SOFTWARE DOCUMENTATION USING CODE SUMMARISATION

```
!pip install datasets
!pip install transformers
```

```
Collecting datasets
  Downloading datasets-2.19.0-py3-none-any.whl (542 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 542.0/542.0 kB 5.7 MB/s eta 0:00:00
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from datasets) (3.13.4)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from datasets) (1.25.2)
Requirement already satisfied: pyarrow>=12.0.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (14.0.2)
Requirement already satisfied: pyarrow-hotfix in /usr/local/lib/python3.10/dist-packages (from datasets) (0.6)
Collecting dill<0.3.9,>=0.3.0 (from datasets)
  Downloading dill-0.3.8-py3-none-any.whl (116 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 116.3/116.3 kB 7.3 MB/s eta 0:00:00
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from datasets) (2.0.3)
Requirement already satisfied: requests>=2.19.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (2.31.0)
Requirement already satisfied: tqdm>=4.62.1 in /usr/local/lib/python3.10/dist-packages (from datasets) (4.66.2)
Collecting xxhash (from datasets)
  Downloading xxhash-3.4.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (194 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 194.1/194.1 kB 6.7 MB/s eta 0:00:00
Collecting multiprocess (from datasets)
  Downloading multiprocess-0.70.16-py310-none-any.whl (134 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 134.8/134.8 kB 7.5 MB/s eta 0:00:00
Requirement already satisfied: fsspec[http]<=2024.3.1,>=2023.1.0 in /usr/local/lib/python3.10/dist-packages (from datase
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-packages (from datasets) (3.9.5)
Collecting huggingface-hub>=0.21.2 (from datasets)
  Downloading huggingface_hub-0.22.2-py3-none-any.whl (388 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 388.9/388.9 kB 9.2 MB/s eta 0:00:00
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from datasets) (24.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from datasets) (6.0.1)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.3
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (23.2.0
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.9.4
Requirement already satisfied: async-timeout<5.0,>=4.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->dataset
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-h
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->datasets)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->dat
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->dat
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2024.1
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas-
Installing collected packages: xxhash, dill, multiprocess, huggingface-hub, datasets
  Attempting uninstall: huggingface-hub
    Found existing installation: huggingface-hub 0.20.3
    Uninstalling huggingface-hub-0.20.3:
      Successfully uninstalled huggingface-hub-0.20.3
Successfully installed datasets-2.19.0 dill-0.3.8 huggingface-hub-0.22.2 multiprocess-0.70.16 xxhash-3.4.1
Requirement already satisfied: transformers in /usr/local/lib/python3.10/dist-packages (4.40.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from transformers) (3.13.4)
Requirement already satisfied: huggingface-hub<1.0,>=0.19.3 in /usr/local/lib/python3.10/dist-packages (from transformer
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (1.25.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from transformers) (24.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (6.0.1)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (2023.12
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from transformers) (2.31.0)
Requirement already satisfied: tokenizers<0.20,>=0.19 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.
Requirement already satisfied: safetensors>=0.4.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.4.3)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-packages (from transformers) (4.66.2)
```

```
from google.colab import drive

# Mount Google Drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
from transformers import BartTokenizer, BartForConditionalGeneration
import os

# Input folder containing Python source files
fine_tuned_model_path =input("Enter the path of the fine-tuned model: ")
input_folder = input("Enter the path of the folder containing Python source files: ")

# Load fine-tuned BART tokenizer and model
bart_tokenizer = BartTokenizer.from_pretrained("facebook/bart-base")
fine_tuned_model = BartForConditionalGeneration.from_pretrained(fine_tuned_model_path)

# Load BART-large-cnn tokenizer and model
bart_large_cnn_tokenizer = BartTokenizer.from_pretrained("facebook/bart-large-cnn")
bart_large_cnn_model = BartForConditionalGeneration.from_pretrained("facebook/bart-large-cnn")

# Step 1: Preprocessing
def preprocess_folder(folder_path):
    source_files = []
    for filename in os.listdir(folder_path):
        if filename.endswith(".py"):
            with open(os.path.join(folder_path, filename), 'r', encoding='utf-8') as file:
                source_files.append(file.read())
    return source_files

# Step 2: Method Summarization using fine-tuned BART model
def summarize_method(method_text, tokenizer, model):
    input_ids = tokenizer.encode("summarize: " + method_text, return_tensors="pt", max_length=1024, truncation=True)
    summary_ids = model.generate(input_ids, max_length=150, min_length=40, length_penalty=2.0, num_beams=4, early_stopping=
    summary = tokenizer.decode(summary_ids[0], skip_special_tokens=True)
    return summary

# Step 3: Source File Summarization using BART-large-cnn model
def summarize_source_file(method_summaries, tokenizer, model):
    input_text = " ".join(method_summaries)
    input_ids = tokenizer.encode("summarize: " + input_text, return_tensors="pt", max_length=1024, truncation=True)
    summary_ids = model.generate(input_ids, max_length=300, min_length=100, length_penalty=2.0, num_beams=4, early_stopping=
    summary = tokenizer.decode(summary_ids[0], skip_special_tokens=True)
    return summary

# Step 4: Overall Project Summarization using BART-large-cnn model
def summarize_project(source_file_summaries, tokenizer, model):
    input_text = " ".join(source_file_summaries)
    input_ids = tokenizer.encode("summarize: " + input_text, return_tensors="pt", max_length=1024, truncation=True)
    summary_ids = model.generate(input_ids, max_length=500, min_length=200, length_penalty=2.0, num_beams=4, early_stopping=
    summary = tokenizer.decode(summary_ids[0], skip_special_tokens=True)
    return summary


# Step 1: Preprocessing
source_files = preprocess_folder(input_folder)

# Step 2: Method Summarization using fine-tuned BART model
method_summaries = [summarize_method(method, bart_tokenizer, fine_tuned_model) for method in source_files]

# Step 3: Source File Summarization using BART-large-cnn model
source_file_summary = summarize_source_file(method_summaries, bart_large_cnn_tokenizer, bart_large_cnn_model)

# Step 4: Overall Project Summarization using BART-large-cnn model
project_summary = summarize_project([source_file_summary], bart_large_cnn_tokenizer, bart_large_cnn_model)


import re

def simplify_code_summary(summary):
    # Split the summary into lines
    lines = summary.split("\n")
    updated_lines = []

    # Process each line individually
    unique_lines = set()
    for line in lines:
        # Remove consecutive repeating words
        line = re.sub(r'\b(\w+)(?:\W+\1\b)+', r'\1', line)

        # Remove words identified by dot (.)
        line = re.sub(r'\b\w+\.\w+\b', '', line)

        # Remove specified words from the line
        line = re.sub(r'\b(return|def|summary)\b', '', line)

        # Remove function calls with arguments
        line = re.sub(r'\b\w+\s*\([^()]*\)\s*', '', line)
```

```
            # Introduce new line whenever a period (.) is encountered
            line = line.replace('.', '.\n')

            # Remove hashes (#)
            line = line.replace('#', '')

            # Add line to updated lines if it's not a repetition
            if line not in unique_lines:
                updated_lines.append(line.strip())
                unique_lines.add(line)

        # Join the updated lines with new lines
        updated_summary = "\n".join(updated_lines)

        return updated_summary.strip()

print("PROJECT SUMMARY")
print(simplify_code_summary(project_summary))
```

```
 Enter the path of the fine-tuned model: /content/drive/MyDrive/finetuned_model
 Enter the path of the folder containing Python source files: /content/drive/MyDrive/chess_game
 PROJECT SUMMARY
 The game engine scales chess pieces to fit the square size of the board.
  The game state is displayed as a black and white image of the chess board.
  It is used to show the state of the game as well as the current state of each piece.
  It also shows the current position of the player and their position on the board at the start and end of a game.
  It can also be used to display the position of different pieces at different times in the game, such as when a move is
  For more information on how to play chess in the UK, visit .
  or go to .
 com.
  In the .
  and Canada, go to  or call the National Chess Association on 1-800-273-8255 or visit a local branch of the NCA.
  In Europe, the NCCA is based at the University of Edinburgh.
```