

```

# -*- coding: utf-8 -*-
"""News Articles Summarizer

Automatically generated by Colab.

Original file is located at
https://colab.research.google.com/drive/1oOL\_bIeipKt3IgsNSbKEzj4VXczVP1cl

And here are the steps described in more detail:

**Install required libraries**: To get started, ensure you have the necessary libraries installed: requests, newspaper3k, and langchain.
**Scrape articles**: Use the requests library to scrape the content of the target news articles from their respective URLs.
**Extract titles and text**: Employ the newspaper library to parse the scraped HTML and extract the titles and text of the articles.
**Preprocess the text**: Clean and preprocess the extracted texts to make them suitable for input to ChatGPT.
**Generate summaries**: Utilize ChatGPT to summarize the extracted articles' text concisely.
**Output the results**: Present the summaries along with the original titles, allowing users to grasp the main points of each article quickly.
"""

!pip install requests newspaper3k

!pip install langchain==0.0.208 deeplake openai tiktoken

!pip install -q newspaper3k python-dotenv

import os
os.environ["OPENAI_API_KEY"] = ''

import requests
from newspaper import Article

headers = {
    'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.36', 'Accept-Language': 'en-US, en'
}

article_url = "https://www.artificialintelligence-news.com/2022/01/25/meta-claims-new-ai-supercomputer-will-set-records/"

session = requests.Session()

try:
    response = session.get(article_url, headers=headers, timeout=10)

    if response.status_code == 200:
        article = Article(article_url)
        article.download()
        article.parse()

        print(f"Title: {article.title}")
        print(f"Text: {article.text}")

    else:
        print(f"Failed to fetch article at {article_url}")
except Exception as e:
    print(f"Error occurred while fetching article at {article_url}: {e}")

from langchain.schema import (
    HumanMessage
)

# we get the article data from the scraping part
article_title = article.title
article_text = article.text

# prepare template for prompt
template = """You are a very good assistant that summarizes online articles.

Here's the article you want to summarize.

=====
Title: {article_title}

{article_text}
=====

Write a summary of the previous article.
"""

prompt = template.format(article_title=article.title, article_text=article.text)

messages = [HumanMessage(content=prompt)]

!pip install -qU \
    langchain==0.0.292 \
    openai==0.28.0 \
    datasets==2.10.1 \
    pinecone-client==2.2.4 \
    tiktoken==0.5.1

from langchain.chat_models import ChatOpenAI

# load the model
chat = ChatOpenAI(openai_api_key = os.environ['OPENAI_API_KEY'], model="gpt-4")

summary = chat(messages)
print(summary.content)

```