Summarization System Documentation

Overview:
This summarization system is designed to extract information from the official websites of companies based on user queries. It utilizes web scraping techniques, PDF extraction, and GPT for summarization. The system is integrated with the Azure Chat OpenAI service for language model-based summarization.

Components:

1. **Initialization Functions**:
    - **initialize_azure_chat_openai()**: Initializes the Azure Chat OpenAI service with required configurations such as API keys and endpoints.
2. **Data Extraction Functions**:
    - **fetch_top_links(query, company_name)**: Uses Google Search API to fetch the top links related to the user query and the company name.
    - **extract_text_from_pdf(url)**: Downloads and extracts text content from PDF files.
    - **scrape_text_from_link(url)**: Scrapes text content from HTML pages.
3. **Summarization Functions**:
    - **summarize_content(content)**: Utilizes Azure Chat OpenAI service to generate a summary of the given text content.
4. **Main Functionality**:
    - **main(query, company_name)**: The main function orchestrating the entire process. It fetches top links, extracts content, generates summaries, and formats the output.

Process Flow:

1. **User Input**:
    - Users provide a query and the name of the company they are interested in summarizing.
2. **Fetching Top Links**:
    - The system utilizes the Google Search API to fetch the top links related to the provided query and company name.
3. **Content Extraction**:
    - For each retrieved link:
        - If the link points to a PDF file, the system extracts text content using PyMuPDF.
        - If the link points to an HTML page, the system scrapes the text content using BeautifulSoup.
4. **Summarization**:
    - The extracted content from each link is summarized using the Azure Chat OpenAI service.
    - Individual summaries for each link are generated.
5. **Collective Summary**:
    - If multiple summaries are available, the system generates a collective summary by combining all individual summaries.
6. **Output Formatting**:

    - The system formats the output by concatenating individual summaries and the collective summary.
7. **Displaying Results**:
    - The formatted output is displayed through the interface for user interaction.

Dependencies:

- **os**, **requests**, **BeautifulSoup**: For web-related operations and file handling.
- **googlesearch**: For fetching top search results.
- **fitz**: PyMuPDF library for PDF handling.
- **transformers**, **sentence_transformers**: For natural language processing tasks.

Usage:

- The system fetches information from relevant sources, summarizes it, and presents the summarized content back to the user.

Limitations:

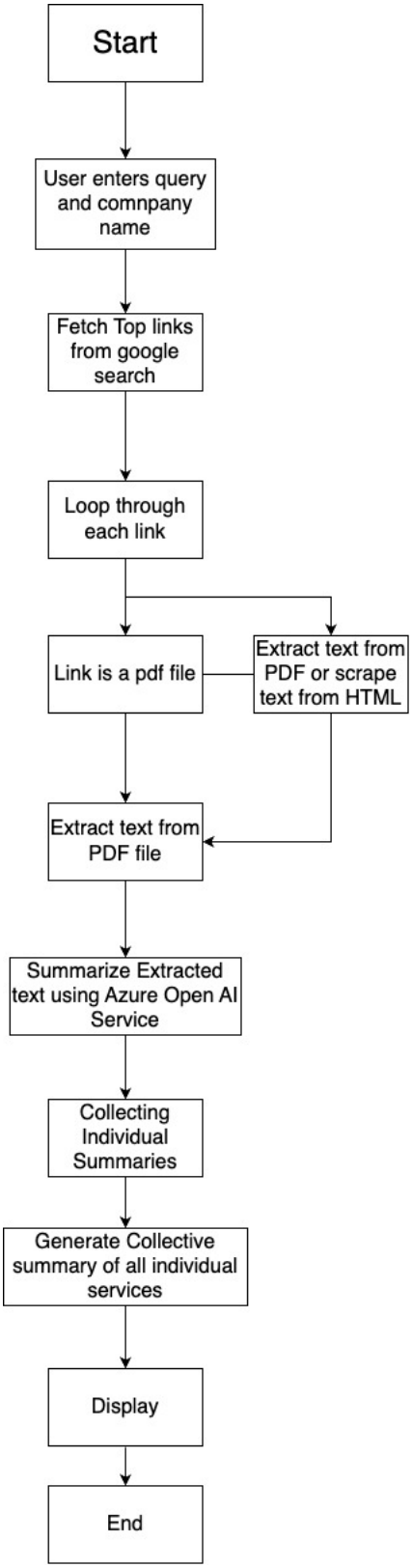
- Dependency on external services (Google Search API, Azure Chat OpenAI) may lead to variations in performance and reliability.
- The effectiveness of summarization heavily depends on the quality and relevance of the extracted content.

Future Improvements:

- Implement error handling and retry mechanisms for robustness.
- Enhance summarization quality through fine-tuning or utilizing different NLP models.
- Provide options for selecting specific sections or types of content for summarization.

The process flow diagram illustrates the following steps:

1. The user enters a query and a company name in the interface.
2. The application fetches the top links from a Google search based on the query and company name.
3. For each link:
    - If the link is a PDF file, the application extracts the text content from the PDF.
    - If the link is an HTML page, the application scrapes the text content from the webpage.
4. The extracted text content from each link is summarized using the Azure OpenAI service.
5. The individual summaries are collected.
6. A collective summary is generated by summarizing the combined text of all individual summaries.
7. The individual summaries and the collective summary are displayed in the interface.
8. The process ends.


Start
User enters query and comnpany name
Fetch Top links from google search
Loop through each link
Link is a pdf file
Extract text from PDF or scrape text from HTML
Extract text from PDF file
Summarize Extracted text using Azure Open AI Service
Collecting Individual Summaries
Generate Collective summary of all individual services
Display
End