Similarities between texts

Objective :

This task aims to explore and compare multiple techniques to measure the similarity between two texts. These techniques can be **s**yntactic (based on word form) or semantic (based on meaning).

We applied the following methods:

- **Cosine Similarity** (syntactic)
- **Jaccard Similarity** (syntactic)
- **BERT-based Similarity** using SentenceTransformer (semantic)

Input Texts (Original Example)

Text 1: *"The astronaut prepared for the mission with intense training."*
Text 2: *"Before flying to space, the cosmonaut trained rigorously."*

These two sentences are semantically similar but use very different vocabulary. This setup allows us to highlight the strengths and weaknesses of different similarity techniques.

Methodology & Code

1. **Cosine Similarity** : was computed using TF-IDF vectors.
   TfidfVectorizer() transforms words into numeric weights based on:

- how often they appear in the document (TF)
- how rare they are across all documents (IDF)

The resulting vectors are compared with cosine_similarity() from Scikit-learn.

2. **Jaccard Similarity** : was computed using set overlap of words.

- Converts both texts into lowercase sets of words.
- Uses Python set operations:

    a. & for intersection
    b. | for union

Limitation: It doesn't consider meaning, only word overlap.

3. **BERT Semantic Similarity** was computed using the all-MiniLM-L6-v2 Sentence BERT model.

- Loads a lightweight, high-performance BERT model (all-MiniLM-L6-v2) from sentence-transformers.
- Converts full sentences into 384-dimensional vectors.
- Measures cosine similarity using util.cos_sim() (optimized for tensor operations).

Advantage: Captures contextual similarity even with totally different words (e.g., "astronaut" vs. "cosmonaut").

Results Summary

Method	Score	Type
Cosine Similarity (TF-IDF)	0.1230	Syntactic
Jaccard Similarity	0.6667	Syntactic
BERT Semantic Similarity	0.6187	Semantic

Analysis

Cosine Similarity (0.1230)

- Based on TF-IDF vectors.
- Low score due to different word usage.
- It captures token overlap, but not meaning.

Jaccard Similarity (0.6667)

- Surprisingly high because of shared common words like "the", "for", "with".

- It doesn't account for word meaning or structure — only set overlap.

BERT Semantic Similarity (0.6187)

- Captures that “astronaut” ≈ “cosmonaut” and “training” ≈ “trained”.
- Despite low word overlap, the sentence meaning is understood.
- Ideal for paraphrase detection or real NLP applications.

Conclusion

- **Syntactic methods (Cosine, Jaccard)** are limited when word choice varies.
- **BERT (Semantic)** is clearly more robust, and closer to human understanding.
- BERT should be favored in any production-level NLP system requiring text understanding.