COMP4801 Final Year Project

Detailed Project Plan

English Speech Coach using AI

Supervisor: Professor Luo, Ping

Student: Lee Cheuk Iu (3035930008)

1. Background

With the ongoing trend of globalization, language learning has become more important than ever [1]. However, it is a time-consuming and challenging task which requires a significant amount of input and output from the learner. This is especially true for learning a second language, which comes with major obstacles such as having limited opportunities for practicing communication using the target language and the lack of feedback on the learner's performance [2, 3]. Due to these limitations, learners may find it difficult to make improvements on their language skills.

As there are many disciplines in language acquisition, this project will focus on assisting the development of speaking skills, specifically in English, which is the most spoken language in the world. To tackle the problems present in language learning in the scope of this project, a web application that allows users to converse with a voice chat bot is proposed. Research has indicated that assisting language learners using chatbots can yield positive effects [4], as it provides users with unlimited opportunities to practice speaking and a wide range of language knowledge, such as new vocabularies and phrases [2, 5]. It is also beneficial to users' learning due to its ability to give personalized and corrective feedback [1, 2].

Moreover, with the advancement of artificial intelligence in recent years, particularly in the field of natural language processing (NLP), automatic speech recognition (ASR) is now capable of adapting to speech from different speakers and is robust enough to be integrated into speech coaching applications [2, 6]. In addition, these applications can benefit from integrating large language models (LLM), which can be used for simulating human-like communication to give learners a more authentic and immersive experience.

In the current market, there are a number of applications and websites for assisting language learning. One of the most notable competitors is ELSA Speak, which utilizes AI chatbots as speaking practice partner for the user [7]. While this app provides comprehensive feedback on the user's pronunciation, fluency and vocabulary, it only gives general suggestions to help improve the learner's spoken English [7]. In addition, it does not provide any help or suggestions for beginners who may struggle to give relevant responses [7]. To make learning more effective for learners with varying English proficiency, some commonly used phrases that fits the context could be suggested and adjustments could be made to the speaker's input so that the user will be aware of their mistakes.

The remainder of this report is structured as follows. First, the objective of this project is highlighted. Then, a detailed description of the deliverable and a discussion about the methods of implementation is offered. Lastly, a schedule of different stages of the project is presented.

2. Objectives

This project aims to provide an alternative learning method that supplements traditional English language education in terms of speaking skills. The proposed web application will provide the user with an experience similar to conversing with a native English speaker by utilizing natural language processing (NLP) technology such as automatic speech recognition (ASR), large language model (LLM) and Text to Speech (TTS). The application will also allow users to make customizations to the chatbot. In particular, the learner can set up specific scenarios for role-playing which prepares users for different types of real-life conversations and helps users learn context-specific vocabulary and expressions. It will also cater to users with lower English proficiency by offering suggestions and demonstrations of a suitable response during the conversation. Furthermore, the application will streamline learning by providing translation options and displaying dictionary entries. It will also make transcripts of past conversations available and will allow learners to create memos for vocabularies and sentences, giving them an easy way to review their progress.

Overall, the main objective of this project is to encourage and to create opportunities for learners to speak English outside of the traditional classroom setting and to prepare them with practical English applicable for various real-life scenarios.

3. Methodology

3.1 Functionalities

3.1.1 Voice Chatbot

A core functionality of the web application will feature an AI-powered chatbot that accepts user's speech as input and returns audio responses. To achieve this, the application will first capture the user's speech using automatic speech recognition (ASR). By passing a user's recorded audio as input, ASR models will return the transcribed text which will be sent to a Generative AI for generating a response. The response will then be converted into synthesized audio by leveraging text to speech (TTS) which will be presented to the user as the response. Using this workflow, the chatbot will be able to simulate the behavior of a human agent that listens to the user and responds verbally.

This is a vital part of the application which will have a significant impact on the user experience. Therefore, in order to increase the naturalness of the human-chatbot interaction, factors such as response delay, accuracy of ASR, quality of the synthesized voice and the relevancy of responses should be optimized by comparing and assessing the capabilities of different ASR, Generative AI and TTS models.

Ideally, the ASR and TTS component will be implemented using Web Speech API as its has the advantage of being provided in most commonly used browsers including Chrome, Edge, and Safari [8]. However, the model only recognizes spoken English with accents that are listed in BCP 47 [8], which is limited and may affect the transcription accuracy for users with non-supported accents. In case of Web Speech API having low accuracy or low-quality synthesized voice, Google Cloud Speech to Text, Google Cloud TTS and other alternatives will be considered.

Regarding the Generative AI component, the chosen model for this project is Gemini developed by Google. Among all model variants, Gemini 1.5 Flash is the most suitable due to its high speed and its support for system instructions [9]. According to its documentation, its behavior can be altered using prompt engineering techniques to define a role for the model, tone and reading level of its output and the goal of the interaction [9]. This is an important feature as it allows for a higher degree of customization which accommodates users with varying learning goals and degrees of English proficiency. Moreover, Gemini is able to gain the most up-to-date information through its web search capability which also allows it to handle a wider range of topics [10].

The chatbot should be expected to generate coherent and context-adhering responses. This can be accomplished through optimizing prompt design by utilizing techniques such as providing context and adding few-shot examples [9], with the context being the role and scenario description while the examples are included within the transcript. In addition, it should be able to detect incoherent or unintelligible input from users and react accordingly by asking for clarification and guiding the user with further questions. When an interaction ends, the chatbot should be prompted with the entire transcript and given the task to analyze the user's overall performance in areas including tone, grammar, vocabulary and relevancy.

3.1.2 Response Suggestions and Feedback

Upon user requesting for assistance during interaction with the chatbot, an appropriate sample response with audio will be generated using the same Generative AI and TTS as the chatbot.

Apart from the feedback given after the conversation, the user can also request for feedback of each of their responses.

3.1.3 Dialogue Translation and Dictionary Entries

On both pages with the chatbot and chat logs, the user will have the option to view the translation of the chatbot's responses in their chosen language. Popular translation APIs such as Google translate and DeepL can be integrated to achieve this functionality. If an alternative method is to be considered, Generative AI models can also be prompted to obtain translations. However, this will not be the preferred method since it may have a longer response time when compared to APIs built specifically for translations.

To further assist users, individual words used in the conversation can be clicked to display its definition by calling a dictionary API such as the ones provided by Merriam-Webster and the Oxford dictionary. According to the official documentation of the two APIs, while both offer limited language options, Oxford dictionary supports definitions in more languages, including Chinese, Hindi, French and more [11, 12], making it more desirable for the purpose of this web application.

3.1.4 Notes

The user will be able to add notes entries during conversation with the chatbot or when viewing chat logs. Depending on whether the entry has a dictionary definition, the application will call a dictionary API or a translation API for providing details in the entry. Users will be allowed to organize these entries in different categories and add their own description for the ease of reviewing their learning.

3.2 System Architecture

3.2.1 User Authentication

Account registration and authentication will be handled using Passport.js, which is flexible and provides both local and OAuth authentication strategies [13]. In order to ensure data security, passwords will undergo hashing using the bcrypt.js library.

3.2.2 Frontend

The user interface (UI) and user experience (UX) design will be done using Figma, which will include both mobile and desktop view. As for frontend development, Vue.js will be used together with the Vuetify UI library. The UI will mainly consist of a login, home, chatbot, chat history and notes page.

3.2.3 Backend

The proposed web application will require the user to login to their account which will keep a record of the user's chat logs and note entries. These records will be retrieved and stored in MongoDB, which is a non-relational database that provides more flexibility and is better suited for storing information such as chat logs which may contain various data types [14]. Furthermore, due to the sensitive nature of login credentials and chat logs, data encryption is crucial to ensure that users' privacy is protected. According to the official MongoDB documentation, extensive support is provided for encrypting data that is in-transit and at-rest [15], making it an ideal choice for this project. The backend will be built using Node.js with the Express framework which will expose a RESTful API for the functionalities proposed in the previous sub-section, database access and authentication. By following REST principles, the API will be more flexible and will streamline the development process.

4. Project Schedule

Date	**Milestones**
September, 2024	Research on feasible approach to development Project planning
1st October, 2024	Detailed project plan Project web page set up
October, 2024	Web application set up UI/UX design AI model feasibility testing
November, 2024	Backend API set up
December, 2024	Backend API set up
13th – 17th January, 2025	First presentation
26th January, 2025	Preliminary implementation Detailed interim report

January, 2025	Frontend development Functionality testing
February, 2025	Frontend development Functionality testing Server setup
March, 2025	System testing Final adjustments
21st April, 2025	Finalized tested implementation Final report
22nd – 26th April, 2025	Final presentation
30th April, 2025	Project exhibition 3-minute video

5. References

[1] Z. R. Eslami and S. Zohoor, “Second language (L2) pragmatics and computer assisted language,” *Technology Assisted Language Education*, vol. 1, no. 3, pp. 1–17, Oct. 2023. Accessed: Sep. 25, 2024. doi: 10.22126/tale.2023.2788. [Online]. Available: https://tale.razi.ac.ir/article_2788.html

[2] R. Shadiev and J. Liu, “Review of research on applications of speech recognition technology to assist language learning,” *ReCALL*, vol. 35, no. 1, pp. 74–88, Jan. 2023. Accessed: Sep. 25, 2024. doi: 10.1017/S095834402200012X. [Online]. Available: https://www-cambridge-org.eproxy.lib.hku.hk/core/journals/recall/article/review-of-research-on-applications-of-speech-recognition-technology-to-assist-language-learning/5E15DEA15B24F210B095A799708AD00B

[3] E. Y. Oh and D. Song, “Developmental research on an interactive application for language speaking practice using speech recognition technology,” *Education Tech Research Dev*, vol. 69, pp. 861–884, Jan. 2021. Accessed: Sep. 25, 2024. doi: 10.1007/s11423-020-09910-1. [Online]. Available: https://link-springer-com.eproxy.lib.hku.hk/article/10.1007/s11423-020-09910-1

[4] S. Zhang, C. Shan, J. S. Y. Lee, S. P. Che and J. H. Kim, “Effect of chatbot-assisted language learning: A meta-analysis,” *Education and Information Technologies*, vol. 28, pp. 15223–15243, Apr. 2023. Accessed: Sep. 25, 2024. doi: 10.1007/s10639-023-11805-6. [Online]. Available: https://link.springer.com/article/10.1007/s10639-023-11805-6

[5] W. Huang, K. F. Hew and L. K. Fryer, “Chatbots for language learning—Are they really useful? A systematic review of chatbot-supported language learning,” *Journal of Computer Assisted Learning*, vol. 38, pp. 237–257, Sep. 2021. Accessed: Sep. 25, 2024. doi: 10.1111/jcal.12610. [Online]. Available: https://onlinelibrary.wiley.com/doi/epdf/10.1111/jcal.12610

[6] X. Chen, D. Zhou, H. Xie and F. Su, “Twenty-five years of computer-assisted language learning: A topic modeling analysis,” *Language Learning & Technology*, vol. 25, no. 3, pp. 151–185, Oct. 2021. Accessed: Sep. 25, 2024. doi: 10125/73454. [Online]. Available: http://hdl.handle.net/10125/73454

[7] Cloud English. " Is Elsa Speak's AI English Practice Partner Good?," *YouTube*, May. 9, 2024 [Video file]. Available: https://www.youtube.com/watch?v=Ok5aiMTiJKg&t=941s. [Accessed: Sep. 25, 2024].

[8] “Web Speech API - MDN Web Docs - Mozilla,” developer.mozilla.org. [Online]. https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API [Accessed: Sep. 25, 2024].

[9] "Gemini models | Gemini API | Google AI for Developers," ai.google.dev. [Online]. https://ai.google.dev/gemini-api/docs/models/gemini [Accessed: Sep. 25, 2024].

[10] N. L. Rane, S. P. Choudhary and J. Rane, "Gemini versus ChatGPT: applications, performance, architecture, capabilities, and implementation," *Journal of Applied Artificial Intelligence*, vol. 5, no. 1, pp. 69–93, Mar. 2024. Accessed: Sep. 25, 2024. doi: 10.48185/jaai.v5i1.1052. [Online]. Available: https://doi.org/10.48185/jaai.v5i1.1052

[11] "JSON Documentation," dictionaryapi.com. [Online]. https://dictionaryapi.com/products/json [Accessed: Sep. 25, 2024].

[12] "Oxford Dictionaries API," developer.oxforddictionaries.com. [Online]. https://developer.oxforddictionaries.com/documentation [Accessed: Sep. 25, 2024].

[13] "Documentation," passportjs.org. [Online]. https://www.passportjs.org/docs/ [Accessed: Sep. 25, 2024].

[14] "What Is A Non-Relational Database?," mongodb.com. [Online]. https://www.mongodb.com/resources/basics/databases/non-relational [Accessed: Sep. 25, 2024].

[15] "MongoDB Data Encryption," mongodb.com. [Online]. https://www.mongodb.com/products/capabilities/security/encryption [Accessed: Sep. 25, 2024].