Customer Support System

Command Line & Web Based

K . Sri Vardhan

Index

Introduction

This project is a Customer Support System that uses web crawling, text embedding, and the OpenAI API to answer questions about web pages' content. It includes two implementations: `Command Line Based` and `Web Based`.

Design

The system is designed to perform the following steps:

1. **Web Crawling:** It crawls web pages, extracts their text content, and stores it for further processing.

2. **Text Embedding:** The extracted text is tokenized and embedded into numerical representations using OpenAI's embedding models.

3. **Question Answering:** Users can ask questions about the crawled web pages, and the system generates responses using the embeddings and OpenAI's API.

Design

Command Line Based

In the command line-based implementation, users interact with the system through the Ubuntu terminal. They can ask questions about the webpages crawled by the system, and the system generates responses based on the embedded text data. This implementation provides a straightforward and text-based interface for users to query web content.

Web-Based (Flask)

The web-based implementation utilizes the Flask framework to provide a user-friendly interface for interacting with the system. Users can access the system through a web browser, making it accessible and convenient. This web interface allows users to input questions and receive responses, enhancing the user experience and accessibility of the system.

Implementation

To run this project on an Ubuntu system,

- Install Python 3.10's virtual environment package, if not installed already

```
chinni@LAPTOP-UVNKR98P:~/check$ sudo apt install python3-venv
[sudo] password for chinni:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  python3-venv
```

- Create a Python virtual environment named 'venv' and activate it

```
chinni@LAPTOP-UVNKR98P:~/check$ python3 -m venv venv
chinni@LAPTOP-UVNKR98P:~/check$ . venv/bin/activate
(venv) chinni@LAPTOP-UVNKR98P:~/check$
```

Implementation

- Install the required Python packages listed in 'requirements.txt'

```
(venv) chinni@LAPTOP-UVNKR98P:~/check$ pip install -r requirements.txt
Collecting autopep8==1.6.0
  Using cached autopep8-1.6.0-py2.py3-none-any.whl (45 kB)
Collecting aiohttp==3.8.3
  Using cached aiohttp-3.8.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.0 MB)
Collecting aiosignal==1.3.1
  Using cached aiosignal-1.3.1-py3-none-any.whl (7.6 kB)
```

- Next, Crawl, Embed and Run the system

```
(venv) chinni@LAPTOP-UVNKR98P:~/check$ python3 crawl.py
https://x.ai/
(venv) chinni@LAPTOP-UVNKR98P:~/check$ python3 embed.py
(venv) chinni@LAPTOP-UVNKR98P:~/check$ python3 app.py
```

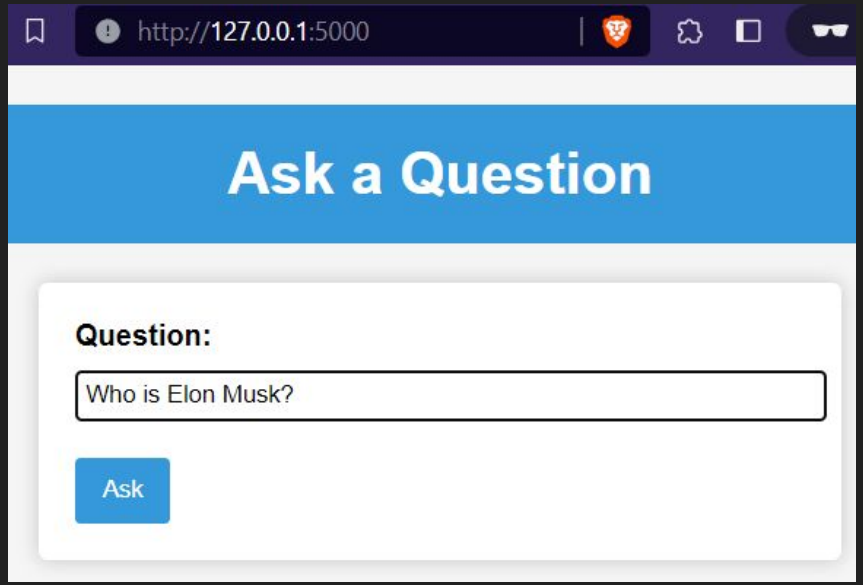
Test (Command Line Based)

I crawled the website www.x.ai

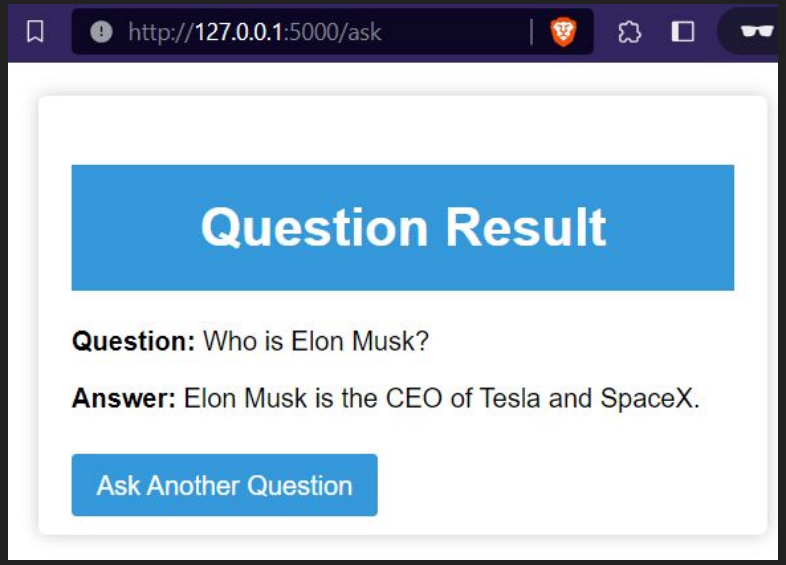
```
(venv) chinni@LAPTOP-UVNKR98P:~/Command based$ python3 app.py
You: Who is the CEO of xAI?
ChatGPT:  Elon Musk
You: What is the salary on Elon Musk?
ChatGPT:  I don't know.
You:
```

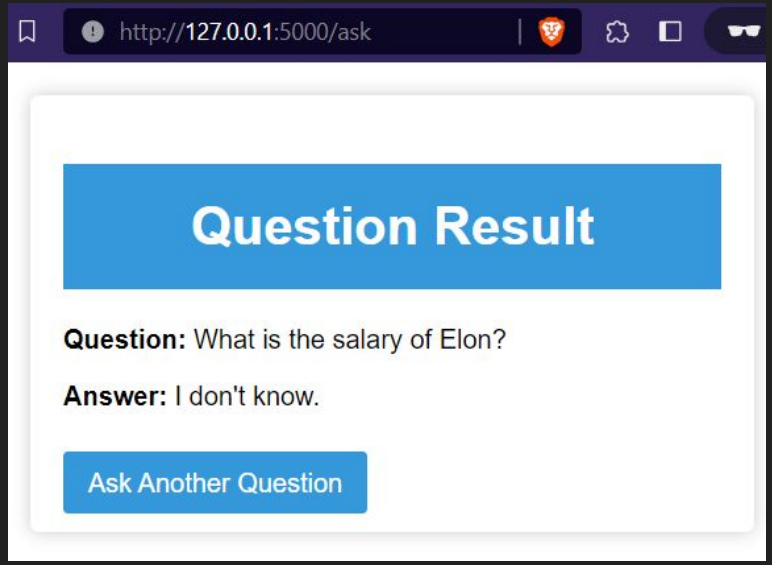
Test (Web Based)

```
 * Serving Flask app 'app' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 133-441-733
```



Test (Web Based)





Enhancement Ideas

- **Multi-Language Support:** Enhance the system to support multiple languages, allowing users to ask questions in various languages and receive answers in the corresponding language.
- **Real-Time Crawling**: Implement real-time web crawling to provide users with the most up-to-date information from webpages.
- **Integration with External APIs:** Integrate with external APIs or databases to provide additional context or information related to the web content.
- **Voice Assistant Integration:** Develop voice command integration, enabling users to ask questions verbally and receive spoken responses.
- **Customizable Embedding Models:** Allow users to choose from a variety of embedding models, including custom-trained models, to improve answer accuracy.

Conclusion

In conclusion, the Customer Support System stands as a testament to the power of innovation in information retrieval. It offers a unique and effective approach to accessing and extracting valuable insights from the vast expanse of web content.

Our project features two distinct interfaces: a command line-based version for those who prefer a straightforward, text-based interaction and a user-friendly web-based version that ensures accessibility to a wider audience. Regardless of the chosen interface, users can seamlessly engage with webpages and obtain accurate answers to their inquiries.

References

- OpenAI
- Professor Chang's Notes and References
- Other