Explanation of Thinking Process for Implementing News Article Question-Answering System

# Objective: Answer user queries using news articles related to Israel and Hamas, supplemented with Wikipedia summaries.

# Process:

**Step 1: Import Required Libraries**

A computer screen shot of text

Description automatically generated

Explanation:

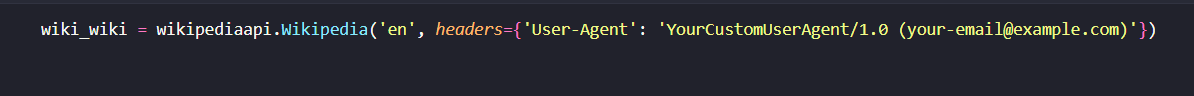
* json: Used for reading JSON data.
* re: Provides support for regular expressions, used for text preprocessing.
* nltk: Natural Language Toolkit for various NLP tasks like tokenization.
* tqdm: Progress bar library for visualizing progress during filtering and preprocessing.
* "BM25Okapi" from "rank\_bm25": BM25 algorithm for information retrieval.
* "pipeline", "AutoTokenizer", "AutoModelForQuestionAnswering" from "transformers": Essential components for working with Hugging Face's
* Transformers library, including pre-trained models and pipelines.
* wikipediaapi: API for interacting with Wikipedia.
* time: Used for adding delays between retries when fetching Wikipedia data.

**Step 2: Load Ddata from news-article.json File**

**A black background with yellow text

Description automatically generated**

**Step 3: Initialize Wikipedia API**

****

**Step 4: Define Pre-processing Function**

**A computer screen shot of text

Description automatically generated**

**Step 5: Filter and Pre-process Articles**

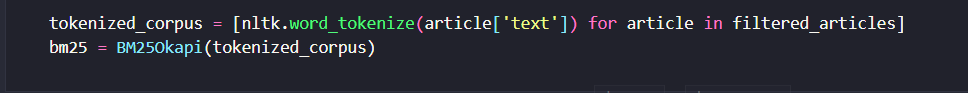
**A screen shot of a computer code

Description automatically generated**

Explanation:

* Filters articles based on whether they contain keywords 'Israel' or 'Hamas' in the article body.
* Pre-processes the filtered articles by converting text to lowercase and removing non-word characters.
* Returns a list of relevant articles.

**Step 6: Tokenize Articles, Initialize BM25 Model and Initialize Question Answering Pipeline**

****

**Step 7: Retrieve Articles Based on Query**

**A computer code with colorful text

Description automatically generated**

Explanation:

* Takes a query and retrieves top articles related to the query using BM25 scoring. Returns a list of relevant articles from filtered\_articles.

**Step 8: Initializing Question Answering Pipeline**

A screen shot of a computer code

Description automatically generated

Explanation:

* Specifies a fine-tuned BERT model (bert-large-uncased-whole-word-masking-finetuned-squad) for question answering using AutoTokenizer and AutoModelForQuestionAnswering from Transformers.

**Step 9: Answer Question Using Articles**

**A computer screen shot of text

Description automatically generated**

**Step 10: Getting Answers from Articles**

A computer screen shot of a program code

Description automatically generated

Explanation:

* Retrieves top articles related to a given question using retrieve\_articles.
* For each article, applies answer\_question to extract the best answer and stores relevant information (article\_title, article\_source, answer, score, start, end) in answers.
* Sorts answers based on the answer's score in descending order and returns the top 3 answers.

**Step 11: Wikipedia Summery**

**A computer screen with text

Description automatically generated**

Explanation**:**

* Takes a question, attempts to fetch the corresponding Wikipedia page (page) using wiki\_wiki.
* If the page exists, retrieves the summary and limits it to max\_summary\_length characters.
* Handles exceptions and retries (max\_retries) with a delay (time.sleep(2)) if necessary.

**Step 12: User Interaction**

**A computer screen with text

Description automatically generated**

Explanation:

* Prompts the user to input a question.
* Retrieves and prints a Wikipedia summary based on the question.
* Retrieves and prints the top 3 answers from the filtered articles based on the question.