represent documents and queries as vectors in a high-dimensional space.
    - The Core Principle: The "closer" two vectors are in this space, the more similar their content is.

The Steps:

1. Represent documents as vectors (using a "bag-of-words").
2. Give weights to the words in the vector (using TF-IDF).
3. Calculate the similarity between the query vector and document vectors.

Term Weighting: TF-IDF

- TF-IDF is the heart of the classic vector space model. It tells us how important a word is to a specific document.
- It's a score composed of two parts:
    - TF (Term Frequency)
    - IDF (Inverse Document Frequency)

TF (Term Frequency)

- **What it is:** How many times a term appears in a document.
- **Intuition:** If a document mentions "monsoon" 10 times, it's probably more about monsoons than a document that mentions it once.
- **Normalization:** A longer document will naturally have higher counts, so we normalize it.
- **Formula :**

  TF(term, doc) = (Number of times term appears in doc) / (Total words in doc)

IDF (Inverse Document Frequency)

What it is: A measure of how rare a term is across the *entire* document collection.

- **Intuition:**
  - Common words ("the," "a," "is") appear everywhere and are not useful for distinguishing documents.
  - Rare words ("Ayurveda," "craniopharyngioma") are very informative.
- We want to give rare, informative words a higher weight.
- Formula:

  IDF(term) = log(Total number of documents / Number of documents containing the term)

The TF-IDF Score

- Final Calculation:
  TF-IDF Score = TF * IDF
- **When is the score HIGH?**
  - When a term appears *frequently* in one document (high TF).
  - ...but *rarely* in the overall collection (high IDF).
- These are the words that best characterize a document.

The Vector Space

Document Vectors:

- After calculating the TF-IDF score for every term, each document becomes a long vector of these scores.
- Doc1_vector = [TF-IDF(term1), TF-IDF(term2), ..., TF-IDF(termN)]

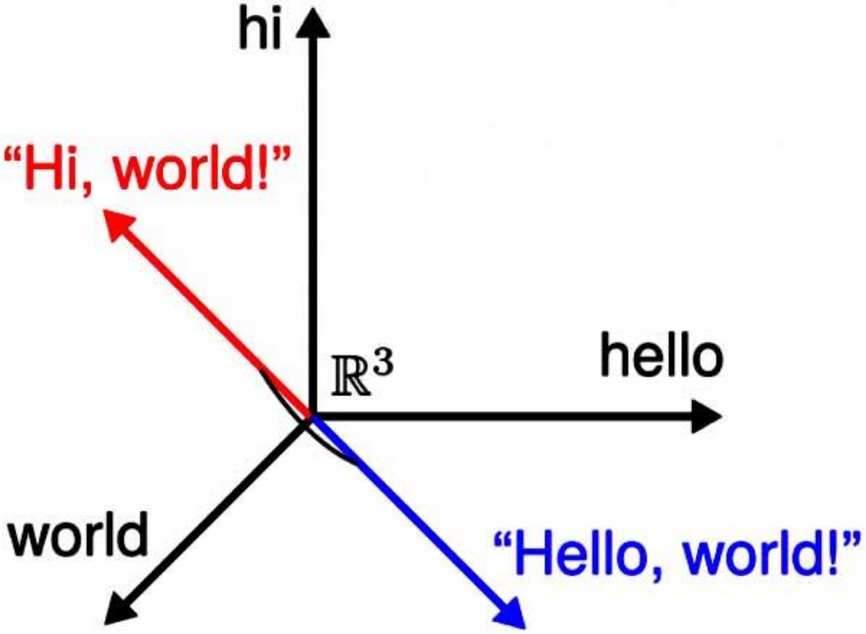
Query Vector:

- We do the exact same TF-IDF process for the user's query.

Ranking with Cosine Similarity

- **How to Compare Vectors?** We measure the *angle* between the query vector and every document vector.
- **Method:** Cosine Similarity
- **Logic:**
  - **Small Angle:** Vectors point in a similar direction. This means the document shares important (high TF-IDF) terms with the query. → **High Relevance**
  - **Large Angle (near 90°):** Vectors are dissimilar. → **Low Relevance**

- **Final Step:** Documents are ranked from highest cosine similarity (most relevant) to lowest.



Cosine Similarity