# **Semantic Similarity Analysis of Textual Data**

## **Steps for Analysis**

Based on research, the following steps have been identified for conducting semantic similarity analysis:

1. **Tokenization and Numerical Representation**
   1. Convert input words or text documents into numerical tokens using OpenAI APIs.
2. **Similarity Analysis Before Embedding Creation**
   1. Perform similarity analysis on the tokens to cluster related vectors in the vector space using the following methods:
      1. Cosine Similarity
      2. Euclidean Similarity
      3. Jaccard Similarity
      4. Pearson/Spearman Correlation
3. **Embedding Generation**
   1. Transform the tokens into embeddings within a high-dimensional vector space using GPI tools.
4. **Visualization**
   1. Use various visualization tools to analyze the relationships and patterns in the data.

## **Key Insights**

* GPT does not directly provide numerical representations like specialized models such as Sentence-BERT.
* Embeddings can be generated at the word or sentence level, depending on the context.
* OpenAI offers an efficient endpoint (/v1/embeddings) for obtaining embeddings directly, which is superior to GPT’s completion endpoint for this purpose.

## **OpenAI Text Similarity Models**

OpenAI provides the following models for text similarity analysis:

* **text-similarity-ada-001**: Cost-effective and suitable for basic tasks.
* **text-similarity-curie-001**: Offers better accuracy but at a higher cost.
* **text-similarity-davinci-002**: The most accurate model, but also the most expensive.
* **text-similarity-babbage-002**: Balances cost and performance effectively.

## **Cost Consideration**

* Among these models, curie-001 and davinci-002 are more expensive compared to ada-001 and babbage-002, making them suitable for tasks where higher accuracy is crucial.