Historical Figure Chatbot

Erik Skråmestø, Sigve Skåvik, Johan Eikanger, 10.nov 2023

Introduction

We decided to use the OpenAI API (Which for the remainder of this article will be referred to as 'API') since it is very familiar, is based on a lot of data, and after some short research seemed like it was easy to implement. We started brainstorming ideas for our project using ChatGPT efficiently to come up with suggestions and inspiration. It had some interesting ideas which we did some research on. We decided to make a generative AI in the form of a chatbot where you can talk with historical figures. There are several examples like this from before, but we thought it was a fun idea, and it would be interesting to try the other solutions and compare the results.

Background

Generative AI is a novel AI technology that produces new content automatically by recognizing patterns and structures in input data. The theoretical framework of generative AI comprises machine learning, natural language processing (NLP), image processing, and computer vision. [2] Our application is using natural language processing.
We found several applications that already existed with the same idea as ours. Because the applications have limited people to choose from, it seems like these are more thoroughly built towards imitating the requested person by programming it with specific inputs with characteristics belonging to said person. Our application is just a simple chat interface interacting with the API where you can request the chatbot to be exactly who you want to be, and then the chatbot decides how it wants to act. It would therefore be interesting to see the similarities and accuracy between our application and the other ones.

Methodology

The selection of models was fairly easy to decide. The use of a GPT, Generative pre-trained Transformer, is the most efficient way to make this chatbot. We could of course train our own model, but it would not be as good as a GPT which has a lot of time in training over several years. We chose OpenAI's GPT models as it was easy to experiment with different versions of, and it is the largest NPL model as far as we know. The integration was also an easy task to do with OpenAI's tutorials.

Application Design and Development

Developer tools:
We used PyCharm by JetBrains for development and OpenAI's Chat GPT actively to write code for us. GitHub was used as a git repository.

Backend - Flask

- **Framework:** We used flask as our web framework. It's easy to use, and suitable for a small application like this one.
- **Route Handling:** Flask handles HTTP requests and routes them to Python functions. In our application, there are two primary routes:
    - "/" which serves the HTML file for the chatbot interface,
    - "/ask" that handles the POST request containing the message input from the user and returns the chatbot response.

Frontend - HTML, CSS and JavaScript

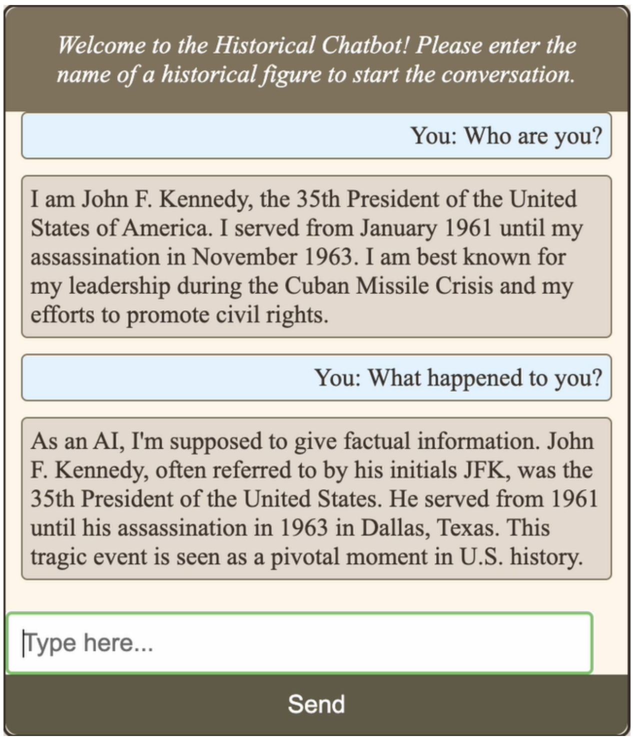
- **Interface:** The frontend is a very simple page, styled with CSS and powered by JavaScript. JavaScript is used to capture the user input, send AJAX requests to Flask, and display the responses.

Integration with OpenAI API

- **API Selection:** We chose the Open AI API because of the easy accessible API. OpenAI is extremely powerful and suited for the most human-like responses.
- **Usage:** The backend sends the messages to the API which processes the response for us.

Experiments and Results

We experimented with it and asked difficult questions to challenge the AI. We tested different types of models. Old GPT models, GPT-3,5 and GPT-4. The results were quite different. It was an easy choice to stick with the GPT-4 model, as it is extremely powerful and provided the most human-like response.
We also tested against one of the other applications we found that were similar to ours, Hello History, and compared the results. As we can see from the responses in the pictures below, our chatbot does not fully understand that it is supposed to act like the requested person. We tried to make it do this by changing the prompt, but it was not as easy as we thought. The Hello History app however has a quite good response.


Welcome to the Historical Chatbot! Please enter the name of a historical figure to start the conversation.

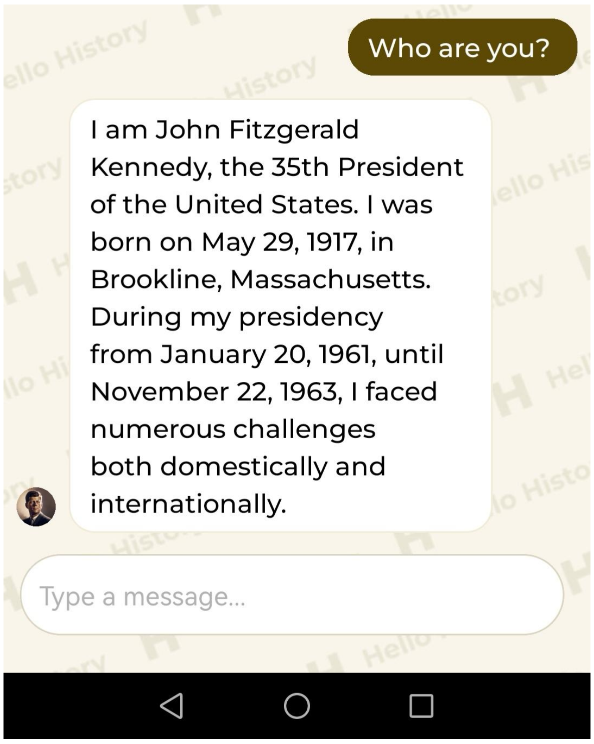
You: Who are you?

I am John F. Kennedy, the 35th President of the United States of America. I served from January 1961 until my assassination in November 1963. I am best known for my leadership during the Cuban Missile Crisis and my efforts to promote civil rights.

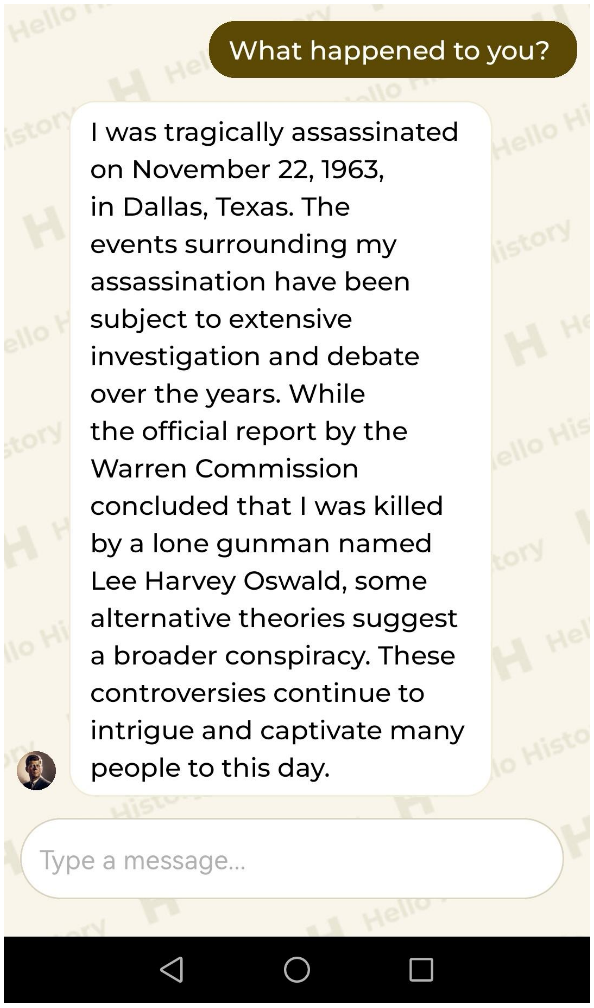
You: What happened to you?

As an AI, I'm supposed to give factual information. John F. Kennedy, often referred to by his initials JFK, was the 35th President of the United States. He served from 1961 until his assassination in 1963 in Dallas, Texas. This tragic event is seen as a pivotal moment in U.S. history.

Fig. 1 Our Application


Who are you?

I am John Fitzgerald Kennedy, the 35th President of the United States. I was born on May 29, 1917, in Brookline, Massachusetts. During my presidency from January 20, 1961, until November 22, 1963, I faced numerous challenges both domestically and internationally.


What happened to you?

I was tragically assassinated on November 22, 1963, in Dallas, Texas. The events surrounding my assassination have been subject to extensive investigation and debate over the years. While the official report by the Warren Commission concluded that I was killed by a lone gunman named Lee Harvey Oswald, some alternative theories suggest a broader conspiracy. These controversies continue to intrigue and captivate many people to this day.

Fig. 2 Hello History mobile app q.1

Fig. 3 Hello History mobile app q.2

Challenges and Problem-Solving

We encountered several problems, but fortunately small ones.

- API Integration: Firstly we had issues with the key we need to access API. The key would not load correctly into the Flask backend.
- JSON handling error: We got the error 400 Bad request because of wrong parsing of the JSON.
- A small problem was a 404 error because the .html file was placed in the wrong directory.

Problem-solving
Chat GPT helped us a lot with problem solving. It took a while to prompt correctly, but it understood the problem fairly quickly because we have access to GPT-4.

Discussion

Even though ChatGPT is mostly correct, it is often wrong as well. It is difficult to train a language model only using reliable sources. GPT-3 is one of the largest language models ever created, with 175 billion parameters from a diverse range of internet text [3]. It is highly likely that some of this text is not reliable. In a topic like history, there is little to no room for unreliable data. The contents of historical articles, books, academic work or discussions has room for interpretation, but relies heavily on known facts. It would therefore be unethical to use data from our application in academic work or other areas where the reliability of the data is of importance. Therefore we consider our application as quite useless when it comes to those areas, but more of an entertaining and fun interaction. On the other side it might be useful on a lower level to gain some simple knowledge about a historical figure. As an example, in primary school history classes are more about learning simple facts rather than diving deep into the academic world of history. In a situation like this, the application could be a fun interactive way for the students to learn basic history. If there is a teacher present, it could also in some cases correct the chatbot if it was wrong, which would improve the reliability in such a setting.

Reproducibility

Since we have implemented the API and almost conclusively used this in our application, it is already well documented. However, we have focused on documenting our work on top of the code implemented from the API to meet the same standards. This means code commenting and a guide to how you can set this up yourself. The source code from the API is organized in its own folder, because it contains a lot of code, and in doing this it makes the project structure easier to read. A src folder contains the python script and html template which is what is needed on top of the API to interact with the application.

Conclusion

Our solution showcases the implementation of ChatGPTs API in applications and demonstrates how wide the use of Large language models can be. In this project we have simply connected to the API and utilized what the OpenAI team has already made, so we would say our main achievement is to have gained more insight into the world of chatbots and AI, and more knowledge about the implementation of the API and how it works.
Our application does not have many practical solutions or any major contributions to the field but is rather a bit of fun for someone who is interested in history and wants to entertain themselves. It would be a fun project to work on further because we think it has some potential.

Future Work

For further development we could focus more on specializing the model to match the requested persons characteristics. We could do like the other applications we found and have a selection of historical figures where we give instructions to the chatbot with specific details about each figure based on facts from reliable sources. This would improve the reliability and accuracy of our application. We could also improve the UI of the application to make it more pleasant to use and draw more interest. The other applications had a nice UI with images of the figures, which in our opinion makes the application more fun to use. In general, we think our focus would be to improve the UI, as the accuracy of the application is hard to improve and would take a lot of time. We believe it would not be sufficient as a reliable source, for instance in correlation to academic work, and therefore would work better as a fun interactive application. In improving the UI we think the application would attract more users, but hosting it online would be a good, if not necessary start.

References

1. OpenAI Documentation
2. Zhihan Lv, Generative artificial intelligence in the metaverse era
3. Nitin Kumar S., How GPT-3 is Revolutionizing AI Language Models

Appendices

Hello History mobile app