Dataset Overview

The Google Scholar dataset contains 6,800 research papers related to Artificial Intelligence (AI) and Machine Learning (ML) collected from Google Scholar. It includes key metadata such as the paper's title, abstract, authors, publication year, URL, number of citations, journal, venue, and publication type.

Data Pre-processing

The data preprocessing steps in the provided code involve the following:

- Selecting the Best Text Column:The code checks if the dataset contains an "Abstract" column. If the "Abstract" column exists, it is chosen as the primary text column. Otherwise, the "Title" column is selected.This ensures that the most informative text field is used for further processing.

- Handling Missing and Duplicate Values:The dataset have missing and duplicated values in the selected text column (Abstract). To ensure meaningful text processing, rows with missing values in the chosen text column are dropped. This step improves the quality of the text data by removing incomplete entries.

Data Embedding

In order to performs text embedding and feature extraction, we employed Sentence-BERT (SBERT) embeddings and TF-IDF vectorization to create a combined feature representation of the dataset. Below is a detailed breakdown of each step:

1- Sentence-BERT (SBERT) Embedding Generation

We used SBERT model (all-MiniLM-L6-v2) from the sentence_transformers library. It converts textual data (abstracts or titles) into dense numerical vectors (embeddings) using this model.

The embeddings are generated using model.encode() which processes the text and returns high-dimensional vector representations.

Benefits of SBERT Embeddings:

- SBERT captures the contextual meaning of sentences rather than just individual words.
- Better for Similarity Search & Clustering which can be used for clustering as they encode textual meaning effectively.

2. TF-IDF Feature Extraction

The used method is TF-IDF Vectorization. The TfidfVectorizer is applied to the same text data.

TF-IDF (Term Frequency-Inverse Document Frequency) is a statistical measure that evaluates the importance of words in a document relative to a collection (corpus).

The vectorizer is configured with:

max_features=2000 → Limits the vocabulary to the top 2000 most frequent words.

ngram_range=(1,2) → Considers both single words (unigrams) and word pairs (bigrams).

Benefits of TF-IDF:

- Captures Word Importance: Helps differentiate between common and rare words in a document.
- Sparse Representation: It results in a high-dimensional sparse matrix that can be useful for classical machine learning models.
- Enhances Text Understanding: Useful for capturing the importance of specific terms in classification tasks.

3 - Combining SBERT Embeddings with TF-IDF Features

The method used is Feature Concatenation which horizontally stacks (concatenates) the SBERT embeddings and TF-IDF features.

Benefits of Combining Both Approaches:

- Rich Representation: Merges deep semantic understanding (SBERT) with word importance (TF-IDF).
- Better Performance: The combined feature set may improve performance in tasks like clustering, classification, and retrieval.
- Balances Contextual and Statistical Information: SBERT captures the meaning, while TF-IDF ensures word importance is preserved.

Dimensionality Reduction

In unsupervised learning, we use PCA (Principal Component Analysis) to reduce the dimensionality of data while preserving as much variance as possible. This helps in visualizing and simplifying complex datasets.

For text embeddings and NLP, validation scores (like accuracy, clustering performance, or similarity effectiveness) are more useful than variance for determining the optimal number of PCA components. Variance is a useful mathematical measure but, in this context, it does not always correlate with better performance in language tasks. Retaining too many dimensions may not improve the downstream NLP like clustering but instead increase noise and overfitting. Asa result, reducing dimensionality while keeping high task performance is preferable over just keeping high variance.

In order to find the best number of components, we employed the method to test the reaction of the clustering technique after reducing the dimensions. We apply clustering (KMeans) as a test case to group similar data points together. The goal of this approach is finding the optimal number of PCA components to retain useful information while removing noise. Also, evaluate how well the clustering structure is formed after PCA

The analysis found that the optimal number of PCA components is 2, meaning most of the dataset's variance is well represented in two dimensions, making visualization and interpretation easier. Clustering using KMeans identified 3 distinct groups, suggesting that the data naturally forms three meaningful clusters. The Silhouette Score of 0.5308 indicates an acceptable clustering quality, meaning the clusters are reasonably well-separated. These findings help reveal underlying patterns in the dataset, providing a strong foundation for further analysis and interpretation.

The transformed data is stored in a new two-dimensional representation, where PC1 captures the highest variance and PC2 captures the second highest variance, making it possible to cluster and interpret relationships between research papers.