Evaluation of Speech Recognition, Text-to-Speech, and Generative Text Artificial Intelligence for English as Foreign Language Learning Speaking Practices

DISSERTATION

Julio Christian Young

Tokyo Denki University

September 2024

Abstract

This research investigates the integration of AI technologies in speaking practice applications, building upon previous studies that explored chatbots as a viable solution for personalized speaking practice. Chatbots offer a promising alternative, fostering a supportive learning environment with reduced anxiety for students. However, the development of chatbot applications entails substantial costs, especially those related to the creation of learning materials and audio used within them. Therefore, this study focuses on evaluating the performance of Text-to-Speech (TTS) technology—specifically WaveNet—for tackling audio production problems, and offline generative AI technology—specifically ChatGPT—for creating dialogue practice materials. Additionally, considering the limited exploration of offline speech recognition (SR) implementation in the domain of English learning, this study also discusses the potential applicability of Vosk as an offline SR. The results of our experiments reveal that, despite WaveNet's TTS-produced materials being perceived as less natural than native speaker audio, they are equally easy to understand for learners. This makes TTS systems a viable option, particularly for beginner learners, especially in situations where native speakers are scarce. In a separate experiment, the implementation of Vosk as an offline SR system demonstrated a reasonably good performance in transcribing the speech of English as a Foreign Language (EFL) students. However, it necessitated a slower pace of speech. Therefore, it could serve as a valuable tool in educational settings where students lack confidence in engaging in speaking practices, and the primary focus is on promoting speaking behavior. Finally, as we explored the generation of learning materials for speaking practices using ChatGPT, it exhibited potential utility, particularly for individuals with a proficiency level equivalent to basic users in English, such as CEFR A1 to A2. Nevertheless, the study uncovered limitations for proficiency levels beyond A2. Building on these promising outcomes, our research expanded to assess the overall effectiveness of integrating these technologies into a speaking practice app aimed at enhancing students' speaking skills. To measure the impact, we conducted a pilot study, employing a pretest and posttest design and utilizing the Oral Proficiency Interview - Computerized (OPIc) test as a proficiency metric. Surprisingly, despite a 6-month learning period with the app, there was no significant improvement in speaking proficiency. These findings suggest that while AI technologies demonstrate promise on an individual level, their combined impact within the developed app fell short of achieving the desired enhancement in overall speaking proficiency. Therefore, further refinement is essential for the effective integration of these technologies into speaking practice applications.

Acknowledgment

I would like to express my sincere gratitude to Prof. Makoto Shishido, my supervisor, for his invaluable mentorship, guidance, and unwavering patience throughout the course of my dissertation. His expertise and support have played a pivotal role in shaping the trajectory of my research journey.

I extend my heartfelt thanks to my fellow research students and colleagues at the university, as well as those I encountered in part-time workplaces. Your camaraderie and shared academic experiences have added richness and depth to my journey, fostering an environment of collaborative learning.

A special acknowledgment goes to the staff at the university's international office for their continuous assistance and support. Their dedication has greatly facilitated various aspects of my academic pursuits.

I am deeply grateful to my family—my mother, father, older brother, and little sister—for their unwavering encouragement, love, and understanding. Their support has been my constant source of strength.

Last but not least, I extend my appreciation to all my friends, both in Tokyo and in Indonesia, for their friendship and companionship. Your presence has made my academic and personal experiences more fulfilling and memorable.

Thank you to each individual who has played a role, big or small, in this academic journey. Your contributions have made a lasting impact, and I am truly grateful for your support.

List of Chapters and Publications

Chapter I: Introduction (1)

↓

Chapter II: TTS for English as Foreign Language Learning (2)

↓

Chapter III: Offline ASR for English as a Foreign Language Learning Applications (3, 4)

↓

Chapter IV: OpenAI's ChatGPT for Generating Chatbot's Dialogue for English as a Foreign Language Practice Materials (5, 6, 7)

↓

Chapter V: Artificial Intelligence-based Speaking Practice Application

↓

Chapter VI: Conclusion

(1) Young, J. C. and Shishido, M., 2022. Natural Language Processing Technologies in English as Second Language Learning Applications A Review. Online, Association of Natural Language Processing.

(2) Young, J. C. and Shishido, M. 2022. Evaluating WaveNet Synthetic Speech for English as Second Language Listening Activities. Ise, Japan, IEEE.

(3) Young, J. C. and Shishido, M. 2022. Evaluation of Offline Automated Speech Recognition for English as Second Language Learning Application. Online, Association for the Advancement of Computing in Education.

(4) Young, J. C. and Shishido, M., 2023. Development, Evaluation, and Further Research of Voice-enabled Chatbot for English as a Foreign Language. Okinawa, Japan, Association of Natural Language Processing.

(5) Young, J. C. & Shishido, M. 2023. Evaluation of the Potential Usage of ChatGPT for Providing Easier Reading Materials for ESL Students. Vienna, Austria: Association for the Advancement of Computing in Education.

(6) Young, J. C., & Shishido, M. 2023. Investigating OpenAI's ChatGPT Potentials in Generating Chatbot's Dialogue for English as a Foreign Language Learning. International Journal of Advanced Computer Science and Applications, 14(6).

(7) Young, J. C. and Shishido, M., 2024. Prompting Brilliance: Unlocking ChatGPT's Potential to Revolutionize EFL Dialogue Practices. Hyogo, Japan, Association of Natural Language Processing.

Table of Contents

Chapter I Introduction

1.1 Background

The globalization of communication and commerce has underscored the importance of English learning in various global contexts. Acknowledging the importance of English proficiency in our interconnected world, Japan, as a prominent nation in Asia, has diligently confronted this challenge over the past decade. The Japanese Ministry of Education, Culture, Sports, Science and Technology (MEXT) has implemented several initiatives to enhance students' English proficiency[1]. Despite relentless efforts to improve the English educational curriculum within schools, Japan consistently finds itself still classified in a low proficiency group[2]. In the context of mastering English language abilities, which involve the core skills of reading, listening, writing, and speaking, Japanese students often experience difficulties, especially in the latter two. Despite special efforts to fortify the English curriculum within schools, speaking proficiency remains a persistent weak point.

Previous studies have explored this issue and proposed that anxiety and a lack of practice contribute to the problem [1]–[3]. Furthermore, the constraints of time and the large number of students in a classroom setting worsen the impracticality of individual dialogue practice [1]. Given these limitations, the traditional classroom learning model falls short in fostering students to develop confidence in their speaking skills, often resulting in demotivation. Addressing this challenge, research indicates that employing a personal speaking practice chatbot can offer a viable solution. This approach could provide students with a dedicated platform for dialogue practice, ensuring each student receives individualized attention and opportunities for improvement [4], [5]. Notably, prior studies [5]–[7] have revealed that students exhibit reduced resistance to practice when engaging with a chatbot, as the feedback received is perceived as less judgmental, creating a more supportive learning environment. In contrast to human partners who may find repeated requests for the same conversation or persistent mistakes to be tedious, a chatbot is willing to engage in such repetitions without any sense of boredom [3], [7].

However, the development of a chatbot application comes with substantial costs. Teachers must invest significant time in crafting dialogues that align with students' proficiency levels, ensuring the content is both relevant and challenging. Additionally, tailoring dialogues to individual skill levels and incorporating diverse responses to prevent monotony requires an extra time investment. Furthermore, the inclusion of native speakers in the creation of high-quality audio content, essential for emulating chatbot responses, adds to the financial and temporal requirements. This process escalates

[1] https://www.mext.go.jp/b_menu/shingi/chousa/shotou/102/houkoku/attach/1352464.htm

[2] https://www.ef.com/wwen/epi/

proportionally with the extent of variation in the dialogues, highlighting the resource-intensive nature of chatbot development.

As Artificial Intelligence (AI) research experiences a surge in advancements, researchers are increasingly exploring its applications in computer-assisted language learning. Remarkably, extensive research has centered on ChatGPT, encompassing diverse applications of English learning, including lesson planning [8], automated assessments and corrective feedback [9], question generation [9], and the creation of English reading materials [10]. Following such promising results, this study posits ChatGPT as a potential solution to the first challenge of generating dialogue materials for a speaking practice chatbot. However, given the existing limited evaluations of its appropriateness, especially in the context of content generation, this study also further identifies the appropriateness and most suitable audience for ChatGPT-generated materials.

Other than ChatGPT, recent progress in AI technology also has empowered Text-to-Speech (TTS) systems to produce increasingly natural-sounding audio materials [11]. In the context of English learning, prior research has revealed the promising potential of TTS-generated content, particularly for perceptual learning through listening comprehension [12], [13], rhyming and synonyms [14], as well as word dictation exercises [15]. Notably, Google's WaveNet [11], a TTS model, has demonstrated exceptional success in producing audio materials that closely mimic natural human speech, surpassing other TTS models in this aspect. Motivated by the need to cut the costly engagement of native speakers for chatbot audio content, this study ventures into the exploration of TTS as an alternative. To underscore the positive outcomes observed in previous research, this study will also assess the appropriateness of TTS, particularly within the context of audio materials employed in dialogue practice.

In addition to the aforementioned objectives, this study also addresses the limited exploration of offline speech recognition (SR) implementation, particularly within the domain of English learning. Prior investigations into SR for English learning predominantly featured online SRs that relied on expensive computational models executed in cloud servers. Despite the positive outcomes demonstrated by various online SR implementations in recognizing users' speech for English learning, the potential of offline SR remains underutilized. Consequently, this research embarks on experimenting with a lightweight SR tailored for in-device inference, even on smartphones, to recognize students' speech input. This attempt seeks to evaluate the performance of offline SR in detecting English as a Foreign Language (EFL) students' speech and aims to stimulate further research in this unexplored area. Through this exploration, we aspire to uncover the viability of offline SR in enhancing EFL students' learning experience.

1.2 Research Objectives

Within the context of English as a Foreign Language (EFL) learning, this research aims to explore the transformative potential of artificial intelligence technologies, specifically Speech

Recognition, Text-to-Speech, and Text Generative AI, in addressing persistent challenges within EFL speaking practice applications. The main objective is to develop a cost-efficient speaking practice application, yet with appropriate content, by implementing, experimenting with, and evaluating these existing AI technologies. The result will be the development of a showcase application, demonstrating the effectiveness of these AI technologies in assisting EFL students to enhance their listening and speaking abilities. Positioned as supplementary materials, the platform intends to allow students to independent practice outside the classroom settings through read-aloud exercises. Notably, the platform's materials will be exclusively generated using Text-to-Speech and Text Generative AI, eliminating the need for direct teacher and native speaker involvement in the app development. In addition, the implementation of automated feedback, which compares students' spoken input to a reference text within a dialogue, aims to elevate the learning experience by offering timely and personalized evaluations.

1.3 Structure of Dissertation

Chapter I serves as an introduction to the dissertation, presenting an overview of the research's background, motivation, and the trajectory it aims to follow. This chapter highlights the significance of speaking practice applications in aiding the English language learning process, underscoring the challenges associated with the high costs of content development for these applications. Following that, we discuss the ongoing implementation of text-generative AI and text-to-speech technologies in EFL learning, emphasizing their potential as a solution to the identified challenges. Additionally, we mentioned the less-explored field of incorporating offline speech recognition (SR) for English learning, emphasizing the scarcity of existing research in this space. Afterward, we establish a direction to suggest and develop a speaking practice app that integrates TTS and Text Generative AI for its content, along with offline SR technology to automatically analyze students' speech input. The proposed app serves as supplementary materials that facilitate students to practice their speaking skills independently. Building upon this direction, a series of experiments are detailed, outlining the implementation, analysis, and evaluation of existing AI technologies incorporated into our system.

In **Chapter II**, the primary focus revolves around conducting an experiment to assess the audio quality generated by Text-to-Speech (TTS) technology for English as a Foreign Language (EFL) learning. The experiment is structured using a supplementary dialogue designed for listening practice derived from an English learning book. Each dialogue line is crafted by a native speaker and matched with its corresponding text within the dialogue in the textbook. Employing WaveNet TTS, we generate TTS versions of each corresponding text. To evaluate the qualities of each audio group, we implement a random shuffling of the audio materials and solicit student judgments on various perceived criteria, which include pronunciation accuracy, comprehensibility, intelligibility, and naturalness for each audio. Subsequently, students are tasked with transcribing each listened audio, allowing us to calculate transcription error rates. Later, we draw a conclusion by analyzing both perceived qualities and transcription error rates for each audio group. Through this process, we aim to discern the appropriateness of TTS-generated audio materials as potential replacements for native speaker involvement in dialogue practice within the realm of EFL learning.

Subsequently, a speaking practice application is developed, comprising read-aloud exercises enriched with WaveNet Text-to-Speech (TTS) for its audio content. Notably, the application integrates offline Speech Recognition (SR) to proficiently recognize students' speech input and offer corrective feedback. This involved a comparison between students' speech transcription results recognized by the SR system and its reference text within a dialogue. Then, we conducted an experiment by utilizing the app to measure the error rate associated with the SR transcription. Other than that, participants' experiences and perspectives concerning the incorporation of the offline SR system within the learning app are collected to analyze qualitative aspects of the app in the evaluation process. All findings related to the adaptation of offline SR implementation will be presented in **Chapter III**.

Chapter IV addresses the challenge in the dialogue creation process that demands a substantial amount of time and effort from educators in the development of speaking practice applications. To address this, the chapter leverages ChatGPT as a powerful tool for generating reference dialogues tailored for read-aloud activities, offering students a platform to enhance their speaking skills. Through a series of experiments, the appropriateness of the materials generated by ChatGPT will be evaluated. Each experiment employs a set of readability metrics to determine the most suitable target audience for the ChatGPT-generated materials. Recognizing the impact of varying prompts on ChatGPT's output quality, the chapter also dedicates effort to analyzing how different prompt engineering techniques might affect the quality of results derived from ChatGPT.

Next, **Chapter V**, marks the integration of materials generated by ChatGPT into the existing speaking practice applications. An experiment is undertaken to assess the application's efficacy in supporting students' speaking practices. Utilizing a pre-post-test design, this chapter examines whether students experience a notable impact due to app usage. Furthermore, students' experiences with the application are actively sought, providing valuable qualitative insights into the user experience.

Finally, **Chapter VI** integrates the findings from individual experiments on each reviewed AI technology with the actual effects of app usage and students' perspectives. This chapter concludes with an analysis of the pros and cons of integrating AI technologies into EFL speaking practices. Additionally, it discusses the needs identified by students after using the app, highlighting areas for improvement. Based on these insights, the chapter suggests directions for future research to develop a more effective AI-enabled speaking practice application.

Chapter II Text-to-Speech Technology for English as a Foreign Language Learning

The area of language acquisition has long been influenced by Stephen Krashen's Input Hypothesis [16], a foundational theory positing that learners progress to the next stage of language acquisition by engaging with slightly more advanced linguistic input than their current proficiency level. This theory has not only shaped the landscape of language education but has also inspired numerous researchers to delve deeper into effective pedagogical practices. Recognizing the importance of listening activities in language development, many teachers advocate listening-centric activities in the classroom to provide learners with the varied input necessary for linguistic growth. However, the traditional teaching methods, constrained by limited time and reliance on the teacher as the sole source of listening input, have struggled to effectively meet these needs. Relying exclusively on the teacher in a classroom setting constrains the diverse linguistic exposure essential for learners to enhance their language proficiency [17].

In response to these challenges, a promising solution emerges through the utilization of Text-to-Speech (TTS) technology. Understanding the potential of TTS as a language learning tool, researchers have explored its capability in various learning settings such as allomorph word acquisition [18], pronunciation practice [15], and listening comprehension [12], [13]. Through the incorporation of TTS, researchers demonstrated how students can access a personalized learning system that aligns with their individual listening needs. Despite the promising outcomes observed in TTS-based learning, the literature is notably sparse on formal evaluations, particularly concerning their efficacy in aiding English as a Foreign Language (EFL) learning. Existing assessments predominantly focus on more traditional TTS implementations, often criticized for producing non-natural and robot-like audio materials [17], [19].

Standing out from customary techniques, a study in [11] introduces a novel TTS method, WaveNet, renowned for its capacity to generate more natural-sounding audio content. Building on the foundations laid by previous research [17], [19], this study takes a step further to assess the speech quality of WaveNet in generating listening materials specifically tailored for EFL learning.

2.1 Related Works

2.2.1 The Applicability of Text-to-Speech for English as a Foreign Language Learning

In recent studies, researchers have explored the impact of Text-to-Speech (TTS) technology on ESL learning, revealing several compelling findings. An investigation by researchers in [15] focused on evaluating the effects of perceptual learning through words' rhyme and synonyms, utilizing TTS synthetic speech in comparison to teacher-led dictation (TLD). The findings suggested that students with lower proficiency exhibited improved learning outcomes when exposed to synthetic speech as opposed to actual human voices. Nevertheless, the study also unveiled that there is no significant performance disparity between TTS and TLD for more proficient students. Additionally, students expressed a preference for teachers as pronunciation models, finding nonnative pronunciation produced by their teacher easier to emulate.

Similarly, a parallel study [14] evaluated TTS implementation to support word dictation activities for EFL students. By closely examining the performance of student groups participating in TTS-led and TLD activities, the researcher observed that students aided by TTS exhibited superior learning performance. Regardless of students' proficiency levels, TTS materials contribute to heightened semantic learning and provide enhanced support for perceptual learning, especially beneficial for those with lower proficiency. The researchers contended that such positive effects come from the more controlled speech quality inherent in TTS technology, characterized by steady reading speech and regular segmentation. These features contribute to making the audio input more comprehensible for students with lower proficiency, thus augmenting their perceptual learning. On the other hand, for higher proficiency students, the easily intelligible audio allows them to concentrate more on the semantic aspects of the material.

The insights gathered from these studies highlight the many benefits of TTS technology in EFL learning. Looking at different research, it's clear that TTS not only helps with specific language challenges but also works well for different levels of proficiency. Notably, TTS stands out as a budget-friendly and efficient tool for creating listening materials, where opportunities to interact with native speakers are limited [14]. Furthermore, various studies underscore that teachers find it easier to adjust the quality and content of listening materials with TTS to fit the needs of their students [13], [20]. Moreover, researchers agree that TTS helps non-native English teachers by providing pronunciation that sounds like a native, making students' language input more varied [12], [13], [18]. In essence, exploring TTS in ESL learning shows that it's a crucial tool for establishing a flexible and enriched learning environment for students with diverse skill levels.

Despite the fruitful outcomes, challenges persist, particularly for more traditional TTS approaches. In an earlier study [15], researchers pointed out that the artificial and emotionless sound created by TTS systems could lead to dull conversations and negatively affect students' motivation to learn. Another concern involves the inability of TTS services to confirm the correct spelling of each word in a given input, rendering materials meaningless in the presence of typos [13]. Moreover, most TTS services struggle with processing punctuation marks, resulting in a lack of intonation differences for declarative, imperative, exclamatory, and interrogative sentences [13].

2.2.2 WaveNet Text-to-Speech

WaveNet is a Text-to-Speech model that creates raw audio waveforms from scratch. The model is a deep neural network trained using a massive volume of speech samples. A trained WaveNet model can generate more natural-sounding speech than other text-to-speech systems. Compared with other text-to-speech technologies, it produces speech audio people prefer. Figure 1 shows the Mean Opinion Score (MOS - voice quality) from WaveNet to other synthetic voices and human speech.

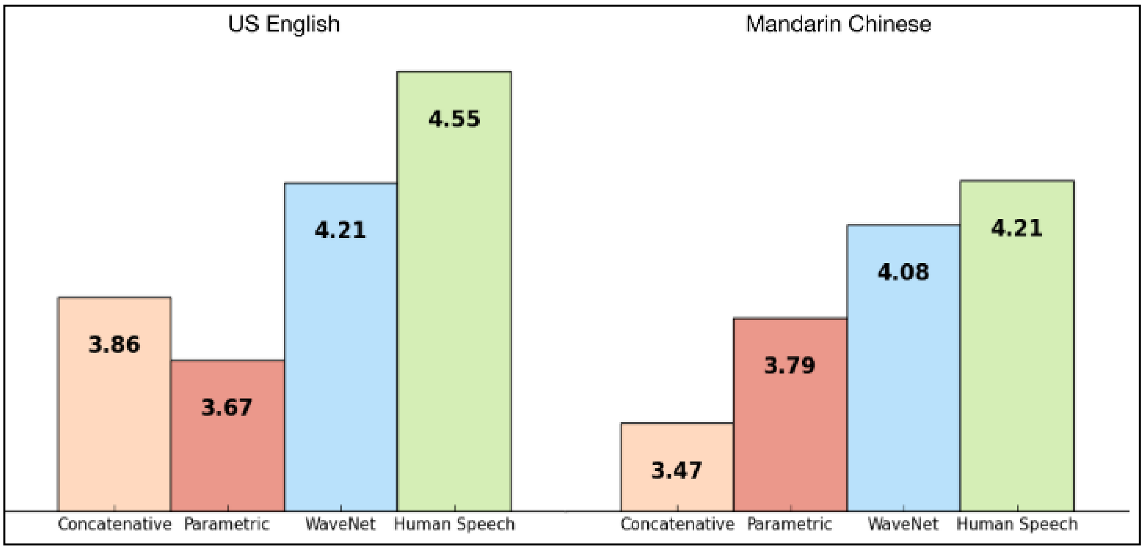

Figure 1 MOS Comparison of WaveNet to Other Synthetic Voices and Human Speech.

With around a 7-11% score difference between WaveNet and other TTS models for English, it is safe to assume that WaveNet might produce better English listening materials.

2.2 Research Methodology

In evaluating the effectiveness of WaveNet-generated materials, our study undertakes a comparison with listening materials featuring a human voice, aligning with methodologies from previous research [17], [19]. The assessment focuses on various dimensions of speech quality, employing specific criteria to gauge the efficacy of WaveNet-generated content. The following criteria are employed to compare WaveNet-generated audio with listening materials produced by real humans.

1. **Pronunciation Accuracy**: The quality regarding how accurately a spoken language, words, or phrases are reproduced.
2. **Comprehensibility**: The quality of how easy it is to understand the given audio
3. **Naturalness**: The believability of the audio as if spoken by a human
4. **Intelligibility**: The quality of how well the audio message is understood

By employing this set of criteria, our experiment aims to understand how well WaveNet creates audio compared to native speakers, as perceived by students. In the upcoming subsections, we discuss the research design, participant details, stimuli and materials, and the analysis process that will be conducted in the experiment.

2.2.1 Participants and Design

Included in the study were sixty undergraduate L2 students who had successfully completed three English courses at their university, demonstrating intermediate to proficient levels of English proficiency based on the cumulative final scores from these courses. The participants, aged between 19 to 25 years, with a mean age of 21.73 and a standard deviation of 1.54, had an average English learning duration of 13.30 years, with a standard deviation of 4.72. Each participant engaged in the experiment by listening to two distinct audio materials—one generated by WaveNet and the other by a native speaker. The experimental design intentionally withheld information about which audio was produced by WaveNet. Post-listening sessions, participants provided feedback on their perceptions of the speech quality of the presented audio. Additionally, participants were tasked with creating transcripts for each audio piece. These transcripts served as an additional dimension for assessing audio quality across the different audio groups.

2.2.2 Stimuli and Materials

The audio materials from WaveNet utilized in the experiment included audio files by both a female (Wavenet-F) and a male speaker (Wavenet-D) of North American English provided by Google Cloud TTS service. These generated materials were then compared to those produced by a native speaker, matching the same speech pattern (dialect) and similar speech properties. Each participant was exposed to a set of twenty sentences, and they rated each sentence on a 6-point scale according to the criteria mentioned earlier during the experiment. The sentences, derived from English learning dialogues available in [14], were carefully selected for relevance. Data collection and analysis were facilitated using Google Forms and Spreadsheets.

2.2.3 Analysis

We conducted descriptive statistics for each audio group, examining the central tendency and variability of speech quality criteria based on the collected data. Additionally, paired-sample t-tests were employed to identify significant differences in speech quality between the WaveNet-generated and native speaker groups. In addition, we calculated the average Word Error Rate (WER) for the collected transcripts as a quantitative measure of audio quality. Before the WER calculation, we implemented text normalization on the transcripts to enhance the relevance of the comparison. This process aimed to eliminate word variations in the transcript, ensuring a more accurate evaluation of WER. The text normalization steps are included.

1. Removing extra whitespace and non-alphanumeric characters.
2. Converting abbreviations to full texts.
3. Converting all letters to lowercase format.
4. Converting all numerals to words.

Under the assumption that lower WER values indicate better audio quality, comparing the WER values between the two groups allows us to assess the audio quality produced by WaveNet.

2.3 Experiment Results

2.3.1 User Perceived Speech Quality

Table 1 shows descriptive statistics per audio source and users' perceived speech quality.

Table 1 User Perceived Speech Quality Score on Human and WaveNet Audio

Audio Source	Indicator	Criteria			
		Pronunciation Accuracy	Naturalness	Intelligibility	Comprehensibility
Human	Mean	5.60	5.33	5.71	5.69
	Std Dev.	0.75	0.92	0.62	0.67
TTS	Mean	5.56	4.79	5.65	5.65
	Std Dev.	0.75	1.34	0.68	0.67

In addition, paired-sample t-tests were conducted with a significance level of 0.05, comparing ratings from native speakers to those from Text-to-Speech (TTS) for each criterion. The results are as follows.

1. Comprehensibility: $t(60)=2.74, p<0.006$
2. Naturalness: $t(60)= 10.66, p<2.03\times10^{-24}$
3. Pronunciation accuracy: $t(60)= 2.90, p<0.004$
4. Intelligibility: $t(60)= 2.64, p<0.008$

The findings indicate a significant difference in each criterion, with statistical significance ($p < 0.05$). This aligns with a previous study [17] involving students with diverse first language backgrounds, where both Text-to-Speech (TTS) and human-produced audio materials were rated. The consistent preference for native speaker-produced audio suggests a general inclination among students. However, since the differences in mean values for naturalness, intelligibility, and comprehensibility are relatively low, there might not be practical significance between audio generated by TTS and native speakers. It is noteworthy that, excluding the naturalness aspect, TTS-produced materials scored exceptionally well, ranging from 5.56 to 5.65 out of 6.00 on three out of four criteria as can be seen in Figure 2.

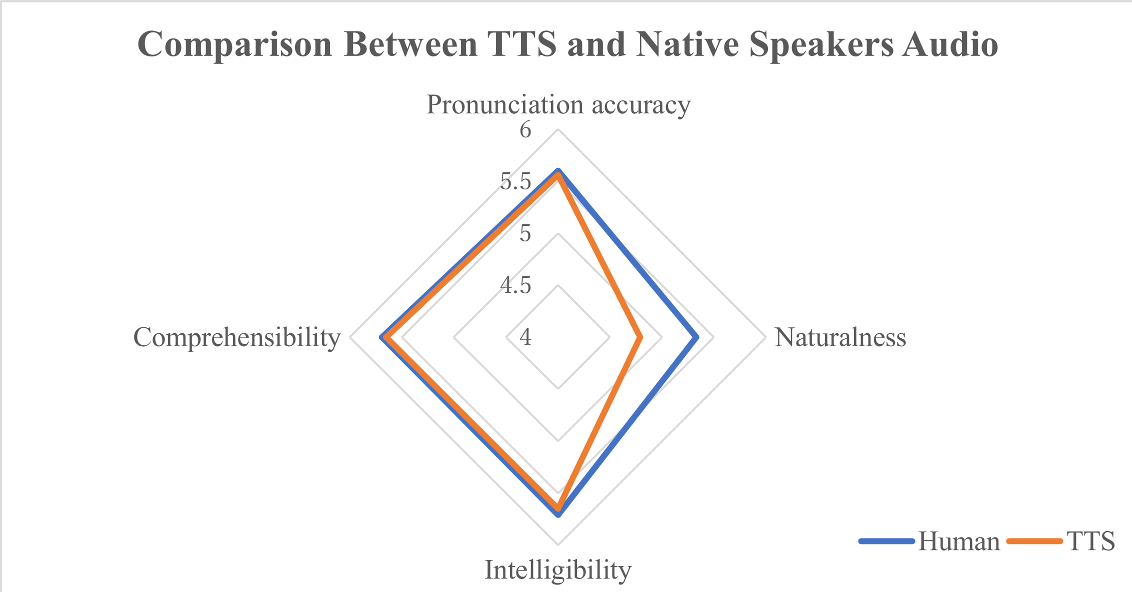

Figure 2 Relatively Small Differences Between TTS and Native Speaker Audio per Criteria (except Naturalness).

Moreover, comparing these results to a previous study [19], the mean rating score difference related to the naturalness aspect between WaveNet-produced and native speaker-produced audio materials is comparatively lower in this study. In the prior study, the mean difference was 1.46, while it is 0.54 in our study. This implies that audio materials generated by WaveNet are perceived as more natural compared to other Text-to-Speech technologies.

2.3.2 Transcriptions Word Error Rate per Audio Group

To complement participants' subjective assessments, we conducted a quantitative analysis by calculating the Word Error Rate (WER) using the transcriptions provided by participants. The WER was computed by comparing the transcribed text with the ground truth text reference for each audio segment. Subsequently, we categorized the WER scores based on the type of audio materials, distinguishing between Text-to-Speech (TTS)-produced materials and those generated by native speakers. The average WER for TTS-produced materials was remarkably low at 0.062, while native speaker-produced materials resulted in a slightly higher WER score of 0.068. In simpler terms, our analysis revealed that only 62 out of 1000 words in the transcript differed from the reference transcript when TTS materials were used, and 68 out of 1000 words for the native speaker-produced materials. The results contradict the earlier findings on participants' perceived comprehensibility aspects for each audio material, as participants favored native speakers' audio materials over TTS-generated ones. However, with the observed significantly low difference in the average perceived comprehensibility between the two audio groups, this outcome suggests that, for language learning purposes, there is only a relatively small to no practical significance between the native speakers' and TTS-generated audio materials.

Next, further investigation was done through a more detailed analysis aimed at addressing two pivotal questions that emerged from the result.

1. What types of mistakes are frequently encountered in learning with synthetic speech?
2. Whether these errors align with errors produced in English listening comprehension activities that involve native speakers?

Given the diverse array of potential mistakes in listening activities, the focus was narrowed to six specific types of errors for the purposes of this study. These include:

1. **Suffix Misidentification (SM)**: This category encompasses errors where participants successfully identified the root word but misinterpreted the associated suffix. Examples include participants misidentifying "damaged" as "damages," "airlines" as "airline," and "economy" as "economic."
2. **Numeral Misidentification (NM)**: Participants in this group misidentified the numeric content within the given audio. Instances include misidentifying "fifty" as "fifteen," "twelve" as "twenty," and "twelve-oh-five" as "twelve four five." Named-Entity Misidentification (NEM): Errors in this group involve participants misidentifying named entities within the given audio. For example, "Kei" might be misheard as "Kay," "Kei" as "Cai," and "Kei" as "Cay."
3. **Complete Word Misidentification (CWM)**: Participants in this category misidentified a word as an entirely different word. Examples include "plus" as "class," "give" as "get," and "counter" as "center."
4. **Typographical Error (TE)**: This group encompasses accidental mistakes made while typing the transcription. Examples include "upgrade" as "upgrad," "culture" as "culturr," and "boarding" as "bording." It is important to note that errors from the previous groups were not considered typographical errors.
5. **Missing Word - Unidentified Words (MW)**: In this group, participants were unable to capture specific words within the given audio. For instance, for the sentence "upgrade your seat to the economy plus," the written transcription only captured "upgrade your seat to economy plus."

Based on the previously mentioned types, Figure 2 illustrates the number of participants' errors per type based on their resulting transcripts. This visual representation serves as a valuable complement to our focused analysis of specific error categories in the subsequent discussion.

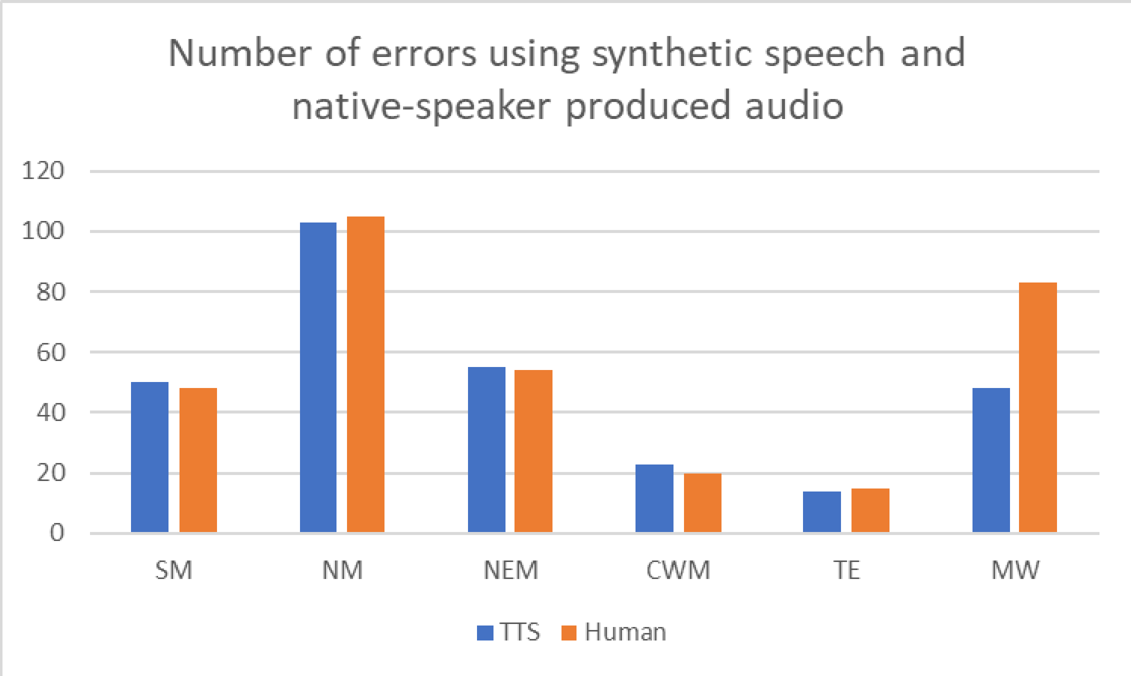

Figure 3 Comparison of Error Counts per Category in Transcription Results Using TTS and Human Audio

Figure 2 showed that learning with synthetic speech and native speaker-produced audio almost had the same number of all types of errors except for MW. Fewer MW errors in transcripts using TTS audio materials suggest that they are easier to listen to. The reason behind this might be because of a more controlled speech quality inherent in TTS technology, characterized by steady reading speech and regular segmentation as previously mentioned in [14]. This controlled delivery allows students to identify words more effectively, as there is relatively no connected speech between words within a sentence that might confuse them. This observation highlights the advantageous features of TTS-generated materials in facilitating clearer and more comprehensible language input for language learners.

2.4 Conclusion

This study aimed to evaluate the quality of speech produced by comparing WaveNet as one of the most natural-sounding TTS systems with human-produced audio. Our findings echo previous research [17], emphasizing a significant preference for native speaker-produced audio materials over TTS-generated ones. Interestingly, aside from the naturalness aspect, TTS-produced materials have a relatively low average score difference compared to native speaker ones, indicating that there might not be practical significance between audio generated by TTS and native speakers. This highlights their potential in language learning environments. Comparing our results to a previous study [19], where WaveNet's naturalness aspect was evaluated against native speaker-produced audio, our findings reveal a comparatively lower mean rating difference (0.54 in this study versus 1.46 in the prior study). This suggests that WaveNet's audio materials are perceived as more natural within the context of TTS technology.

Moreover, an in-depth analysis of error types revealed comparable error frequencies between synthetic speech and native speaker-produced audio, except for Missing Words (MW), in which TTS audio materials performed better. These results suggest that TTS-generated audio is easier to listen to, making it potentially more suitable for students with lower English proficiency. As previously mentioned in [14], the controlled speech quality characterized by steady reading speech and regular segmentation in TTS systems enables students to identify each word within a sentence more effectively.

Chapter III Offline ASR for English as a Foreign Language Learning Applications

Speaking proficiency is a pivotal component in achieving fluency and mastery of a targeted language. The ability to engage in spoken communication not only accelerates language acquisition but also enhances students' confidence in utilizing the language effectively. However, for students residing in environments with limited opportunities for regular language practice, particularly in speaking, the challenge of engaging in public conversations can be daunting. Xenoglossophobia, commonly known as foreign language anxiety, coupled with the fear of making mistakes and the apprehension of receiving negative feedback, often acts as significant barriers, impeding students from actively participating in speaking practice.

Recent advancements in machine learning technology have opened up new possibilities for using intelligent-based computer-assisted language learning (CALL). Intelligent-based CALL methods are becoming recognized as practical and handy learning tools that go beyond regular classrooms [21]–[23]. These methods aim to boost students' exposure to the language they're learning, providing creative solutions to the aforementioned challenges.

As one of the AI technologies, Automated Speech Recognition (ASR) stands out as a promising tool within the context of intelligent-based CALL, showcasing encouraging results in EFL learning. In a study conducted by Moxon [24], the implementation of ASR was shown to empower students by providing opportunities for language practice outside the traditional classroom environment, contributing to overall linguistic proficiency. Similarly, another research [4], [5], [25] highlights the effectiveness of ASR applications as speaking partner substitutes, providing learners with a valuable method to practice and refine oral communication skills, and outlining various benefits over traditional classroom speaking practice.

However, despite the promising results observed in previous studies, there are significant privacy and implementation concerns associated with this technology. Notably, it highlights the susceptibility of online ASR to profiling attacks, allowing potential inference of a user's presence through intercepted traffic [26], [27]. Moreover, the data collection practices of online ASR providers, even with certain data-sharing options, raise concerns about users' lack of control over the usage and potential inferences drawn from their collected data. On top of that, the reliance on Internet technology for processing each audio input in online ASR also results in higher bandwidth, particularly problematic for extended use on handheld devices.

Considering these challenges, the potential of offline ASR technology emerges as an alternative solution for EFL learning applications. Although research on offline ASR is limited, attributed to its comparatively lower performance than online counterparts, this study aims to assess its viability, particularly within the domain of EFL learning. Additionally, the evaluation extends to capturing students' perspectives on the appropriateness of ASR technology for EFL learning.

3.1 Related Works

3.2.1 Automated Speech Recognition for English as a Foreign Language Learning

The integration of Automated Speech Recognition (ASR) technology within English as a Foreign Language (EFL) education has garnered significant attention, particularly in its application to enhance pronunciation skills. Researchers have investigated diverse aspects of its utilization, ranging from targeted exercises such as single vowel pronunciation [21] to the evaluation of students' phonetic accuracy and fluency[24]. Additionally, ASR has been employed for broader language practice, including general utterance exercises [4], [5], [25]. Another noteworthy application is the real-time transcription of EFL teachers' speech, providing learners with an additional visual aid in the classroom [28].

The benefits of ASR technology in EFL learning contexts are multifaceted. Notably, its integration into a speaking practice application reduces the need for one-on-one tutoring sessions that traditionally require native speakers, thereby saving human and financial resources [24], [25]. Furthermore, it creates a more comfortable learning environment by mitigating the fear of judgment and embarrassment that learners may encounter in traditional classroom settings [4]. On the other hand, the dual-modality approach of providing a written transcription alongside the audio output caters to diverse learning preferences, contributing to a better educational experience [28].

Quantitative studies have consistently reported positive perceptions of ASR applications among students. Research findings indicate that ASR can reduce anxiety during speaking exercises, align with learner preferences, and enhance willingness to engage in both in-class and out-of-class language activities [4], [5], [25]. Notably, participants in studies focusing on real-time transcription processes have praised ASR for its role in improving listening comprehension skills [28]. Additionally, evidence suggests that the use of ASR in EFL learning may result in enhanced speaking performance compared to traditional methods [24].

However, the implementation of ASR in EFL learning is not without its challenges. In the context of speaking practice applications, the nature of ASR may inadvertently encourage learners to speak more slowly and with exaggerated enunciation to ensure clearer transcription, potentially deviating from natural speech patterns [24]. Another significant challenge stems from the fact that current ASR systems are predominantly trained with data from native speakers, leading to lower recognition rates for non-native speakers and subsequent translation errors [25].

3.2.2 Vosk as Offline Automatic Speech Recognition System

Vosk is an open-source automatic speech recognition (ASR) system that provides a lightweight recognition model to convert spoken language into written text [29]. Vosk gained popularity [30]–[33] for its ability to achieve accurate speech recognition results while using a relatively small amount of memory, making it suitable for applications on devices like smartphones and other resource-constrained platforms. The library supports multiple languages and offers models that can be adapted to specific vocabularies, allowing users to fine-tune the recognition system for their needs.

One of the standout features of Vosk is its lightweight model, which operates with a memory footprint of less than 400MB, this model challenges the norm by achieving a Word Error Rate (WER) of 9.85%, surpassing the performance of larger models that typically require around 16GB of memory to achieve a WER of 5.6%[3]. This optimization allows Vosk to bring efficient and accurate speech recognition capabilities to resource-limited devices. Furthermore, in its implementation of a lightweight model, Vosk introduces the dynamic vocabulary reconfiguration feature, allowing users to adapt the recognition system's vocabulary dynamically. This enables the system to focus on a specific set of keywords or phrases before transcribing speech audio. Allowing users to define and refine the set of words actively recognized ensures that the ASR system can prioritize and accurately transcribe the intended content. This attribute proves invaluable in applications where precision in understanding specific terms is paramount for optimal performance and user satisfaction.

In the realm of language learning, this feature takes on particular significance, especially in supporting students with low self-esteem. While such a feature may compromise precise pronunciation assessment, the ability of the system to recognize input effectively can serve as a motivational tool, encouraging students to speak more confidently and fostering a positive learning environment. Additionally, it is noteworthy that even though the Vosk system is configured to concentrate on a specific set of keywords or phrases before transcribing speech audio, it exhibits a thoughtful design feature. In cases where the system encounters input it cannot accurately recognize, it gracefully returns a token labeled "[UNK]" (unknown). This not only ensures the system remains applicable in the realm of language learning but also enhances the authenticity of the learning experience, especially when a student pronounces a word significantly differently or poorly, leading to instances where the system may not recognize the input.

3.2 Research Methodology

This section presents an overview of the methodology employed to evaluate Vosk as an offline ASR technology for the EFL learning process. Divided into key components, the chapter encompasses the Stimuli and Materials, followed by the Participants and Procedures subsection. The Stimuli and Materials subsection outlines the tools and resources used to facilitate our experiment. Subsequently, the Participants and Procedures subsection provides insights into how the individuals engage with the application in the research and the step-by-step methodology employed throughout the study.

3.2.1 Stimuli and Materials

In this section, the stimuli and materials utilized for the research are detailed, encompassing the development of an android-based guided conversational practice application with Text-to-Speech

---

[3] https://alphacephei.com/vosk/models

(TTS) capabilities and offline ASR. Figure 1 illustrates the primary interface of the developed application. The conversational practice application was designed for Android platforms, featuring guided interactions.

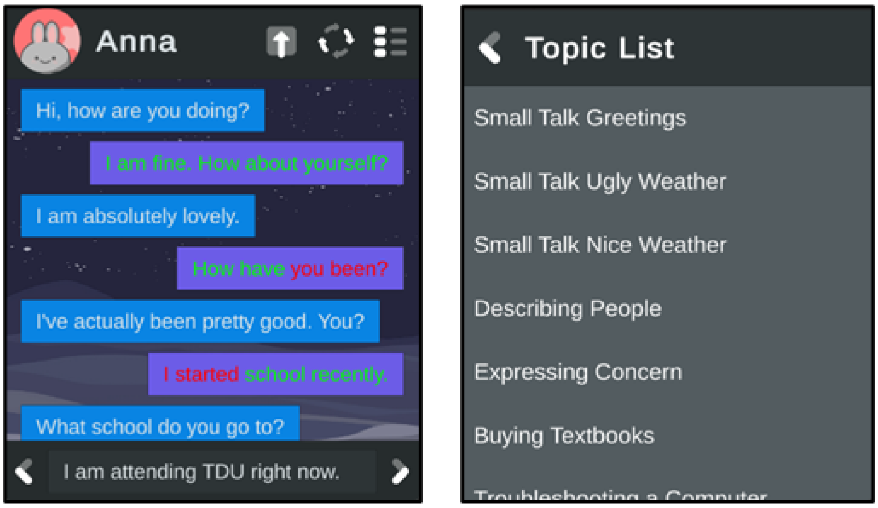

Figure 4 One of a bot and student conversational example within the application (Left) and several conversational topics available (Right)

To generate TTS materials, the Google WaveNet TTS [11] was employed. This advanced technology ensured high-quality audio output, enhancing the overall user experience. For offline ASR capabilities, the Vosk lightweight wideband model, specifically designed for low computational resource devices was integrated. This allowed users to engage in conversational practice without relying on continuous internet connectivity. Additionally, the conversational dialogues presented within the application were sourced from the ESL Fast platform[4], a reputable online English learning resource. With all the previously mentioned components, the user interaction process within the application can be comprehensively understood through the following steps.

a. The conversation begins with the bot reading aloud a transcription text (marked in red in Figure 5).

[4] https://www.eslfast.com/robot/

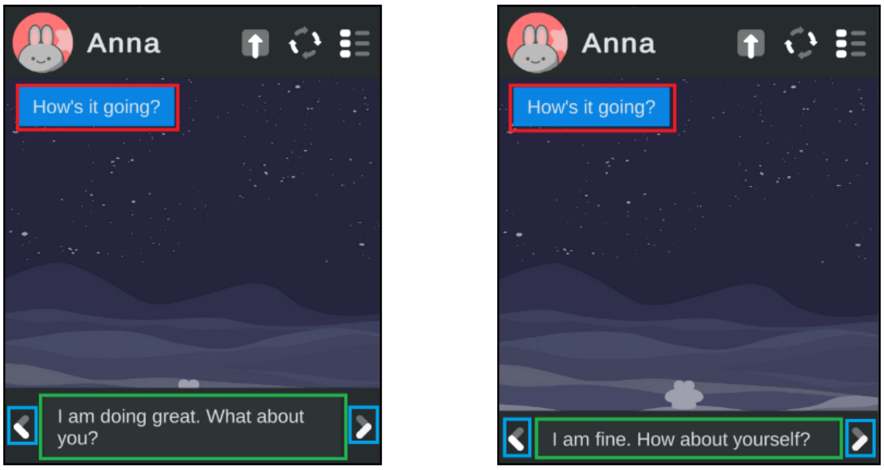

Figure 5 Chatbot will start the conversation (red) and the user can select their response (green) by using the left and right arrow keys (blue)

b. Users navigate through available responses using arrow keys (marked in blue in Figure 5) and select a response by tapping on the corresponding text.

c. The selected response is then rendered as a new chat bubble in the conversation panel (marked in yellow in Figure 6). Users can listen to the selected response and proceed by tapping on the new chat bubble.

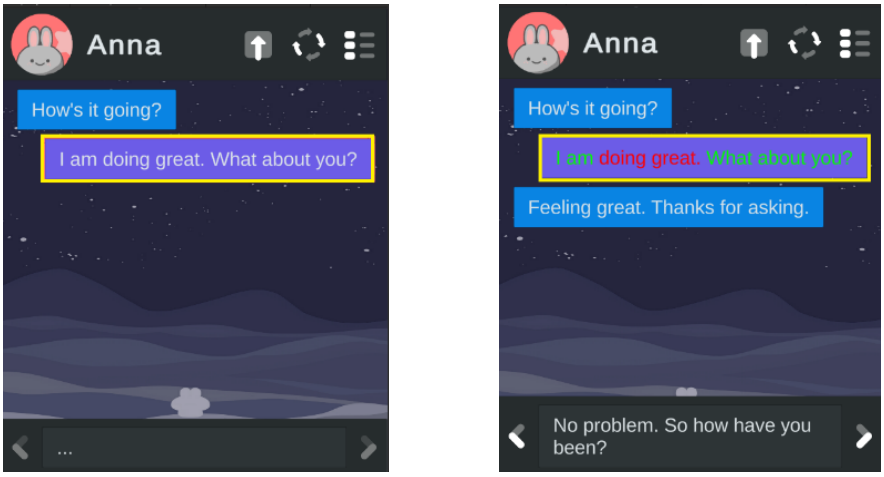

Figure 6 The application compared the ASR resulting transcription with the actual text and re-rendered the user's response

d. The ASR module processes the user's speech audio, returning a transcription.

e. The text within the selected chat bubble is re-rendered, with words appearing in green if present in the transcription and red otherwise (Figure 6).

f. The bot responds based on the user's selection, suggesting new topics if required. The application flow continues if the bot can still respond, proceeding to the next step in the described sequence.

In the developed application, dynamic vocabulary reconfiguration (DVR) features were implemented by constraining the reference keyword detection in the ASR functionality to encompass all unique words found in alternative responses across the application. To illustrate, consider multiple responses available to the prompt "Thanks for asking," such as "No problem," "You are welcome," "No worries," and "Do not mind it." When a user opts to practice any of these responses, the system focuses on detecting specific words like "no," "problem," "you," "are," "welcome," "worries," "do," "not," "mind," and "it." Additionally, the ASR system with DVR features appends the word "[unk]" within the resulting transcript if it encounters difficulties processing certain parts within the given audio.

3.2.2 Participants and Procedures

In this study, 15 EFL students who have completed three semesters of English courses offered by the university were recruited as participants. Before this, the participants had also undergone approximately 12 years of English education, starting in elementary school. With the primary goal of evaluating the performance of offline ASR within the app, participants were asked to engage with the application, and the resulting transcriptions from their interactions were recorded. Subsequently, the analysis of offline ASR was conducted using the Word Error Rate (WER) based on the transcription results. WER is computed by comparing the actual reference text with the transcriptions obtained. It is defined by Equation 1, where S, I, and D represent the numbers of substitutions, insertions, and deletions needed to align the resulting transcription with the reference text, and N denotes the total number of words in the reference text.

$$W = \frac{S + I + D}{N} \quad (1)$$

This method of assessment was chosen to provide an understanding of the ASR technology's accuracy. The resulting WER serves as a valuable metric considering the implementation of offline ASR in EFL learning applications.

Afterward, a post-activity questionnaire was employed to gauge participants' perspectives on the perceived usefulness (PU), perceived ease of use (PE), and attitude towards the use (TA) of offline ASR within the application. The questionnaire consists of six questions, each contributing to a specific aspect of the participants' experience.

1. Has ASR correctly recognized and evaluated what you said? (PU-1)
2. Does practicing speaking skills with ASR give you a more controllable and personalized learning environment? (PU-2)
3. Do you feel comfortable practicing your speaking skills with the help of ASR?" (PE-1)
4. Do you need to emphasize clarity (speaking slowly) when you practice using ASR?" (PE-2)
5. How interested are you in a learning system with ASR that can help practice your English

speaking skills?" (TA-1)

6. Do you want to adopt the learning system with ASR to practice your English speaking skills?" (TA-2)

All questions were formatted as 6-point Likert scales, ranging from one to six, representing a response continuum from 'strongly disagree' to 'strongly agree.' Notably, for Question 4, where a higher scale indicates a more challenging or less comfortable experience, the values presented in the results section have been inverted for readers' ease of understanding.

3.3 Results

3.3.1 Participants' Perspectives Towards the Learning Application with Offline ASR

In Table 2, the mean scores of participants' perspectives on perceived usefulness (PU), perceived ease of use (PE), and attitude towards the use (TA) of the offline ASR application are presented.

Table 2 Participants' Perspectives towards The Offline ASR Application

Question	PU-1	PU-2	PE-1	PE-2	TA-1	TA-2
Mean	5.06	5.13	5.33	4.13	5.20	5.26
Final Mean	5.1		4.73		5.23	

Examining the final mean score for the perceived usefulness criteria (PU-1 and PU-2) in Table 1, it is evident that the majority of participants agreed that the offline ASR used within the application accurately recognized their speech, facilitating a more personalized learning environment for practicing English-speaking skills.

Furthermore, the mean score from the first perceived ease of use criterion (PE-1) indicates that participants felt comfortable practicing their speaking skills with the assistance of ASR technology. This positive response suggests that the application could effectively contribute to a user-friendly experience, aligning with the intended goal of enhancing ease in language learning. However, it is noteworthy that the mean score from the second perceived ease of use criterion (PE-2) suggests that some participants found it necessary to speak more slowly to ensure the ASR functionality accurately detected their speech. This insight provides valuable feedback for the potential target audience of the application. As intermediate to advanced learners often have a words-per-minute rate higher than beginners, the use of offline ASR applications might be inappropriate for them, as they might feel uncomfortable speaking slowly.

Lastly, the final mean score for attitudes towards using the application (TA-1 and TA-2) reveals that a significant number of participants expressed interest in using the application and were inclined to adopt it for practicing their English-speaking skills. This positive attitude towards adoption is a key indicator of the offline ASR's potential success in engaging learners and fostering enthusiasm for language practice.

Building on these findings, we conducted an open-ended group discussion with participants to delve deeper into their learning experiences while using the app. During the discussion, a participant expressed a diminishing level of enthusiasm after using the application for an extended period. Notably, this participant suggested that incorporating additional features beyond speech evaluation could enhance the overall user experience. This sentiment gained resonance as other participants echoed similar views, expressing agreement that the inclusion of various types of exercises could introduce a higher level of challenge and contribute to a more enjoyable learning experience. These insights underscore the importance of continuous innovation and adaptability in the design of language learning applications, emphasizing the need to address user engagement over prolonged usage periods.

3.3.2 Word Error Rate of Offline ASR Transcriptions with DVR Features

The evaluation of offline ASR performance within the application was conducted by analyzing users' resulting transcriptions. The Word Error Rate (WER) served as the metric to quantify performance, computed by comparing users' transcriptions with a reference text. In this experiment, utilizing 308 resulting transcriptions, the offline ASR achieved an impressive WER of 7.42%. To provide a more intuitive understanding, this WER of 7.42% indicates that, based on 100 words spoken by the participants, the offline ASR failed to detect approximately 7 or 8 of them. This notably low WER suggests that, with the incorporation of DVR features, the offline ASR system demonstrated comparable performance to Cloud-based ASR services. Notably, DVR features played a pivotal role in enhancing the accuracy and effectiveness of the offline ASR, contributing to its competitive performance.

To gain further insights into the nature of errors impacting offline ASR performance, a detailed analysis of resulting transcriptions was conducted. The investigation revealed recurring patterns of errors, including the tendency to skip or misrecognize the first and last parts of speech. Interestingly, for other sentences when the same word appeared in the middle of the speech, the offline ASR system demonstrated accurate detection. Additionally, the system exhibited difficulty in correctly recognizing named entities, even when they were English-named entities. Finally, the speed at which a user pronounced a sentence emerged as a crucial factor influencing transcription accuracy. This observation aligns with the lower average response to question PE-2, shedding light on the nuanced relationship between pronunciation speed and transcription precision.

3.4 Conclusion

In summary of the experiments, key insights have surfaced, offering valuable perspectives on the effectiveness and considerations linked to the incorporation of Automatic Speech Recognition (ASR) technology in English-learning applications.

1. **Preference for ASR Practice**: Most EFL students participating in this study expressed a preference for practicing with ASR technology, attributing its appeal to the creation of a more controllable and personalized learning environment. This, in turn, contributed to an enhanced and comfortable learning process. The positive reception underscores the potential

of ASR technology to positively impact the language learning experience for students.

2. **Offline ASR with DVR Features**: The implementation of offline ASR with dynamic vocabulary reconfiguration (DVR) features yielded commendable recognition accuracy, aligning with the specific needs of our application. The seamless learning experience offered by this implementation, without relying on continuous internet connectivity or additional servers as in Cloud ASR, signifies a valuable alternative for learners seeking a flexible and accessible language practice tool.
3. **Considerations for Effective Use**: The findings highlight essential considerations when employing offline ASR technology in an English-learning application. The recognition accuracy was notably influenced by the speaking speed of students, indicating that the technology may be better suited for learners with relatively lower language skills. Moreover, addressing the tendency of the chosen offline ASR (Vosk) to skip or misrecognize the first and last parts of speech can enhance overall performance. The suggestion to introduce small audio gaps before the transcription process serves as a practical refinement. Notably, the use of DVR features, while advantageous in certain aspects, may compromise pronunciation accuracy. Therefore, careful consideration is required when selecting features, especially in applications where pronunciation accuracy is paramount and cannot be compromised.

Chapter IV OpenAI's ChatGPT for Generating Chatbot's Dialogue for English as a Foreign Language Practice Materials

A successful chatbot system for language learning typically involves several key components, including a speech recognition (SR) module, audio content, and reference dialogue content. In the previous two chapters, we covered subjects related to a speech recognition module and audio content for a chatbot system for helping English as a Foreign Language (EFL) students learn English. Therefore, this chapter will focus on the potential of text-generative artificial intelligence (AI) technology for generating reference dialogue in the EFL chatbot system. While numerous previous studies have extensively explored speech recognition and technology for developing audio content, the chatbot's dialogue content is often still sourced from existing materials produced by humans. With the recent advancements in text-generative AI, it is now possible for machines to generate readable and contextually appropriate content. Using machine-generated content could reduce reliance on human-produced content in the development process, thus reducing the cost and time needed significantly.

In examining text-generative AI technologies, our focus narrows down to the evaluation of one remarkable innovation that has received significant attention on the internet: Chat Generative Pre-Trained Transformer (ChatGPT) by OpenAI. ChatGPT is a novel chatbot implementation with impressive abilities to deliver coherent and contextually appropriate responses based on user requests. By leveraging vast amounts of text data, ChatGPT can generate text in various styles and tones, making it a promising option for creating content suitable for numerous purposes. In the realm of EFL learning, many researchers believe that the content generated by ChatGPT could be beneficial for EFL students in their learning process.

Despite the growing body of literature on ChatGPT's potential in language education [34]–[38], formal evaluations of its role in producing high-quality learning materials for students remain scarce. To address this gap, instead of directly assessing ChatGPT's potential for generating reference dialogue for practice, we initially conducted a small experiment on the quality of materials generated by ChatGPT. In this experiment, we analyzed its potential use in providing suitable reading materials for EFL readers. We presented ChatGPT with reading material and requested it to generate an easier version. Subsequently, we measured several metrics to determine its capabilities in doing so. Through this formal evaluation, we aimed to gain a deeper understanding of the potential benefits and limitations of using ChatGPT in EFL education, facilitating informed decisions about its adoption.

Following this, further experiments were conducted to use ChatGPT for producing dialogue practice materials, employing multiple readability metrics for comprehensive analysis. Gaining insights into the characteristics of ChatGPT-generated dialogue aims to identify the most appropriate audience to maximize learning benefits. Determining the target audience that can benefit the most from these materials enables optimization of their use, enhancing the effectiveness of language learning experiences. Lastly, considering the demonstrated potential of prompt engineering techniques in enhancing the performance of large language models, another experiment explores and applies several prompt engineering techniques in the context of EFL dialogue practice content generation. By employing these techniques in a combinatorial manner, we will generate EFL dialogue practice materials tailored to a specific learning scenario. Subsequently, quantitative measurements assess the quality and appropriateness of the generated materials, aiming to uncover valuable insights on effectively prompting ChatGPT to generate the best EFL dialogue practice materials.

4.1 Related Works

4.1.1 Applications of Natural Language Generation in English Education Context

Natural Language Generation (NLG) stands as a distinct field within artificial intelligence, focused on the automatic creation of text with human-like characteristics. Its primary goal is to generate written or spoken language that appears natural, coherent, and imbued with meaning. NLG finds applications across various domains, including the development of chatbots, text summarization, and text paraphrasing [39]. In the sphere of English learning, the integration of NLG technologies holds promise for delivering interactive, personalized, and captivating learning experiences [40]–[42]. For instance, chatbots can help learners with prompt feedback on their writing, while text summarization facilitates a quick comprehension of key points within a given text. In the following subsections, we will discuss chatbots, automatic text summarization, and automatic text paraphrasing as specific implementations of NLG.

A . Chatbot

Implementing chatbots for EFL learning offers students valuable conversation practice. Tailored to specific purposes, a chatbot can process and respond with either text or speech, engaging learners interactively. Voice-enabled chatbots often integrate additional artificial intelligence technologies like speech recognition and text-to-speech to facilitate natural spoken interactions [5]. Research indicates a notable advantage of chatbots lies in their accessibility, allowing language practice anytime and anywhere, a convenience especially beneficial for learners [43]. These implementations prove particularly significant for EFL learners who might lack opportunities for language-speaking practice in traditional classroom settings. Engaging with a chatbot enables students to receive real-time feedback, fostering the development of language confidence through conversational interaction [5], [6].

Numerous studies have consistently demonstrated that students find a comfortable environment for practicing and making mistakes when engaging with a chatbot [4]–[6], [28]. Particularly in EFL settings where access to native speakers is limited, chatbots emerge as valuable tools for offering students essential conversation practice. Furthermore, an array of studies has assessed the impact of chatbot-assisted learning on student performance, revealing that chatbots contribute to enhancing both perception skills (reading and listening) and production skills (writing and speaking) [43]. Despite these advantages, it's crucial to acknowledge the potential pitfalls. As previously mentioned, free-text input chatbots may occasionally lead to communication breakdowns, causing frustration and demotivation for learners [44]. Similarly, chatbots employing predefined dialogue conversations are often criticized for feeling monotonous and uninteresting [43].

B . Automatic Text Summarization

While numerous studies have explored automatic text summarization, there remains a limited amount of research specifically discussing its application in English learning. Nonetheless, a handful of studies have concentrated on adapting automatic text summarization for the broader learning process. Previous research, as exemplified by studies such as those by Cagliero et al. [45] and Pramudianto et al. [46], suggests that text summarization can bolster student learning motivation by furnishing a condensed rendition of extensive text materials, such as lecture notes or discussion threads. The task of reading lengthy texts can be demotivating for students, demanding more time and effort. However, the process of summarization renders these materials more accessible, thereby amplifying students' motivation to engage with the content. By simplifying complex text materials, text summarization for learning ensures that students find the material more manageable, thus enhancing their motivation to learn about the subject matter. Consequently, in the context of EFL learning, text summarization's ability to provide simplified versions of materials could be beneficial for the learning process. The adaptability of text summarization to tailor the difficulty of materials to match the student's level of understanding holds the potential to significantly enhance student engagement and motivation for effective learning outcomes.

C . Automatic Text Paraphrasing

Paraphrasing stands as a valuable asset for EFL students navigating the intricacies of language acquisition. This practice involves rephrasing written content, allowing students to comprehend the meaning behind words and express the information in their own words [42]. Not only does paraphrasing contribute to an expansion of vocabulary and language proficiency, but it also nurtures confidence in English writing. Proficiency in paraphrasing is particularly crucial in academic writing, showcasing a profound understanding of the material and the ability to use language effectively [47].

Within the context of EFL education, the integration of automatic text paraphrasing can significantly enhance students' writing skills. By offering diverse renditions of the same information, this technology aids in language comprehension and the development of a more extensive vocabulary.

A study conducted by Ariyanti [47] evaluated the impact of incorporating paraphrasing tools on students' writing performance, revealing positive outcomes. Notably, students displayed a favorable attitude toward utilizing automatic paraphrasing tools. Another study by Sulistyaningrum [42] further underscores the utility of online paraphrasing tools in assisting students in overcoming challenges associated with paraphrasing. These findings collectively highlight the transformative potential of automatic text paraphrasing in augmenting the writing skills and attitudes of EFL students.

4.1.2 ChatGPT in the field of English Language Education

The release of ChatGPT in November 2022 sparked widespread interest on the Internet, captivating researchers and the general public alike with its remarkable capabilities. As ChatGPT gained popularity, researchers began exploring its potential applications in education. Several studies have investigated the integration of ChatGPT in language learning, recognizing its vast potential to enhance learning experiences by offering a more personalized, seamless, and engaging approach, while also acknowledging and discussing potential challenges that may arise [34]–[38], [48]–[53]. Despite the optimistic outlook, a handful of studies [54], [55] have raised significant concerns about its adaptation, emphasizing the need for careful consideration of potential issues. Therefore, in the next subsection, we aim to provide a summary of the challenges and limitations, followed by the opportunities and applications of ChatGPT in the realm of education.

A . Challenges and Limitations

One major issue in the implementation of ChatGPT is the challenge of authorship attribution [36], [37], [56], [48]–[55]. As an AI language model, ChatGPT generates responses based on patterns and information from its training data, without explicitly citing the original authors or sources. This lack of proper attribution raises questions about intellectual property [36], [52] and plagiarism [37], [54], [55], as it becomes difficult to determine the origin of the information generated by ChatGPT. Without clear mechanisms for tracing the sources of information generated by ChatGPT, there is a risk of inadvertently promoting intellectual dishonesty or failing to acknowledge the work of original authors.

Another significant concern raised by researchers is the potential for ChatGPT to generate inaccurate or biased information in a convincingly human-like manner [38], [48], [51], [52]. Being trained on vast amounts of data from the internet, ChatGPT can inadvertently perpetuate misinformation or biases present in its training corpus [36]. Other than that, researchers also mentioned a serious issue regarding the over-reliance on its use that might hinder the development of students' creativity and critical thinking skills, as they may become passive recipients of information rather than active participants [48], [50], [53], [56].

Furthermore, the current landscape of conversational AI technologies is dominated by proprietary products developed by a handful of large technology companies due to the immense computational resources required [52], [55]. However, this concentration of power raises concerns within the research community, as it presents immediate challenges related to transparency and open science [52]. The lack of transparency in proprietary systems hinders the ability of researchers to fully understand and evaluate the inner workings of these models, limiting collaboration, reproducibility, and innovation. Additionally, the prevalence of monopolistic practices within the industry creates inequities in access and control over conversational AI technologies, potentially stifling competition and hindering the development of diverse and inclusive solutions [52].

B . Opportunities and Applications

Despite the skepticism and concerns toward the impact of ChatGPT in the field of education, many researchers believe that it has the potential to solve difficult problems and unlock numerous opportunities. A survey study conducted by Ali et al. [34] collected opinions from students and teachers regarding their perceptions of ChatGPT. The results showed overwhelmingly positive attitudes toward its adaptation, with most students agreeing that learning with ChatGPT would bring a sense of fun and enjoyment to their language-learning journey. In another study by Jeon and Lee [53], researchers identified four key roles where ChatGPT could be highly useful in an educational context. These roles include that of an interlocutor, content provider, teaching assistant, and evaluator. Rather than resisting the adaptation of AI in education, the study suggests that it is necessary to embrace LLM technology while maintaining a balance between the use of technology and human guidance. This approach ensures that the integration of ChatGPT and similar models serves as a supportive tool for teachers, augmenting their instruction and facilitating personalized learning experiences for students.

Similarly, previous studies conducted by researchers in the field [35], [52] have advocated for embracing a ChatGPT mindset in education instead of outright banning its use. The study emphasizes the need for a shift from purely quantitative assessment methods to a more balanced approach that combines qualitative and quantitative assessment [35]. Researchers argued that the focus should be on student's ability to synthesize essential content rather than solely evaluating the final learning product. Therefore, there needs to be a corresponding change in the teaching and learning process, aligning it with the digital technologies that have become integrated into many aspects of students' lives. By integrating ChatGPT and similar technologies into the educational context, researchers believe that educators can free themselves from unnecessary tasks and redirect their efforts toward more important aspects that can drive innovation and breakthroughs [52].

Specifically, ChatGPT can serve as a valuable teacher-assistive technology, offering a range of benefits for language learning tasks. In [53], researchers have highlighted that ChatGPT diversifies the options available to teachers in terms of learning materials. With the ability to generate content in a human-like manner, ChatGPT can provide a variety of resources that cater to different learning styles and preferences [49], [51]. By tailoring the generated materials to each student's specific needs, teachers can provide targeted and personalized learning experiences [49]. Additionally, the integration of ChatGPT can also alleviate the heavy marking load on teachers thus allowing them to spend more time on lesson planning [49].

Furthermore, various research studies have explored the potential of ChatGPT as a teaching assistant, providing additional support to educators in various ways [35], [38], [49], [51], [53]. For instance, ChatGPT can assist in correcting student grammar and explaining the meanings of difficult words, offering instant definitions and clarifications to enhance students' vocabulary acquisition [53]. Moreover, ChatGPT can generate essays and questions on specific topics, fostering students' critical thinking and analytical skills [51], [53]. The materials generated from ChatGPT allow teachers to generate personalized material that addresses students' specific needs [49], [51]. Lastly, ChatGPT has the potential to automate the grading process for assignments and assessments [53]. This automated grading system offers efficiency and consistency, enabling teachers to provide timely feedback to students and focus on more interactive and individualized instruction.

4.1.3 Applications of Natural Language Generation in English Education Context

Readability metrics provide quantitative measures that help evaluate the complexity and difficulty level of written texts. These metrics take into account various linguistic and structural features of the text to estimate its readability. In the field of education, assessing the readability of educational materials is crucial to ensure that they are appropriate and comprehensible for the target audience [57]–[60]. By utilizing these metrics, educators can gauge the readability of educational resources and make informed decisions about adapting or modifying texts to match the reading abilities of their students.

Readability metrics employ specific calculations to assess the readability of a text. For example, Flesch-Kincaid calculates readability by using two key factors: sentence length and word length. The formula assumes that longer sentences and words with multiple syllables require more cognitive effort to understand. Several other commonly used readability metrics, such as the Automated Readability Index (ARI), Coleman-Liau Index, and Simple Measure of Gobbledygook (SMOG) Index, follow a similar principle. ARI and Coleman-Liau, for instance, calculate the text's difficulty based on the average number of characters per word and the average number of words per sentence. On the other hand, the SMOG Index focuses solely on the number of polysyllabic words in the text.

However, readability metrics can go beyond assessing sentence and word length. The Dale-Chall Readability Formula, for example, employs a predefined list of "difficult" words to evaluate readability. This formula determines the text's difficulty by counting the number of challenging words that may be unfamiliar to a particular audience. By incorporating such information, the formula provides an assessment that accounts for the vocabulary demands of the text. While each readability metric offers valuable insights on its own, several research showed that using multiple metrics in combination can yield a more comprehensive analysis of a text's readability.

Several studies in the past have demonstrated the usefulness of readability metrics in the context of English as a Foreign Language (EFL) learning. These metrics provide valuable insights into the appropriateness by showing the level of difficulty of text materials, enabling educators and curriculum developers to make informed decisions about the selection and adaptation of reading resources for EFL learners. In a study conducted by researchers in [60], the Flesch Reading Ease (FRE) formula was employed to compare the readability levels of two English textbooks. By analyzing the reading materials within each book using the formula, researchers were able to determine the intended target audience of the textbooks. By identifying the intended learner proficiency levels, teachers can ensure a better alignment between students' reading abilities and the difficulty of the materials.

Similarly, previous studies in [57], and [59] also utilized readability metrics to evaluate and select English textbooks for senior high school students. However, instead of relying on a single metric, these studies employed a combination of multiple readability metrics. By considering various syntactic factors through multiple metrics, these studies aimed to provide a more comprehensive and accurate assessment of text difficulty. In [59], in addition to using the Flesch-Kincaid metric, an additional readability metric called Coh-Metrix was utilized. The inclusion of Coh-Metrix aimed to provide more detailed results regarding the complexity of the textbook. The findings of this study revealed that the selected textbook was below the expected standard and considered too easy for senior high school students, prompting the need for a more appropriate choice of materials.

In another study [57], researchers employed seven different readability metrics to evaluate the difficulty of each reading text within an English textbook. Based on the resulting scores, a consensus was reached to determine the final difficulty level of each text. The overall appropriateness of the English textbook was then concluded by calculating the average difficulty level across all the reading texts within it. Doing so allowed researchers to assess the collective difficulty of the texts and make informed judgments about the textbook's suitability for a particular audience proficiency level.

4.2 Evaluation of ChatGPT for Providing Easier Reading Materials for EFL Students

4.2.1 Stimuli and Materials

In this study, the primary source materials were sourced from "The Jakarta Post," a local online English-language newspaper based in Indonesia. First, one hundred news articles were selected specifically from the front-page section of the newspaper. This section encompasses a wide array of subjects, including but not limited to politics, economy, tourism, and culinary, ensuring a comprehensive representation of diverse topics. Opting for an online newspaper as the data source was motivated by its convenient accessibility. Additionally, previous research, as highlighted by [61], [62], underscores the efficacy of newspapers as valuable reading materials for EFL students.

The materials produced by ChatGPT, tailored for enhanced comprehension by EFL students, were generated using the ChatGPT version as of January 30th, 2022. It is crucial to acknowledge that variations in ChatGPT versions can yield different outcomes. For the sake of experiment reproducibility, all generated materials will be meticulously preserved and publicly accessible within this repository [63]. To generate these materials, ChatGPT was initially initiated through a web browser. Subsequently, a specific prompt – "Please rewrite the content below to enhance understanding for English as a Foreign Language readers" – was presented to the bot. Following this prompt, each material underwent processing by ChatGPT, and the resultant responses were documented. While it was feasible to request the response length by employing an additional prompt before providing the actual content, this step was intentionally omitted in the experiment to maintain simplicity in the adjustment process. Although incorporating more prompts into the interaction with the bot could potentially enhance the quality of output, such an approach would be impractical in real-world scenarios. The addition of more prompts might pose challenges for students in terms of memorizing required prompts and customizing parameters in the pre-prompt, rendering it less user-friendly and unrealistic.

4.2.2 Measurement Procedure

In this experiment, a range of text complexity metrics will be employed, spanning from straightforward measures to more intricate ones. The simpler metrics encompass the average number of words per sentence and the average number of sentences per text content. These metrics were selected to offer a high-level, abstract depiction of text complexity. In contrast, the McAlpine EFLAW and Gunning Fog readability metrics were chosen as more complex measures for this study, aiming to provide a nuanced, detailed analysis of text complexity.

The McAlpine EFLAW readability metric was specifically chosen for its focus on mini-word clusters in wordy cliches, colloquial expressions, and phrasal verbs – elements that often pose challenges for EFL readers. This metric facilitates a detailed understanding of text complexity at the word level, making it particularly suitable for analyzing materials intended for EFL students. To ensure a comprehensive evaluation of text complexity, the Gunning Fog readability metric was also incorporated into the research. Unlike the McAlpine EFLAW metric, which concentrates solely on word-level analysis, the Gunning Fog formula considers the number of syllables in each word, offering a more granular analysis of the text's complexity.

While alternative readability metrics, such as Flesch and Dale-Chall, could have been considered, the Gunning Fog formula was specifically chosen due to its initial design for use in newspapers – a format of the materials utilized in this study. The combination of both simple and complex metrics aims to furnish a holistic understanding of the complexity inherent in both the original text and its AI-generated simplified version.

4.3 Results and Discussion

4.3.1 Average Number of Sentences per Material's Type

The analysis began by calculating the number of sentences in each text individually. Subsequently, we calculated the average number of sentences for each group. The outcomes of these calculations are visually presented in the histogram depicted in Figure 7.

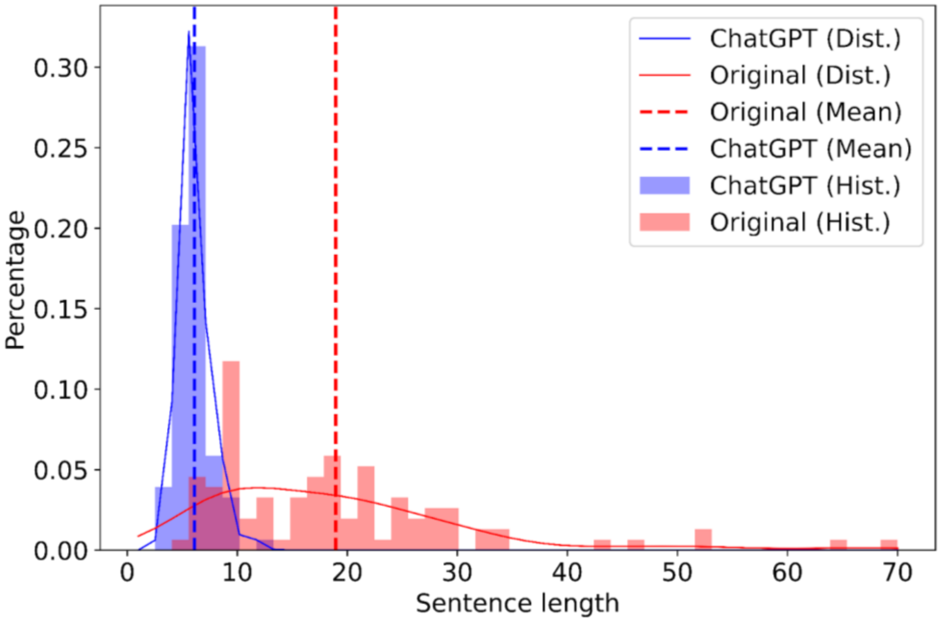

Figure 7 Sentence Count per Material's Type: Distribution Patterns and Group Averages

Figure 7 provides a clear visualization of the impact of ChatGPT's adjustments, revealing that the ChatGPT-generated content, exhibits a notably reduced average length compared to the original content. Specifically, the average length of ChatGPT content stands at 6.09 sentences, while the original content averages 18.94 sentences. Furthermore, an examination of the histogram illustrates that the distribution of sentence lengths in ChatGPT displays reduced variability, indicating a more focused range around the center when compared to its original counterpart. This is notably reflected in the standard deviation values, with the ChatGPT content registering at 1.40, in contrast to the substantial deviation observed in the original content at 11.91.

The findings suggest that, despite the absence of explicit commands for summarization during the ChatGPT content generation process, ChatGPT implicitly undertakes a summarization procedure to render the content more accessible for EFL readers. Notably, the distribution of sentence lengths in ChatGPT content consistently centers around four to eight sentences, in contrast to the original content, which exhibits a wide range of sentence lengths spanning approximately seven to 30 sentences per content. This trend signifies ChatGPT's consistent delivery of content tailored to a concise and manageable length, aligning to enhance readability for EFL learners.

Observing the consistent trend of ChatGPT returning content with a more uniform length, it is crucial to recognize a potential trade-off between concise readability and the preservation of detailed information, particularly as the length of the original content increases. It is noteworthy that while ChatGPT successfully narrows down the content length, there is a likelihood of information loss, particularly concerning nuanced details and complexities within the original text. As illustrated by the histogram in Figure 1, the distribution of sentence lengths in ChatGPT content is notably concentrated, indicating a tendency to simplify and condense the information. Therefore, for an English study that necessitates intricate details within the content, relying solely on ChatGPT for simplifying reading materials might not be optimal. The potential loss of detailed information could hinder the effectiveness of the materials in addressing the specific requirements of such a study. In these cases, it may be advisable to consider alternative strategies or a customized approach that balances readability improvements with the retention of critical content details.

4.3.2 Average Sentence's Length per Materials Type

Besides the average number of sentences, the number of words per sentence (sentence's length) also significantly influences the difficulty level of the material. For instance, having fewer sentences with longer word counts can present challenges for readers in understanding the content, as the extended structure may lead to cognitive overload and reduced clarity. Hence, to guarantee that the intended simplification by ChatGPT not only focuses on reducing the number of sentences but also promotes readability through appropriately balanced sentence lengths, we conducted an analysis of the number of words per sentence in each reading content. Figure 8 provides a visual representation of the distribution of sentence length in each content type, complete with their corresponding average values.

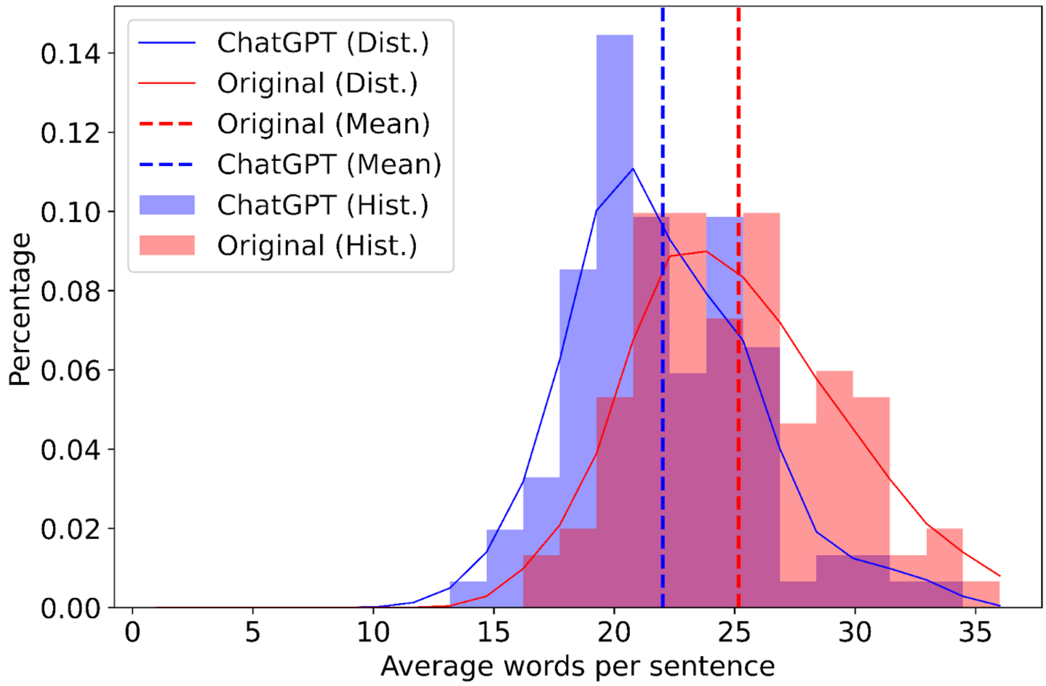

Figure 8 Sentence Length per Material's Type: Distribution Patterns and Group Averages

Analyzing Figure 8 reveals a distinction in the average sentence length between the content generated by ChatGPT and the original content. Specifically, the ChatGPT content exhibits a lower average sentence length at 22.01, in contrast to the original content's average of 25.17. Additionally, Figure 8 illustrates a consistent sentence length distribution per sentence for both contents. The standard deviation of word length per sentence in ChatGPT content is 3.66, while the original content records a value of 4.14. These results imply that the reader-friendly nature of content generated by ChatGPT extends beyond the paragraph level, resonating at the sentence level as well. Not only does it consistently uphold a lower average word length, but it also exhibits a more uniform distribution, further enhancing its overall ease of comprehension.

4.3.3 McAlpine EFLAW and Gunning Fog Readability Scores per Material's Type

The comparative analysis was extended using two readability metrics, namely the McAlpine EFLAW and Gunning Fog readability metrics. The McAlpine EFLAW metric assesses the number of mini-words in a sentence, acknowledging that sentences with fewer mini-words are generally more comprehensible to English as a Foreign Language (EFL) readers. Conversely, the Gunning Fog metric evaluates the ratio of complex to simple words in reading material, designating words with more than two syllables as complex. The distribution of scores and the average values for both metrics per material type are visually presented in Figure 9.

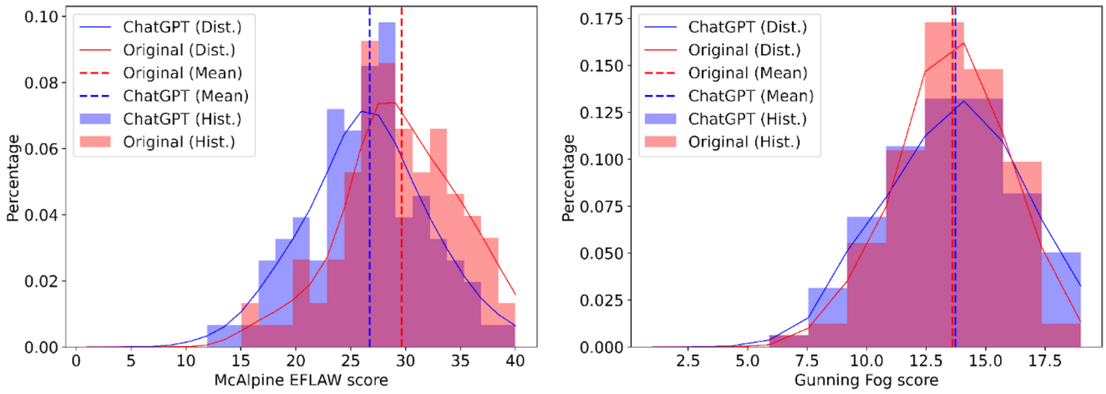

Figure 9 Readability Metrics per Material's Type: Distribution Patterns and Group Averages

Figure 3 shows that the ChatGPT content maintains a slightly lower average McAlpine EFLAW score compared to the original content. Specifically, the average McAlpine EFLAW score for ChatGPT content is 26.73, while the original content records 29.63. In contrast, the Gunning Fog scores for both types of content exhibit a marginal difference in average values. The ChatGPT and original content have average scores of 13.73 and 13.61, respectively. Additionally, the distribution of scores in both content types demonstrates similar patterns. Furthermore, examining the standard deviations provides insights into the variability within the scores. The standard deviation of the McAlpine EFLAW score for ChatGPT and original content is 5.7 and 5.3, respectively. Similarly, the standard deviation of the Gunning Fog score is 2.27 and 2.76 for ChatGPT and original content, respectively.

From the results presented above, it is evident that although ChatGPT engages in abstractive summarization—requiring the reformation and paraphrasing of sentences rather than mere extraction—it adeptly mirrors the difficulty level of the original text. Despite introducing its own vocabulary, the word choices exhibit a comparable syllabic complexity to the original content. It is important to note, however, that a fraction of the materials generated by ChatGPT is perceived as more challenging than the original content, with approximately 5% having Gunning Fog scores surpassing 17.5. This suggests a tendency toward the use of longer words in certain materials. Nevertheless, it's crucial to consider that the distribution, notably leaning towards lower scores in McAlpine EFLAW (tending to the left), presents an alternative interpretation. This tendency indicates that ChatGPT produces more materials with less complex structures, potentially posing challenges for international readers due to the reduced presence of mini-word clusters.

In conclusion, the analysis suggests that ChatGPT employs a strategy of utilizing shorter words to diminish the presence of mini-word clusters. However, as a trade-off, this approach leads to the production of materials featuring lengthier words, contributing to a higher Gunning Fog score in approximately 5% of the generated content. The nuanced interplay between these two complexity metrics underscores the intricate balancing act that ChatGPT navigates in its abstractive summarization process.

4.3.4 Discussion

In summary, the experiment showcased the adaptability of ChatGPT in tailoring reading materials for English as a Foreign Language (EFL) readers. Without explicit commands for summarization, ChatGPT intuitively engaged in such a process, simplifying the materials by reducing the number of sentences and adjusting sentence length. The consistent output of content with a lower number of sentences (averaging four to eight) demonstrates the potential benefits for general EFL learning. This consistency in complexity allows for diverse reading materials with similar cognitive loads, fostering effective teaching strategies. However, it is essential to acknowledge that for English studies requiring intricate details within the content, relying solely on ChatGPT for simplifying reading materials may not be optimal.

Moreover, the calculation of McAlpine EFLAW scores uncovered a notable shift in distribution within ChatGPT's content, displaying lower scores compared to the original content. This shift suggests enhanced comprehension for EFL readers, attributed to the reduced presence of mini words. However, in exchange for this beneficial effect, the approach results in the creation of materials with longer words. This contributes to a higher Gunning Fog score observed in around 5% of the generated content.

Despite the promising outcomes, it is important to underscore the necessity for additional research directly involving English as a Foreign Language (EFL) students. This research could investigate the effectiveness and acceptance of content generated by ChatGPT, aiming to capture EFL students' perceptions and preferences concerning adjusted reading materials. This nuanced understanding is essential for better integration of AI-generated content for EFL learning, ensuring it suits learners' distinct needs and preferences.

4.4 Evaluation of ChatGPT for Generating Chatbot's Dialogue for English as a Foreign Language Learning

4.4.1 Stimuli and Materials

To evaluate the possibility of using AI-generated materials for EFL learning, we generated a set of dialogues by using OpenAI’s ChatGPT. The dialogues were produced by inputting the following prompt to the bot: "Please help me to make a dialogue to help EFL students practice their English. The dialogue is between A and B. A is an undergraduate student at TDU University. B is an exchange student from Italy. The topic is {{topic name}}", where {{topic name}} was selected from Table 3.

Table 3 List of Topics for the AI-generated materials

Topic (1st – 5th)	**Topic (6th – 10th)**	**Topic (11th – 15th)**
Greet new exchange student	Fermented foods	Learn programming
Lunch Invitation	Sumo wrestling	Summer’s vacation
Play arcade on weekends	Coin Laundry	Traveling to Kyoto
Foods and hobbies	Favorite snacks	Buying new clothes
Learn to use chopsticks	Sightseeing in Tokyo	Last week in Tokyo

In the prompt above, the lines "The dialogue is between A and B. A is an undergraduate student at TDU. B is an exchange student from Italy" are intended to give context to the AI so it could create a livelier dialogue related to students. Furthermore, a series of topics in Table 4 means to test whether the ChatGPT can produce various topics for students to practice. On top of that, we asked ChatGPT to give two or three alternative lines of dialogue for each line in the produced dialogue. Later, in the experimentation, using a single dialogue from ChatGPT, 30 unique dialogue choices will be simulated. Therefore, 450 unique sample combinations of dialogues will be analyzed.

4.4.2 Measurement Procedure

The analysis procedure began by calculating three readability metrics: Flesch Reading Ease, McAlpine EFLAW, and Dale-Chall based on each simulated dialogue. The Flesch Reading Ease metric measures dialogue difficulty by examining the ratio of polysyllables to all words, with a higher count indicating assumed complexity. Conversely, the McAlpine EFLAW metric assesses complexity through the inclusion of mini-words, assuming higher complexity with increased use of such terms. Lastly, the Dale-Chall metric incorporates a list of challenging words from prior studies.

The utilization of these three metrics serves to address the individual limitations inherent in each metric. While the Flesch Reading Ease metric solely focuses on the count of polysyllables in its calculation, it falls short of recognizing how a cluster of mini-words might pose challenges for international readers and impede translation efforts. Consequently, the McAlpine EFLAW readability score calculation process is incorporated, considering such intricacies. Additionally, the Dale-Chall metric plays a role in assessing the text's difficulty level, particularly emphasizing words with few polysyllables that remain challenging to comprehend, such as "abide," "deem," and "quail."

Building upon the scores generated by each metric, we further the process of interpreting the difficulty level of the dialogue by visually exploring the distribution of difficulty levels within the simulated dialogues. Through an analysis of the results obtained from the visualization, we delve into a more detailed examination to ensure the complexity of the dialogues generated by ChatGPT.

4.4.3 Results and Discussion

Based on the simulated dialogues, the Flesch Reading Ease score for each sample was first calculated. Then, through the resulting scores, a visualization was carried out to show the scores' central tendency and distribution from all samples. Figure 3 shows the distribution of scores from all samples.

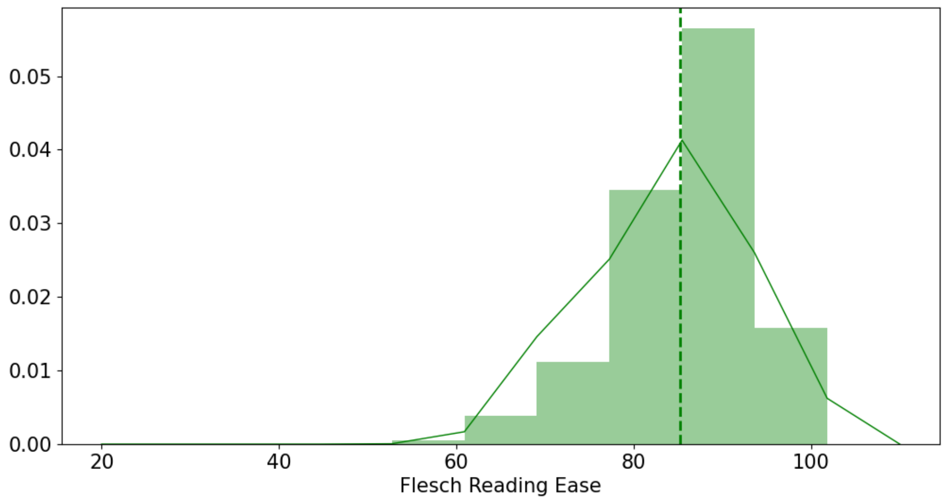

Figure 10 Flesch Reading Ease Scores from Generated Materials: Distribution Patterns and Average

Figure 3 illustrates a score distribution among the sample dialogues ranging from 60 to 100. The majority of samples fall within the 80-90 range, boasting an average score of 84.96 and a standard deviation of 8.48. Consequently, it is discerned that most simulated dialogues are easily comprehensible for sixth-grade elementary school students. Notably, a small yet significant portion, constituting about 5%, falls within the 60-80 score range, making them suitable stimuli for junior high school students.

However, it is worth noting that these generated materials may not be optimal for senior high school students or those in higher education, as their comprehension level surpasses the challenge presented by such content. This interpretation aligns with a broader consideration for English as a Foreign Language (EFL) students, corroborated by insights from a previous study [64]. Given that the majority of samples fall within the 80-90 score range, students with a Common European Framework of Reference for Languages (CEFR) level of A2 (elementary) stand to benefit the most. While the materials might still be suitable for students with CEFR levels A1 (beginner) and B1 (intermediate), those at levels B2 (upper intermediate) to C2 (advanced) might find the materials less challenging and overly simplistic.

Subsequently, readability scores according to the Dale-Chall formula were computed for all sample dialogues to assess various complexity criteria. Figure 4 depicts the distribution of the resulting scores across all samples.

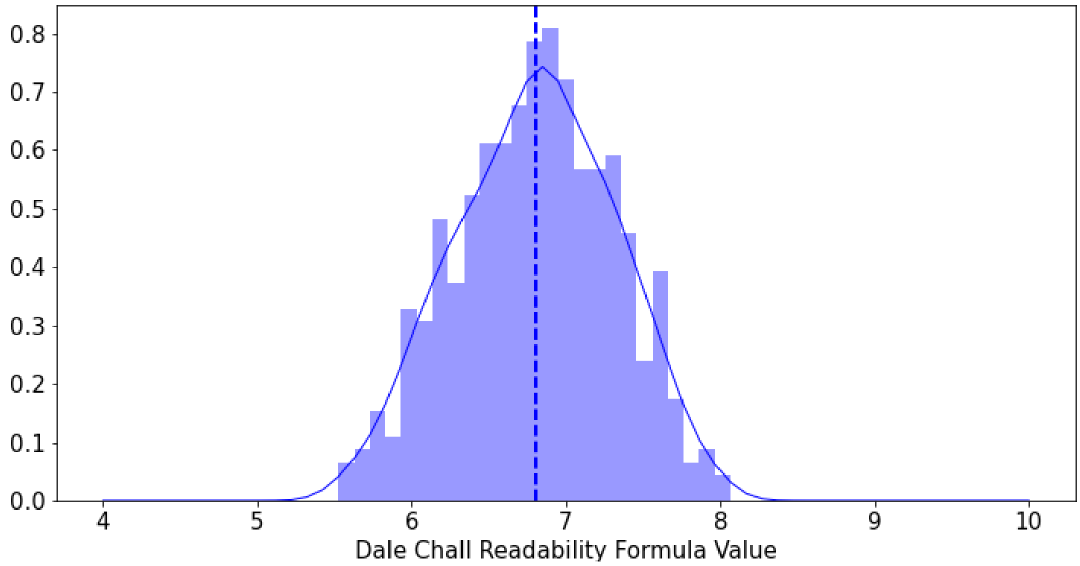

Figure 11 Dale Chall Scores from Generated Materials: Distribution Pattern and Average

Similarly, to the previous interpretation, based on the distribution of resulting scores shown in Figure 4, it can be argued that the generated materials are most suitable for sixth-grade elementary school students or students in the early years of junior high school (CEFR A2 and B1). Moreover, the absence of samples with Dale-Chall scores above 8.0 confirms that the generated materials are unsuitable for students with CEFR levels B2 to C2.

Finally, the McAlpine EFLAW score was calculated for each simulated conversation. The visualization of the score distribution can be observed in Figure 5.

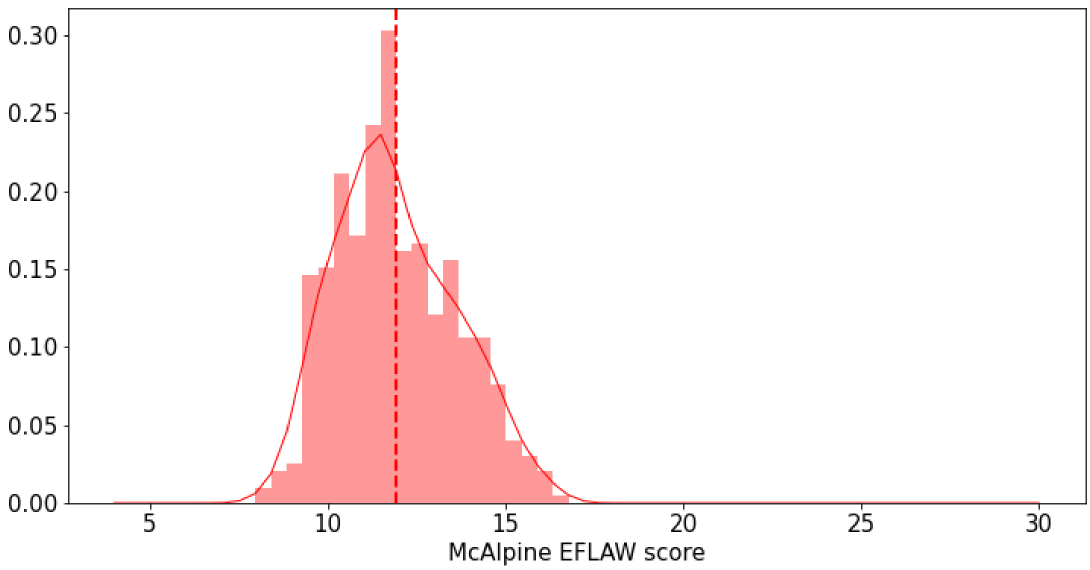

Figure 12 McAlpine EFLAW Scores from Generated Materials: Distribution Pattern and Average

Referring to the resulting scores in Figure 5, as none of the generated material has a score above 20, it can be interpreted that the resulting materials do not extensively utilize mini-words that could confuse EFL students when consuming them. The result even suggests that all simulated dialogue is very easy to understand.

Combining these results altogether, several conclusions can be drawn regarding the suitability of ChatGPT-generated materials as EFL chatbot reference dialogues. Firstly, the minimal McAlpine EFLAW score observed in all simulated conversations suggests that the dialogues generated by ChatGPT do not contain excessive use of mini-words. This indicates that wordy clichés, colloquial expressions, and phrasal verbs, which could potentially confuse international readers, were avoided in the resulting dialogue. The consistently low McAlpine EFLAW scores across all simulated dialogues indicate that EFL students can easily comprehend and understand the content. These findings provide confidence in the appropriateness of ChatGPT-generated materials as reference dialogues for EFL chatbot systems.

Furthermore, the Flesch Reading Ease scores obtained affirm that the majority of ChatGPT-generated materials are most suitable for students with CEFR levels A2. This interpretation is further reinforced by the Dale-Chall scores derived from the simulated dialogues. Despite the Dale-Chall score calculation considering different criteria than the Flesch Reading Ease formula, a parallel interpretation is drawn, supporting the alignment of ChatGPT-generated materials with the proficiency levels outlined in CEFR level A2. Delving deeper into the characteristics defined for CEFR level A2, these materials are particularly well-suited for students who exhibit the following proficiencies.

1. Vocabulary: Grasp most everyday words and phrases related to personal information and basic needs, along with many words and phrases associated with hobbies, travel, and work.

2. Grammar: Understand simple grammatical structures (e.g., present and past tenses) and basic question forms.
3. Reading: Proficient in reading short and straightforward texts, such as simple stories, with the aid of a dictionary.
4. Writing: Capable of composing basic sentences and short texts detailing personal experiences or daily routines.
5. Listening: Comprehend simple and direct information in everyday conversations or short speeches on familiar topics.
6. Speaking: Engage in basic conversations, asking and answering questions about personal details, preferences, requests, or suggestions.

4.5 Evaluation of ChatGPT for Generating Chatbot's Dialogue for English as a Foreign Language Learning

Building upon the promising outcomes of the previous experiment, this experiment digs into the analysis of various prompt techniques and their impact on the quality and characteristics of generated dialogues. The focus is particularly on creating dialogues for a specific target audience within a specific learning context. For this purpose, imagine a scenario where our aim is to craft dialogues for the practice of senior high school students. These students, having studied English as their second language from elementary through senior high school, possess an English proficiency level of approximately CEFR B2.

In the context of paired practice sessions, it becomes imperative that the generated dialogues involve precisely two individuals. Considering the time constraints inherent in the course, the objective is to create dialogues with a duration of 2-3 minutes. This duration allows all students to engage in practice in front of the class, enabling their peers to attentively listen and take notes for each pair. Therefore, in such scenarios, it is anticipated that each dialogue will consist of precisely 14 lines, aligning with the specified goal. Furthermore, to enhance comprehensibility and maintain manageable pacing, a maximum of approximately 10 words per line is stipulated.

In accordance with the described context, the following dialogue specifications are established:

1. The dialogue should exclusively involve a conversation between two participants.
2. The generated dialogue should be suitable for EFL students with CEFR B2 proficiency.
3. The dialogue should consist of exactly 14 lines.
4. Each line of dialogue should contain a maximum of approximately 10 words.

However, our preliminary experimentation has revealed that ChatGPT naturally generates dialogues for two participants, even without explicitly specifying the number. Consequently, we will set aside the first criterion. Other than that, the remaining specifications will serve as guiding principles to evaluate how different prompts can impact the quality of ChatGPT-generated dialogues.

4.5.1 Prompt Strategies for Dialogue Generation

To generate dialogues that are suitable for our target audience at the CEFR B2 level, it is crucial to specify appropriate topics for the bot. Rather than relying solely on our own brainstorming process to determine suitable topics, we will leverage the capabilities of ChatGPT itself. By requesting ChatGPT to generate a list of topics for us, we employ a specific prompt technique known as meta prompting. This technique involves utilizing the generated list of topics as part of the subsequent prompt for dialogue generation. This approach not only saves time and effort but also has the potential to enhance the quality of the generated dialogues. Since the model itself produces the list of topics, it is likely already familiar with them, which can contribute to more coherent and contextually appropriate dialogues. By employing the prompt, "Generate 30 topics that are suitable for CEFR B2 student's dialogue practice" we obtained a set of topics from ChatGPT. The following is the list of topics generated by the model in response to the prompt.

Table 4 List of Topics Generated Through Meta-prompting

No.	**Topic Name**	**No.**	**Topic Name**
1.	Discussing your favorite book or movie	16.	Planning a dinner party or potluck
2.	Planning a weekend getaway	17.	Discussing a controversial political topic
3.	Sharing your experiences with public transportation	18.	Sharing your experiences with a health issue
4.	Debating the pros and cons of globalization	19.	Talking about your favorite cuisine or restaurant
5.	Talking about your career aspirations	20.	Planning a fundraising event for a charity
6.	Discussing a current event or news story	21.	Discussing your views on animal rights
7.	Planning a party or event	22.	Sharing your experiences with volunteering
8.	Sharing your thoughts on climate change	23.	Talking about your favorite type of music
9.	Discussing personal finance and money management	24.	Planning a DIY project
10.	Talking about your favorite hobby or pastime	25.	Discussing a personal challenge or setback
11.	Planning a trip abroad	26.	Sharing your thoughts on gender roles and equality
12.	Debating the merits of different forms of exercise	27.	Talking about your favorite TV show or series
13.	Discussing your views on education	28.	Planning a camping or outdoor adventure
14.	Sharing your experiences with social media	29.	Discussing your views on social justice issues

15.	Talking about your favorite form of art (painting, music, theater)	30.	Sharing your experiences with immigration or cultural exchange'

Aside from meta prompting, we will explore and experiment with various other prompt techniques including direct task specifications, task demonstration, mimetic proxy, and constraining behavior. By employing these different prompt techniques, our objective is to assess their influence on the dialogues generated according to the predefined specifications.

To initiate the experiment, we will start by utilizing a Base Prompt as shown in Table 5. This prompt aims to simulate a typical request made by teachers to ChatGPT for the generation of dialogue practice materials. We intentionally omitted specific information about the target audience's proficiency level and other dialogue specifications to highlight the significant differences in dialogue results that can occur when such details are not provided.

Table 5 List of Prompt Techniques and Their Definitions

Prompt Technique	**Text Provided to ChatGPT**
Base Prompt	*Hi ChatGPT, please help me to generate a dialogue between A & B. The dialogue topic is {topic_name}. The dialogue will be used for dialogue practice between two students.* * {topic_name} will be replaced by topics from 4.
Direct Task Specification	*Do a task with the following details.* *Task: "Generate a dialogue between A & B"* *Topic: "{topic_name}"*
Task Demonstration	*Do a task with the following details.* *Task: "Generate a dialogue between A & B"* *Topic: "{topic_name}"* *Example: "A: Hey, B, have you got a minute? I've got a small favor to ask.* *B: Right.* *A: So, how would you feel about DJing at the office party next week?* *B: Office party? I don't usually do work parties.* *A: Oh, right. So where do you usually DJ then?* *B: No, I mean I don't usually go to work parties, let alone DJ at them.* *A: Come on, I think you'd be brilliant at it!* *B: Oh, I don't know. I think I'm probably busy that day anyway.* *A: Come on! There's nothing to lose!* *B: Except for my reputation and credibility.* *A: Paul, you'd really be helping me out.* *B: OK, I'll think about it.*

	A: Great! Thanks, B!
Mimetic Proxy	*Do a task with the following details.* *Act as: "EFL teacher"* *Task: "Generate a dialogue between A & B"* *Topic: "{topic_name}"*
Constraining Behaviour	*Do a task with the following details.* *Task: "Generate a dialogue between A & B"* *Topic: "{0}"* *Audience English Level: "CEFR B2"* *Criteria of the line of dialogue: "about 10 words max"* *Total number of lines: "14 lines"*

Later, we will compare materials generated using different prompt techniques with Base Prompt resulting materials. This comparative analysis will be conducted across a range of dialogue specifications, allowing us to measure the benefits of prompt techniques.

From Table 4, it is apparent that we have adopted a DTS-like format for each prompt employed in the task demonstration, mimetic proxy, and constraining behavior techniques. This deliberate decision stems from the fact that these prompt techniques can be categorized as direct task specifications, as they provide explicit guidance to the model for achieving the desired results. Moreover, for the Task Demonstration prompt, the dialogue included within is specifically designed for CEFR B2 level practice. By presenting this dialogue as a demonstration, we aim to guide ChatGPT in generating responses that align with the given task. As for the Mimetic Proxy prompt, the additional information of "Role: EFL teachers" is included to guide ChatGPT in adopting a specific behavior and response style consistent with the designated role. By explicitly specifying the role of EFL teachers, we aim to encourage ChatGPT to generate dialogues that reflect the language, tone, and instructional approach typically employed by EFL teachers. In the case of the Constraining Behavior prompt, we take a more direct approach by explicitly defining all the specifications and guidelines that ChatGPT needs to adhere to during the dialogue generation process.

Additionally, besides evaluating each prompt technique individually, we have designed two additional prompts that combine multiple techniques as can be seen in Table 6. As shown in Table 6, we incorporate all the prompt techniques. The purpose of this prompt is to observe the performance and output quality when all prompt techniques are used together. In the second prompt, we intentionally exclude the Task Demonstration technique while using the remaining prompt techniques, based on the assumption that the inclusion of Task Demonstration may introduce confusion or potential bias to the model's responses.

Table 6 Two Additional Prompts that Combine Multiple Prompt Techniques

Prompt Technique	**Text Provided to ChatGPT**
Combination of all prompt techniques	*Do a task with the following details.* *Task: "Generate a dialogue between A & B"* *Topic: "{0}"* *Act as: "EFL teacher"* *Audience English Level: "CEFR B2"* *Criteria of the line of dialogue: "about 10 words max"* *Total number of lines: "14 lines"* *Example: "A: Hey, B, have you got a minute? I've got a small favor to ask.* *B: Go on then.* *A: So, how would you feel about DJing at the office party next week?* *B: Office party? I don't usually do work parties.* *A: Oh, right. So where do you usually DJ then?* *B: No, I mean I don't usually go to work parties, let alone DJ at them.* *A: Come on, I think you'd be brilliant at it!* *B: Oh, I don't know. I think I'm probably busy that day anyway.* *A: Come on! There's nothing to lose!* *B: Except for my reputation and credibility.* *A: Paul, you'd really be helping me out.* *B: OK, I'll think about it.* *A: Great! Thanks, B!"*
Combination of all prompt techniques except Task Demonstration	*Do a task with the following details.* *Task: "Generate a dialogue between A & B"* *Topic: "{0}"* *Act as: "EFL teacher"* *Audience English Level: "CEFR B2"* *Criteria of the line of dialogue: "about 10 words max"* *Total number of lines: "14 lines"*

Moreover, please refer to Figure 1 to understand more about the experiment setup.

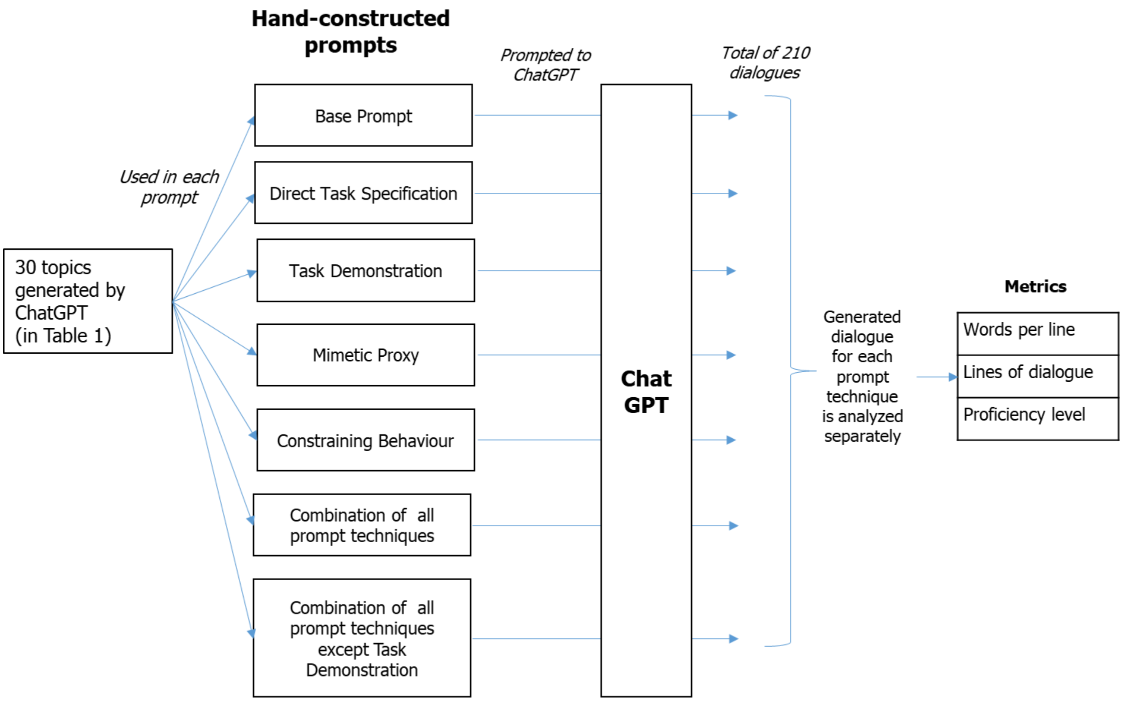

Figure 13 Visualization of The Experiment Setup

Through meta prompting, we curated a list of 30 suitable dialogue topics tailored to the CEFR B2 level, ensuring their relevance and appropriateness for language learners. Our analysis will encompass a total of 210 dialogues, examining seven distinct prompt settings. These settings include the Base Prompt, four individual prompt techniques (Direct Task Specification, Task Demonstration, Mimetic Proxy, and Constraining Behavior), as well as two combined prompts—one incorporating all techniques and the other excluding Task Demonstration. This approach allows us to thoroughly evaluate the impact of each prompt technique on the quality of the generated dialogues and how they could be used altogether.

4.5.2 Measurement Procedure

When evaluating the appropriateness of generated dialogues for a specific CEFR (Common European Framework of Reference for Languages) level, there is no direct implementation that can precisely measure the CEFR level based on the input text. Therefore, we employ a set of readability metrics to assess the suitability of the generated dialogues. These metrics include Flesch Reading Ease, SMOG, Coleman Liau, Automated Readability Index, Dale Chall, Linsear Write, and Gunning Fog. By subjecting the dialogues to these readability metrics, we generate scores that capture various aspects of text complexity. Subsequently, a consensus-based approach is employed to ascertain the optimal target audience for each dialogue. For this purpose, we utilize the Python implementation of the textstat library. Since the readability metrics primarily return recommended school grade levels, we establish an assumption for the CEFR level equivalence. Specifically, we consider school grades below the sixth grade as equivalent to the CEFR A2 level, junior high school grades as equivalent to the CEFR B1 level, and senior high school grades and higher education as equivalent to the CEFR B2 level. This mapping enables us to align the readability metrics' results with the appropriate CEFR levels.

4.5.3 Results and Discussion

The following table presents the experimentation results, showcasing the performance of different prompt techniques for generating EFL dialogue. In the table, we will use the following abbreviations to represent the performance results for different prompt techniques.

BP: Base Prompt
DTS: Direct Task Specification
TD: Task Demonstration
MP: Mimetic Proxy
CB: Constraining Behavior
ALL: All Prompt Combinations
ALL-TD: All Prompts except Task Demonstration

Table 7 Performance of Different Prompt Techniques for Generating EFL Dialogue

	BP	**DTS**	**TD**	**MP**	**CB**	**ALL**	**ALL-TD**
Words Per Line							
- Average	19.24	20.28	16.88	19.47	**9.44**	9.88	9.80
- Median	18	19	16	17	**9**	**9**	**9**
- Below 10 (%)	17.07	15.64	18.54	17.46	**69.35**	62.96	63.50
Lines of Dialogue							
- Exactly 14 (Count)	6	1	4	4	6	7	**10**
- Exactly 14 (%)	20	3	13.33	13.33	20	23.33	**33.33**
Proficiency Level							
- School Grade (Average)	6.0	**6.43**	5.97	6.13	5.8	5.83	5.53
- A2 (Count)	16	17	18	16	**20**	18	18
- B1 (Count)	**14**	10	10	12	8	10	11
- B2 (Count)	0	**3**	2	2	2	2	1

Table 7 presents various quantitative metrics according to the dialogue specifications for different prompt techniques. The results from the CB prompt indicate ChatGPT's understanding of the maximum words per line specification. In contrast, in other cases where the specification is not explicitly mentioned, the average number of words per line is significantly higher. However, it is important to note that at best (The CB case), only 69.3% of the dialogues generated met this specified criterion. Furthermore, as additional specifications are introduced through different techniques (A1 and A2), we observe that ChatGPT starts producing more dialogue with lines of dialogue more than

10 words. This suggests that ChatGPT may start to compromise on the maximum words per line criteria to fulfill other specified criteria.

Meanwhile, the results for words per line in the TD prompt indicate that ChatGPT struggles to learn the implicit criteria provided in the example dialogue. Despite each line of the example dialogue having a word count below 10, the resulting dialogues from the TD prompt exhibit an average number of words per line that is significantly higher (with a median of 16 and only 18.54% of lines containing fewer than 10 words). These findings highlight how ChatGPT struggles to fully grasp and replicate the implicit criteria from an example dialogue.

Similarly, the higher percentage of dialogues that met 14 lines of dialogue criteria in CB, C1, and C2 prompts demonstrate ChatGPT's understanding of explicit dialogue specifications. Interestingly, while the CB prompt yielded the best dialogue results in terms of word-per-line criterion, more dialogues from the A2 prompt met the number of lines of dialogue criterion (33.3%). This result might suggest that ChatGPT might prioritize the specified criteria in the prompt differently when extra context is provided (role as EFL teacher in A2 prompt). Nonetheless, even the most effective prompt employed in the experiment only led to 10 dialogues that met the specific line count. This implies that ChatGPT may better comprehend lower-level criteria better than higher ones.

Moreover, by looking at the distribution of the appropriateness of proficiency levels from the generated materials across all prompts, none of the prompt techniques used were able to generate dialogues suitable for the intended proficiency level. The dialogues produced by the prompt technique that achieved the best results, DS, only reached an intended school grade of 6.43, which aligns with sixth-grade to first-year junior high school students (A2 - B1 level). Other prompt techniques resulted in dialogues that were even further below the intended proficiency level. Despite any techniques used, ChatGPT failed to learn and replicate the implicit requirements. Out of the resulting dialogues, only about one or two dialogues were identified as potentially suitable for CEFR B2. This number is even smaller than the DS prompt which doesn’t contain any information regarding the intended target audience. Therefore, we concluded that ChatGPT couldn’t understand the concept of proficiency level.

The findings indicate that using prompt techniques can influence the quality of the generated dialogues but only for lower-level criteria. When the maximum words per line specification were explicitly provided, ChatGPT could produce more materials that met the required criterion. However, it is important to note that not all generated dialogue lines adhered to this criterion (with at max only 69.35% compliance). Moreover, the TD prompt, which aimed to implicitly instruct the criteria from an example dialogue, posed challenges for ChatGPT. The resulting dialogues exhibited a higher average words per line count, indicating the struggle to grasp and replicate the implicit criteria accurately. Nonetheless, when compared to the results from a regular prompt, the TD prompt performed slightly better.

Meanwhile, for higher-level criteria, despite the prompt techniques used, none could generate a satisfying result. ChatGPT failed to produce dialogues at the intended proficiency even when explicit instruction was given. Such limitations could be attributed to the training process of ChatGPT. As ChatGPT is trained to facilitate conversations with a diverse audience, it may unintentionally receive positive reinforcement for providing more straightforward, easily comprehensible responses. Consequently, this unintentional inclination towards simplicity could limit ChatGPT's ability to offer more advanced materials to higher-proficiency students.

Chapter V Artificial Intelligence-based Speaking Practice Application

Over the course of the three preceding chapters, our investigation traversed the landscapes of text-to-speech (TTS), speech recognition (SR), and generative AI, revealing their potential to support English as a Foreign Language (EFL) students in their learning process. The audio quality produced by a TTS system, specifically WaveNet TTS, was compared to that of native speakers, revealing a surprising finding—despite being perceived as less natural, TTS audios proved to be more comprehensible than their human counterparts. Additionally, we assessed the capabilities of Vosk, an offline speech recognition system, revealing its accuracy in successfully detecting the speech of EFL students. Interestingly, feedback from participants uncovered a notable challenge: the unease experienced during speech practice with speech recognition (SR), leading to the need for a slower speech rate to enhance clarity. Moreover, the next experimentation involving ChatGPT for EFL dialogue practice revealed the most appropriate proficiency levels for the generated materials, primarily tailored for A1 and A2 dialogue practice. Further investigation into controlling the generated output, with a specific emphasis on dialogue specifications, highlighted difficulties in prompting ChatGPT to produce dialogues at a B2 proficiency level. Nevertheless, success was achieved by concentrating on more straightforward criteria such as word count and total lines within a dialogue.

In this chapter, we integrate various AI technologies into a unified application, demonstrating how they collectively assist EFL students. By employing ChatGPT, the application generates a series of dialogues, enabling students to choose topics and refine their speaking abilities using Vosk as a speech recognition system. The application provides a comprehensive language learning experience by responding to students in both text and audio formats, utilizing WaveNet TTS. To assess the effectiveness of the application, a pilot study will be conducted. Participants will be invited for a specific period of experimentation, during which their English-speaking proficiency will be evaluated before and after engaging with the application. In the upcoming chapters, detailed information regarding the implementation and test settings will be further explored.

5.1 Related Works

There are two types of chatbots: text-based and voice-enabled chatbots classified based on their modality. Voice-enabled chatbots have been proposed as a helpful tool for learning and practicing a second language (L2) speaking skills. A study in [65] discovered that L2 learners appreciated the chatbot's capacity to expose them to various conversational expressions and vocabulary and enable repetitive practice. On top of that, L2 learners prefer chatbots over human partners due to their fear of making mistakes and appearing incompetent during interactions with human partners [43].

A recent study by Han [7] showed how Alexa, a general voice-enabled chatbot, could help students by engaging them in conversations to practice their speaking skills. The experimentation indicated that chatbot-assisted learning improved students' pronunciation and language fluency. Moreover, post-questionnaire responses showed that the integration of such chatbot positively impacted students' interest in learning and enhanced their motivation to learn. Similarly, using readily available chatbots such as Google Assistants [66], [67], and Alexa [68], [69] also led to positive improvements in students' language proficiency. Researchers noted that students felt less embarrassed and anxious when practicing with a chatbot [66], [67]. Furthermore, chatbots promote self-directed foreign language learning outside school settings where native speakers are hard to find [68], [69].

Although general chatbots may seem appealing, several studies have suggested that such system adaptation may be less effective for L2 learners as it may not cater to their specific needs [25], [26]. Therefore, several criteria should be considered when designing a chatbot for language learning, such as language learning potential, learners' suitability, and authenticity [25]. The language learning potential criterion can be further broken down into components like interactional modification and task focus. Secondly, the learners' suitability criterion should consider various factors, such as language proficiency, age, learning style, and individual characteristics. Lastly, the authenticity criterion indicates that the materials presented within the chatbot should imitate real-life tasks that learners are likely to encounter.

A previous study [25] that implemented a task-oriented chatbot for helping students in their learning journey yielded promising results. The chatbot could maintain lengthy English conversations and engage in L2 problem-solving tasks with participants. Researchers noted that this type of speaking experience is hard to provide in regular EFL classes due to class size and time constraints within the curriculum. Similarly, an evaluation of a specifically designed EFL chatbot in [27] demonstrated the significant potential of its adaptation. The study found that the chatbot matched students' learning styles and enabled them to learn ubiquitously, thus making them enjoy their learning experience. Regardless, the study's pre-test and post-test settings revealed no significant improvement in students' Oral Proficiency Interview – Computer (OPIc) scores after the system adoption. The mixed findings in chatbot research indicate a need for further investigation.

5.2 Research Methodology

To assess the effectiveness of the AI-based speaking practice application, we will begin by presenting detailed information about the application in the Stimuli and Materials subsection. Following that, we will present specifics regarding the participants involved in the experiment and their engagement with the application in the subsequent subsection. Finally, the last subsection will provide an explanation of the criteria and procedures utilized to measure the app's effectiveness.

5.2.1 Stimuli and Materials

Figure 14 provides an initial glimpse of the application, showcasing its overall interface.

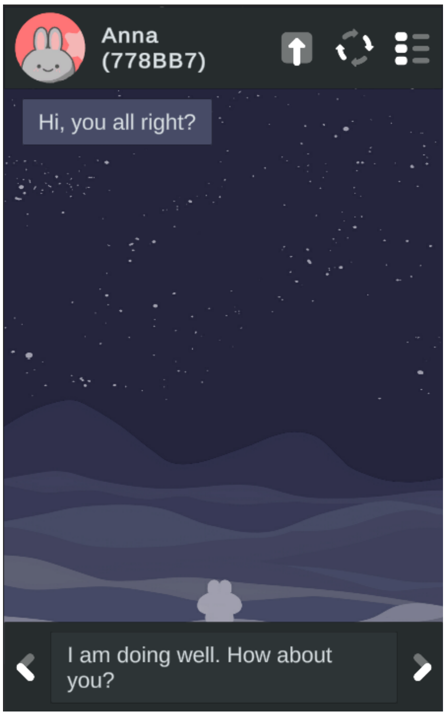

Figure 14 First Look into The Application

As depicted in this figure, the application follows a straightforward design reminiscent of a messenger app, facilitating communication between two individuals. Moreover, as depicted in this figure, the bot consistently takes the lead by always initiating the conversation with the user. Notably, the bot responses are presented in left-aligned bubble chats, while, as observed later, the user responses are positioned in right-aligned bubble chats.

Aside from the text response, the app also provides an auditory response. With the appearance of a text response within a bubble chat, the application will also audibly read aloud the text content, allowing students to listen and engage with the spoken material. All audio responses generated by the bot were pre-recorded using Google Wavenet TTS with an American English accent and female-like characteristics. The resulting audio files were saved in MP3 format. Although Wavenet allows adjustment of the words per minute (speech rate) using the 'speaking_rate' parameter, during audio generation, the parameter was maintained at its default value of 1.0. This decision was based on earlier small trials, which indicated that altering the speech rate either slower or faster resulted in more unnatural audio qualities.

Returning to the top part of the screen, a unique user ID is visible alongside the bot's name. This unique identifier serves to trace user engagement without compromising any significant user information. This approach ensures privacy while still allowing for effective app usage analysis. Additionally, three icons occupy the interface's top section, each serving a distinct function. From left to right, these icons enable users to navigate to the top of the screen, restart a conversation with the bot, and select a specific conversation topic.

In the app, user interaction with the bot involves utilizing the left and right arrow buttons that are located at the bottom part of the screen. These buttons allow users to select their desired response, facilitating a more dynamic conversation as illustrated in the figure below.

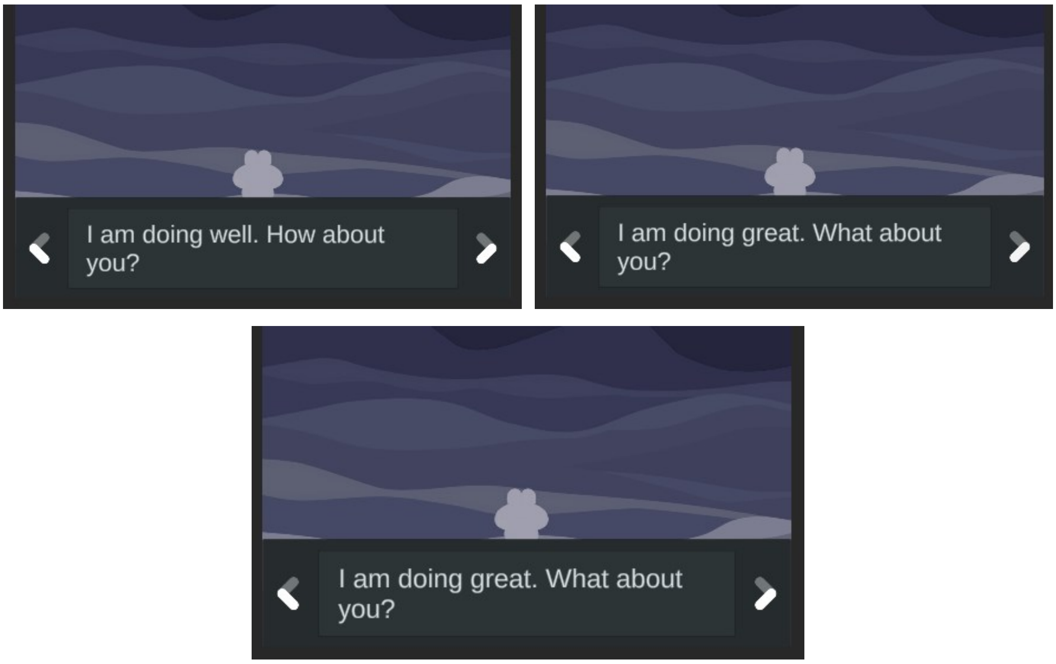

Figure 15 The Visualization of User's Response Selection

Once the user has made their selection, they can seamlessly proceed with the interaction by tapping the text field and a new right-aligned bubble chat will appear on the screen as in Figure 16.

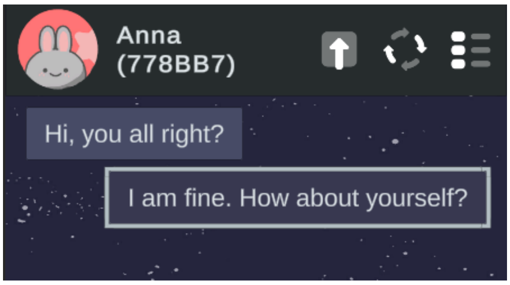

Figure 16 The App Renders User's Selection into a New Right-aligned Bubble Chat

After selecting their response and tapping the newly appeared right-aligned bubble chat (as demonstrated in Figure 16), users can seamlessly transition to the read-aloud activity. Upon tapping the bubble chat, a sound visualizer panel emerges from the bottom part of the screen, signaling the recording and transcription process for the user's voice input as can be seen in Figure 16.

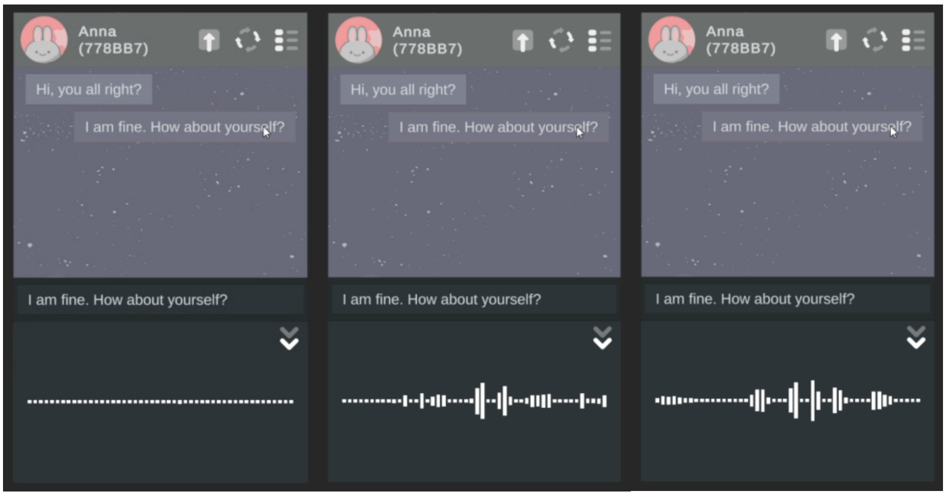

Figure 17 Sound Visualizer for User’s Voice Input Visualization

Once the users finish speaking, users can proceed to tap the double-down arrow icon situated at the top right of the panel, initiating a feedback mechanism for their speech input. The application responds by re-rendering the corresponding bubble chat, employing different colors to denote the accuracy of the user's pronunciation in comparison to the transcription result, as illustrated in Figure 18.

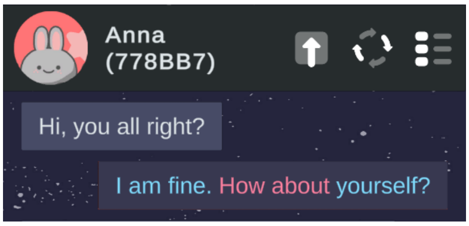

Figure 18 Different Colors to Denote the Accuracy of User’s Pronunciation

In this color-coded feedback system, words recognized accurately by the speech recognition (SR) model are rendered in blue. On the other hand, words present in the bubble chat but absent in the transcription result are rendered in red. This color scheme serves as a visual feedback mechanism, encouraging users to focus on and evaluate their pronunciation of words that the application couldn't recognize. This interactive approach aims to enhance the user's self-assessment and pronunciation refinement during the language learning process.

To accomplish accurate transcription, the application employs the Vosk Lightweight wideband model designed for Android and RPi, boasting a compact size of 67.6 MB and requiring approximately 300 MB of runtime memory. Throughout the recording process, the model actively transcribes the user's speech in real-time, ensuring a swift return of the transcription result as soon as the user concludes their speech. Notably, the model considers three alternative phrases while processing the audio input, with the app proceeding with the one having the highest probability. To accommodate the short format of all responses, a maximum recording length of 30 is set. Despite the short format of all available responses, the maximum recording length is set to account for additional buffer time in case the user speaks slowly.

After receiving feedback on their speech from the app, users can seamlessly proceed with the conversation. The bot will dynamically continue the interaction, allowing users to once again choose their responses through the intuitive interface described earlier. This conversational flow persists until the conclusion of the interaction, as shown in Figure 19.

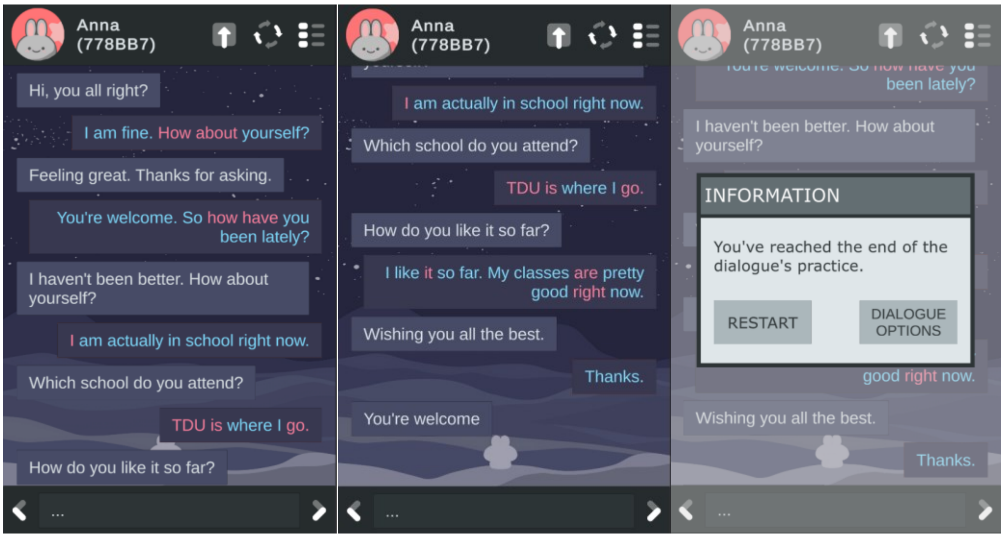

Figure 19 Different Colors to Denote the Accuracy of User’s Pronunciation

When reaching the end of the conversation, a panel will appear, providing users with options for further engagement. Users can decide to either restart the conversation, reinforce their practice with the same topic, or opt for a new conversation topic to diversify their learning experience.

If the users choose to explore new topics, they can tap the "DIALOGUE OPTIONS" button within the panel. This action triggers the appearance of a topic list panel from the right side of the screen, seamlessly covering the entire display as can be seen in Figure 20.

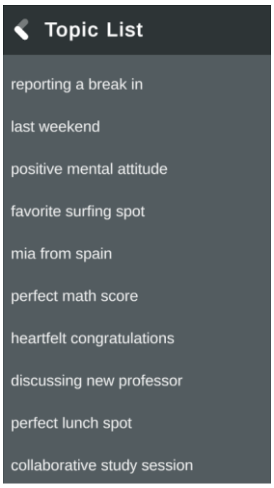

Figure 20 List of Topics Available for Users to Practice

This list comprises a total of 23 dialogue options within the app. Users can easily navigate through the list by scrolling up and down, allowing for a diverse selection of conversation topics.

The first ten dialogues are curated from ESLFast, an English Teaching and Learning platform renowned for its conversational practice resources. The remaining dialogues are generated using ChatGPT, with specific topic keywords prompting the model to create 12 to 16 lines of dialogue between two individuals. ChatGPT was further prompted to generate three alternative responses for each individual line while retaining the entities discussed. In instances where ChatGPT produced text outputs not meet our expectations, a filtering process was done, and the request was restarted.

To emphasize how each technology works within the app, Figure 21 provides a visual summary of the key technologies utilized in the application. The Vosk model ensures accurate real-time speech transcription, the Google WaveNet TTS delivers high-quality, natural-sounding audio responses, and ChatGPT generates diverse and contextually appropriate dialogue options, enhancing the overall user experience.

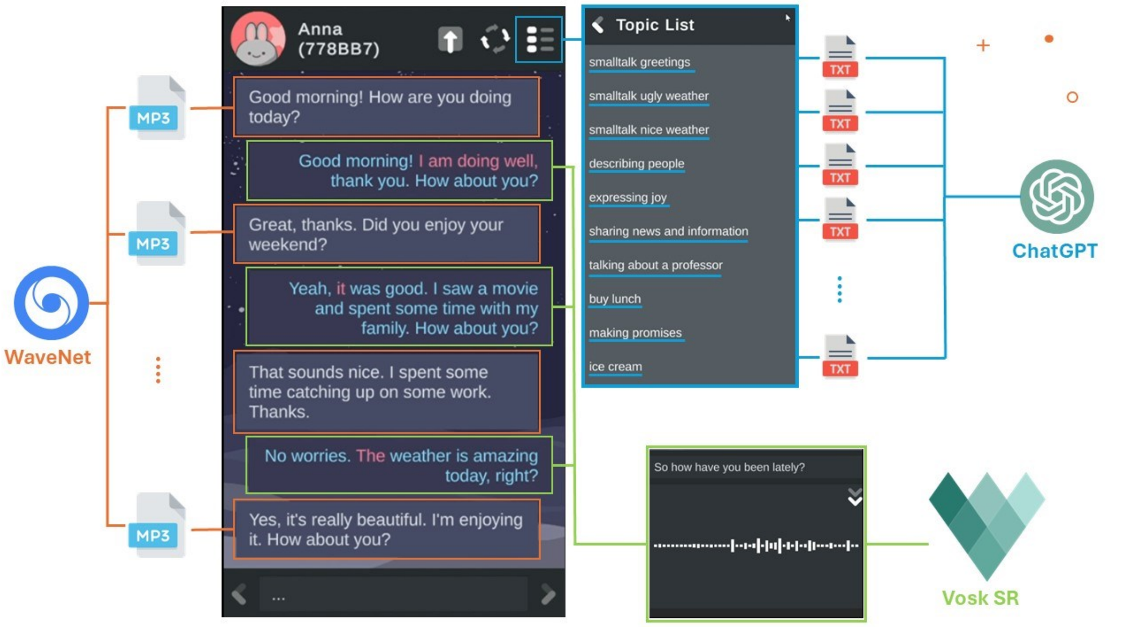

Figure 21 System Diagram to Demonstrate How Each Technology Contributes

5.2.2 Participants and Procedure

In order to assess the effectiveness of the developed language learning applications for English as a Foreign Language (EFL) students, a total of 10 EFL students were invited to participate in a six-month learning period utilizing a pretest-posttest study design. The study started in mid-June 2023 and concluded in the first week of December 2023. During the first week of the study, participants received assistance in installing the applications on their personal devices, and thorough checks were conducted to ensure that all application functionalities operated as intended. Recognizing that most participants did not possess Android mobile devices, the application was ported into a desktop version to accommodate the devices used by the participants.

Participants were instructed to engage with the learning apps by practicing with the provided materials at least once a week over the 24-week period. They had the flexibility to choose topics of interest for their practice, and the option to simultaneously practice on multiple topics. To establish a baseline for participants' English-speaking proficiency, an online Oral Proficiency Interview – Computer (OPIc) test was conducted. The results indicated that most participants exhibited an average proficiency level of novice high, with the lowest proficiency recorded as novice low and the highest as intermediate medium, according to the American Council on the Teaching of Foreign Languages (ACTFL) proficiency scale. This level of proficiency suggested that participants could only communicate minimally using individual words or phrases, even in daily life conversations, and were not yet ready to independently produce their own sentences.

Given their proficiency levels, the participants represented an ideal target group for the learning apps, as the applications do not require users to independently generate complete sentences. This design aligns well with the participants' current proficiency, providing an appropriate and targeted learning environment. At the end of the six-month learning period with the apps, participants will undergo another online OPIc test to measure any improvements in their English speaking proficiency, providing valuable insights into the efficacy of the learning applications.

5.2.3 Criteria and Measurement

In evaluating the effectiveness of the learning applications, we employed a multifaceted approach, utilizing both quantitative and qualitative measures. Initially, we measured participants' speaking proficiency progression by comparing their OPIc scores obtained before and after the six-month learning period with the apps. This provided a quantitative metric for assessing any improvements in their English-speaking proficiency. Aside from this, we designed a structured instrument to gather qualitative data on participants' language learning experiences. The instrument aimed to capture insights into their English-speaking confidence, practice frequency, and common challenges encountered during speaking skill practice. To achieve this, participants were asked a series of targeted questions, as outlined in Table 8.

Table 8 List of Questions to Measure Participants’ Confidence Level, Practice Frequency, and Common Challenges for English Speaking Practice

No	Question
1.	How confident are you in your English speaking skills? (**Likert scale**: 1[Very not confidence] – 7[Very confidence])
2.	How often do you practice English on a daily basis? (**Options**: Not at all, Rarely, Infrequently, Occasionally, Sometimes, Frequently, Every day)
3.	Do you struggle to improve your English speaking skills due to a lack of opportunities to practice? (**Likert scale**: 1[Very disagree] – 7[Very agree])

4.	Do you struggle to improve your English speaking skills due to a lack of confidence in your abilities and a fear of being judged? (**Likert scale**: 1[Very diasgree] – 7[Very agree])
5.	What other factors do you believe make it challenging to improve your speaking skills? (**Free-text question**)

Then, the questionnaire featured an additional segment focusing on participants' perspectives regarding the use of chatbots for English learning. This segment jumped into participants' beliefs regarding the potential of chatbots to replace human partners in language practice. To gain a deeper understanding, we also inquired about the benefits of using chatbots for language practice from several previous studies. A detailed breakdown of the questions in this section is provided in Table 9.

Table 9 List of Questions to Gain Participants' Perspectives Regarding the Use of Chatbots for English Learning

No	Question
1.	Chatbots can replicate the smoothness of a conversation with a real human. (**Likert scale**: 1[Very disagree] – 7[Very agree])
2.	Chatbots can help me to improve my language skills effectively (**Likert scale**: 1[Very disagree] – 7[Very agree])
3.	I feel less judged when making a mistake with a chatbot as opposed to conversing with a person (**Likert scale**: 1[Very disagree] – 7[Very agree])
4.	I find it easier to practice my speaking skills with a chatbot than with a human (**Likert scale**: 1[Very disagree] – 7[Very agree])
5.	Starting a conversation with a chatbot reduces my anxiety compared to conversing with a person (**Likert scale**: 1[Very disagree] – 7[Very agree])

Finally, the final section of the instrument was dedicated to obtaining crucial insights into participants' perceptions of AI-related features embedded within the learning applications. This section specifically delved into participants' perspectives on the potential helpfulness of these AI features for their English learning journey. The detailed breakdown of questions in this section is outlined in Table 10.

Table 10 List of Questions to Gain Participants' Perceptions of AI-related Features within the Application

No	Question
1.	The content and topics provided in the app align with my learning interests (**Likert scale**: 1[Very disagree] – 7[Very agree])
2.	I easily understood the content (words, phrases, sentences, and grammar) in the application (**Likert scale**: 1[Very disagree] – 7[Very agree])
3.	The app accurately recognized the words I said (**Likert scale**: 1[Very disagree] – 7[Very

	agree])
4.	The app provides feedback based on my utterance quickly (**Likert scale**: 1[Very disagree] – 7[Very agree])
5.	The chatbot helps me improve my pronunciation by pointing out the parts that you mispronounced (**Likert scale**: 1[Very disagree] – 7[Very agree])
6.	I can learn how to pronounce a word by mimicking the chatbot voice output (**Likert scale**: 1[Very disagree] – 7[Very agree])
7.	I can understand the voice outputted by the chatbot well (**Likert scale**: 1[Very disagree] – 7[Very agree])
8.	The voice outputted by the chatbot sounds natural to me (**Likert scale**: 1[Very disagree] – 7[Very agree])
9.	The dialogue options within the app are diverse enough for my needs (**Likert scale**: 1[Very disagree] – 7[Very agree])
10.	Please share any features you didn't like in the app (**Free-text question**)
11.	Please list any features you wish the app had to enhance your English learning experience (**Free-text question**)

5.3 Results and Discussion

After the 6-month learning period with the language learning application, our examination of participants' speaking proficiency, as measured by the OPIc scores before and after the app usage, revealed a diverse range of outcomes, as illustrated in Figure 21.

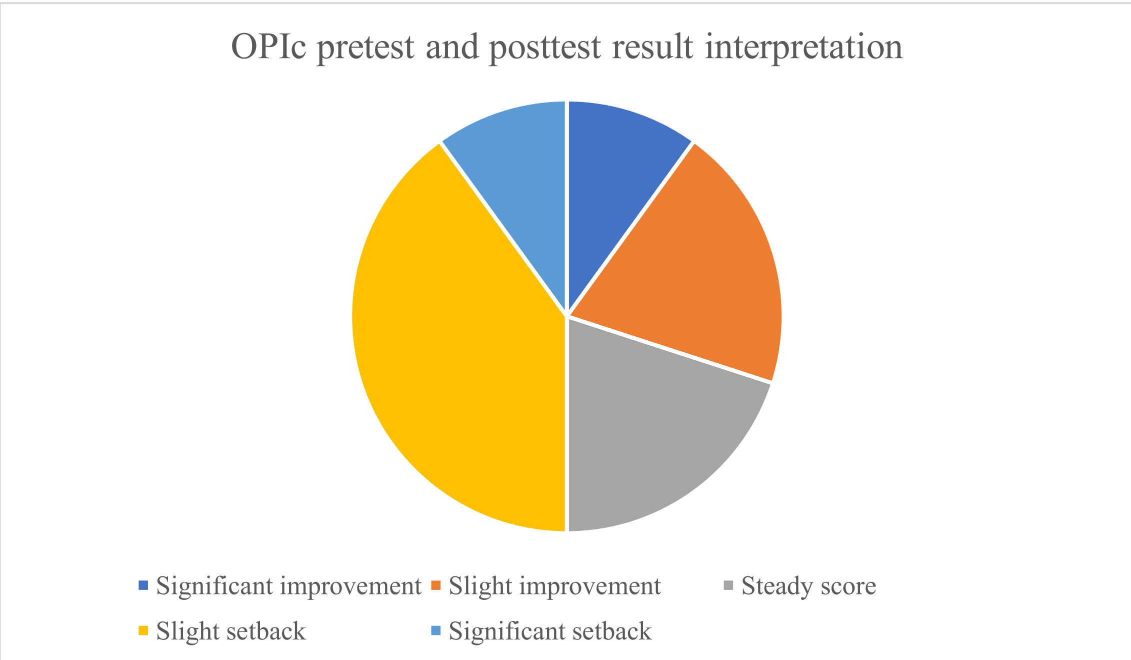

Figure 22 Mixed Results Regarding How the Application Affects Students' Speaking Proficiency.

As can be seen from Figure 22, only three students exhibited an improvement in their oral proficiency after using the app. Notably, one student experienced a low or insignificant improvement, while the other two demonstrated more substantial advancements. The criterion for determining the significance of improvement was set at an increase of two or more proficiency levels from their initial score, with increases of only one level categorized as insignificant. Conversely, two students displayed no change in their proficiency scores, indicating that the app had no discernible impact on their speaking skills. Surprisingly, the remaining students demonstrated a decrease in their test scores after using the app. Among them, four students experienced an insignificant setback, while one student faced a more significant decline. To identify the significance of setbacks, we utilized a criterion that marked setbacks as significant when the posttest score decreased by two or more proficiency levels compared to the pretest score. Conversely, a decrease of only one proficiency level was categorized as an insignificant setback.

An overall analysis of the participants' oral proficiency as depicted in the figure below reveals no significant improvement after the 6-month learning period using the app.

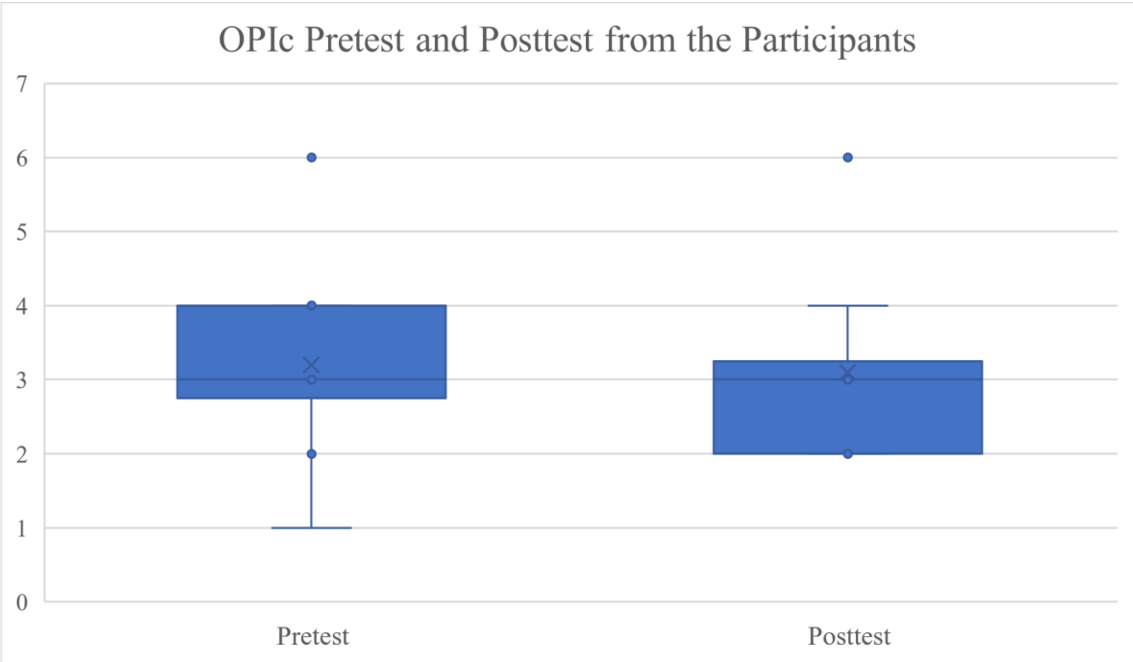

Figure 23 OPIc Pretest and Posttest from the Participants

Before participants engaged with the application, the mean OPIc score was 3.2 with a standard deviation of 1.32. Post-application usage, the mean of the posttest scores slightly decreased to 3.1, with a standard deviation of 1.20. Both pretest and posttest mean scores align within the novice high proficiency range. Consequently, it can be concluded that there is no significant improvement in participants' proficiency after utilizing the app over the 6-month period. Building upon this outcome, we sought to obtain valuable insights from participants by gathering feedback on their learning experiences with the app.

In response to participants' experiences and feedback, the examination first investigated the individual factors influencing English proficiency and practice habits. Notably, none of the participants reported feeling confident in their English-speaking proficiency, and a significant majority (9 out of 10 participants) admitted to infrequent practice. Following that, participants' perspectives on the integration of a chatbot into their language-learning journey were explored. Specifically, the investigation focuses on participants' views regarding the perceived smoothness, effectiveness, convenience, and comfortability (considering factors like reduced judgment and decreased anxiety) in utilizing a chatbot for English-speaking practice.

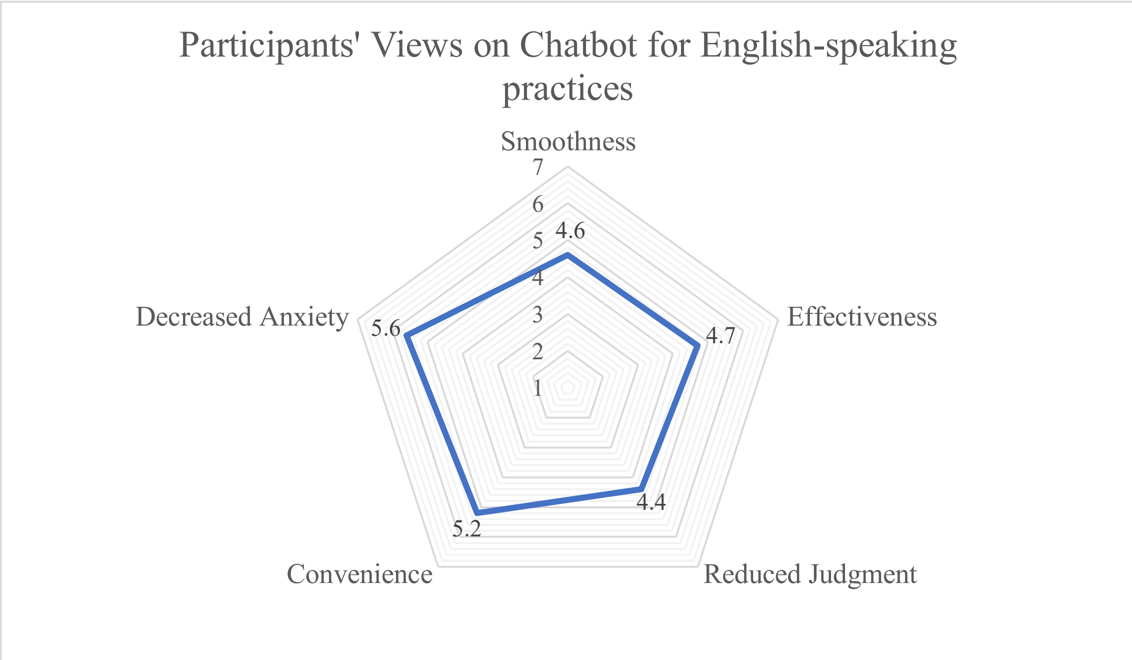

Figure 24 Participants' Views on Chatbot for English-speaking practices.

Figure 24 reveals a positive inclination among participants toward the adaptation of learning with the chatbot with all measured criteria scoring above 4 on a scale of 1 to 7. In this scale, where 1 signifies a strong negative and 7 represents a strong positive response, we could safely assume that scores above 4 indicate a positive inclination. Furthermore, the examination of participants' views pinpointed two significant strengths of the app: decreased anxiety (5.6 out of 7) and convenience (5.2 out of 7). Learners expressed feeling less anxious initiating conversations with the chatbot compared to practicing with a real human. Additionally, they found it more convenient to practice with the chatbot than with a real human partner. Connecting these insights with previous findings on participants' perceived lack of opportunities to practice and confidence issues, it becomes evident that the chatbot addresses these challenges by offering more practice opportunities for beginners, despite the presence of a real human partner.

However, for the remaining three criteria—smoothness, effectiveness, and reduced judgment—there is only a slight positive inclination. To understand these nuances, a detailed analysis was conducted, focusing on several criteria related to content appropriateness (preference, variety, and ease of understanding), speech-enabled learning, recognition accuracy, and responsiveness, as well as audio qualities (ease of understanding, naturalness, and audio-enhanced learning) within the application.

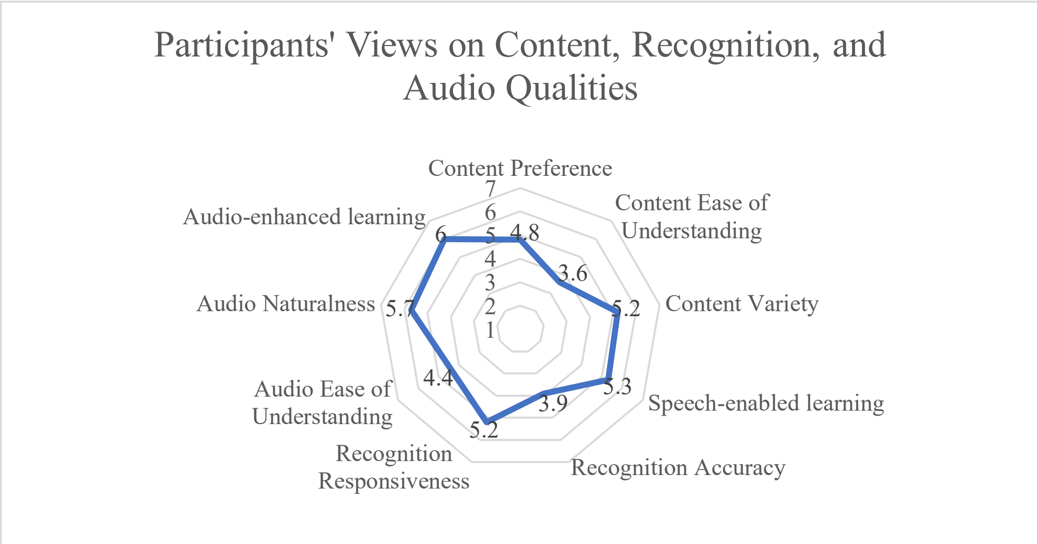

Figure 25 Participants' Views on Content, Recognition, and Audio Qualities within the Application

Continuing the analysis of participants' perceptions, Figure 25 provides insights into their views on various qualities related to the implementation of AI technologies within the app. Connecting these insights to the previously noted weak points in participants' views on the smoothness, effectiveness, and reduced judgment of the chatbot, several key findings emerge.

Within the app, two criteria received the lowest scores: content ease of understanding (3.6 out of 7) and recognition accuracy (3.9 out of 7). The lower score in recognition accuracy, affecting the app's ability to provide accurate correction feedback, may contribute to the perceived reduced judgment quality of learning with the chatbot. The implemented speech recognition, struggling to correctly identify words in participants' speech, might result in more corrections, potentially perceived as judgment by the students. Furthermore, the low score regarding content ease of understanding could impact the smoothness, effectiveness, and reduced judgment of learning with a chatbot. The participants perceived difficulty in understanding the content might indicate that they struggle to recognize certain words within the app, affecting the overall smoothness and effectiveness of using the application. If participants need to consult additional learning materials to understand these words, it may contribute to a perception of the app being less smooth and effective. Additionally, unfamiliarity with new words might make pronunciation challenging, further hindering the app's ability to recognize their utterances, and potentially leading to increased perceived judgment.

In addition to content ease of understanding and recognition accuracy, audio ease of understanding ranks third from the lowest position (4.4 out of 7). This factor may also influence the smoothness of the app, as participants might find it challenging to comprehend newly introduced words in the audio, potentially impacting their overall learning experience.

Continuing the evaluation of various criteria within the app, it's noteworthy that other than the three previously mentioned criteria, the remaining aspects received relatively favorable scores, ranging from 4.8 to 6 out of 7. Particularly, the two highest-scoring criteria are the potentiality of audio-enhanced learning (6 out of 7) and audio naturalness (5.7 out of 7). The high score in the audio-enhanced learning criterion suggests that there is substantial potential for users to learn pronunciation by listening to text-to-speech (TTS)-generated audio materials. Additionally, the positive inclination toward audio naturalness indicates a favorable response to TTS-generated materials for English learning, as users find the audio to be natural and authentic.

Moreover, a relatively positive score for speech-enabled learning (5.3 out of 7) and recognition responsiveness (5.2 out of 7) suggests a favorable response to the integration of speech recognition technology for English learning. Users seem to appreciate the ability of the app to recognize their speech and provide responsive feedback. Similarly, positive inclinations toward content variety (5.2 out of 7) and content preference (4.8 out of 7) suggest a favorable response to the use of generated content through the utilization of generative AI, such as ChatGPT, for language learning. Participants find value in the diversity and personal preferences catered to by the generative AI within the app.

Lastly, free-text responses were collected from participants on key areas for future development, emphasizing features that align more closely with users' needs.

1. Participants highlighted the importance of flexibility in responding to the chatbot. They suggested incorporating an additional freeform text input alongside the default response text. This input would allow users to compose their own sentences, fostering the development of personalized language production skills. To seamlessly integrate the suggested features into the current system that relies on pre-generated content, the application can employ a strategy to suggest main keyword topics that must be included in the user's freeform text input. This ensures that the chatbot's subsequent response remains coherent and relevant to the chosen topic. By doing so, learners can still benefit from the advantages of pre-generated content while having the flexibility to compose their own sentences.
2. Participants suggested UI improvements to enhance the learning progression with the app. They recommended a feature that marks topics chosen by learners and provides a history of their previous interactions with the chatbot. This functionality allows users to reflect on their learning progress and facilitates a more streamlined experience.
3. Participants expressed interest in customizability regarding the chatbot's output. They proposed an option to hide the default textual output, focusing solely on audio responses. This customization aims to serve learners who prioritize improving their listening skills.
4. Participants suggested enhancements to the learning experience, including explanations of

sentence structures in responses and vocabulary translations. In addition, they expressed the belief that brief explanations related to the selected topic would provide context for a more informed and accessible listening experience.

5.4 Conclusion

In conclusion, this chapter explored the potential benefits of integrating Text-to-Speech (TTS), Speech Recognition (SR), and ChatGPT as generative AI technologies to enhance students' speaking practice. The conducted pilot study, employing a pretest and posttest design with the Oral Proficiency Interview - Computerized (OPIc) test as a proficiency metric, aimed to evaluate the effectiveness of the developed apps over a 6-month learning period. Additionally, an instrument capturing participants' challenges, perspectives on English speaking practices, and attitudes towards AI-based technology implementation was utilized. The results indicate that, despite the 6-month learning period, there was no significant improvement in speaking proficiency among participants who used the app. The participants maintained an average proficiency level of Novice High, both before and after app usage. Connecting these findings with the questionnaire results revealed notable insights.

Firstly, the application demonstrated its potential as a convenient platform for speaking practices, especially for students with limited opportunities. Secondly, as participants perceived less anxiety while practicing with the app, it could serve as a comfortable starting point for students before engaging in conversations with real human partners. However, the lack of significant proficiency improvement suggests areas for enhancement in future application development. Consequently, focusing on the enhancement of three key aspects—content ease of understanding, audio ease of understanding, and Speech Recognition (SR) accuracy—is imperative for refining the application in its future iterations.

In relation to the content ease of understanding in the app, despite prior experimentation indicating that the materials produced by ChatGPT suit basic English users (CEFR A1 and A2), participants faced challenges in comprehending the content, as reflected by a low score in this aspect. To address this, incorporating features for an enhanced learning experience, such as translation support and a more detailed context regarding the topics offered in the application, could prove beneficial. This additional information would allow them to anticipate and better engage in upcoming conversations within the app. Providing a more detailed context regarding the topics can also contribute to a more informed and accessible listening practice experience for students, thereby facilitating ease of audio understanding. On top of that, the low score regarding recognition accuracy could also be a result of unfamiliarity with the word presented within the app. As the participants still struggle to understand the content provided within the app, they are likely unfamiliar with several words presented. This lack of familiarity further hinders the app's ability to recognize their utterances, resulting in low recognition accuracy. To fix this issue, an optional feature that allows the user to listen to how their selected response is pronounced before actually practicing their speaking skills could be implemented. Therefore, they could learn how to pronounce such a word by mimicking the provided example.

Chapter VI Conclusion

In this research, our primary focus centered on the integration of artificial intelligence (AI) technologies to address prevalent challenges in the development of speaking practice applications. Our efforts were concentrated on two key aspects: the implementation of AI for content production, encompassing learning materials and audio files, and the evaluation of offline speech recognition (SR) technology for assessing students' input in their speaking practices. To achieve this, we carefully selected and assessed a range of AI technologies relevant to English learning, gathering insights from students to understand their perspectives on such adaptations. The outcomes of our experiments have led to several notable conclusions:

1. **TTS for Audio Production**: To address challenges in audio production, WaveNet emerged as a promising Text-to-Speech (TTS) technology for our study. Therefore, we compared audio materials generated by WaveNet and those from native to evaluate the qualities of TTS-produced materials. Aside from the naturalness aspect, the TTS-produced materials exhibited a relatively low average score difference (pronunciation accuracy, comprehensibility, intelligibility) when compared to those produced by native speakers. This finding suggests that there may not be practical significance in terms of audio quality between the TTS-generated and native speaker audio. Interestingly, despite participants perceiving native speakers' sounds as slightly easier to understand, our analysis revealed that participants made fewer transcription errors when exposed to TTS-produced materials. This indicates that, in settings where native speakers are scarce, and their involvement in developing a speaking practice application might be costly, TTS systems offer a viable option, delivering comparable or even slightly superior understanding for learners.
2. **Applicability of Offline SR**: In another experiment, Vosk, as the offline Speech Recognition (SR), demonstrated promising results with a relatively low Word Error Rate (WER) of 7.42% when transcribing English as a Foreign Language (EFL) students' speech. It is important to note that this low WER was achieved with restricted vocabularies tailored for recognition through the Dynamic Vocabularies Reconfiguration feature. Also, our findings revealed that students using this system need to adopt a slower speech pace and prioritize speech clarity to optimize accuracy. Consequently, for learners with advanced language skills, the implementation of offline SR might not be the most suitable choice due to these constraints. However, in educational settings where students exhibit low confidence in engaging in speaking practices and the primary focus is on promoting speaking behavior and pronunciation accuracy could be compromised, offline SR technology proves to be a valuable tool.
3. **Generative AI for Learning Materials**: In our exploration of generating learning materials for

speaking practices with ChatGPT, it demonstrates potential utility, especially for individuals with a proficiency level equivalent to basic users in English or CEFR A1 to A2. Even without employing specific prompt techniques, ChatGPT demonstrated an instinctive ability to create content suited for individuals with basic English proficiency. Subsequent experiments, focused on the exploration of prompt techniques to tailor the produced materials, consistently confirmed that most materials generated by ChatGPT align with the intended target audience of English basic users. However, this also implies that for CEFR levels above A2, the materials produced by ChatGPT might not be suitable, and further exploration is needed.

Following these promising outcomes, our research extended to assess the overall effectiveness of integrating such technologies into a speaking practice app for improving students' speaking skills. To evaluate the impact, we implemented a comprehensive pilot study utilizing a pretest and posttest design, incorporating the Oral Proficiency Interview - Computerized (OPIc) test as a proficiency metric. Despite the reasonable performance achieved by individual technologies in isolated experiments, the results post-6-month learning period revealed no significant enhancement in participants' speaking proficiency. Surprisingly, participants maintained an average proficiency level of Novice High both before and after app usage. This unexpected outcome suggests that while AI technologies showed promise in specific contexts, their collective impact within the developed app did not lead to the desired improvement in overall speaking proficiency.

Subsequent to these results and a thorough analysis of participants' viewpoints on the developed application, it has become evident that there are specific areas that require further refinement to create a more effective speaking practice application. Despite the intended use of ChatGPT-produced materials for basic English users, participants expressed difficulty in understanding the dialogue presented in the app. This lack of understanding makes it challenging for them to listen to and pronounce words and phrases introduced in the materials, subsequently impacting their perception of the audio and the recognition accuracy of the app. Participants have suggested potential solutions, including incorporating L1 translations and providing further explanations for the text content within the app. This proposed enhancement aims to make the learning experience smoother, allowing users to comprehend materials without relying on external resources. Additionally, participants recommended the inclusion of optional sentence pronunciation examples to assist them in pronouncing unfamiliar words clearly, particularly when encountering new vocabulary. These insights from participants shed light on specific areas for improvement to enhance the effectiveness of the speaking practice application.

References

[1] H. C. Nguyen, "Motivation in Learning English Language: A Case Study at Vietnam National University, Hanoi," *Eur. J. Educ. Sci.*, vol. 06, no. 01, pp. 49–65, 2019, doi: 10.19044/ejes.v6no1a4.

[2] S. G. B. MacWhinnie and C. Mitchell, "English classroom reforms in Japan: a study of Japanese university EFL student anxiety and motivation," *Asian-Pacific J. Second Foreign Lang. Educ.*, vol. 2, no. 1, 2017, doi: 10.1186/s40862-017-0030-2.

[3] Q. Yin and M. Satar, "English as a Foreign Language Learner Interactions with Chatbots: Negotiation for Meaning.," *Int. Online J. Educ. Teach.*, vol. 7, no. 2, pp. 390–410, 2020.

[4] M. Shishido, "Developing and Evaluating an E-learning Material for Speaking Practice with the Latest AI Technology," in *The IAFOR International Conference on Education – Hawaii 2021*, 2021. [Online]. Available: https://doi.org/10.22492/issn.2189-1036.2021.5

[5] M. Shishido, "Evaluating e-learning system for English conversation practice with speech recognition and future development using AI Introducing the E-learning system with speech recognition," in *Proceedings of EdMedia + Innovate Learning*, 2019, pp. 213–218.

[6] N. Kim, "Chatbots and Korean EFL Students ' English Vocabulary Learning," *J. Digit. Converg.*, vol. 16, no. 2, pp. 1–7, 2018.

[7] D.-E. Han, "The Effects of Voice-based AI Chatbots on Korean EFL Middle School Students' Speaking Competence and Affective Domains," *Asia-Pacific J. Converg. Res. Interchange.*, vol. 6, no. 7, pp. 71–80, 2020, doi: 10.47116/apjcri.2020.07.07.

[8] G. van den Berg and E. du Plessis, "ChatGPT and Generative AI: Possibilities for Its Contribution to Lesson Planning, Critical Thinking and Openness in Teacher Education," *Educ. Sci.*, vol. 13, no. 10, 2023, doi: 10.3390/educsci13100998.

[9] O. Koraishi, "Teaching English in the age of AI: Embracing ChatGPT to optimize EFL materials and assessment," *Lang. Educ. Technol.*, vol. 3, no. 1, pp. 55–72, 2023, [Online]. Available: https://bit.ly/43en7e1

[10] T. N. Fitria, "Artificial intelligence (AI) technology in OpenAI ChatGPT application: A review of ChatGPT in writing English essay," *ELT Forum J. English Lang. Teach.*, vol. 12, no. 1, pp. 44–58, 2023, doi: 10.15294/elt.v12i1.64069.

[11] A. van den Oord *et al.*, "Wavenet: A generative model for raw audio," *arXiv Prepr. arXiv1609.03499*, 2016.

[12] T. Andriani, Y. Herawati, and T. Sulistyo, "Text-to-Speech Application for Foreign Language Learner' Listening Comprehension in Indonesia," in *The 2nd International Conference on Quran and Hadith Studies Information Technology and Media in Conjunction with the 1st International Conference on Islam, Science and Technology, ICONQUHAS & ICONIST*, 2020. doi:

10.4108/eai.2-10-2018.2295544.

[13] D. Oktalia and N. A. Drajati, "English teachers' perceptions of text to speech software and Google site in an EFL Classroom: What English teachers really think and know," *Int. J. Educ. Dev. Using Inf. Commun. Technol.*, vol. 14, no. 3, pp. 183–192, 2018.

[14] N. Matsuda, "Evidence of Effects of Text-to-Speech Synthetic Speech to Improve Second Language Learning," *JACET J.*, vol. 61, pp. 149–164, 2017.

[15] H.-H. Chiang, "A Comparison Between Teacher-Led and Online Text-to-Speech Dictation for Students' Vocabulary Performance," *English Lang. Teach.*, vol. 12, no. 3, p. 77, 2019, doi: 10.5539/elt.v12n3p77.

[16] S. Krashen, "Some issues relating to the monitor model," *Teach. Learn. English as a Second Language. Trends Res. Pract. TESOL '77 Sel. Pap. from Elev. Annu. Conv. Teach. English to Speak. Other Lang. Miami, Florida*, no. Brown, H; Yorio, Carlos; Crymes, Ruth (eds.), pp. 144–158, 1977.

[17] W. Cardoso, G. Smith, and C. Garcia Fuentes, "Evaluating text-to-speech synthesizers," no. 2015, pp. 108–113, 2015, doi: 10.14705/rpnet.2015.000318.

[18] W. Cardoso, "Learning L2 pronunciation with a text-to-speech synthesizer," in *Future-proof CALL: language learning as exploration and encounters – short papers from EUROCALL 2018*, 2018, pp. 16–21. doi: 10.14705/rpnet.2018.26.806.

[19] J. Grimshaw, T. Bione, and W. Cardoso, "Who's got talent? Comparing TTS systems for comprehensibility, naturalness, and intelligibility," in *Future-proof CALL: language learning as exploration and encounters – short papers from EUROCALL 2018*, 2018, pp. 83–88. doi: 10.14705/rpnet.2018.26.817.

[20] H. Mulyono and D. N. Vebriyanti, "Developing Native-Like Listening Comprehension Materials Perceptions of a Digital Approach," *J. ELT Res.*, vol. 1, no. 1, p. 1, 2016, doi: 10.22236/jer_vol1issue1pp1-20.

[21] C. Tejedor-Garcia, V. Cardeñoso-Payo, and D. Escudero-Mancebo, "Design and Evaluation of Two Mobile Computer-Assisted Pronunciation Training Tools To Favor Autonomous Pronunciation Training of English As a Foreign Language," in *EDULEARN20 Proceedings*, 2020, no. December 2022, pp. 7639–7646. doi: 10.21125/edulearn.2020.1936.

[22] C. S. C. Dalim, M. S. Sunar, A. Dey, and M. Billinghurst, "Using augmented reality with speech input for non-native children's language learning," *Int. J. Hum. Comput. Stud.*, vol. 134, pp. 44–64, 2020.

[23] E. Junining, S. Alif, and N. Setiarini, "Automatic speech recognition in computer-assisted language learning for individual learning in speaking," *JEES (Journal English Educ. Soc.*, vol. 5, no. 2, pp. 219–223, 2020.

[24] S. Moxon, "Exploring the effects of automated pronunciation evaluation on L2 students in Thailand," *IAFOR J. Educ.*, vol. 9, no. 3, pp. 41–56, 2021, doi: 10.22492/ije.9.3.03.

[25] M. Shishido, “Developing e-learning system for English conversation practice using speech recognition and artificial intelligence,” in *Proceedings of EdMedia: World Conference on Educational Media and Technology*, 2018, pp. 226–231.

[26] J. S. Edu, J. M. Such, and G. Suarez-Tangil, “Smart home personal assistants: a security and privacy review,” *ACM Comput. Surv.*, vol. 53, no. 6, pp. 1–36, 2020.

[27] D. Alsmadi and V. Prybutok, “Sharing and storage behavior via cloud computing: Security and privacy in research and practice,” *Comput. Human Behav.*, vol. 85, pp. 218–226, 2018.

[28] J. Lee and Y. Hwang, “A Meta-analysis of the Effects of Using AI Chatbot in Korean EFL Education,” 영어영문학연구, vol. 48, no. 1, pp. 213–243, 2022.

[29] AlphaCephei, “Vosk: A Speech Recognition Toolkit,” 2019. https://alphacephei.com/vosk/ (accessed Jan. 23, 2023).

[30] T. F. Pereira *et al.*, “A web-based Voice Interaction framework proposal for enhancing Information Systems user experience,” *Procedia Comput. Sci.*, vol. 196, pp. 235–244, 2022.

[31] H. Hübert, V. Taliaronak, and H. S. Yun, “AI BASED GESTURE AND SPEECH RECOGNITION TOOL FLOW FOR EDUCATORS,” in *ICERI2022 Proceedings*, 2022, pp. 6395–6400.

[32] H. Lee, C.-C. Hsia, A. Tsoy, S. Choi, H. Hou, and S. Ni, “VisionARy: Exploratory research on Contextual Language Learning using AR glasses with ChatGPT,” in *Proceedings of the 15th Biannual Conference of the Italian SIGCHI Chapter*, 2023, pp. 1–6.

[33] A. Trabelsi, S. Warichet, Y. Aajaoun, and S. Soussilane, “Evaluation of the efficiency of state-of-the-art Speech Recognition engines,” *Procedia Comput. Sci.*, vol. 207, pp. 2242–2252, 2022.

[34] J. K. M. Ali, M. A. A. Shamsan, T. A. Hezam, and A. A. Q. Mohammed, “Impact of ChatGPT on Learning Motivation: Teachers and Students’ Voices,” *J. English Stud. Arab. Felix*, vol. 2, no. 1, pp. 41–49, 2023, doi: 10.56540/jesaf.v2i1.51.

[35] M. S. S. Moqbel and A. M. T. Al-Kadi, “Foreign language learning assessment in the age of ChatGPT: A theoretical account,” *J. English Stud. Arab. Felix*, vol. 2, no. 1, pp. 71–84, 2023.

[36] T. Adiguzel, M. H. Kaya, and F. K. Cansu, “Revolutionizing education with AI: Exploring the transformative potential of ChatGPT,” *Contemp. Educ. Technol.*, vol. 15, no. 3, p. ep429, 2023.

[37] J. Qadir, “Engineering Education in the Era of ChatGPT: Promise and Pitfalls of Generative AI for Education,” in *2023 IEEE Global Engineering Education Conference (EDUCON)*, 2023, pp. 1–9. doi: 10.1109/EDUCON54358.2023.10125121.

[38] F. R. Baskara and F. X. Mukarto, “Exploring the Implications of ChatGPT for Language Learning in Higher Education,” *IJELTAL (Indonesian J. English Lang. Teach. Appl. Linguist.*, vol. 7, no. 2, pp. 343–358, 2023.

[39] A. Gatt and E. Krahmer, “Survey of the state of the art in natural language generation: Core tasks, applications and evaluation,” *J. Artif. Intell. Res.*, vol. 61, pp. 65–170, 2018.

[40] N. Borrero, “Nurturing students’ strengths: The impact of a school-based student interpreter

program on Latino/a students' reading comprehension and English language development," *Urban Educ.*, vol. 46, no. 4, pp. 663–688, 2011.

[41] J. Jeon, "Exploring AI chatbot affordances in the EFL classroom: Young learners' experiences and perspectives," *Comput. Assist. Lang. Learn.*, pp. 1–26, 2021.

[42] S. D. Sulistyaningrum, "Employing online paraphrasing tools to overcome students' difficulties in paraphrasing," *Stairs English Lang. Educ. J.*, vol. 2, no. 1, pp. 52–59, 2021.

[43] L. K. Fryer, D. Coniam, R. Carpenter, and D. Lăpușneanu, "Bots for language learning now: Current and future directions," *Lang. Learn. Technol.*, vol. 24, no. 2, pp. 8–22, 2020.

[44] N.-Y. Kim, "A study on the use of artificial intelligence chatbots for improving English grammar skills," *J. Digit. Converg.*, vol. 17, no. 8, pp. 37–46, 2019.

[45] L. Cagliero, L. Farinetti, and E. Baralis, "Recommending personalized summaries of teaching materials," *IEEE Access*, vol. 7, pp. 22729–22739, 2019.

[46] F. Pramudianto, T. Chhabra, E. F. Gehringer, and C. Maynards, "Assessing the Quality of Automatic Summarization for Peer Review in Education.," in *EDM (Workshops)*, 2016.

[47] A. Ariyanti, "Technology-Enhanced Paraphrasing Tool to Improve EFL Students' Writing Achievement and Enjoyment," *J. English Lang. Teach. Linguist.*, vol. 6, no. 3, pp. 715–726, 2021.

[48] J. S. Barrot, "Using ChatGPT for second language writing: Pitfalls and potentials," *Assess. Writ.*, vol. 57, p. 100745, 2023, doi: https://doi.org/10.1016/j.asw.2023.100745.

[49] W. C. H. Hong, "The impact of ChatGPT on foreign language teaching and learning: opportunities in education and research," *J. Educ. Technol. Innov.*, vol. 5, no. 1, 2023.

[50] M. M. Rahman and Y. Watanobe, "ChatGPT for Education and Research: Opportunities, Threats, and Strategies," *Appl. Sci.*, vol. 13, no. 9, 2023, doi: 10.3390/app13095783.

[51] L. Kohnke, B. L. Moorhouse, and D. Zou, "ChatGPT for Language Teaching and Learning," *RELC J.*, p. 00336882231162868, Apr. 2023, doi: 10.1177/00336882231162868.

[52] E. A. M. Van Dis, J. Bollen, W. Zuidema, R. van Rooij, and C. L. Bockting, "ChatGPT: five priorities for research," *Nature*, vol. 614, no. 7947, pp. 224–226, 2023.

[53] J. Jeon and S. Lee, "Large language models in education: A focus on the complementary relationship between human teachers and ChatGPT," *Educ. Inf. Technol.*, 2023, doi: 10.1007/s10639-023-11834-1.

[54] H. H. Thorp, "ChatGPT is fun, but not an author," *Science*, vol. 379, no. 6630. American Association for the Advancement of Science, p. 313, 2023.

[55] M. Perkins, "Academic Integrity Considerations of AI Large Language Models in the post-pandemic era: ChatGPT and beyond," *J. Univ. Teach. Learn. Pract.*, vol. 20, no. 2, p. 7, 2023.

[56] O. P. Pfeffer *et al.*, "ChatGPT for Good? On Opportunities and Challenges of Large Language Models for Education".

[57] N. Hambali, Muslih and Mirizon, Soni, and Heryana, "Difficulty Index and Cognitive Skills of

English Textbook for Senior High School," *Indones. J. EFL Linguist.*, vol. 6, no. 1, pp. 17–28, 2021, [Online]. Available: http://www.indonesian-efl-journal.org/index.php/ijefll/article/view/331

[58] P. C. Gómez and Á. A. S. Lafuente, "Readability indices for the assessment of textbooks: a feasibility study in the context of EFL," *Vigo Int. J. Appl. Linguist.*, no. 16, pp. 31–52, 2019.

[59] A. A. Hakim, E. Setyaningsih, and D. Cahyaningrum, "Examining the readability level of reading texts in English textbook for Indonesian senior high school," *J. English Lang. Stud.*, vol. 6, no. 1, pp. 18–35, 2021.

[60] I. A. Fata, E. Komariah, and A. R. Alya, "Assessment of Readability Level of Reading Materials in Indonesia EFL Textbooks," *Ling. Cult.*, vol. 16, no. 1, pp. 97–104, 2022.

[61] F. Khodabandeh and M. H. Tharirian, "Exploring the Impact of Blended, Flipped, and Traditional Teaching Strategies for Teaching Grammar on Iranian EFL Learners'''through English Newspaper Articles," *Teach. English as a Second Language. Q. (Formerly J. Teach. Lang. Ski.*, vol. 39, no. 3.1, pp. 89–129, 2020.

[62] G. Shakibaei, E. Namaziandost, and F. Shahamat, "The effect of using authentic texts on Iranian EFL learners' incidental vocabulary learning: The case of English newspaper," *Int. J. Linguist. Lit. Transl.*, 2019.

[63] J. C. Young and M. Shishido, "Adjusting Reading Materials for EFL Learning using ChatGPT," 2023. https://github.com/bwbkwk/chatgpt-for-efl-learning (accessed Feb. 21, 2023).

[64] D. Yao, "A Comparative Study of Test Takers' Performance on Computer-Based Test and Paper-Based Test across Different CEFR Levels.," *English Lang. Teach.*, vol. 13, no. 1, pp. 124–133, 2020.

Appendix

Application Code - JSONToDialogueConverter

```
using System.Collections;
using System.Collections.Generic;
using System.Text;
using System.Linq;
using UnityEngine;

namespace DialogueConverter
{
    public class JSONtoDialogueConverter : MonoBehaviour
    {
        private ChatbotDialogue chatbotDialogue;
        private string[] listOfTopics;
        private Dictionary<string, AudioClip> topicAudioClips;
        private Dictionary<string,ChoiceAudioInfo> audioInfos;
        private AudioSource audioSource;

        public string[] ListOfTopics { get => listOfTopics; }

        private void Awake()
        {
            TextAsset jsonDialogue =
Resources.Load<TextAsset>("Text/chatbot-dialogue");
            chatbotDialogue =
JsonUtility.FromJson<ChatbotDialogue>(jsonDialogue.text);
            // print(chatbotDialogue.dialogueGroups[1].toRiveScript());
            listOfTopics = new
string[chatbotDialogue.dialogueGroups.Length];
            for (int i = 0; i< chatbotDialogue.dialogueGroups.Length;i++)
            {
                listOfTopics[i] = chatbotDialogue.dialogueGroups[i].topic;
            }

            audioSource = GetComponent<AudioSource>();

            topicAudioClips = new Dictionary<string, AudioClip>();
            foreach(string topic in listOfTopics)
            {

topicAudioClips.Add(topic,Resources.Load<AudioClip>(string.Format("Audio/{
0}", topic)));
            }

            audioInfos = new Dictionary<string,ChoiceAudioInfo>();

            foreach(DialogueGroup group in chatbotDialogue.dialogueGroups)
            {
```

```
            foreach(DialogueItem item in group.items)
            {
                if (item.audioInfos != null)
                {
                    for (int i = 0; i < item.choices.Length; i++)
                    {
                        audioInfos[Utility.NormalizeString(group.topic + " 
"+ item.choices[i])] = item.audioInfos[i];
                    }
                }
            }
        }
    }

    public void PlayAudio(int topicIndex, string sentence)
    {
        string audioReference = 
Utility.NormalizeString(ListOfTopics[topicIndex]) + " " + sentence;


        if (topicIndex < listOfTopics.Length && 
audioInfos.ContainsKey(audioReference))
        {
            ChoiceAudioInfo info = audioInfos[audioReference];
            audioSource.clip = 
topicAudioClips[ListOfTopics[topicIndex]];
            audioSource.time = info.start;
            audioSource.Play();
            float audioLength = (info.end - info.start);
            audioSource.SetScheduledEndTime(AudioSettings.dspTime + 
audioLength);
        }

    }
    public float GetAudioLength(int topicIndex, string sentence)
    {
        if (topicIndex < listOfTopics.Length && 
audioInfos.ContainsKey(sentence))
        {
            ChoiceAudioInfo info = audioInfos[sentence];
            audioSource.clip = 
topicAudioClips[ListOfTopics[topicIndex]];
            audioSource.time = info.start;
            float audioLength = (info.end - info.start);
            return audioLength;
        }

        return 0;
    }
```

```
        public void StopAudio()
        {
            audioSource.Stop();
        }

        public List<string> GetListOfWordsByTopic(int selectedIndex)
        {
            HashSet<string> words = new HashSet<string>();
            foreach (DialogueItem item in
chatbotDialogue.dialogueGroups[selectedIndex].items)
            {
                for (int i = 0; i < item.choices.Length; i++)
                {
                    foreach(string word in
Utility.NormalizeString(item.choices[i]).Split(' '))
                    {
                        words.Add(word);
                    }
                }
            }

            if (words.Contains(""))
                words.Remove("");

            return words.ToList();
        }
        public List<string> GetListOfWordsByTopicAndSequence(int
topicIndex, int sequenceIndex = 1)
        {
            HashSet<string> words = new HashSet<string>();
            foreach (string sentence in
chatbotDialogue.dialogueGroups[topicIndex].items[sequenceIndex].choices)
            {
                print(sentence);
                foreach (string word in
Utility.NormalizeString(sentence).Split(' '))
                {
                    words.Add(word);
                }
            }

            if (words.Contains(""))
                words.Remove("");

            return words.ToList();
        }

        public string GetFullRiveScript()
        {
            StringBuilder sb = new StringBuilder();
```

```
            foreach(DialogueGroup group in chatbotDialogue.dialogueGroups)
            {
                // print(group);
                sb.Append(group.toRiveScript()+"¥n");
                sb.Append("¥n");
            }
            return sb.ToString();
        }
    }
}
```

Application Code - ChatbotController

```
using UnityEngine;
using DialogueConverter;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.IO;

public class ChatbotController : MonoBehaviour
{
    private RiveScript.RiveScript botBrain;

    [SerializeField]
    private DialogueController dialogueController;
    private TopicListController topicListController;
    private VoskSpeechToText voskSTT;

    private FormDialogueController dialogueFinishController;
    private JSONtoDialogueConverter converter;
    private string chatbotCommandListOfChoices = "chatbot command topic
{0} dialogue with id {1} list of choices";
    private string chatbotCommandHowToReply = "chatbot command topic {0}
how to reply {1}";
    private int selectedTopicIndex = 0, nextTopicIndex = 0;
    private SFXController sfxController;

    public ActionDelay delay = new ActionDelay(0.25f, 0.5f, 0.25f,0.25f);

    public struct ActionDelay
    {
        public WaitForSeconds preAddChatDelay, inBetweenChatAndSpeechDelay;
        public float postSpeechExtraDelay, totalDelay;

        public ActionDelay(float preAddChatDelay, float postAddChatDelay,
float preSpeechDelay, float postSpeechExtraDelay)
        {
            this.preAddChatDelay = new WaitForSeconds(preAddChatDelay);
```

```
            inBetweenChatAndSpeechDelay = new
WaitForSeconds(postAddChatDelay+ preSpeechDelay);
            this.postSpeechExtraDelay = postSpeechExtraDelay;

            totalDelay = preAddChatDelay + postAddChatDelay + preSpeechDelay
+ postSpeechExtraDelay;
        }

    }

    private void Awake()
    {
        sfxController =
GameObject.Find("SFXController").GetComponent<SFXController>();
        dialogueController =
GameObject.Find("DialogueController").GetComponent<DialogueController>();
        dialogueFinishController = GameObject.Find("Canvas").transform.

Find("BaseImage").Find("DialogueFinishPanel").GetComponent<FormDialogueCon
troller>();
        topicListController = GameObject.Find("Canvas").transform.

Find("BaseImage").Find("SideNavigation").Find("TopicMasterPanel").
                                GetComponent<TopicListController>();
        converter = GetComponent<JSONtoDialogueConverter>();

        string transcript = converter.GetFullRiveScript();

        transcript += "¥n + *¥n - NOT YET IMPLEMENTED";


        botBrain = new RiveScript.RiveScript(true,true,false);
        botBrain.stream(transcript);
        botBrain.sortReplies();

        voskSTT =
GameObject.Find("STTController").GetComponent<VoskSpeechToText>();
    }

    public void PrepareChangeTopic(int nextTopicIndex)
    {
        this.nextTopicIndex = nextTopicIndex;
    }
    public void ChangeTopic()
    {
        this.selectedTopicIndex = nextTopicIndex;
        converter.StopAudio();
        dialogueController.RestartSession();
```

```
        float audioLength =
ReplyToBot(string.Format(chatbotCommandListOfChoices,
converter.ListOfTopics[selectedTopicIndex], 1));

dialogueController.ChangeEvent(DialogueController.EventName.ChatbotTTS,
audioLength);

    }


    public float ReplyToBot(string message, bool isActualUserResponse =
false)
    {
        string normalizedMessage = Utility.NormalizeString(message);
        string result = "";
        if (isActualUserResponse)
        {
            result = botBrain.reply("user", string.Format("{0} xox {1} xox",
normalizedMessage, converter.ListOfTopics[selectedTopicIndex]));
        }
        else
        {
            result = botBrain.reply("user", normalizedMessage);
        }
        string normalizedResult = Utility.NormalizeString(result);
        float audioLength = converter.GetAudioLength(selectedTopicIndex,
normalizedResult);
        StartCoroutine(ReplyToBotActions(result, normalizedResult,
audioLength));
        return audioLength + delay.totalDelay;
    }
    public IEnumerator ReplyToBotActions(string botResponse, string
normalizedBotResponse, float audioLength )
    {
        if (botResponse.Equals("NOT YET IMPLEMENTED"))
        {
            dialogueController.DialogueSuggestions = new string[] { "..." };
            dialogueFinishController.ToggleActivation();
            yield break;
        }
        yield return delay.preAddChatDelay;
        sfxController.PlayBubbleSFX();
        dialogueController.AddChat(botResponse);
        yield return delay.inBetweenChatAndSpeechDelay;
        converter.PlayAudio(selectedTopicIndex, normalizedBotResponse);
        yield return new WaitForSeconds(audioLength +
delay.postSpeechExtraDelay);
        sfxController.PlayBubbleSFX();
```

```
        GetDialogueSuggestions(string.Format(chatbotCommandHowToReply,
converter.ListOfTopics[selectedTopicIndex], normalizedBotResponse));
        yield break;
    }

    public void GetDialogueSuggestions(string query)
    {
        string result = botBrain.reply("user", query);
        if(!result.Equals("NOT YET IMPLEMENTED"))
        {
            dialogueController.DialogueSuggestions = result.Split('~');
            HashSet<string> words = new HashSet<string>();
            foreach (string sentence in
dialogueController.DialogueSuggestions)
            {
                foreach (string word in
Utility.NormalizeString(sentence).Split(' '))
                {
                    words.Add(word);
                }
            }

            if (words.Contains(""))
                words.Remove("");

            voskSTT.PrepareVocabularies(words.ToList());
        }
        else
        {
            dialogueController.DialogueSuggestions = new string[] { "..." };
            dialogueFinishController.ToggleActivation();
        }

    }

    void Start()
    {
        ChangeTopic();
        topicListController.GenerateTopicItems(converter.ListOfTopics);
    }

    public void RestartDialogue()
    {
        nextTopicIndex = this.selectedTopicIndex;
        ChangeTopic();
    }
}
```

Application Code - DialogueController

```
using System.Collections;
```

```
using System.Collections.Generic;
using System.Text;
using TMPro;
using UnityEngine;
using UnityEngine.UI;

public class DialogueController : MonoBehaviour
{
    [SerializeField]
    private int chatBubblePadding = 30;
    [SerializeField]
    private int chatBubbleHorizontalOffset = 60;
    [SerializeField]
    private int chatBubbleVerticalOffset = 40;
    [SerializeField]
    private int chatBubbleFontSize = 80;
    [SerializeField]
    private int chatBubbleEffectSize = 15;
    [SerializeField]
    private GameObject chatContentPanel;
    private GameObject userChatOption;
    private UserChatController userChatController;
    private ChatbotController chatbotController;
    private ExternalLoggerController externalLoggerController;
    private VoskSpeechToText voskSTT;

    private TextMeshProUGUI userChatOptionTxt;
    private GameObject chatBotChatContent;
    private GameObject userChatContent;
    private GameObject wrapper;

    private TextMeshProUGUI userLastResponseTxt;

    private int dialogueSuggestionIndex = 0;
    private string[] dialogueSuggestions;

    [SerializeField]
    private float eventTransmitionTime;
    [HideInInspector]
    public float eventStartingTime,eventLength;
    public EventName currentEvent;
    public enum EventName { Nothing, ChatbotTTS, WaitForSpeech,
STTRunning };
    private SFXController sfxController;
    public Coroutine animateNewDialogue;

    public bool enableChatSelectionField;
    public void ChangeEvent(EventName name, float eventLength = -1)
    {
        if(currentEvent != EventName.Nothing && name == EventName.Nothing)
```

```
        {
            userChatController.SetChatOptionFunctionality(true);
        } else if(currentEvent == EventName.Nothing && name !=
EventName.Nothing)
        {
            LoadingForNewEvent();
        }
        currentEvent = name;
        this.eventLength = eventLength + eventTransmitionTime;
        eventStartingTime = Time.time;

    }

    public string[] DialogueSuggestions
    {
        get => dialogueSuggestions;
        set
        {
            dialogueSuggestions = value;
            if(currentEvent == EventName.Nothing)
RenderDialogueSuggestion(NextDialogueSuggestion);
        }
    }

    private string currentDialogueSuggestion = "";

    public string NextDialogueSuggestion {
        get
        {
            dialogueSuggestionIndex = (dialogueSuggestionIndex + 1) %
dialogueSuggestions.Length;
            currentDialogueSuggestion =
dialogueSuggestions[dialogueSuggestionIndex];
            return currentDialogueSuggestion;
        }
    }

    public string PreviousDialogueSuggestion
    {
        get
        {
            dialogueSuggestionIndex = (dialogueSuggestionIndex - 1);
            if (dialogueSuggestionIndex == -1)
            {
                dialogueSuggestionIndex = DialogueSuggestions.Length - 1;
            }
            currentDialogueSuggestion =
dialogueSuggestions[dialogueSuggestionIndex];
            return currentDialogueSuggestion;
```

```
        }
    }

    private void Awake()
    {
        externalLoggerController =
GetComponent<ExternalLoggerController>();
        sfxController =
GameObject.Find("SFXController").GetComponent<SFXController>();
        currentEvent = EventName.Nothing;
        chatBotChatContent = Resources.Load<GameObject>("Prefabs/ChatBot-
ChatContent");
        RectOffset ro = new RectOffset(chatBubbleHorizontalOffset,
chatBubbleHorizontalOffset, chatBubbleVerticalOffset,
chatBubbleVerticalOffset);
        chatBotChatContent.GetComponent<VerticalLayoutGroup>().padding =
ro;

chatBotChatContent.transform.Find("Text").GetComponent<TextMeshProUGUI>().
fontSize = chatBubbleFontSize;


        userChatContent = Resources.Load<GameObject>("Prefabs/User-
ChatContent");
        userChatContent.GetComponent<VerticalLayoutGroup>().padding = ro;

userChatContent.transform.Find("Text").GetComponent<TextMeshProUGUI>().fon
tSize = chatBubbleFontSize;
        userChatContent.GetComponent<Outline>().effectDistance = new
Vector2(chatBubbleEffectSize, chatBubbleEffectSize);

        wrapper = Resources.Load<GameObject>("Prefabs/Wrapper");

        enableChatSelectionField = true;
        voskSTT =
GameObject.Find("STTController").GetComponent<VoskSpeechToText>();
        voskSTT.OnTranscriptionResult += OnTranscriptionResult;

        chatbotController =
GameObject.Find("ChatbotController").GetComponent<ChatbotController>();
        userChatOption =
GameObject.Find("Canvas").transform.Find("BaseImage").Find("UserChatOption
Panel").gameObject;

        userChatOptionTxt =
userChatOption.transform.Find("ChatOption").GetChild(0).GetComponent<TextM
eshProUGUI>();
        userChatController =
userChatOption.GetComponent<UserChatController>();
```

```
    }

    private void OnTranscriptionResult(string obj)
    {
        var voskInterpretation = new RecognitionResult(obj);
        string userSpeechTranscription =
voskInterpretation.Phrases[0].Text;

StartCoroutine(externalLoggerController.SendDataToLogger(currentDialogueSu
ggestion, userSpeechTranscription));
        RerenderSelectedDialogue(userSpeechTranscription);
    }

    public bool ToggleSTTModule()
    {

        bool isRecording = voskSTT.ToggleRecording();
        if (animateNewDialogue != null && isRecording)
        {
            StopCoroutine(animateNewDialogue);
            userLastResponseTxtBorder.effectDistance = Vector2.zero;
        }

        if (isRecording) {
            ChangeEvent(EventName.STTRunning);
        }
        return isRecording;
    }


    public void AddChat(string message, bool isFromUser = false)
    {
        GameObject chatContent = (isFromUser) ? userChatContent :
chatBotChatContent;
        GameObject newWrapper = Instantiate<GameObject>(wrapper);

        chatContent = Instantiate<GameObject>(chatContent);
        TextMeshProUGUI textMeshProUGUI =
chatContent.transform.GetChild(0).GetComponent<TextMeshProUGUI>();
        textMeshProUGUI.text = message;

        chatContent.transform.SetParent(newWrapper.transform, false);
        newWrapper.transform.SetParent(chatContentPanel.transform, false);

        HorizontalLayoutGroup hlg =
newWrapper.GetComponent<HorizontalLayoutGroup>();
        RectOffset ro = hlg.padding;
        if (isFromUser)
        {
            userLastResponseTxt = textMeshProUGUI;
```

```
            userLastResponseTxtBorder = chatContent.GetComponent<Outline>();
            ro.right = chatBubblePadding;
            ro.left = 3 * chatBubblePadding;
            ChangeEvent(EventName.WaitForSpeech);
            animateNewDialogue = StartCoroutine(AnimateNewDialogue());
        }
        else
        {
            ro.left = chatBubblePadding;
            ro.right = 3 * chatBubblePadding;
            chatContent.transform.SetSiblingIndex(0);
        }
    }
    Outline userLastResponseTxtBorder = null;
    Color[] borderColors = new Color[]
        {
            new Color(255f/255f,118f/255f,117f/255f),
            new Color(178f/255f,190f/255f,195f/255f),
            new Color(162f/255f,155f/255f,254f/255f),
            new Color(255f/255f,234f/255f,167f/255f)
        };
    public IEnumerator AnimateNewDialogue()
    {
        WaitForSeconds delay = new WaitForSeconds(0.01f * Time.timeScale);

        int colorIdx = 0;
        while (true)
        {
            for (int i = 0; i < 100; i++) {
                userLastResponseTxtBorder.effectColor =
Color.Lerp(userLastResponseTxtBorder.effectColor,
borderColors[colorIdx],i/100f);
                yield return delay;
            }
            colorIdx++;
            colorIdx %= borderColors.Length;
        }
    }

    private void RenderDialogueSuggestion(string message)
    {
        userChatOptionTxt.text = message;
    }

    public void RenderDialogueSuggestion(bool isNextSuggestion = true)
    {
        string message = "";
        if (!isNextSuggestion)
        {
            message = PreviousDialogueSuggestion;
```

```
        } else
        {
            message = NextDialogueSuggestion;
        }
        RenderDialogueSuggestion(message);
    }

    public IEnumerator RenderSelectedDialogue()
    {
        enableChatSelectionField = false;
        sfxController.PlayBubbleSFX();
        yield return chatbotController.delay.preAddChatDelay;
        AddChat(currentDialogueSuggestion, true);
        userChatOptionTxt.text = "...";
        enableChatSelectionField = true;
        yield break;
    }

    public void RerenderSelectedDialogue(string STTResult)
    {
        string actualText = currentDialogueSuggestion;
        HashSet<string> STTWords = new HashSet<string>(STTResult.Split('
'));

        StringBuilder sb = new StringBuilder();
        string normalizedWords = null;
        foreach (string word in actualText.Split(' ')) {

            normalizedWords = Utility.NormalizeString(word);
            if (STTWords.Contains(normalizedWords)) {
                sb.Append("<color=#81deff>");
            } else
            {
                sb.Append("<color=#ff82a0>");
            }
            sb.Append(word);
            sb.Append("</color> ");
        }
        userLastResponseTxt.text = sb.ToString();

        ShowBotReply();
    }
    public void ShowBotReply()
    {
        ChangeEvent(EventName.ChatbotTTS,
chatbotController.ReplyToBot(currentDialogueSuggestion,true));
    }

    public void RestartSession()
    {
```

```
        if(animateNewDialogue != null) { StopCoroutine(animateNewDialogue);
animateNewDialogue = null; }
        switch (currentEvent)
        {
            case EventName.ChatbotTTS:
                eventStartingTime = 0;
                eventLength = 0;
                break;
        }
        ClearChatContent();
        currentEvent = EventName.Nothing;
        userChatController.SetChatOptionFunctionality(true);
    }

    public void ClearChatContent()
    {
        for (int i = 0; i < chatContentPanel.transform.childCount; i++) {
            Destroy(chatContentPanel.transform.GetChild(i).gameObject);
        }
    }

    void Update()
    {
        switch (currentEvent)
        {
            case EventName.ChatbotTTS:
                if(Time.time > eventStartingTime + eventLength)
                {
                    RenderDialogueSuggestion(NextDialogueSuggestion);
                    ChangeEvent(EventName.Nothing);
                }
                break;
        }
    }

    void LoadingForNewEvent()
    {
        userChatOptionTxt.text = "...";
        userChatController.SetChatOptionFunctionality(false);
    }
}
```

Application Code – VoskSpeechToText

```
using System;
using System.Collections;
using System.Collections.Concurrent;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Threading;
```

```
using System.Threading.Tasks;
using Ionic.Zip;
using Unity.Profiling;
using UnityEngine;
using UnityEngine.Networking;
using UnityEngine.UI;
using Vosk;

public class VoskSpeechToText : MonoBehaviour
{
    [Tooltip("Location of the model, relative to the Streaming Assets
folder.")]
    public string ModelPath = "";

    [Tooltip("The source of the microphone input.")]

    public VoiceProcessor VoiceProcessor;
    [Tooltip("The Max number of alternatives that will be processed.")]
    public int MaxAlternatives = 3;

    [Tooltip("How long should we record before restarting?")]
    public float MaxRecordLength = 5;

    [Tooltip("The phrases that will be detected. If left empty, all words
will be detected.")]
    public List<string> KeyPhrases = new List<string>();

    //Cached version of the Vosk Model.
    private Model _model;

    //Cached version of the Vosk recognizer.
    private VoskRecognizer _recognizer;

    //Conditional flag to see if a recognizer has already been created.
    //TODO: Allow for runtime changes to the recognizer.
    private bool _recognizerReady;

    //Holds all of the audio data until the user stops talking.
    private readonly List<short> _buffer = new List<short>();

    //Called when the the state of the controller changes.
    public Action<string> OnStatusUpdated;

    //Called after the user is done speaking and vosk processes the audio.
    public Action<string> OnTranscriptionResult;

    //The absolute path to the decompressed model folder.
    private string _decompressedModelPath;

    //A string that contains the keywords in Json Array format
```

```
    private string _grammar = "";

    //Flag that is used to wait for the model file to decompress
successfully.
    private bool _isDecompressing;

    //Flag that is used to wait for the the script to start successfully.
    private bool _isInitializing;

    //Flag that is used to check if Vosk was started.
    private bool _didInit;

    private float _startRecordTime;

    //Threading Logic

    //Lock for the string result
    private readonly object _resultLock = new object();

    //The json string that was returned from Vosk
    private string _threadedRecognitionResult;

    //The result that was called in the Recognition event.
    private string _result;

    //Thread safe queue of microphone data.
    private readonly ConcurrentQueue<short[]> _threadedBufferQueue = new
ConcurrentQueue<short[]>();

    //lock for StreamingIsBusy flag.
    private int _threadSafeBoolBackValue = 0;

    //Flag to see if we are processing speech to text data.
    public bool StreamingIsBusy
    {
        get => (Interlocked.CompareExchange(ref _threadSafeBoolBackValue,
1, 1) == 1);
        set
        {
            if (value) Interlocked.CompareExchange(ref
_threadSafeBoolBackValue, 1, 0);
            else Interlocked.CompareExchange(ref _threadSafeBoolBackValue,
0, 1);
        }
    }

    static readonly ProfilerMarker voskRecognizerCreateMarker = new
ProfilerMarker("VoskRecognizer.Create");
    static readonly ProfilerMarker voskRecognizerReadMarker = new
ProfilerMarker("VoskRecognizer.AcceptWaveform");
```

```
    private void Awake()
    {
        // TextAsset dictionary =
Resources.Load<TextAsset>("Text/dictionary");
        // KeyPhrases = dictionary.text.Split(',').ToList();

    }

    public void PrepareVocabularies(List<string> vocabularies)
    {
        KeyPhrases = vocabularies;
    }
    //If Auto start is enabled, starts vosk speech to text.
    void Start()
    {
        // TextAsset dictionary =
Resources.Load<TextAsset>("Text/dictionary");

            // StartVoskStt(dictionary.text.Split('¥n').ToList());
        StartVoskStt();
    }

    /// <summary>
    /// Start Vosk Speech to text
    /// </summary>
    /// <param name="keyPhrases">A list of keywords/phrases. Keywords need
to exist in the models dictionary, so some words like "webview" are better
detected as two more common words "web view".</param>
    /// <param name="modelPath">The path to the model folder relative to
StreamingAssets. If the path has a .zip ending, it will be decompressed
into the application data persistent folder.</param>
    /// <param name="startMicrophone">"Should the microphone after vosk
initializes?</param>
    /// <param name="maxAlternatives">The maximum number of alternative
phrases detected</param>
    public void StartVoskStt( List<string> keyPhrases= null, string
modelPath = default, bool startMicrophone = false, int maxAlternatives =
3)
    {
        if (_isInitializing)
        {
            Debug.LogError("Initializing in progress!");
            return;
        }
        if (_didInit)
        {
            Debug.LogError("Vosk has already been initialized!");
            return;
```

```
        }

        if (!string.IsNullOrEmpty(modelPath))
        {
            ModelPath = modelPath;
        }

        if (keyPhrases != null)
        {
            KeyPhrases = keyPhrases;
        }

        MaxAlternatives = maxAlternatives;
        StartCoroutine(DoStartVoskStt(startMicrophone));
    }

    //Decompress model, load settings, start Vosk and optionally start the
microphone
    private IEnumerator DoStartVoskStt(bool startMicrophone)
    {
        _isInitializing = true;
        yield return WaitForMicrophoneInput();

        yield return Decompress();

        OnStatusUpdated?.Invoke("Loading Model from: " +
_decompressedModelPath);
        Vosk.Vosk.SetLogLevel(0);
        _model = new Model(_decompressedModelPath);

        yield return null;

        OnStatusUpdated?.Invoke("Initialized");
        VoiceProcessor.OnFrameCaptured += VoiceProcessorOnOnFrameCaptured;
        VoiceProcessor.OnRecordingStop += VoiceProcessorOnOnRecordingStop;

        if (startMicrophone)
            VoiceProcessor.StartRecording();

        _isInitializing = false;
        _didInit = true;
    }

    //Translates the KeyPhraseses into a json array and appends the
`[unk]` keyword at the end to tell vosk to filter other phrases.
    private void UpdateGrammar()
    {
        if (KeyPhrases.Count == 0)
        {
            _grammar = "";
```

```
            return;
        }

        JSONArray keywords = new JSONArray();
        foreach (string keyphrase in KeyPhrases)
        {
            keywords.Add(new JSONString(keyphrase.ToLower()));
        }

        keywords.Add(new JSONString("[unk]"));

        _grammar = keywords.ToString();

        // print(_grammar);
    }

    //Decompress the model zip file or return the location of the
decompressed files.
    private IEnumerator Decompress()
    {
        if (!Path.HasExtension(ModelPath)
            || Directory.Exists(
                Path.Combine(Application.persistentDataPath,
Path.GetFileNameWithoutExtension(ModelPath))))
        {
            OnStatusUpdated?.Invoke("Using existing decompressed model.");
            _decompressedModelPath =
                Path.Combine(Application.persistentDataPath,
Path.GetFileNameWithoutExtension(ModelPath));
            Debug.Log(_decompressedModelPath);

            yield break;
        }

        OnStatusUpdated?.Invoke("Decompressing model...");
        string dataPath = Path.Combine(Application.streamingAssetsPath,
ModelPath);

        Stream dataStream;
        // Read data from the streaming assets path. You cannot access the
streaming assets directly on Android.
        if (dataPath.Contains("://"))
        {
            UnityWebRequest www = UnityWebRequest.Get(dataPath);
            www.SendWebRequest();
            while (!www.isDone)
            {
                yield return null;
            }
            dataStream = new MemoryStream(www.downloadHandler.data);
```

```
        }
        // Read the file directly on valid platforms.
        else
        {
            dataStream = File.OpenRead(dataPath);
        }

        //Read the Zip File
        var zipFile = ZipFile.Read(dataStream);

        //Listen for the zip file to complete extraction
        zipFile.ExtractProgress += ZipFileOnExtractProgress;

        //Update status text
        OnStatusUpdated?.Invoke("Reading Zip file");

        //Start Extraction
        zipFile.ExtractAll(Application.persistentDataPath);

        //Wait until it's complete
        while (_isDecompressing == false)
        {
            yield return null;
        }
        //_decompressedModelPath =
        //Update status text
        OnStatusUpdated?.Invoke("Decompressing complete!");

        //Wait a second in case we need to initialize another object.
        yield return new WaitForSeconds(1);
        //Dispose the zipfile reader.
        zipFile.Dispose();
    }

    ///The function that is called when the zip file extraction process is
updated.
    private void ZipFileOnExtractProgress(object sender,
ExtractProgressEventArgs e)
    {
        if (e.EventType == ZipProgressEventType.Extracting_AfterExtractAll)
        {
            _isDecompressing = true;
            _decompressedModelPath =
Path.Combine(Application.persistentDataPath,
Path.GetFileNameWithoutExtension(ModelPath));
        }
    }

    //Wait until microphones are initialized
    private IEnumerator WaitForMicrophoneInput()
```

```
    {
        while (Microphone.devices.Length <= 0)
            yield return null;
    }

    //Can be called from a script or a GUI button to start detection.
    public bool ToggleRecording()
    {
        if (!VoiceProcessor.IsRecording)
        {
            VoiceProcessor.StartRecording();
            return true;
        }

        VoiceProcessor.StopRecording();
        return false;
    }

    //Calls the On Phrase Recognized event on the Unity Thread
    void Update()
    {
        lock (_resultLock)
        {
            if (_result != _threadedRecognitionResult)
            {
                OnStatusUpdated?.Invoke("Received Result");
                _result = _threadedRecognitionResult;
                OnTranscriptionResult?.Invoke(_result);
            }
        }
    }

    //Callback from the voice processor when new audio is detected
    private void VoiceProcessorOnOnFrameCaptured(short[] samples)
    {
        //Only change the state if we are starting fresh
        if (StreamingIsBusy == false && _buffer.Count == 0)
        {
            _startRecordTime = Time.time;
            OnStatusUpdated?.Invoke("Listening");
        }

        if (Time.time - _startRecordTime > MaxRecordLength)
        {
            VoiceProcessorOnOnRecordingStop();
            return;
        }
        else
        {
            _buffer.AddRange(samples);
```

```
    }

}

//Callback from the voice processor when recording stops
private void VoiceProcessorOnOnRecordingStop()
{
    if (StreamingIsBusy)
        return;

    OnStatusUpdated?.Invoke("Fetching Result");
    StreamingIsBusy = true;
    _threadedBufferQueue.Enqueue(_buffer.ToArray());
    Task.Run(ThreadedWork).ConfigureAwait(false);

    _buffer.Clear();
}

//Feeds the autio logic into the vosk recorgnizer
private async Task ThreadedWork()
{
    StreamingIsBusy = true;
    voskRecognizerCreateMarker.Begin();

    UpdateGrammar();
    _recognizer = new VoskRecognizer(_model, 16000.0f, _grammar);

    _recognizer.SetMaxAlternatives(MaxAlternatives);
    _recognizer.SetWords(true);
    await Task.Delay(100);

    voskRecognizerCreateMarker.End();

    voskRecognizerReadMarker.Begin();

    while (_threadedBufferQueue.Count > 0)
    {
        if (_threadedBufferQueue.TryDequeue(out short[] voiceResult))
        {
            _recognizer.AcceptWaveform(voiceResult, voiceResult.Length);
                lock (_resultLock)
                {
                    _threadedRecognitionResult = _recognizer.Result();
                }
        }
    }

    voskRecognizerReadMarker.End();
```

```
        //We wait 2seconds to avoid getting a partial result when
processing audio immediately after.
        await Task.Delay(2000);
        StreamingIsBusy = false;

    }
}
```

Application Code - ExternalLoggerController

```
using UnityEngine.Networking;
using UnityEngine;
using System.Collections;
using System.Text;

public class ExternalLoggerController : MonoBehaviour
{
    public bool isConnectedToInternet = false;

    public void Start()
    {
        StartCoroutine(CheckInternetConnection());
    }

    public IEnumerator CheckInternetConnection()
    {
        using(UnityWebRequest webRequest =
UnityWebRequest.Get("https://www.google.com/"))
        {
            yield return webRequest.SendWebRequest();

            switch (webRequest.result)
            {
                case UnityWebRequest.Result.Success:
                    isConnectedToInternet = true;
                    StartCoroutine(WarmUpLogServer());
                    break;
                default:
                    isConnectedToInternet=false;
                    break;
            }
        }
        yield break;
    }
    public IEnumerator WarmUpLogServer()
    {
        using (UnityWebRequest webRequest =
UnityWebRequest.Get("https://anna-logger.herokuapp.com/fire-up"))
        {
            yield return webRequest.SendWebRequest();
        }
```

```
        yield break;
    }
    public IEnumerator SendDataToLogger(string referenceText, string
detectedText)
    {
        detectedText += "###"+IDGeneratorController.Instance.appID;
        if (!isConnectedToInternet)
        {
            yield break;
        }

        LogItem logItem = new LogItem(referenceText, detectedText);
        UnityWebRequest webRequest =
            new UnityWebRequest("https://anna-logger.herokuapp.com/log",
UnityWebRequest.kHttpVerbPOST)
            {
                uploadHandler = new UploadHandlerRaw(
                    Encoding.UTF8.GetBytes(
                        JsonUtility.ToJson(logItem)
                    )
                ),
                downloadHandler = new DownloadHandlerBuffer()
            };

        webRequest.SetRequestHeader("Content-Type", "application/json");

        yield return webRequest.SendWebRequest();

        switch (webRequest.result)
        {
            case UnityWebRequest.Result.Success:
                print(webRequest.downloadHandler.text);
                yield break;
            default:
                print("Unsuccessful");
                break;
        }
    }
}
```

Application Code - VoiceProcessor

```
/* * * * *
 * A unity voice processor
 * ------------------------------
 *
 * A Unity script for recording and delivering frames of audio for real-
time processing
 *
 * Written by Picovoice
 * 2021-02-19
```

```
 *
 * Apache License
 *
 * Copyright (c) 2021 Picovoice
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 *   you may not use this file except in compliance with the License.
 *   You may obtain a copy of the License at
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 *   Unless required by applicable law or agreed to in writing, software
 *   distributed under the License is distributed on an "AS IS" BASIS,
 *   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
 *   See the License for the specific language governing permissions and
 *   limitations under the License.
 *
 * * * * */
using System;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;


/// <summary>
/// Class that records audio and delivers frames for real-time audio
processing
/// </summary>
public class VoiceProcessor : MonoBehaviour
{

    void Awake()
    {

        frequencyBands.Add(0, 4);
        frequencyBands.Add(16, 12);
        frequencyBands.Add(24, 24);
        frequencyBands.Add(32, 12);
        frequencyBands.Add(40, 8);
#if UNITY_WEBGL && !UNITY_EDITOR
        Microphone.Init();
        Microphone.QueryAudioInput();
#endif
        UpdateDevices();
        // print(CurrentDeviceIndex);
        // print(CurrentDeviceName);
    }

    void Update()
```

```
    {
#if UNITY_EDITOR
        if (CurrentDeviceIndex != MicrophoneIndex)
        {
            ChangeDevice(MicrophoneIndex);
        }
#endif
#if UNITY_WEBGL && !UNITY_EDITOR
        Microphone.Update();
#endif

    }
    /// <summary>
    /// Indicates whether microphone is capturing or not
    /// </summary>
    public bool IsRecording
    {
        get { return _audioClip != null &&
Microphone.IsRecording(CurrentDeviceName); }
    }

    [SerializeField] private int MicrophoneIndex;

    /// <summary>
    /// Sample rate of recorded audio
    /// </summary>
    public int SampleRate { get; private set; }

    /// <summary>
    /// Size of audio frames that are delivered
    /// </summary>
    public int FrameLength { get; private set; }

    /// <summary>
    /// Event where frames of audio are delivered
    /// </summary>
    public event Action<short[]> OnFrameCaptured;

    /// <summary>
    /// Event when audio capture thread stops
    /// </summary>
    public event Action OnRecordingStop;

    /// <summary>
    /// Event when audio capture thread starts
    /// </summary>
    public event Action OnRecordingStart;

    /// <summary>
    /// Available audio recording devices
```

```
    /// </summary>
    public List<string> Devices { get; private set; }

    /// <summary>
    /// Index of selected audio recording device
    /// </summary>
    public int CurrentDeviceIndex { get; private set; }

    /// <summary>
    /// Name of selected audio recording device
    /// </summary>
    public string CurrentDeviceName
    {
        get
        {
            if (CurrentDeviceIndex < 0 || CurrentDeviceIndex >=
Microphone.devices.Length)
                return string.Empty;
            return Devices[CurrentDeviceIndex];
        }
    }

    [Header("Voice Detection Settings")]
    [SerializeField, Tooltip("The minimum volume to detect voice input
for"), Range(0.0f, 1.0f)]
    private float _minimumSpeakingSampleValue = 0.05f;

    [SerializeField, Tooltip("Time in seconds of detected silence before
voice request is sent")]
    private float _silenceTimer = 1.0f;

    [SerializeField, Tooltip("Auto detect speech using the volume
threshold.")]
    private bool _autoDetect;

    private float _timeAtSilenceBegan;
    private bool _audioDetected;
    private bool _didDetect;
    private bool _transmit;


    AudioClip _audioClip;
    private event Action RestartRecording;

    public static float[] sampleBuffer = new float[512];
    public static float[] frequencyBand = new float[48];


    Dictionary<int, int> frequencyBands = new Dictionary<int, int>();
```

```
void MakeFrequencyBands()
{
    int count = 0;
    int sampleCount = 8;

    for (int i = 0; i < 48; i++)
    {
        float average = 0;

        if (frequencyBands.ContainsKey(i))
        {
            sampleCount = frequencyBands[i];
        }
        for (int j = 0; j < sampleCount; j++)
        {
            average += sampleBuffer[count] * (count + 1);
            count++;
        }
        average /= count;
        frequencyBand[i] = average * 80;
    }
}


/// <summary>
/// Updates list of available audio devices
/// </summary>
public void UpdateDevices()
{
    Devices = new List<string>();
    foreach (var device in Microphone.devices)
    {
        Devices.Add(device);
    }

    if (Devices == null || Devices.Count == 0)
    {
        CurrentDeviceIndex = -1;
        Debug.LogError("There is no valid recording device connected");
        return;
    }

    CurrentDeviceIndex = MicrophoneIndex;
}

/// <summary>
/// Change audio recording device
/// </summary>
```

```
    /// <param name="deviceIndex">Index of the new audio capture
device</param>
    public void ChangeDevice(int deviceIndex)
    {
        if (deviceIndex < 0 || deviceIndex >= Devices.Count)
        {
            Debug.LogError(string.Format("Specified device index {0} is not
a valid recording device", deviceIndex));
            return;
        }

        if (IsRecording)
        {
            // one time event to restart recording with the new device
            // the moment the last session has completed
            RestartRecording += () =>
            {
                CurrentDeviceIndex = deviceIndex;
                StartRecording(SampleRate, FrameLength);
                RestartRecording = null;
            };
            StopRecording();
        }
        else
        {
            CurrentDeviceIndex = deviceIndex;
        }
    }

    /// <summary>
    /// Start recording audio
    /// </summary>
    /// <param name="sampleRate">Sample rate to record at</param>
    /// <param name="frameSize">Size of audio frames to be
delivered</param>
    /// <param name="autoDetect">Should the audio continuously record
based on the volume</param>
    public void StartRecording(int sampleRate = 16000, int frameSize =
512, bool ?autoDetect = null)
    {
        if (autoDetect != null)
        {
            _autoDetect = (bool) autoDetect;
        }

        if (IsRecording)
        {
            // if sample rate or frame size have changed, restart recording
            if (sampleRate != SampleRate || frameSize != FrameLength)
            {
```

```
                RestartRecording += () =>
                {
                    StartRecording(SampleRate, FrameLength, autoDetect);
                    RestartRecording = null;
                };
                StopRecording();
            }

            return;
        }

        SampleRate = sampleRate;
        FrameLength = frameSize;

        _audioClip = Microphone.Start(CurrentDeviceName, true, 1,
sampleRate);

        StartCoroutine(RecordData());
    }

    /// <summary>
    /// Stops recording audio
    /// </summary>
    public void StopRecording()
    {
        if (!IsRecording)
            return;

        Microphone.End(CurrentDeviceName);
        Destroy(_audioClip);
        _audioClip = null;
        _didDetect = false;

        StopCoroutine(RecordData());
    }

    /// <summary>
    /// Loop for buffering incoming audio data and delivering frames
    /// </summary>
    IEnumerator RecordData()
    {
        sampleBuffer = new float[FrameLength];
        int startReadPos = 0;

        if (OnRecordingStart != null)
            OnRecordingStart.Invoke();

        while (IsRecording)
        {
            int curClipPos = Microphone.GetPosition(CurrentDeviceName);
```

```
            // print(curClipPos);
            if (curClipPos < startReadPos)
                curClipPos += _audioClip.samples;
            int samplesAvailable = curClipPos - startReadPos;
            if (samplesAvailable < FrameLength)
            {
                yield return null;
                continue;
            }

            int endReadPos = startReadPos + FrameLength;
            if (endReadPos > _audioClip.samples)
            {
                // fragmented read (wraps around to beginning of clip)
                // read bit at end of clip
                int numSamplesClipEnd = _audioClip.samples - startReadPos;
                float[] endClipSamples = new float[numSamplesClipEnd];
                _audioClip.GetData(endClipSamples, startReadPos);

                // read bit at start of clip
                int numSamplesClipStart = endReadPos - _audioClip.samples;
                float[] startClipSamples = new float[numSamplesClipStart];
                _audioClip.GetData(startClipSamples, 0);

                // combine to form full frame
                Buffer.BlockCopy(endClipSamples, 0, sampleBuffer, 0,
numSamplesClipEnd);
                Buffer.BlockCopy(startClipSamples, 0, sampleBuffer,
numSamplesClipEnd, numSamplesClipStart);
            }
            else
            {
                _audioClip.GetData(sampleBuffer, startReadPos);
            }

            MakeFrequencyBands();

            startReadPos = endReadPos % _audioClip.samples;


            if (_autoDetect == false)
            {
                _transmit =_audioDetected = true;
            }
            else
            {
                float maxVolume = 0.0f;

                for (int i = 0; i < sampleBuffer.Length; i++)
                {
```

```
                if (sampleBuffer[i] > maxVolume)
                {
                    maxVolume = sampleBuffer[i];
                }
            }

            if (maxVolume >= _minimumSpeakingSampleValue)
            {
                _transmit= _audioDetected = true;
                _timeAtSilenceBegan = Time.time;
            }
            else
            {
                _transmit = false;

                if (_audioDetected && Time.time - _timeAtSilenceBegan >
_silenceTimer)
                {
                    _audioDetected = false;
                }
            }
        }

        if (_audioDetected)
        {
            _didDetect = true;
            // converts to 16-bit int samples
            short[] pcmBuffer = new short[sampleBuffer.Length];
            for (int i = 0; i < FrameLength; i++)
            {
                pcmBuffer[i] = (short) Math.Floor(sampleBuffer[i] *
short.MaxValue);
            }

            // raise buffer event
            if (OnFrameCaptured != null && _transmit)
                OnFrameCaptured.Invoke(pcmBuffer);
        }
        else
        {
            if (_didDetect)
            {
                if (OnRecordingStop != null)
                    OnRecordingStop.Invoke();
                _didDetect = false;
            }
        }
    }
```

```
        if (OnRecordingStop != null)
            OnRecordingStop.Invoke();
        if (RestartRecording != null)
            RestartRecording.Invoke();
    }

}
```

Application Code - LogItem

```
using System;
using System.Text;


[Serializable]
public class LogItem
{
    public Content content;
    public string signature;

    private static String secretKey;

    public static string SecretKey {
        get {
            if(secretKey == null)
            {
                string input = "this-is-anna-logger-not-so-secret-key-ayyy-
123456-!@#$%^&";

                secretKey = SHA1DigestString(input);
            }
            return secretKey;
        }
    }

    public LogItem(string referenceText, string detectedText)
    {
        content = new Content(referenceText, detectedText);
        this.signature =
SHA1DigestString(SHA1DigestString(content.ToString()) + SecretKey);
    }

    public static string SHA1DigestString(string input)
    {

        ASCIIEncoding encoding = new ASCIIEncoding();
        byte[] bytes = encoding.GetBytes(input);

        System.Security.Cryptography.SHA1CryptoServiceProvider sha = new
System.Security.Cryptography.SHA1CryptoServiceProvider();
```

```
        return BitConverter.ToString(sha.ComputeHash(bytes)).Replace("-",
String.Empty).ToLower();
    }


    [Serializable]
    public class Content
    {
        public string referenceText, detectedText;

        public Content(string referenceText, string detectedText)
        {
            this.referenceText = referenceText;
            this.detectedText = detectedText;
        }
        public override string ToString()
        {
            return String.Format("{{¥"referenceText¥": ¥"{0}¥",
¥"detectedText¥": ¥"{1}¥"}}", referenceText, detectedText).Replace("
",String.Empty);
        }
    }
}
```

Application Code - UserChatController

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.EventSystems;
using UnityEngine.UI;

public class UserChatController : MonoBehaviour
{
    [SerializeField]
    private DialogueController dialogueController;
    private EventTrigger chatOptionEvent;
    private Button btnPreviousOption, btnNextOption;
    public bool isInteractive;
    private void Awake()
    {
        isInteractive = true;
        dialogueController =
GameObject.Find("DialogueController").GetComponent<DialogueController>();
        btnPreviousOption =
transform.Find("BtnPreviousOption").GetComponent<Button>();
        chatOptionEvent =
transform.Find("ChatOption").GetComponent<EventTrigger>();
        btnNextOption =
transform.Find("BtnNextOption").GetComponent<Button>();
    }
```

```
    public void BtnNextDialogueEvent()
    {
        dialogueController.RenderDialogueSuggestion();
    }

    public void BtnPreviousDialogueEvent()
    {
        dialogueController.RenderDialogueSuggestion(false);
    }

    public void BtnSelectDialogueEvent()
    {
        if(dialogueController.enableChatSelectionField)
            StartCoroutine(dialogueController.RenderSelectedDialogue());
    }

    public void SetChatOptionFunctionality(bool enability)
    {
        chatOptionEvent.enabled = enability;
        btnPreviousOption.interactable = enability;
        btnNextOption.interactable = enability;
    }
}
```

Application Code - UserResponseController

```
using System.Collections;
using System.Collections.Generic;
using TMPro;
using UnityEngine;
using UnityEngine.EventSystems;

public class UserResponseController : MonoBehaviour
{
    private EventTrigger eventTrigger;
    private void Awake()
    {
        eventTrigger = GetComponent<EventTrigger>();
        EventTrigger.Entry entry = new EventTrigger.Entry();
        entry.eventID = EventTriggerType.PointerUp;
        entry.callback.AddListener((eventData) => { string selectedText =
transform.Find("Text").GetComponent<TextMeshProUGUI>().text;
AudioVisualizerController.Instance.OpenAVPanelEvent(selectedText);
Destroy(eventTrigger); Destroy(this);  });
        eventTrigger.triggers.Add(entry);
    }
}
```

Application Code - Utility

```
using System.Collections.Generic;
```

```
using System.Text;

public class Utility
{
    public static Dictionary<string, string[]> processedAbbrevations = new
Dictionary<string, string[]>()
    {
        {"tdu", new string[]{"t","d","u" } },
        {"i¥'m", new string[] {"i","m"} }
    };
    public static string NormalizeString(string inputString)
    {
        string normalizedString = inputString.ToLower();

        var sb = new StringBuilder();

        foreach (char c in normalizedString)
        {
            if (!char.IsPunctuation(c))
                sb.Append(c);
        }

        return sb.ToString();
    }

    public static bool WordInTokens(string word, HashSet<string> tokens)
    {
        string normalizedWord = word.ToLower();

        if (!processedAbbrevations.ContainsKey(normalizedWord))
            return tokens.Contains(NormalizeString(normalizedWord));
        foreach (string element in processedAbbrevations[normalizedWord])
        {
            if (!tokens.Contains(element))
            {
                return false;
            }
        }
        return true;

    }
}
```

Application Code - AudioVisualizerController

```
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using TMPro;
using UnityEngine;
```

```
using UnityEngine.UI;

public class AudioVisualizerController : MonoBehaviour
{
    DialogueController dialogueController;
    Animator animator;
    public static AudioVisualizerController _instance;
    public static AudioVisualizerController Instance { get {
            if(_instance == null) _instance =
FindObjectOfType<AudioVisualizerController>();
            return _instance;
        }
    }

    private GameObject screenOutPanel;
    public AudioClip recordingStart, recordingStop;
    private AudioSource audioSource;

    private RectTransform[] visualBars;

    public float barScaler = 1.0f;
    public float barMaxSize = 460f;
    public float barMinSize = 20f;

    private bool isPanelActive=false;

    GameObject userSelectedDialogue;
    TextMeshProUGUI userSelectedDialogueText;
    // Start is called before the first frame update
    void Start()
    {
        userSelectedDialogue =
transform.Find("UserSelectedDialogue").gameObject;
        userSelectedDialogueText =
userSelectedDialogue.transform.Find("ChatOption").Find("Text").GetComponen
t<TextMeshProUGUI>();
        userSelectedDialogue.SetActive(false);
        dialogueController =
GameObject.Find("DialogueController").GetComponent<DialogueController>();
        animator = GetComponent<Animator>();

        screenOutPanel =
GameObject.Find("Canvas").transform.Find("BaseImage").Find("ScreenOutPanel
").gameObject;
        audioSource = GetComponent<AudioSource>();

        var tempVisualBars = new HashSet<RectTransform>(

transform.Find("VisualBarContainer").GetComponentsInChildren<RectTransform
>()
```

```
        );

tempVisualBars.Remove(transform.Find("VisualBarContainer").GetComponent<Re
ctTransform>());
        visualBars = tempVisualBars.ToArray();
    }

    // Update is called once per frame
    void Update()
    {
        if (isPanelActive)
        {
            for (int i = 0; i < VoiceProcessor.frequencyBand.Length; i++)
            {
                Vector2 size = visualBars[i].sizeDelta;
                size.y =
Mathf.Min(barMaxSize,Mathf.Max(barMinSize,VoiceProcessor.frequencyBand[i]
* barScaler));
                visualBars[i].sizeDelta = size;

            }
        }

    }
    public void EndRecordingAndCloseAVEvent(bool playRecordingStopAudio =
true)
    {
        StartCoroutine(CloseAVPanel(playRecordingStopAudio));
        isPanelActive = false;
    }

    public void OpenAVPanelEvent(string selectedText)
    {
        isPanelActive = true;
        userSelectedDialogue.SetActive(true);
        userSelectedDialogueText.text = selectedText;
        StartCoroutine(OpenAVPanel());
    }
    public IEnumerator CloseAVPanel(bool playRecordingStopAudio)
    {
        yield return new WaitForSeconds(0.5f);
        dialogueController.ToggleSTTModule();
        if (playRecordingStopAudio) audioSource.PlayOneShot(recordingStop);
        animator.SetBool("InScreen", !animator.GetBool("InScreen"));
        screenOutPanel.gameObject.SetActive(false);
        userSelectedDialogue.SetActive(false);
        yield break;
    }
    public IEnumerator OpenAVPanel()
    {
```

```
        dialogueController.ToggleSTTModule();
        screenOutPanel.gameObject.SetActive(true);
        animator.SetBool("InScreen", !animator.GetBool("InScreen"));
        yield return new WaitForSeconds(0.5f);
        audioSource.PlayOneShot(recordingStart);
        yield break;
    }
}
```

Application Code - FormDialogueController

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class FormDialogueController : MonoBehaviour
{
    GameObject backdropPanel, headerPanel, bodyPanel;

    public void Start()
    {
        backdropPanel = transform.parent.Find("ScreenOutPanel").gameObject;
        headerPanel = transform.Find("Header").gameObject;
        bodyPanel = transform.Find("Body").gameObject;

        ToggleActivation();
    }
    public void ToggleActivation()
    {
        backdropPanel.SetActive(!backdropPanel.activeSelf);
        headerPanel.SetActive(!headerPanel.activeSelf);
        bodyPanel.SetActive(!bodyPanel.activeSelf);
    }
}
```

Application Code - IDGeneratorController

```
using System.Collections;
using System.Collections.Generic;
using TMPro;
using UnityEngine;

public class IDGeneratorController : MonoBehaviour
{
    public string appID = "";

    private static IDGeneratorController _instance;

    public static IDGeneratorController Instance { get {
```

```
            if(_instance == null)
            {
                _instance = 
GameObject.FindObjectOfType<IDGeneratorController>();
            }
            return _instance;
        }
    }

    // Start is called before the first frame update
    void Start()
    {

        appID = PlayerPrefs.GetString("ApplicationID");
        if(appID.Equals(""))
        {
            appID = ColorUtility.ToHtmlStringRGB(Random.ColorHSV());
            PlayerPrefs.SetString("ApplicationID", appID);
        }

        this.GetComponent<TextMeshProUGUI>().text += " ("+appID+")";
    }

    // Update is called once per frame
    void Update()
    {

    }
}
```

Application Code - SFXController

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class SFXController : MonoBehaviour
{
    public AudioClip bubbleSFX;

    private AudioSource audioSource;
    // Start is called before the first frame update
    void Start()
    {
        audioSource = GetComponent<AudioSource>();
    }

    public void PlayBubbleSFX()
    {
        audioSource.PlayOneShot(bubbleSFX);
```

```
    }
}
```

Application Code - SideNavigationController

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class SideNavigationController : MonoBehaviour
{
    Animator animator;
    bool isActive = false;
    private void Awake()
    {
        animator = GetComponent<Animator>();
    }

    // Update is called once per frame
    void Update()
    {
        animator = GetComponent<Animator>();
    }

    public void ToggleSideNavigation()
    {
        isActive = !isActive;
        animator.SetBool("active", isActive);
    }
}
```

Application Code - TopicListController

```
using System.Collections;
using System.Collections.Generic;
using TMPro;
using UnityEngine;
using UnityEngine.UI;

public class TopicListController : MonoBehaviour
```

```
{
    private GameObject topicItem;
    private Transform topicContentPanel;
    private ChatbotController chatbotController;
    private FormDialogueController formDialogueController;

    [SerializeField]
    private float topicItemFontSize = 90f;
    [SerializeField]
    private float topicItemHeight = 250f;
    private void Awake()
    {
        topicContentPanel = transform.Find("TopicContentPanel").transform;
        topicItem = Resources.Load<GameObject>("Prefabs/TopicItem");


topicItem.transform.Find("Name").GetComponent<TextMeshProUGUI>().fontSize
= topicItemFontSize;

        chatbotController =
GameObject.Find("ChatbotController").GetComponent<ChatbotController>();
        formDialogueController = GameObject.Find("Canvas").transform.

Find("BaseImage").Find("ChangeDialoguePanel").GetComponent<FormDialogueCon
troller>();


        float panelWidth =
transform.Find("TopicContentPanel").GetComponent<RectTransform>().sizeDelt
a.x;

        LayoutElement topicItemLE =
topicItem.GetComponent<LayoutElement>();

        topicItemLE.minWidth = panelWidth;
        topicItemLE.minHeight = topicItemHeight;
    }
    public void GenerateTopicItems(string[] items)
    {
        for (int idx = 0; idx < items.Length; idx++)
        {
            GameObject topicItemGO = Instantiate(topicItem);
            topicItemGO.transform.SetParent(topicContentPanel, false);

topicItemGO.transform.Find("Name").GetComponent<TextMeshProUGUI>().text =
items[idx];
            int currentIndex = idx;

            topicItemGO.GetComponent<Button>().onClick.AddListener(() =>
TopicOnSelect(currentIndex));
```

```
        }
    }

    public void TopicOnSelect(int index)
    {
        chatbotController.PrepareChangeTopic(index);
        formDialogueController.ToggleActivation();
    }
}
```