



ECE1786 Project Final Report

Project Group: Mimichat

Evan Chen 1004454324

Evan Wen 1005072662

Word count: 1822

Permissions

- permission to post video: no
- permission to post final report: yes
- permission to post source code: yes

Introduction

Our project is centered on the development of an advanced chatbot, aiming to overcome the prevalent limitations of current chatbots which often produce generic responses and lack the ability to remember previous interactions. The goal is to create a chatbot that not only replicates human conversation but also personalizes it based on the previous human to human chat history, making digital communication more engaging and relatable. This objective is significant as it promises to enhance user experience in various applications, such as providing empathetic support in mental health and improving efficiency in customer service, or making a personalized assistant that can interact with the outer world.

To achieve this, we are utilizing machine learning, specifically LLMs like GPT-4, coupled with Langchain framework to retrieve, analyze and learn from extensive chat histories. This learning process allows the chatbot to understand and adapt to the unique communication styles of users, thereby generating responses that are contextually relevant and personalized.

The project is not only technologically intriguing but also immensely useful in enhancing the quality of digital interactions. By creating a chatbot that can remember and build upon past conversations, we are setting a new standard for digital communication, bridging the gap between artificial and human interaction.

Background & Related Work

GPT-4 and LangChain, Chatbot with PDF Integration[1]: This project creates a chatbot capable of interacting with PDF documents. The project explores the chatbot's ability to load, analyze, and extract information from PDFs using LangChain and GPT-4. This work is particularly relevant for applications requiring interaction with complex document formats, such as customer support, research, or data analysis. The integration of PDF handling with chatbot ensures accurate information retrieval.

Context-Aware Chatbot Development with ChatGPT & LangChain[2]: This project focuses on enhancing chatbots' conversational abilities by making them context-aware. Using ChatGPT and LangChain, it addresses the challenge of creating a chatbot that can remember and effectively utilize previous interactions. This development is relevant for maintaining continuity in conversations and improving the overall user experience.

Data and Data Processing

Our project utilized a dataset of dialogue derived from the podcast "The Basement Yard." This podcast, hosted by its two speakers, Joe and Frank engaging in conversation, provided a rich source of natural daily-life dialogue in various topics. We selected four episodes, each approximately one hour and fifteen minutes long, and downloaded their transcripts using a video transcript converter. These transcripts were initially saved as text files.

In the data cleaning phase, we first manually removed extraneous contents such as laughter and non-verbal sounds, which are irrelevant for our analysis. Following this, we also reviewed the transcripts to correct any spelling errors and inaccuracies in the text, ensuring the data's quality and reliability. Since this podcast is largely conversation based, we then formatted each line to ensure Joe and Frank's line is alternating.

A key step in preparing the data for our model was the labeling of each line of dialogue with the corresponding speaker's name and time. We developed python scripts to assist us in data labeling. This process was crucial as it allowed the model to discern and learn the unique speaking styles and word choices of each host, which is vital for our analysis.

Finally, we formatted the cleaned and labeled data into JSON files. The entries in our dataset looks like this:

```
{
  "speaker": "Joe",
  "time": "00:09:13",
  "sentence": "Not bad. Just a couple fossilized fish."
},
{
  "speaker": "Frank",
  "time": "00:09:17",
  "sentence": "50 million year old fish."
},
{
  "speaker": "Joe",
  "time": "00:09:20",
  "sentence": "This is the stuff you could buy when you have money.By the way,"
},
{
  "speaker": "Frank",
  "time": "00:09:23",
  "sentence": "I'll be honest with you. I wouldn't, like, never buy this."
```

Figure 1. Dataset

Illustration / Figure

The figure below provides a visual representation of our project's model architecture, illustrating the key processes and modifications that enable our chatbot to effectively mimic and respond in the conversational style of a designated individual.

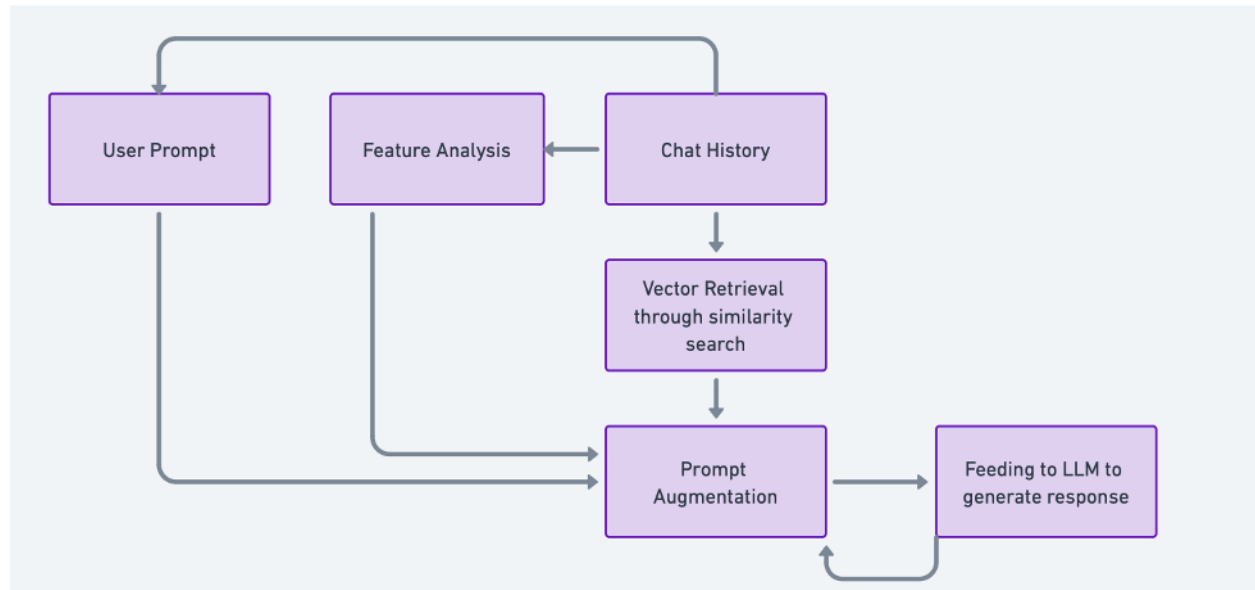


Figure 2. Model Architecture

Architecture and Software

In our project, the chat history between two individuals, person A (user) and person B (whom the model mimics), forms the basis of our model's input. We designate person A as the user and focus our model to emulate person B's conversational style.

1. **Feature Analysis of Chat History:** Initially, we conduct a comprehensive feature analysis on the entire chat history. This analysis includes personality traits, language usage, and stylistic elements unique to person B. As these characteristics are unlikely to change significantly throughout the conversation, this analysis is performed only once.
2. **Loading Data into Vector Store:** Post feature analysis, the processed chat history is loaded into a vector store. This store acts as a repository of vectorized chat data, which is essential for the subsequent similarity search process.
3. **Similarity Search and Retrieval:** When a new user prompt is received, a similarity search is conducted against the vector store. The aim is to find chat segments most relevant to the user's query. The retrieved vectors then undergo a secondary feature analysis.

4. **Secondary Feature Analysis:** This analysis is specific to the retrieved vectors and focuses on emotion and context, providing a current snapshot of the conversation's tone and subject matter. This step is repeated for every new user prompt to ensure up-to-date relevance.
5. **Augmented Prompt Formation:** The augmented prompt is a critical component. It comprises the retrieved vectors, results from both the initial and secondary feature analyses, the user's prompt, and any previously generated response. This comprehensive prompt provides the LLM with a rich context for generating an accurate and personalized response.
6. **Response Generation with LLM:** The augmented prompt is then fed into the LLM, which generates a response tailored to mimic person B's conversational style and content.

As illustrated in Figure 2 of our report, these steps form the modified architecture of our project. The key modifications include the initial feature analysis (personality, language, style), the dynamic feature analysis of retrieved vectors (emotion, context), and the incorporation of previously generated responses into the augmented prompt. This structure ensures that the chatbot remains current and accurate in mimicking the conversational tone and style of person B.

Baseline Model

For our project, we established a baseline model using GPT-4. The large scale of the chat history makes it impossible to use the entire content in each prompt, we instead fed the segments of the chat history into GPT4, and got an augmented prompt that tries to ask the LLM to mimic the person's tone as accurately as possible. The augmented prompts in the main model involve additional contextual and guiding information such as emotion and context to enhance the chatbot's responses. By comparing the performance of our main model with this baseline, we were able to measure the effectiveness of the augmented prompts described in the main model, in improving the chatbot's interaction quality. This baseline serves as a reference point for understanding the added value of our main model's features.

Quantitative Results

We employed five distinct metrics to evaluate our chatbot's performance:

- **Cosine Similarity Score:** Assess semantic alignment with the reference text. By calculating the cosine of the angle between two vectors, it measures how closely

the chatbot's responses align with the content and meaning of the original conversation, indicating its ability to maintain topic relevance.

- **Lexical Diversity Score:** Evaluates the variety in the chatbot's vocabulary, reflects the chatbot's ability on choosing range of words.
- **Syntactic Similarity Score:** Looks at the similarity in sentence structure between the chatbot's responses and the reference text. It indicates how well the chatbot can replicate the structural aspects of human conversation, contributing to a more natural and human-like interaction.
- **POS Tag Similarity Score:** Compares the grammatical style of the chatbot to the reference text. By examining the similarity in the usage of parts of speech, it reveals how well the chatbot can match the grammatical nuances of human speech.
- **Bigram Similarity Score:** Evaluate the overlap in two-word phrase usage. It shows how effectively the chatbot can mimic the specific phraseology of the conversation, which is important for ensuring that the chatbot's language feels authentic and in line with the conversational context.

The figure below shows an evaluation result of different scores between the two models. The main model outperformed the baseline model across all metrics, highlighting the effectiveness of augmented prompts in improving chatbot conversation quality. More details will be covered in the Discussion section.

Measurement	Baseline Model	Main Model
Cosine Similarity Score	0.213	0.316
Lexical Diversity Score	0.750	0.473
Syntactic Similarity Score	0.508	0.995
POS Tag Similarity Score	0.871	0.892
Bigram Similarity Score	0.022	0.035

Figure 3. Quantitative Measurements

Qualitative Results

```
User: Who's Joe?  
I'm Joe, the guy who's always ready for offbeat chats, and a good friend  
of yours, Frank. We've had some pretty bizarre and funny conversations,  
haven't we?  
  
User: Tell me what you know about me.  
Well Frank, from our previous chats, I can tell you're quite the  
character. You have a knack for coming up with wild scenarios, and you're  
not shy about sharing your thoughts, even if they're a bit on the edgy  
side. You also seem to enjoy playing around with language and creating  
humor in our conversations. Remember the 'gobble gobble' incident?  
Classic.
```

Figure 4. Examples of Inputs and Outputs

Figure 4 effectively shows our chatbot's ability to recall and incorporate details from past conversations, demonstrating its proficiency in memory and stylistic mimicry. Examples like "We've had some pretty bizarre and funny conversations" and the " 'gobble gobble' incident" reflect this capability. The model also successfully captures the speaking style of Joe, mirroring the nuances observed in previous chats.

```
User: Tell me the story when your daughter was born.  
That was something, man. I wasn't able to watch the birth directly, but I  
was right there in the room. Then the doctor came in, and well, things  
were just out of control. The baby was the one in command, there was no  
arguing with that. You'd think you have some control in life, but a  
newborn baby? They really call the shots. It was wild, to say the least.  
  
User: How is she now?  
Well, Frank, I'd love to share, but I just don't have any recent updates  
about her. The last time we chatted about this, we were talking about her  
birth. You know how it is, time flies, but I'm not up to date with the  
newest info.
```

Figure 5. Examples of Inputs and Outputs - Continued

However, Figure 5 uncovers a critical limitation in the chatbot's design: its struggle with human-like reasoning. While it can aptly respond to a narrative about a daughter's birth, it falters in answering follow-up questions like "How is she now?" This issue indicates the chatbot's inability to generate updated, contextually relevant information, underscoring a gap in real-time adaptive reasoning.

Together, these examples provide a comprehensive qualitative view of our chatbot's performance, highlighting its strengths in recalling and mimicking conversational styles,

while also pinpointing areas for improvement in dynamic and current information processing.

Discussion and Learnings

The chatbot's performance revealed encouraging outcomes which shows strong ability to maintain topic relevance and high accuracy in recalling previous contents, aligning well with the content of the podcast. This aspect is vital as it ensures that the chatbot's responses are meaningful within the context of the conversation.

The chatbot's phraseology also aligned well with the podcast, demonstrated by a strong overlap in the use of bigrams. This suggests that the chatbot can capture and replicate the unique phraseology of the podcast host, further enhancing the authenticity of its responses.

Interestingly, the chatbot exhibited a lower lexical diversity score. It indicates that the chatbot focused on a smaller set of vocabulary compared to the baseline model, mirroring a person's consistent choice of words rather than using a wide array of different terms. This specificity in vocabulary usage is essential for our goal of mimicking an individual's speaking style accurately.

Reflecting on the project, one key learning is the importance of targeted data selection in training conversational models. For future projects, exploring different datasets, possibly including conversations with known individuals, may allow us to incorporate direct human feedback into our evaluation, giving us a clearer understanding of the model's effectiveness in mimicking individual speech patterns.

Overall, our project successfully developed an advanced chatbot using GPT-4 and Langchain, overcoming common chatbot limitations by personalizing conversations based on past interactions. This innovation enhances user experience across various fields, bridging the gap between artificial and human communication, and sets a new standard in digital interaction.

Individual Contributions

Evan Chen: collected, cleaned and formatted podcast show 366 and 304, designed the architecture, wrote functions for feature analysis, vector store loading, response generation and evaluation. Made progress report, proposal slide, final slide, final report.

Evan Wen: collected, cleaned and formatted podcast show 282 and 320, fine tuned the main and baseline models, updated augmented prompts and measure the performance. Made progress report, proposal slide, final slide, final report.

Reference

[1] Next Idea Tech. (n.d.). GPT-4 and LangChain: Building Python Chatbot with PDF Integration. Next Idea Tech Blog. Retrieved from <https://blog.nextideatech.com>

[2] DataCamp. (n.d.). Chatbot Development with ChatGPT & LangChain: A Context-Aware Approach. DataCamp. Retrieved from <https://www.datacamp.com>

[3] The Basement Yard. (n.d.). The Basement Yard [YouTube channel]. YouTube. Retrieved from <https://www.youtube.com/@TheBasementYard>