**Task4 implementation process**

Task 4: Find out the similarity between Github answers and Github Copilot answers:

Convert the texts into vectors and calculate the cosine similarity of the two answers.

**Step 1.** **Text preprocessing**

Before comparing the similarity between human and AI answers, we observed that directly computing cosine similarity on raw AI responses led to abnormally low results. To improve accuracy, we applied a preprocessing pipeline using NLTK, including:

Tokenization: Splitting text into individual tokens.

Stop Words Removal: Filtering out common words with little semantic value (e.g., “the”, “in”).

Stemming & Lemmatization: Reducing words to their base form (e.g., “running” → “run”).

**Step 2.** **Model selection and similarity calculation**

**The calculation formula of cosine similarity:**

图示

AI 生成的内容可能不正确。

文本

AI 生成的内容可能不正确。

After discussion and learning, our group decided to use TF-IDF Weighted Cosine Similarity and SBERT Sentence Cosine Similarity to calculate the similarity between manual answers and copilot answers. and finally decided to use SBERT Sentence Cosine Similarity because of results.

**TF-IDF Weighted Cosine Similarity (Term Frequency-Inverse Document Frequency)**:

TF-IDF converts text into sparse vectors where each dimension corresponds to a word, and the value indicates its importance. This model does not need to be pre-trained and is good for keyword-based short text.

The cosine similarity calculation results obtained by the model are stored in the task 4 named ‘TF-IDF Cosine Similarity Output file’

**SBERT Sentence Cosine Similarity (Sentence-BERT):**

SBERT (Sentence-BERT) is a sentence embedding model based on Transformers. ‘**all-MiniLM-L6-v2**’ is a lightweight, fast, and high-performing sentence embedding model of SBERT. It converts text into sparse vectors where each dimension corresponds to a word, and the value indicates its importance. This model allow users to compute cosine similarity on semantically aligned vectors.

In the **all-MiniLM-L6-v2** model, each token in a sentence is represented by a 384-dimensional vector.Therefore, if a sentence has T tokens after tokenization, the model's output will be a matrix of shape T × 384, where each row corresponds to the contextual embedding of a token.

Mean Pooling：

图示

AI 生成的内容可能不正确。

The cosine similarity calculation results obtained by the model are stored in the task 4 file named ‘SBERT Cosine Similarity Output.xlsx’