**Language Learning AI Tutor Assistant App: LightTalk**

**Author:** Natasha Wong

**Presenter:** Natasha Wong

**Advisor:** Yu Sun

1 College of Science, Cal Poly Pomona
2 Computer Science Department, Cal Poly Pomona

**INTRODUCTION**

LightTalk is an AI chatbot which lets users first specify information about themselves including the language they want to learn, their experience in the language, points for improvement, their goals such as learning more vocabulary or improving reading. LightTalk provides suggestions on exercises that may aid them in their language-learning journey. Various learning topics such as conversation practice, reading comprehension, pronunciation practice, or vocabulary practice produce different types of exercises. The assistant receives feedback from the user on how easy or difficult they found the exercise to be and analyzes that data to test the effectiveness of the chatbot and tailor improved responses for the user’s skill level.

**OBJECTIVES**

As a child of immigrants who was never taught my heritage language, I understand the unique challenges faced by individuals navigating through multiple languages and cultures. I hope to provide individuals a safe and supportive environment to help their journey in reclaiming their linguistic identities and bridging communication gaps through generations. To overcome barriers to language-learning to people of all ages, this app allows resources to become more accessible, personalized, and empowering regardless of one’s previous experience or background with language-learning.

**METHODOLOGY**

Materials: Python, Flet, OpenAI Assistant API, Visual Studio Code

1. Utilize the OpenAI Assistant API to create an Assistant incorporating a base model, instructions, and context documents.

2. Engineer and define custom instructions for the language-learning tutor Assistant, identifying topics for user assistance on their language-learning journey.

3. Include instructions for tracking user language proficiency throughout conversation and providing constructive feedback.

4. Implement LightTalk Assistant into a multi-platform app using Flet, a framework powered by Flutter. The provided Python script utilizes the Flet library to create a chat interface for the language-learning assistant.

**RESULTS**

The chatbot suggests a personalized curriculum based on the language the user wants to learn. This curriculum is meticulously designed to assist the user in learning the language efficiently, taking into account their individual learning preferences and progress. Users are provided with a variety of interactive exercises aimed at enhancing language proficiency. These exercises are thoughtfully crafted to be engaging and enjoyable, encouraging active participation and making the learning process more dynamic and effective. The chatbot effectively tracks the user’s language proficiency throughout conversation history. This feature enables users to monitor their progress over time, offering insights into their strengths and areas for improvement. Furthermore, the chatbot utilizes this data to adapt the curriculum and exercises according to the user’s evolving skill level, ensuring a tailored learning experience.

During experimentation, challenges arose in accessing and displaying user and assistant messages separately on the chat page, which required the creation of specialized functions to extract and distinguish message content. Additionally, crafting appropriate responses for language-learning prompts posed a challenge, necessitating precise formulation of instructions to ensure effective user engagement and assistance.

**CONCLUSIONS**

The implemented language-learning assistant successfully incorporates several key features including tailored curriculum, interactive exercises, and language proficiency tracking to enhance the user’s learning experience. In future iterations, users could be enabled to attach files of their language courses for the chatbot to reference. Upon user request, the chatbot would then generate personalized lessons and exercises based on the provided text, offering users more tailored assistance in line with their current language-learning endeavors. This addition would enhance the adaptability and customization of the language-learning assistant, further optimizing the user experience.

**ACKNOWLEDGEMENTS**

Flet - Creating realtime chat app in Python | Flet
Ilan Bigio - Assistants API Overview (Python SDK) | OpenAI Cookbook
OpenAI Assistants API
Tobias Solis - 9 Ways to Use ChatGPT for Language Learning