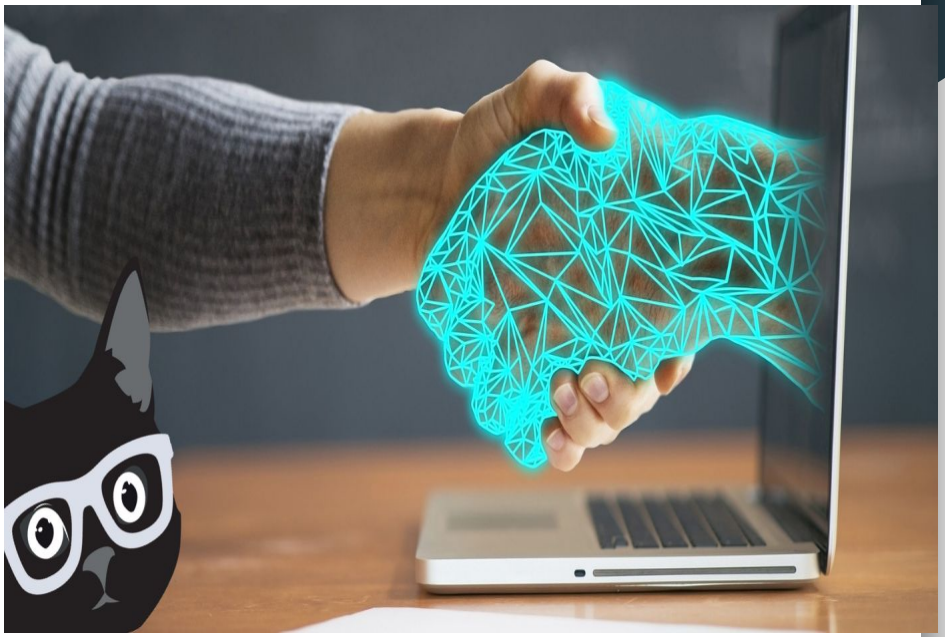
Customer Support System using ChatGPT

By
Tejasri Vaitla
Id: 19749

Table of Contents

Introduction

- This project is designed to create a Customer Support System that utilizes web crawling, text embedding, and the OpenAI API to provide answers to questions based on webpage content. The system follows a structured design to achieve its goals.



Design

The system is designed to perform the following steps:

1. Web Crawling:

It scans webpages, extracts their text content, and stores it for future processing.

2. Text Embedding:

The extracted text is tokenized and transformed into numerical representations using OpenAI's embedding models.

3. Question Answering:

Users can inquire about the crawled webpages, and the system generates responses by employing the embeddings and OpenAI's API.

Implementation

- The project is implemented in two distinct ways: Command Line Based and Web-Based (Flask). These implementations offer different interaction methods for users to access and query web content.

- **Command Line Based:**

  In the command line-based implementation, users interact with the system via 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 offers a straightforward and text-based interface for users to access web content.

- **Web-Based (Python Flask)**

  The web-based implementation utilizes the Flask framework to offer a user-friendly interface for interacting with the system. Users can access the system via a web browser, making it user-friendly and accessible. This web interface enables users to input questions and obtain responses, enhancing the overall user experience and accessibility of the system.

Implementation

Ubuntu Terminal Setup

- Create crawl.py, embed.py, app.py files
- Install Python 3.10's virtual environment package, if not already installed:

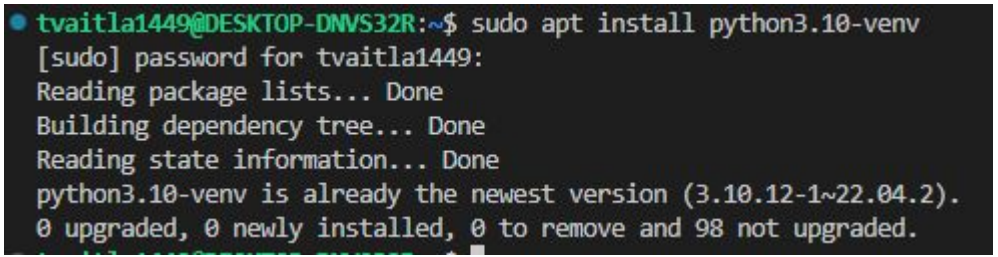
  $ sudo apt install python3.10-venv

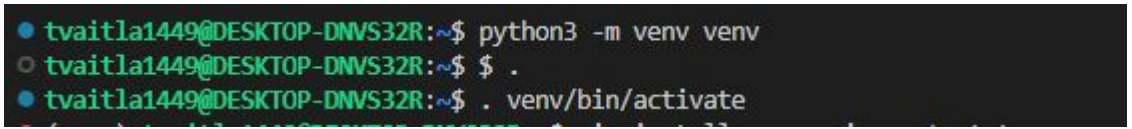
- Create a Python virtual environment named 'venv'

  $ python3 -m venv venv

- Activate the virtual environment

  $. venv/bin/activate





Implementation

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

  $ pip install -r requirements.txt

```
(venv) tvaitla1449@DESKTOP-DNVS32R:~$ 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)
Collecting appnope==0.1.3
  Using cached appnope-0.1.3-py2.py3-none-any.whl (4.4 kB)
Collecting asttokens==2.2.1
```

- To crawl data from webpages, run:

  $ python3 crawl.py

```
(venv) tvaitla1449@DESKTOP-DNVS32R:~$ python3 crawl.py
https://openai.com/
/home/tvaitla1449/crawl.py:163: FutureWarning: The default value of regex will change from True to False in a future version.
  serie = serie.str.replace('\\n', ' ')
```

Implementation

- To embed the crawled data, run:

  $ python3 embed.py

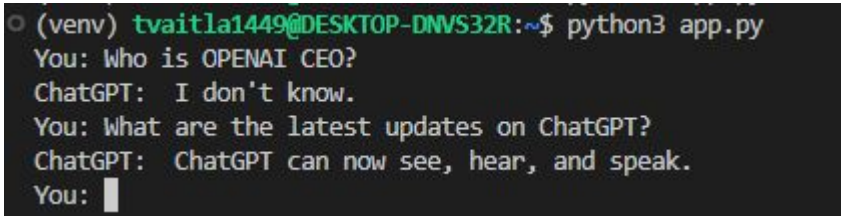
- To run the system, run:

  $ python3 app.py

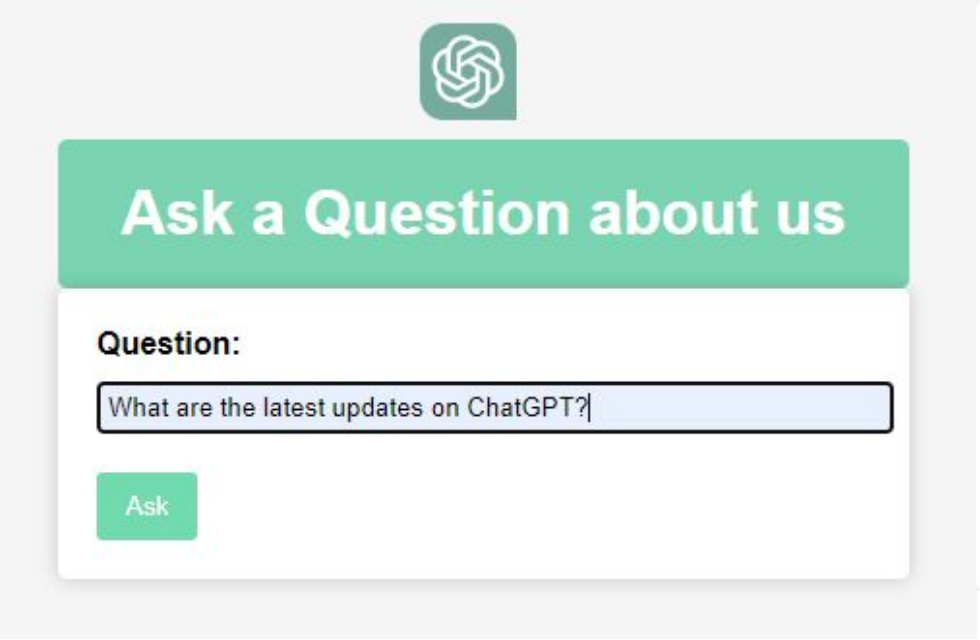
```
(venv) tvaitla1449@DESKTOP-DNVS32R:~$ python3 embed.py
```

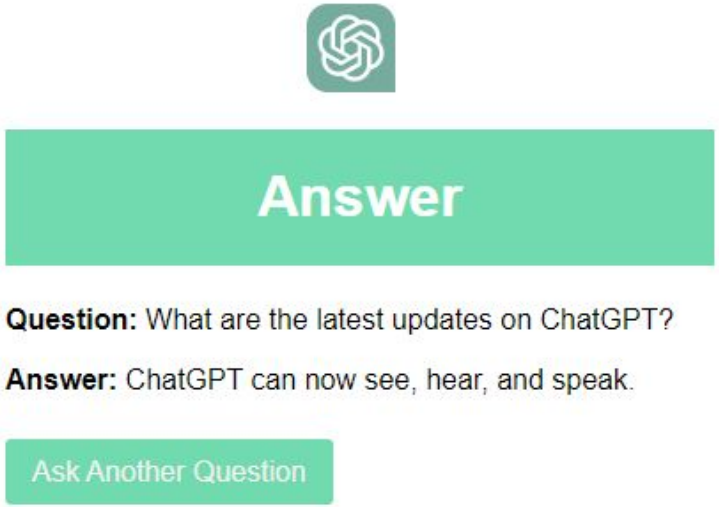
```
(venv) tvaitla1449@DESKTOP-DNVS32R:~$ python3 app.py
You: Who is OPENAI CEO?
ChatGPT:  I don't know.
You: What are the latest updates on ChatGPT?
ChatGPT:  ChatGPT can now see, hear, and speak.
You:
```

Test(Command Based)

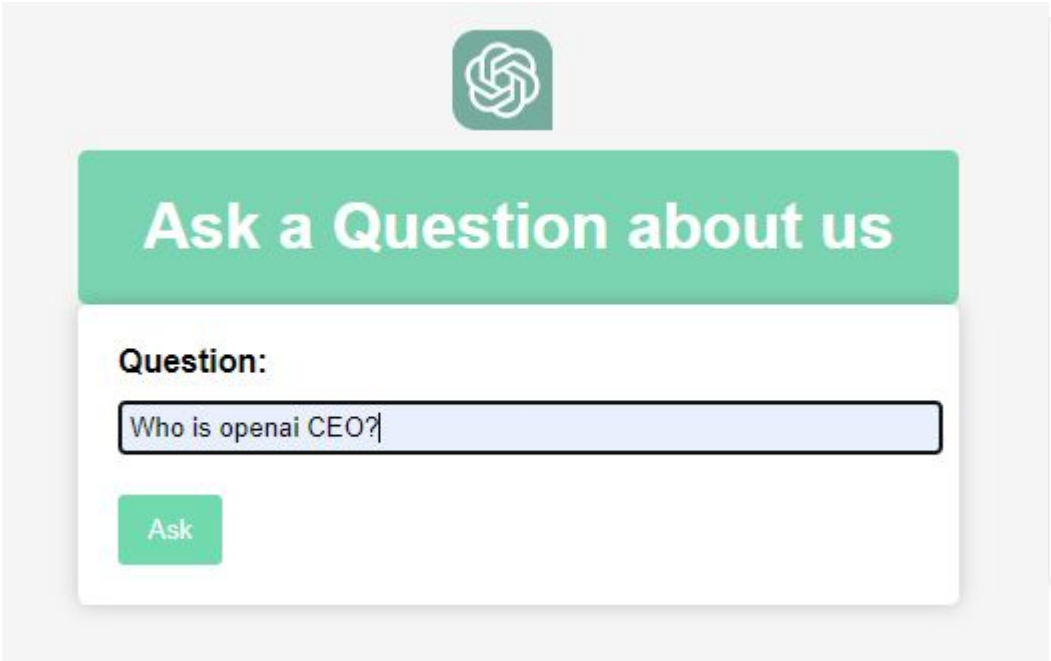


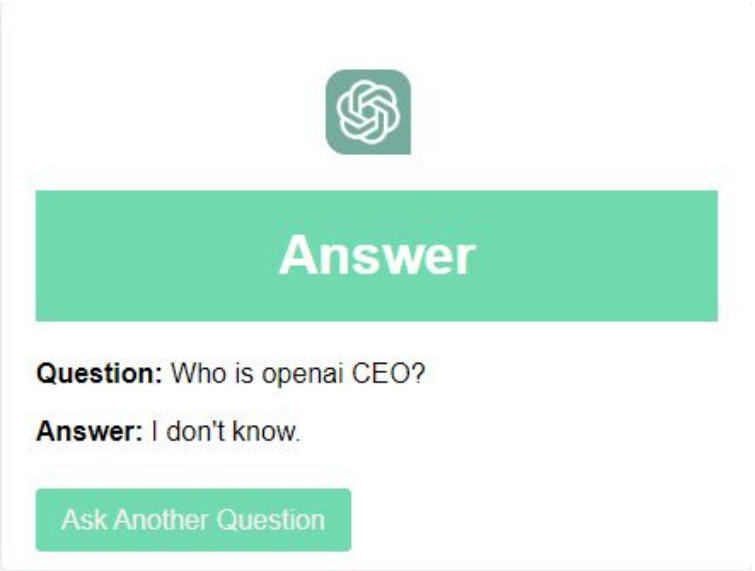
Test(Web Based)





Test(Web Based)





Enhancement Ideas

- Enhance the NLP capabilities of the system to better understand user queries and provide more context-aware responses
- Continuously fine-tune the AI models used for text embedding and question answering to enhance accuracy.
- Integrate a feedback mechanism where users can rate the accuracy and helpfulness of responses. Use this feedback to continuously improve the system.

Conclusion

In conclusion, the Customer Support System using web crawling and text embedding offers a versatile approach to extracting and delivering information from webpages. The two implementations provide flexibility in how users can interact with the system, making it a valuable tool for web-based information retrieval.

References

https://platform.openai.com/docs/tutorials/web-qa-embeddings

https://hc.labnet.sfbu.edu/~henry/sfbu/course/machine_learning/chatgpt/slide/exercise_chatgpt.html