Nhóm 9

Web Crawler in Python

Thành viên nhóm:

- Nguyễn Trường Duy
- Nguyễn Anh Nguyên

Nhóm 9

Web Crawler là gì?

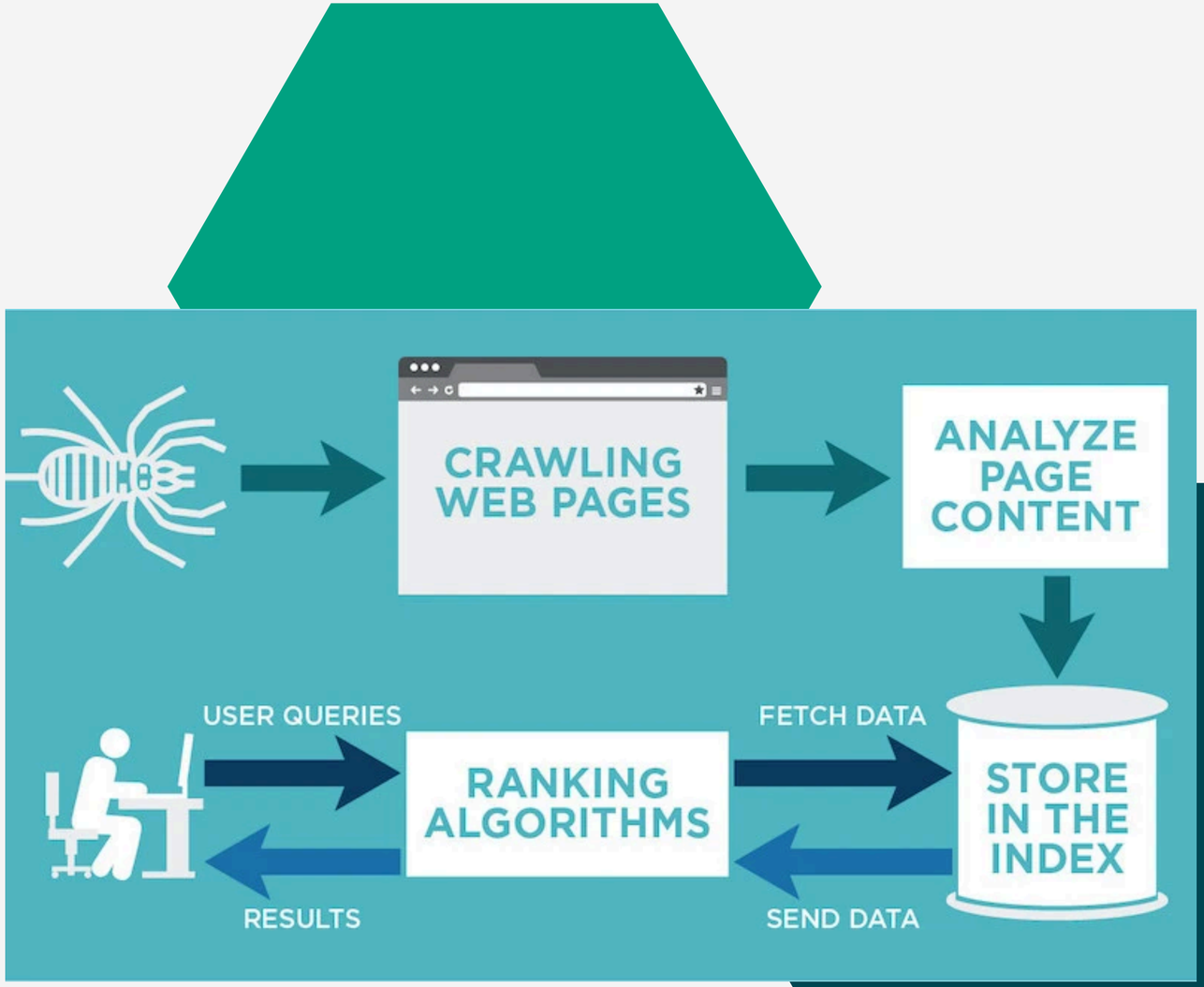
Web crawler được hiểu là các website có khả năng tiếp nhận tự động các dữ liệu trên internet thông qua các trang world wide web có sẵn. Nói một cách dễ hiểu thì web crawler chính là một con bot của công cụ tìm kiếm, có thể thu thập và lập chỉ mục nội dung cho tất cả các website có sẵn.

Quay lại Trang Chương trình

Nhóm 9

Cách thức hoạt động của web crawler

- Web crawler sẽ tìm hiểu từng URL trên internet và tiến hành phân loại các URL có cùng nội dung chủ đề vào cùng một nhóm. Sau đó, web crawler sẽ thêm các siêu liên kết trên một website bất kỳ vào danh sách cần thu thập thông tin.



Quay lại Trang Chương trình

Crawl with python

Quay lại Trang Chương trình

Crawl A Web Page with Scrapy

- Scrapy là một trong những thư viện quét Python phổ biến và mạnh mẽ nhất; nó sử dụng cách tiếp cận "bao gồm pin" để crawl, nghĩa là nó xử lý rất nhiều chức năng phổ biến mà tất cả các máy cạp cần để các nhà phát triển không phải phát minh lại bánh xe mỗi lần.

Crawl A Web Page with Selenium

- Selenium ban đầu là một công cụ được tạo ra để kiểm tra hoạt động của trang web, nhưng nhanh chóng, nhu cầu quét web bằng Selenium đã tăng lên
- Công cụ này khá phổ biến và có khả năng tự động hóa các trình duyệt khác nhau như Chrome, Firefox, Opera và thậm chí cả Internet Explorer thông qua phần mềm trung gian được kiểm soát có tên là Selenium webdriver.

Crawl A Web Page with Scrapy

Quay lại Trang Chương trình

Step 1 — Creating a Basic Scraper

Scraping có 2 bước chính:

1. Tìm kiếm và tải xuống các trang web một cách có hệ thống.
2. Trích xuất thông tin từ các trang.

Step 2 — Extracting Data from a Page

Scraping là một quá trình gồm hai bước:
1.Đầu tiên, lấy từng trích dẫn bằng cách tìm kiếm các phần của trang có dữ liệu chúng tôi muốn.
2.Sau đó, đối với mỗi trích dẫn, hãy lấy dữ liệu mà chúng tôi muốn từ đó bằng cách lấy dữ liệu ra khỏi thẻ HTML

Step 3 — Crawling Multiple Pages

Lấy dữ liệu nhiều page từ domain muốn crawl

Quay lại Trang Chương trình

Cách thực hiện

Install scrapy

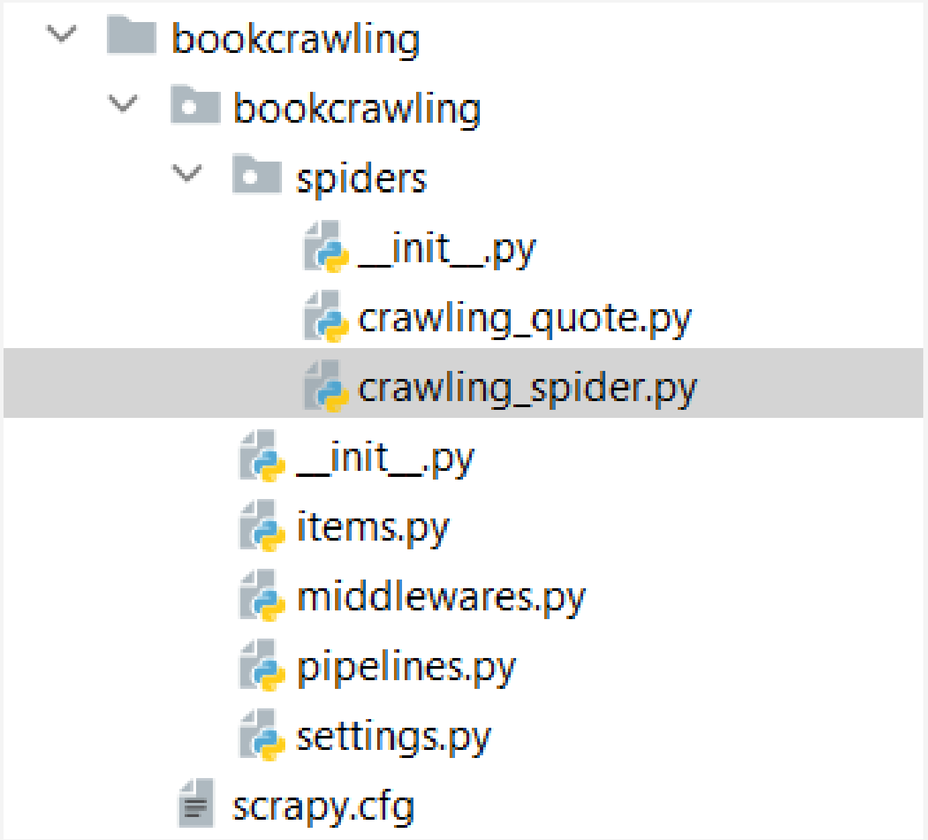
pip install scrapy

Tạo project

scrapy startproject myproject
Thư mục của mình tên bookcrawling

Tạo file py trong thư mục spiders

Example:crawling_quote.py



Cách thực hiện

Quay lại Trang Chương trình

Các attribute trong class

- name — just a name for the spider.
- allowed_domains — a list of domains
- start_urls — a list of URLs that you start to crawl from. We'll start with one URL.

Các class hỗ trợ

- Rule: This class is used to define rules for crawling. It specifies how to follow links from one page to another.
- LinkExtractor: This class is used to extract links from web pages based on certain criteria.

```
class CrawlingSpider(CrawlSpider):
    name = "myfirstcrawler"
    allowed_domains = ["toscrape.com"]
    start_urls = ["https://books.toscrape.com/"]

    #PROXY_SERVER = "127.0.0.1"
    rules = (
        Rule(LinkExtractor(allow="catalogue/category")),
        Rule(LinkExtractor(allow="catalogue",deny="category"), callback="parse_item")
    )
```

Cách thực hiện

Quay lại Trang Chương trình

Một số lệnh thông dụng

- scrapy genspider spider_name example.com
- scrapy crawl myfirstcrawler
- scrapy crawl myfirstcrawler -o output.json
- scrapy crawl myfirstcrawler -s LOG_FILE=scrapy.log
- scrapy shell "https://example.com"

```
#PROXY_SERVER = "127.0.0.1"
rules = (
    Rule(LinkExtractor(allow="catalogue/category")),
    Rule(LinkExtractor(allow="catalogue",deny="category"), callback="parse_item")
)
def parse_item(self,response):
    yield {
        "title":response.css(".product_main h1::text").get(),
        "price":response.css(".price_color::text").get(),
        "availability":response.css(".availability::text")[1].get().replace("\n","").replace(" ","")
    }
```

Phát triển thêm CHATPDF

Step 1 — Sử dụng streamlit

Sử dụng streamlit để tạo một trang web đơn giản chưa giao diện chat bot

Step 2 — Xử lý dữ liệu được crawl

- Dữ liệu được xuất theo dạng file json sẽ được chuyển hoá sang pdf
- Sử dụng langchain và chromadb để lưu dữ liệu dưới dạng vectordb

Step 3 — Tạo model chat bot

- Sử dụng ChatGoogleGenerativeAI từ google_ai
- Xử lý thông tin từ dữ liệu được lưu trữ trong vectordb để trả lời người dùng

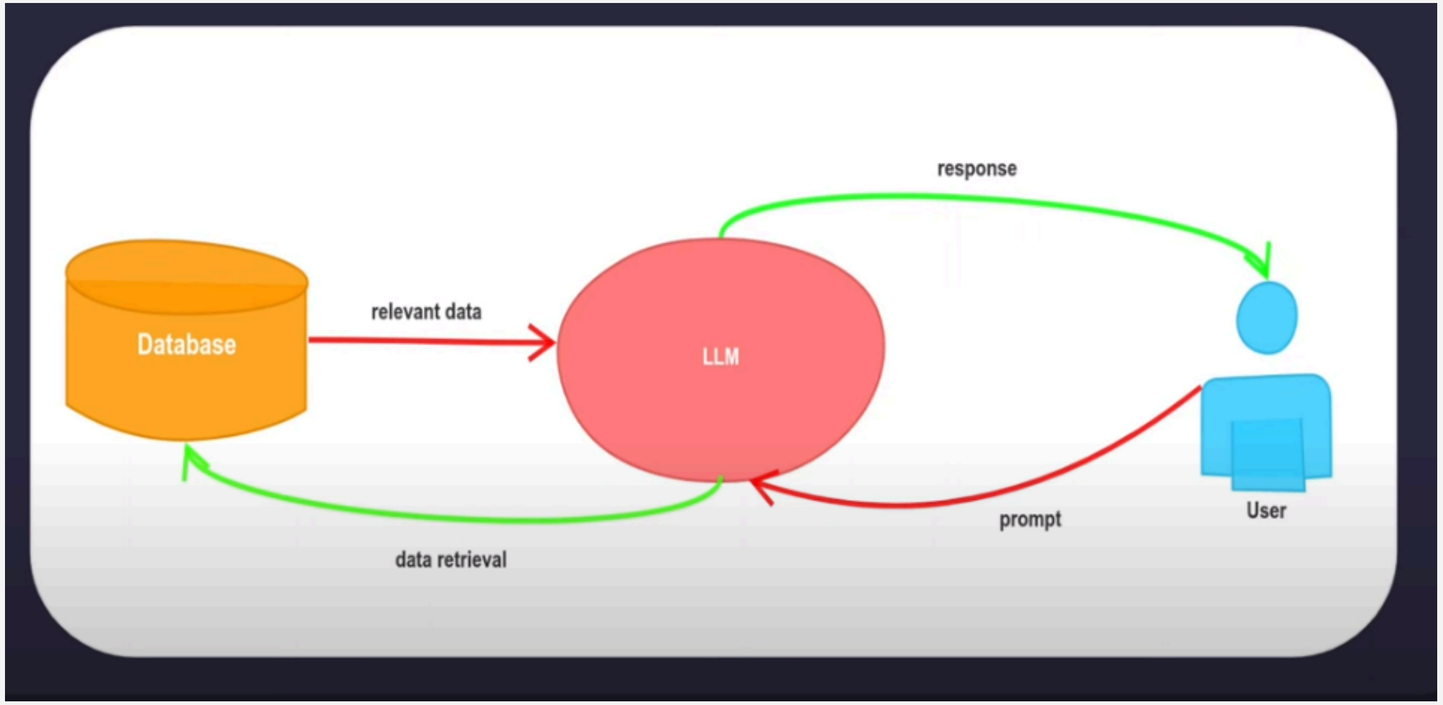
Quay lại Trang Chương trình

Phát triển thêm CHATPDF

Quay lại Trang Chương trình

Mô hình miêu tả tương tác với chat bot

Người dùng sau khi tương tác với chatbot, basic Large language model sẽ retrieve từ trong dữ liệu của database trả về data cho llm, llm sẽ trả về thông tin người dùng theo một template nhất định

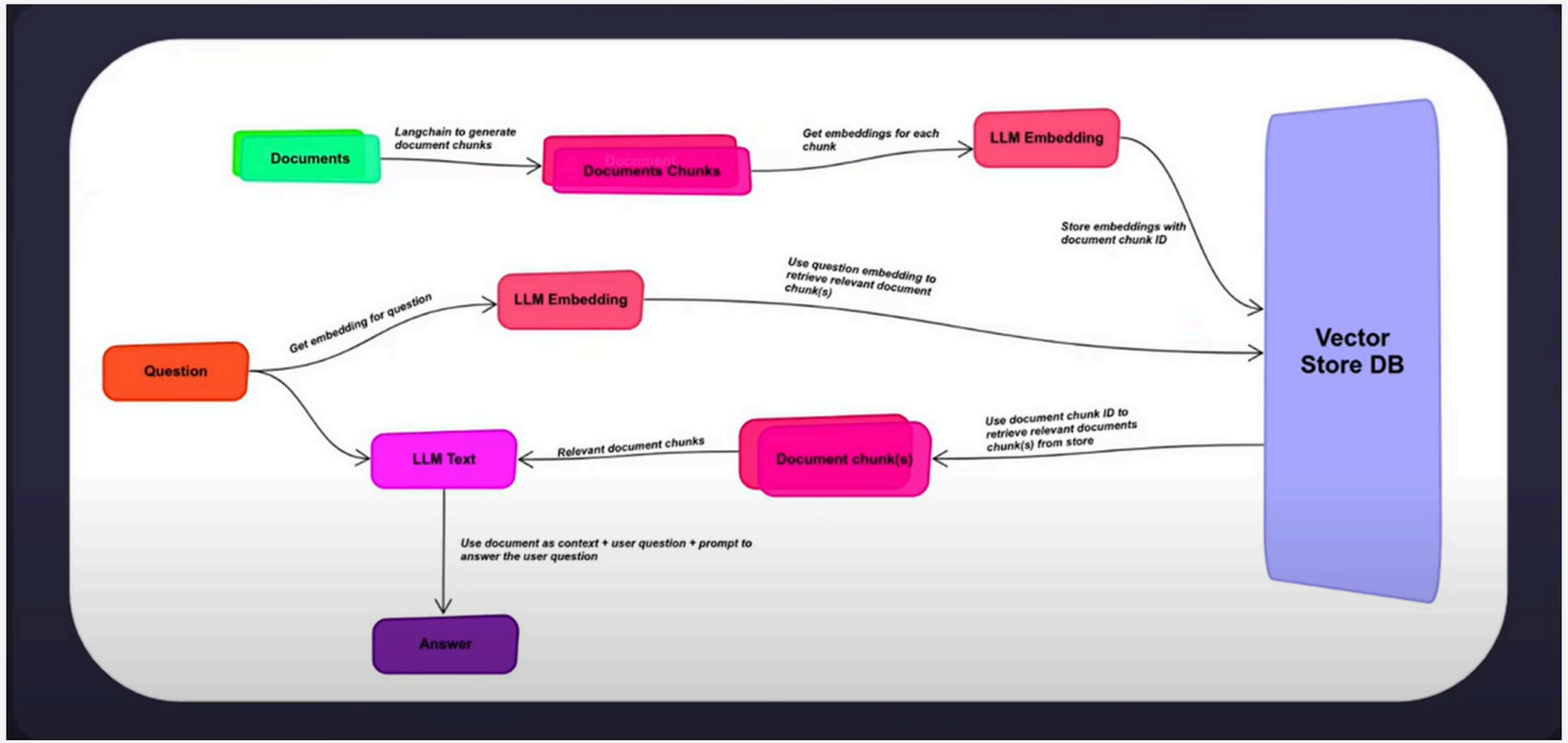


Phát triển thêm CHATPDF

Quay lại Trang Chương trình

Tổng quan quá trình thực hiện

Từ file tài liệu chúng được chia nhỏ ra thành các chunks để tạo ra các biểu diễn (embeddings) cho các câu sử dụng mô hình BERT sau đó lưu vào vectordb, quá trình còn lại là phần tương tác với câu hỏi của người dùng của llm(ta có thể retrieve data từ vectordb bằng nhiều kỹ thuật khác nhau



Một số kỹ thuật retrieve

Quay lại Trang Chương trình

Sematic Similarity Search

- Tìm kiếm dựa trên sự tương tự của từ(trong vector embeddings)

Maximum Marginal Relevance

- MMR kết hợp cả hai thành phần(độ liên quan và độ nhận dạng) thành một điểm tổng thể chọn ra các mục tốt nhất cho danh sách kết quả

Contextual Compression Retriever

- Tìm kiếm tương tự tài liệu trong kho lưu trữ vectơ bằng cách tính đến ngữ cảnh từ truy vấn

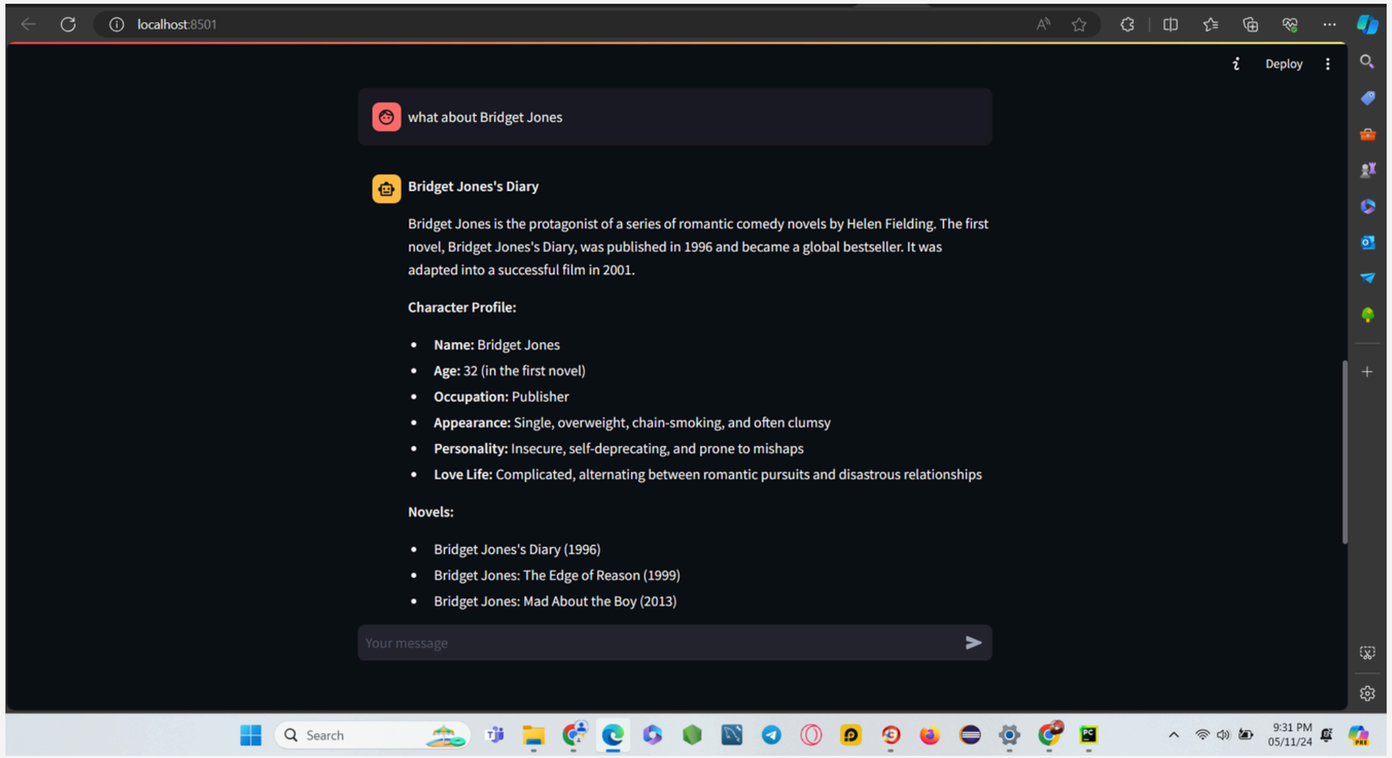
Crawl A Web Page with Selenium

- Sử dụng thông tin từ một phần của tài liệu hoặc ngữ cảnh hiện tại để truy vấn dữ liệu khác

Quay lại Trang Chương trình

Kết quả thực hiện

- Thực hiện được thành công việc crawl thông tin sách
- Sử dụng công cụ tạo được một chatbot để truy xuất về dữ liệu đã thu thập
- Tạo được giao diện tương tác với người dùng trên nên tảng web sử dụng streamlit



Kết quả thực hiện của crawl data thông tin sách

Quay lại Trang Chương trình

-Các trường tên đề sách, giá, còn hàng hay không, mô tả
-Từ các thông tin đó chuyển thành pdf và lưu vào vectordb