**Detailed Explanation of the Document Similarity Checker**

**1. Introduction to Document Similarity:**

* Document similarity measures how similar two pieces of text are to each other.
* One common approach to measure similarity is using cosine similarity, which is based on the cosine of the angle between two vectors in a multi-dimensional space.

**2. Vector Space Model (VSM):**

* Documents are represented as vectors in a multi-dimensional space.
* Each dimension corresponds to a unique term (word) in the document.
* The value in each dimension is typically the term frequency (TF) or term frequency-inverse document frequency (TF-IDF).

**3. TF-IDF Vectorization:**

* **Term Frequency (TF):** Measures the frequency of a term in a document.
* **Inverse Document Frequency (IDF):** Measures the importance of a term across the entire document corpus.
* **TF-IDF:** Combines TF and IDF to weigh terms by their importance.
* Mathematically, TF-IDF is defined as:

* + t is the term.
  + d is the document.
  + N is the total number of documents.
  + DF (t) is the number of documents containing term t.

**4. Cosine Similarity:**

* Cosine similarity measures the cosine of the angle between two vectors.
* It ranges from -1 to 1, where:
  + 1 means the vectors are identical.
  + 0 means the vectors are orthogonal (no similarity).
  + -1 means the vectors are diametrically opposed.
* The formula for cosine similarity between two vectors A and B is:

Cosine Similarity (A, B) =A⋅B / ∥A∥∥BA∥

Where:

* + A⋅B is the dot product of vectors A and B.
  + ∥A∥ is the magnitude (norm) of vector A.
  + ∥B∥ is the magnitude (norm) of vector B.

**5. Dot Product:**

* The dot product of two vectors A and B is calculated as:

Where:

* + Ai and Bi are the components of vectors A and B, respectively.
  + n is the number of dimensions (terms).

**6. Magnitude (Norm) Calculation:**

* The magnitude (or Euclidean norm) of a vector AAA is calculated as:

∥A∥=

Where:

* + Ai are the components of vector A.

**7. Implementation in Python:**

* **Text Extraction:**
  + Extract text from PDF using PyMuPDF.
  + Extract text from DOCX using python-docx.
* **TF-IDF Vectorization:**
  + Use TfidfVectorizer from scikit-learn to convert the documents into TF-IDF vectors.
* **Cosine Similarity Calculation:**
  + Calculate the cosine similarity using the dot product and magnitude of the TF-IDF vectors.

**Summary:**

* **Vector Representation:** Documents are converted into vectors using TF-IDF.
* **Dot Product:** Calculates the sum of the products of corresponding terms in the document vectors.
* **Cosine Similarity:** Measures the cosine of the angle between the document vectors to determine similarity.
* **Implementation:** Combined these mathematical concepts in a Python application to create a Document Similarity Checker using FastAPI and Gradio for a user-friendly interface.

This explanation highlights how linear algebra concepts like dot product and cosine similarity are utilized in building the Document Similarity Checker application.