

[AI Lang Master - Github](#)

Team Members: Tejas Pinjala, Thanh Nguyen, Faizul Anis

CS 6320.002 Fall 2024

Instructor: Tatiana Erekhinskaya

12/01/2024

Abstract

The purpose of this project is to implement a conversational AI that helps users learn and practice a new language. Compared to traditional language learning applications with predefined lessons, we aim to develop an AI that offers natural, simulated conversations to improve the language learning experience through practice. By integrating GPT models with prompt engineering, a user-friendly web application was developed with features including real-time grammar, phrases, and vocabulary feedback. The challenges, such as off-topic responses and time-consuming model training, were mitigated through targeted prompt tuning and fine-tuning pre-built models.

Table of Contents

1. [Abstract](#)
2. [Table of Contents](#)
3. [Introduction](#)
4. [Contributions](#)
5. [Related Work](#)
6. [Scope and Methodology](#)
7. [Challenges and Solutions](#)
8. [Experiments and Evaluations](#)
9. [Lessons Learned](#)
10. [Future Work](#)
11. [Conclusion](#)
12. [References](#)

Introduction

Our big goal is to create a conversational AI that helps users practice and learn languages through simulated conversations focusing on speaking and context-based learning. Traditional language learning platforms rely heavily on predefined lessons, limiting flexibility and lacking natural conversational interaction. Our AI addresses this gap by simulating a conversation-like lesson to learn a language, as immersion is proven to be the most effective language-learning method. We aim to provide a user-friendly environment for multilingual practice and support various languages with instant feedback on grammar and vocabulary on various topics.

Contributions

Tejas Pinjala:

- Implemented interactive chat functionalities, enhancing user engagement and real-time communication within the application.
- Applied prompt engineering techniques and fine-tuned various AI models to improve accuracy, responsiveness, and overall performance.
- Developed and curated comprehensive datasets to support machine learning model training, ensuring data quality.

Thanh Nguyen:

- Led the integration of React.js, Django, and Docker to build a full-stack web application.
- Implemented GPT-based features to enhance user interactions.
- Designed the chatbot workflow and robust API endpoints to ensure seamless frontend-backend communication.

Faizul Anis:

- Conducted comprehensive debugging and performed quality assurance of AI models.
- Facilitated the transition of the full-stack application model to support both Windows and Mac operating systems.
- Designed participant surveys, analyzed feedback, and developed detailed documentation to support project workflows and insights.

Related Work

Great heights in language learning have been achieved in relation to technological integration, especially in Artificial Intelligence and Natural Language Processing. Some well-known applications like Duolingo, Babbel, and Rosetta Stone take an organized structure in lessons that explain and teach languages. These tools have contributed to the success of language learning technology, entailing several areas for improvement in bringing natural conversational abilities into our modern days.

Emerging Conversational AI Systems:

Recent developments in Conversational AI have sought to overcome the limitations that traditional language learning applications present. These systems use various existing LLMs, like GPT, BERT, and others, to generate responses similar to those made by humans. Even though showing promising potential, these AI models have yet to be widely adopted into mainstream language-learning platforms.

1. **LLMs for Education:** GPT-3 and GPT-4 models are remarkable examples that have performed impressively in generating coherent and contextually relevant text. Their applications in educational tools have been explored; however, these applications often need more customization for language learning, such as real-time grammatical feedback and vocabulary enhancement.
2. **Language-Specific Adaptations:** Most conversational AI systems are designed for general-purpose interactions. Adapting these models for language-specific nuances, such as idiomatic expressions and cultural contexts, remains an ongoing challenge.

3. **Commercial Implementations:** Products like Google's AI assistant and OpenAI's ChatGPT offer conversational capabilities but are not specifically optimized for structured language learning or educational contexts.

Research in Language Learning and AI:

1. **Conversational Agents in Language Learning:** Various conversational agents have been found to increase the intensity of language learning through better interactive practice. However, their effectiveness is often limited to the degree of reliance on pre-scripted dialogues rather than generative models. Studies show learners benefit most from dynamic systems adapting to users, simulating natural human conversations.
2. **Fine-Tuning Pre-built Models:** Studies investigate the process of fine-tuning a large pre-trained LM for different tasks, including language refinement. During their fine-tuning process, models learn the application of grammar rules and vocabulary specific to a particular target language. For instance, several pilot studies have used fine-tuned GPT models for grammar correction and vocabulary enrichment with great success, improving user engagement and learning results.
3. **Dynamic Conversation Challenges:** Handling off-topic responses is one of the main challenges in building conversational AI for language learning. Research highlights the importance of prompt engineering and training classifiers to keep the focus on the context of conversations. On the other hand, some error analyses reveal the inability to capture regional accents and cultural references, such that these systems need further refinement.

Scope and Methodology

This project builds on the advancements and limitations of the existing tools by moving beyond the predefined lessons to simulate free-form, real-world conversations. It leverages fine-tuned GPT models to generate contextually relevant and grammatically accurate responses with a feedback mechanism for any corrections in grammar or vocabulary. To reduce the off-topic responses, we have used prompt engineering techniques to make sure that the conversation remains educational and engaging.

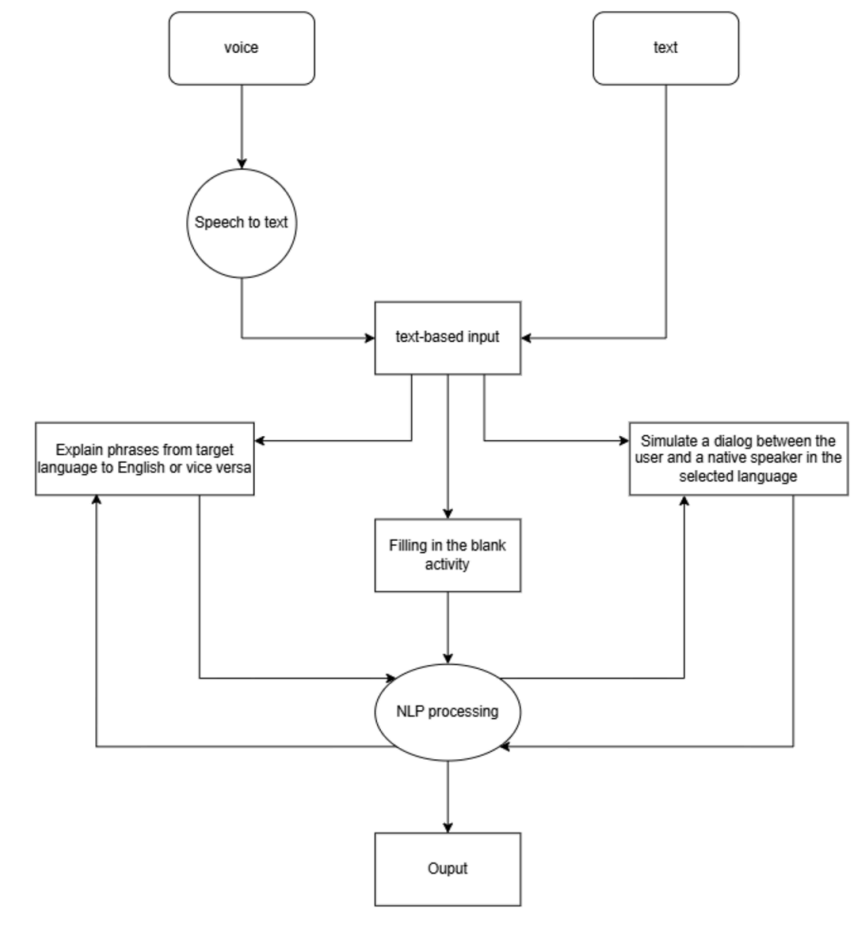


Figure 1: Task Flow Diagram

We have designed a full-fledged web application to introduce users to conversational language learning by bridging the gap between traditional lesson-based language learning

methods and immersive conversational practices using the most recent NLP techniques. First, we developed the full-stack web application infrastructure, which includes a friendly Chat UI with the possibility of text and voice input on the front-end. On the back-end, it will be powered by the Django framework for processing input from the front-end, while interacting with AI models that build up responses. All app components are containerized in Docker to build good communication between server and client.

Going further to the core of the back-end, this is a development toward an AI capable of eliciting better responses. The GPT-based model should render coherent, contextually relevant outputs that are to be advanced by refined prompt engineering techniques, cueing the AI in generating responses that will achieve such goals of the projects, whether these are grammatically correct and vocabulary-enhancing kinds of feedback. It determines the backend mechanisms needed to analyze user input, process the specifics of the language element, and keep the AI on conversational topics relevant to language learning.

Complementing these backend enhancements are refinements in usability and aesthetic appeal on the front-end of the Chat UI. The design of UI functionalities makes the interaction more intuitive, and styles will be fitted to increase the user experience. The Chat UI will handle text and voice inputs with equal ease, affording users flexibility in how they want to engage with the application.

Hence, the project proceeds with discussing the development of the AI model using an ensemble-based approach. The primary approach consisted of fine-tuning prebuilt language models like GPTs, strengthening the system in the multi-model approach to absorb all the different facets: conversational AI, tokenization, recognizing entities, reasoning processes, and response generation. While fine-tuning uses pre-trained knowledge for efficiency and

adaptability, prompt engineering may limit responses off-topic and guarantee conversational relevance.

Even resource research has featured a lot in collecting this dataset to make such datasets suitable for conversational language learning. Further development is based on day-to-day life and academic topics so that conversations fall within the bounds of practical applications. Iterative testing and error analysis in refining performance are also part of this project. Testing was done internally amongst members and also externally with seven users in order to get a variety of responses. The outdoor testing exercise helps identify the existing challenges, while the efficacy of AI in making language-appropriate responses has been established.

Challenges and Solutions

This project faced many different challenges during its life cycle, each requiring a different kind of solution so that the group could move on to the result. Choosing the proper AI model, refining the answers, and integrating the system components were the few challenges that defined the development process and honed the end result.

The first challenge was selecting an appropriate AI model for language learning conversationally. Training this model from scratch would require lots of computational resources and, therefore, is extremely time-consuming. Different types of pre-built models, including open-sourced ones, were tested based on GPT. The results were much better: meaningful, contextually appropriate, and grammatically correct. The GPT 4o Mini model had better overall contextual and grammatically correct responses and would, therefore, be much more cost-effective than many other options. This was chosen as a source to start fine-tuning, which the team would adapt to this particular use case of the project. This fine-tuning had to be done in that the model needed adaptation to the conversational setting of language learning, which it did not have from the pre-knowledge encoded in GPT 4o Mini, while drastically reducing resource and development time.

Another major challenge we faced was handling off-topic or irrelevant responses from the AI. By default, conversational AI systems are very dependent on context and struggle with ambiguous or unexpected user inputs; therefore, the team used some quite advanced techniques of prompt engineering. In this iteration and crafting of input prompts, this model got the responses aligned with the goals of this project. Thus, personalized and customized prompts directed the AI to stay on track and generate topic-relevant, educationally adequate results.

Prompt engineering has become one of the main strategies for reducing irrelevant responses and adding to the reliability of AI in interactions.

Another challenge lies in assessing the quality of the AI responses. Unlike structured tasks, which have well-defined metrics, the accuracy of a conversation is fundamentally subjective, depending on context. Here, the team needed to provide clear guidelines for evaluating responses against grammatical accuracy, choice of words, and contextual relevance. These criteria have been applied through various test cycles and by incorporating external user feedback for further refinements.

Preparation of the data was also a challenge since we could not find the proper data according to our purposes to fine-tune our model. We solved this problem by creating and processing our own dataset, focusing only on practical examples of conversational speech concerning everyday life and academic purposes. This was done in an effort to make sure that the training data truly reflected the kind of interactions users would go through in real life, further improving the abilities of the AI to create meaningful responses. Finally, the integration of the front-end Chat UI, back-end APIs, and AI models into one system added a lot of technical complexity. Such integration called for thorough planning and iterative testing to make the components communicate smoothly. Some Docker containers were used here to create a consistent development and deployment environment that prevented possible bottlenecks in integration. It allowed not just smooth communication between the front and back ends but also provided a scaling ability for the same or any future enhancements.

We had one last issue that had taken a lot of time to fix. It was giving a WebSocket 1011 error when we tried to run it for the first time. The chatbot would give an error to every message sent because the web sockets were closed. So we had to double-check and reinstall the program

to see if it ran again, but it just ended up having the same issue. Once we realized that another member's code was not also working, let it be an update issue. As OpenAI had recently updated the API calls, we would have to make sure we were on the latest version, and that had finally fixed the error.

Each of these challenges allowed for refinement in one way or another to build upon the final product of a more robust and effective language learning platform. Solutions-not just in model choices and tuning but in how the prompts were engineered and integrations were made-meant the project reached its objectives regarding accuracy, usability, and scalability.

Experiments and Evaluations

In the experiment, we had gathered 7 participants and ran a study on how effective the AI language tutor really is. We tested the following areas: overall experience, effectiveness, feedback on mistakes, ability to adjust to learning pace, and improvements we could make. In testing, we targeted English learning Spanish over other languages because it was easier to decipher. All 7 participants have various experiences with Spanish and have taken results from.

The preparation phase involved ensuring the application was fully made and all the components, such as the front-end chat UI and the back-end system that uses Django, ran perfectly. It also involved making sure the Docker containers were properly set up and ready for participants to interact with the website. The primary objective was to watch how the model learned and how well we had trained it to provide contextually relevant results.

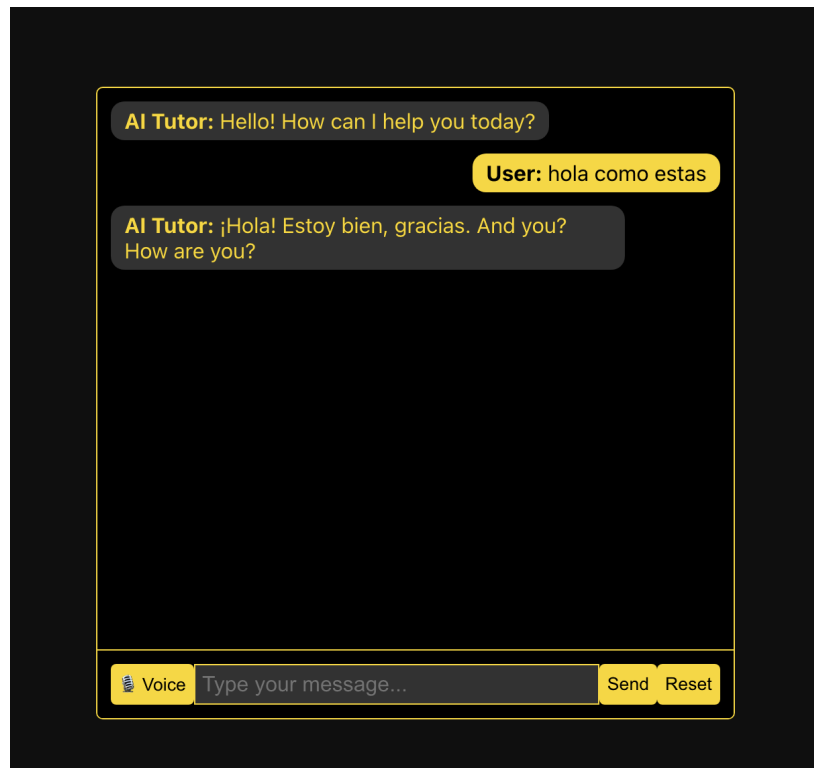


Figure 2: AI Tutor Webpage showing User Interaction

In the testing phase, the participants interacted with the AI for 5-10 minutes. They were encouraged to talk to the bot like any other tutor or Spanish teacher to facilitate a good user relationship further. Using the microphone, the user can click the icon, speak, and send speech data to be sent to the model. The data will then be analyzed, and a narrated response will be given based on the information. After interacting with the chatbot, they are asked to complete a post-evaluation survey.

Data collection is an important step of the process, as the qualitative and quantitative feedback is recorded. First, we asked the participants about their overall experience out of 10 to see if the chatbot was working as intended. Next is how effective the chatbot is in helping to improve your skills, showing if the chatbot is good at its job. If the chatbot met the user's expectations, how well did it respond to mistakes, and how easy was it to interact with the chatbot? Was the chatbot able to adjust to the user's learning pace or not? Lastly, some qualitative questions we asked were the improvements or additional features we should add and comments on the whole experiment.

Ratings from different candidates

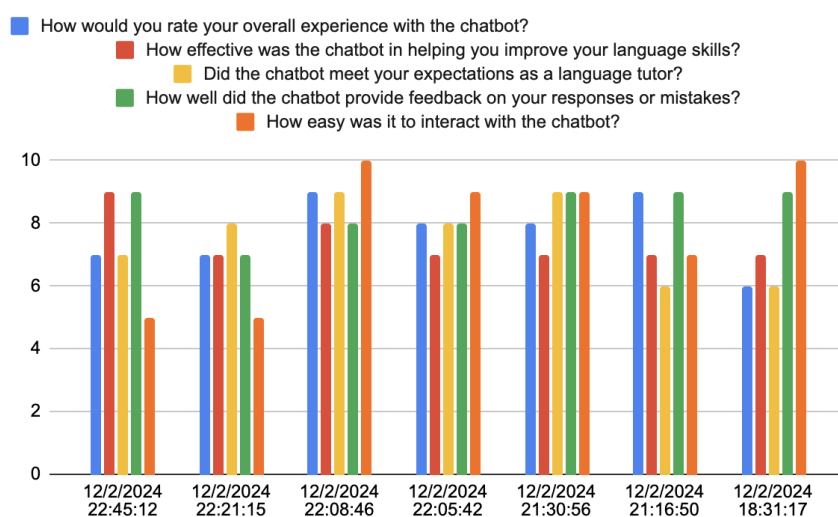


Figure 3a: AI Tutor Survey results

Was the chatbot able to adjust to your learning pace?

7 responses

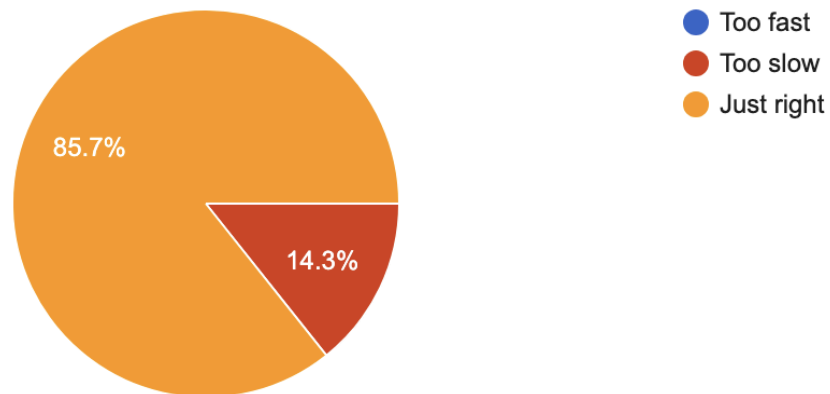


Figure 3b: AI Tutor Learning Survey results

The results of survey data, Figure 3a shows the responses to the different parts of AI language tutor application. They show ratings of each of the categories and effectiveness in improving language skill across the board. These were all mostly tested on a 1-10 scale while the majority of the survey results were positive and around the ratings of 8-9. This shows that the AI Tutor showed an effective learning experience for most users with a small margin of error. The performance of the bot was positively criticized, showing its deep reliability and user satisfaction.

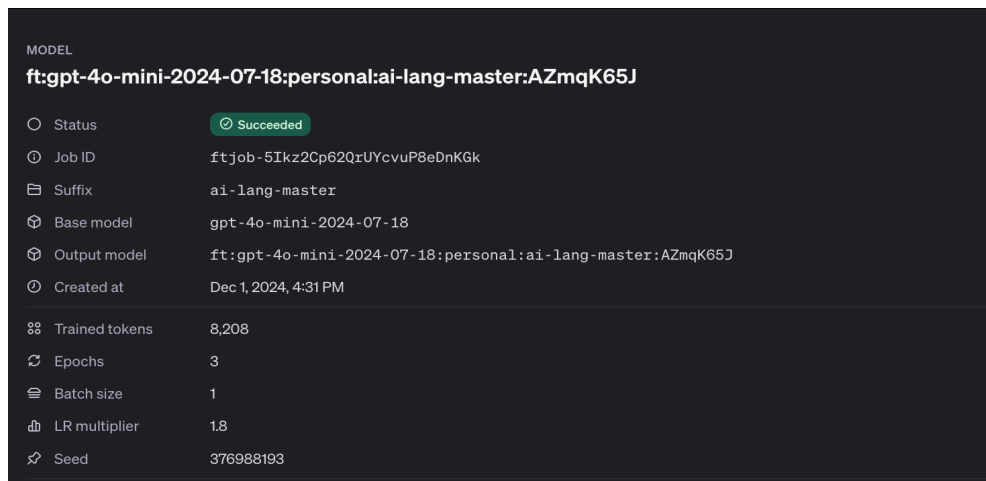
Figure 3b shows how the AI Tutor adapted to the pace of learning for each individual. It shows that 85.7% of the users felt that the chatbot that we've made adapted perfectly to their own learning pace. Only 14.3% said that the pace was “too slow” while nobody thought the pace was “too fast.” This shows that the project was very successful in completing its goal of teaching users needs. But the 14.3% who felt that the pace was slow said in the comments that we should focus on a system for faster learners as well.

Lessons Learned

We have learned a lot from this experiment and the results of the survey.

First is the entirety of this experiment. We learned a lot of lessons when fine-tuning the model.

We learned about token training and how it affects the quality of the AI's output. Tokens are the smallest forms of data that the model processes and serve as the focal point of how the AI generates responses. We created specific tokens that made the model produce contextually necessary and accurate responses. The training on these tokens improves the model's language-specific context, such as Spanish. The model was trained in 3 epochs, which was vital to its success.



The image shows a dark-themed user interface for a model training process. At the top, it identifies the model as 'ft:gpt-4o-mini-2024-07-18:personal:ai-lang-master:AZmqK65J'. Below this, a list of training parameters is displayed, each with an icon and a value. The 'Status' is 'Succeeded' in a green pill. The 'Job ID' is a long alphanumeric string. The 'Suffix' is 'ai-lang-master'. The 'Base model' is 'gpt-4o-mini-2024-07-18'. The 'Output model' is the same as the top header. The 'Created at' timestamp is 'Dec 1, 2024, 4:31 PM'. A horizontal line separates these from the training metrics: 'Trained tokens' (8,208), 'Epochs' (3), 'Batch size' (1), 'LR multiplier' (1.8), and 'Seed' (376988193).

MODEL	
ft:gpt-4o-mini-2024-07-18:personal:ai-lang-master:AZmqK65J	
○ Status	🟢 Succeeded
🔑 Job ID	ftjob-5Ikz2Cp62QrUYcvuP8eDnKGk
📁 Suffix	ai-lang-master
🔗 Base model	gpt-4o-mini-2024-07-18
🔗 Output model	ft:gpt-4o-mini-2024-07-18:personal:ai-lang-master:AZmqK65J
🕒 Created at	Dec 1, 2024, 4:31 PM
🔢 Trained tokens	8,208
🔄 Epochs	3
📏 Batch size	1
📈 LR multiplier	1.8
🌱 Seed	376988193

Figure 4: Trained tokens and Epochs

We learned a lot about front-end chat interfaces, back-end APIs, and AI models trained and tested by developers. Docker was an excellent tool to run multiple environments and handle them properly and seamlessly. We learned that technical performance was not enough, as we needed the ease of interaction, feedback, and the AI's ability to adapt to the user's learning pace, which are all factors vital to having a valid system. As well as the iterative nature of the development process also was needed to reinforce the context of the language deciphering.

Future Work

After this, we plan to enhance the chatbot's agility and effectiveness to ensure it can take in various languages, like Chinese, and cater to the user's needs. The next step would be to expand the dataset to add expressions and cultural formalities, such as the differences between Spanish and Portuguese and English and British English. Making a chatbot is a wonderful context-aware language learning experience. The AI adjusts its lesson plans based on the user's proficiency and progress and areas that need improvement.

Another area of work we could do is improving the feedback of the real-time translations by integrating pronunciation and grammatical analysis. Next, we could add VR/AR components to simulate talking with a natural digitized person. Offering environments to practice real-world scenarios with the chatbot, like ordering food at a restaurant or navigating a big city, would foster greater engagement and effectiveness.

Conclusion

In this project, we have created a conversational AI chatbot that combines language teachers and tutors with advanced AI models to foster real-world conversational practice. Using fine-tuned GPT models, a user-centric web app, and advanced prompt engineering, we discussed key challenges in NLP learning, like the need for natural interactions, context-filled responses, and real-time feedback. With the use of epochs and processing, as well as lots of testing and feedback, we were able to create a well-made, scalable solution to the issue.

The outcome of our study made it so that the AI chatbot was effective and working as intended as a promising future of conversational AI in language education. There are some areas we could have improved, for example, more languages that we could have supported and enhanced the adaptability. This project gives a good foundation for future advancements in making language learning more effective, thought-inducing, and engaging to learners worldwide.

References

Author links open overlay panelMarcello M. Mariani a b, et al. “Artificial Intelligence Empowered Conversational Agents: A Systematic Literature Review and Research Agenda.” *Journal of Business Research*, Elsevier, 21 Mar. 2023, www.sciencedirect.com/science/article/pii/S0148296323001960.

Chen, Kaiping, et al. “Conversational AI and Equity through Assessing GPT-3’s Communication with Diverse Social Groups on Contentious Topics.” *Nature News*, Nature Publishing Group, 18 Jan. 2024, www.nature.com/articles/s41598-024-51969-w.

Conversational AI: An Overview of Methodologies, Applications & Future Scope | *IEEE Conference Publication* | *IEEE Xplore*, ieeexplore.ieee.org/document/9129347/. Accessed 2 Dec. 2024.

LeewayHertz. “How to Train a GPT Model: A Comprehensive Guide.” *Medium*, Javarevisited, 8 Aug. 2023, medium.com/javarevisited/how-to-train-a-gpt-model-a-comprehensive-guide-cd77d8db2693.

Models - Hugging Face, huggingface.co/models. Accessed 2 Dec. 2024.

OpenAI API Reference, platform.openai.com/docs/api-reference/introduction. Accessed 2 Dec. 2024.