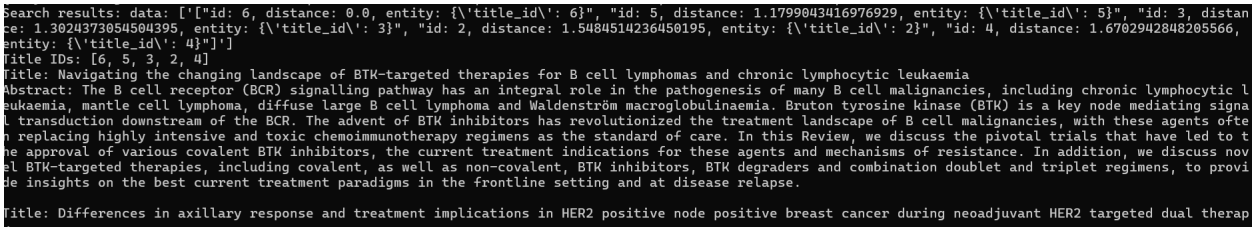
# Documentation for Code in main.py

Output: **The lesser the distance. The more relevant the article is to the query given.**




---

## Overview

This script automates the process of scraping articles from the specified website, storing them in a MySQL database, generating text embeddings, storing the embeddings in a Milvus vector database for efficient retrieval, and performing similarity searches. Each part of the process is modularized into functions for easy understanding and maintenance.

## Step-by-Step Explanation

### 1. Configuration Loading

- The script reads configuration settings from a config.ini file using the configparser module.
- The config.ini file is expected to store MySQL and Milvus connection details, which are then loaded and used throughout the script.

### 2. Database Connection: MySQL

- MySQL Database Connection Setup: Configures MySQL database connection using settings from config.ini.
- This is used in various functions to save, retrieve, and manage article data.

## 3. Milvus Connection

- Connecting to Milvus: Establishes a connection to a Milvus server using parameters from the configuration file.
- Milvus is a vector database used here for efficient similarity search of embedded article titles.

## 4. Function save_to_database

Purpose: Saves article details into the MySQL database, specifically into the journals table.

Process:

Connects to MySQL.

Checks the current maximum id in the journals table to assign a sequential id to each new record.

Inserts the article's title, authors, published_date, and abstract as a new record.

Error Handling: Catches MySQL errors and ensures that connections are properly closed.

## 5. Function get_article_details

Purpose: Extracts article details from a specific article URL and calls save_to_database to store them.

Process:

Fetches the page using requests.

Parses the HTML to extract:

Title

Authors

Published Date

Abstract

Calls save_to_database to store these details.

## 6. Function get_latest_research_urls

Purpose: Scrapes a main URL to find links to the latest research articles.

Process:

Fetches the main page.

Finds the "Latest Research and Reviews" section.

For each article link found, calls get_article_details to retrieve and store its details.

Structure: Uses a fixed base URL (https://www.nature.com) and concatenates it with relative links for complete URLs.

## 7. Function recreate_milvus_collection

Purpose: Ensures that the Milvus collection for storing article embeddings is set up.

Process:

Checks if the journal_titles collection exists and deletes it if it does.

Defines the schema for a new collection with two fields:

title_id: ID of the title, primary key.

title_embedding: 384-dimensional vector field to store embeddings.

Creates a new collection in Milvus based on this schema.

## 8. Function embed_and_store_in_milvus

Purpose: Generates embeddings for each article title and stores them in Milvus.

Process:

Retrieves all titles from the MySQL journals table.

For each title:

Generates an embedding using SentenceTransformer.

Stores both the title ID and embedding in Milvus.

Output: Displays the ID and title being stored, as seen in the output logs you shared.

## 9. Function create_index

Purpose: Builds an index on the title embeddings in Milvus to speed up similarity searches.

Process:

Defines indexing parameters.

Creates an index on the title_embedding field using IVF_FLAT with L2 distance metric.

Loads the collection into memory for fast searching.

## 10. Function search_articles

Purpose: Searches for articles similar to a given query based on title embeddings.

Process:

Generates an embedding for the query using SentenceTransformer.

Uses this embedding to search for the most similar articles in the Milvus collection.

Retrieves the top K matches and then fetches their details from MySQL by matching IDs.

## Output:

Displays the query embedding and found title IDs.

Prints details of the top matching articles, including titles and abstracts.

11. Execution in main Block

Data Retrieval: Calls get_latest_research_urls to fetch and store the latest articles.

Milvus Setup and Embedding:

Calls recreate_milvus_collection to set up the Milvus collection.

Calls embed_and_store_in_milvus to store title embeddings.

Calls create_index to index embeddings.

Search Example: Runs a sample query using search_articles.

Explanation of the Provided Output

The output provides feedback during the embedding and search processes:

Embedding Storage Messages:

"Storing title: [Title] with ID: [ID]": Shows that the script is generating and storing embeddings for each title in the database, confirming which title (and corresponding ID) is processed.

Embedding Storage Confirmation:

"Embeddings stored in Milvus": Indicates all title embeddings have been successfully stored in the Milvus vector database.

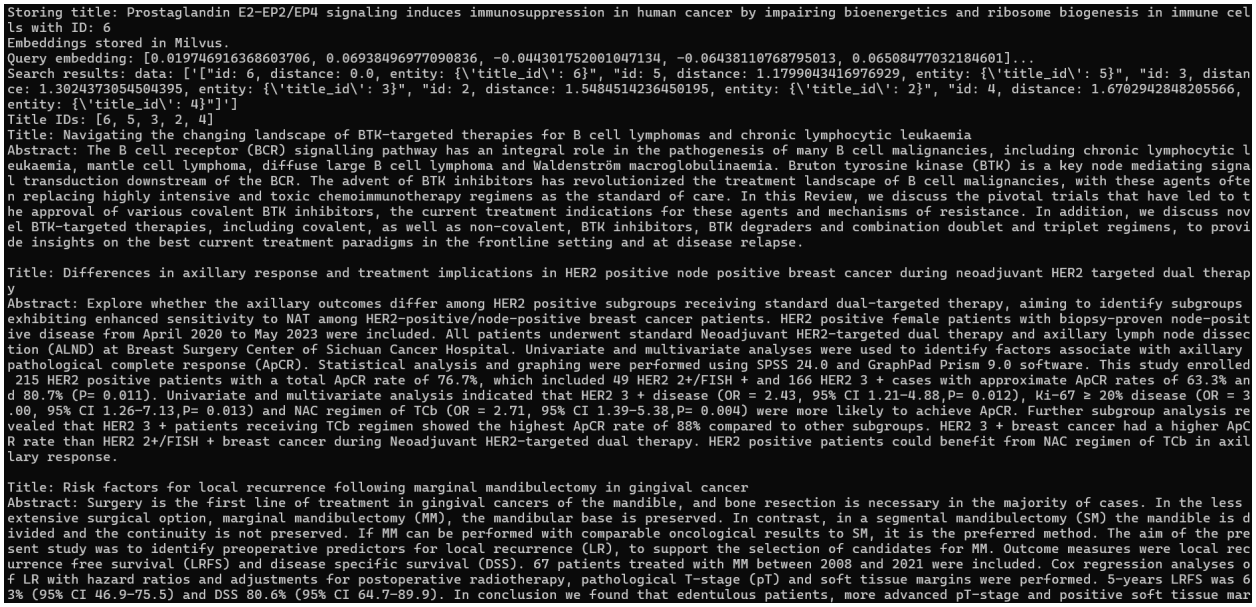
Query Embedding and Search Results:

"Query embedding: [First few values]...": Shows the embedding generated from the search query to help confirm the embedding model's output.

"Search results: data: [...], Title IDs: [IDs]": Lists the IDs of the top matching titles from the search. The distance values indicate similarity, with lower values being more similar.

Retrieved Articles:

Each "Title: [Title] \nAbstract: [Abstract]" section shows a matching article found in the search, with its title and abstract. This provides the user with context and detail on similar articles based on the original search query.




**The lesser distance. The more relevant the article is to the query given.**