# Generative Artificial Intelligence Chatbot Integration for Virtual Reality Language Learning

**Conference Paper**

**Author(s):**
Cheng, Long; Gisler, Joy; Lao, Kenli; Kunz, Andreas

**Publication date:**
2025

**Permanent link:**
https://doi.org/https://doi.org/10.3929/ethz-b-000733441



**Originally published in:**
https://doi.org/10.1109/vrw66409.2025.00035

# Generative Artificial Intelligence Chatbot Integration for Virtual Reality Language Learning

Long Cheng[*†]
ETH Zurich
RhySearch

Joy Gisler [*‡]
ETH Zurich

Kenli Lao[§]
ETH Zurich

Andreas Kunz[¶]
ETH Zurich

(a) (b) (c)

Figure 1: First person view of interaction with AI agent in different languages and contexts: (a) English (b) German (c) Japanese

**Abstract**

This paper describes the potential of integrating generative artificial intelligence (GenAI) based chatbots into virtual reality (VR) for language learning. Existing VR and AI-based systems primarily focus on practicing vocabulary or written skills, often neglecting speaking practice. To address this gap, a VR-enabled conversation trainer, chatbot AI for VR (CHAI) [1], is introduced. The system integrates speech-to-text, a large language model, text-to-speech, and an avatar to enable natural conversations with CHAI. It is evaluated for its language error detection capabilities in English and German. The results show high accuracy in detecting language errors and demonstrate substantial reproducibility in error detection.

**Index Terms:** Virtual Reality, Large Language Model, Chatbot

## 1 Introduction

The ability to speak foreign languages has become a necessity for work, personal relationships, academic pursuits, and cross-cultural understanding [12]. However, learning a new language is often accompanied by several challenges. These challenges include a lack of time for in-person courses or a fear of making mistakes, which subsequently leads to neglecting practicing the new language. Given the challenges in language learning and the potential of technologies such as Virtual Reality (VR), new opportunities are emerging. VR allows users to immerse in a foreign culture without the need to travel. However, current interactions within VR are limited by pre-scripted or hard-coded dialogues. The integration of Artificial Intelligence (AI), such as Large Language Models (LLM), can overcome this limitation by enabling more natural, dynamic conversations. LLM-powered chatbots act as virtual teachers, providing real-time feedback. Combining VR with LLMs provides language learners an opportunity to practice conversational skills in an engaging, motivating, and low-pressure setting.

[*]contributed equally
[†]e-mail: locheng@ethz.ch
[‡]e-mail: gj@ethz.ch
[§]e-mail: kenlao@ethz.ch
[¶]e-mail: kunz@ethz.ch

[1]We demo the application in https://youtu.be/QHouH0A23eM

## 2 Related Work

Research and studies on VR assisted language learning gained interest in recent years [3, 10], demonstrating the increasing popularity of VR for language learning. In particular games in VR for language learning have gained popularity [1, 9, 18, 6]. The results from those works indicate that VR is a valuable tool for language learners, particularly for those who find traditional methods challenging. The VR system significantly improved learners' communicative ability. Learners also reported a sense of presence and positive perceptions of VR's realness. Further, Kaplan & Guber [13] stated that a VR training environment lowers the "Foreign Language Anxiety" compared to online lessons on Zoom.

An increasing number of VR learning environments also involves chatbots, starting from a scripted version "ELIZA" [2] to recent ones that are based on large language models (LLM) such as GPT-3 [19]. Its ability to generate natural language, remember previous inputs, and respond to diverse prompts made it widely popular. In response, competitors like Google's Gemini [7], Meta's Llama [22], and Mistral AI [17] emerged, further advancing LLM technology. The rise of LLM also opened new paths for research. Kohnke et al. [15] performed a technology review to examine the potential of ChatGPT for language teaching and learning. Their review underlines ChatGPT's ability to simulate interactions, providing language learners with opportunities to practice. Key features include defining words in context, correcting language mistakes, and offering translations. As stated by Zhang & Huang [24], AI chatbots significantly enhanced vocabulary acquisition, showing strong long-term retention for vocabulary. This was confirmed by Karatas et al. [14] and Li et al. [16], who stated that ChatGPT positively impacts students' writing, grammar, and vocabulary skills, while also boosting motivation and engagement. Young & Shishido [23] explored the potential of ChatGPT to generate dialogues for a

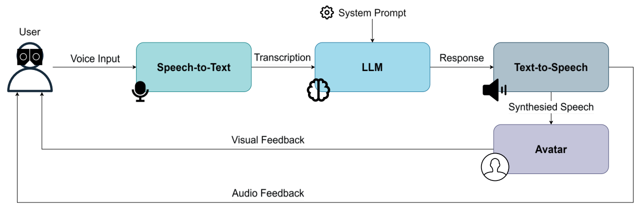


Figure 2: System design of CHAI.

chatbot to help students to practice their speaking skills. The dialogues generated by ChatGPT are appropriate for students at the A2 level (elementary) according to the Common European Framework of Reference for Languages (CEFR). The true potential of current AI chatbots lies in their adaptability to users. Research on the multilingual capabilities of ChatGPT has shown that while it performs well in high-resource languages. Not only do they support language learning, but they also offer a stress-free, anxiety-tolerant, and readily accessible platform for interaction [8].

As shown in the above, LLM-based VR applications for language learning are efficient, immersive, and well-received by users. However, recent studies often focus on vocabulary acquisition and general language skills rather than conversational and speaking practice, resulting in a limited focus on conversational practice, especially for improving speaking skills. Hence, there is currently no VR-enabled, LLM-based chatbot specifically for practicing speaking skills in the context of language learning, which is the focus of this paper. We first introduce the system design of an LLM-based AI chatbot. The system integrates four components, Speech-to-Text (STT), LLM, Text-to-Speech (TTS), and a VR environment with an avatar. These components work together to enable natural interaction for language speaking practice. Based on this, we evaluated five models of the STT together with three established LLM. After introducing the developed prototype, a sensitivity analysis was conducted to assess it's error detection capability. The analysis focused on two languages, German and English, testing the system's ability to detect language mistakes. This gave insight into the prototype's effectiveness for language learning.

## 3 System Design

The Chatbot AI for VR (CHAI) implemented in this paper must enable users to engage in natural conversations with a LLM in a virtual environment. Additionally, the AI chatbot must provide feedback regarding potential language errors made by the user. Therefore, it must possess the following capabilities:

- Speech recognition: CHAI must recognize the user's speech and convert it into a format suitable for the LLM.
- LLM processing: The LLM must generate appropriate responses based on the user's input.
- Output synthesis: The LLM's output must be transformed into a form that feels human-like and natural to the user, both visually and audibly.

Based on these requirements, the system design shown in Fig. 2 was devised. CHAI integrates four main components: STT, LLM, TTS, and an avatar within a virtual environment.

**STT**: OpenAI's general-speech recognition Whisper model family was selected for evaluation for the STT component. Five different model sizes (tiny, base, small, medium, and large) were tested based on their word rate (WR), word error rate (WER), and Character Error Rate (CER). Each model's performance was analyzed to determine which size offered the best balance between transcription speed and accuracy. The model with the best performance in these metrics was selected for integration into the final system. To assess the STT performance, three participants were asked to record themselves saying 10 example sentences in three languages (English, German, and Japanese). These recordings were processed by the STT, and the transcribed sentences were then evaluated.

WR is calculated based on the number of transcribed words over the elapsed time and quantifies how fast the model can recognize and transcribe audio inputs into corresponding words. WER is used to determine how accurate a speech recognition system can transcribe audio input into text. This is achieved by dividing the number of incorrectly transcribed words by the total number of words in the input. For non-Roman script-based languages, such as Chinese or Japanese, a more fine-grained metric – CER should be used. Word error rate and character error rate are defined in Equation 1.

$$\text{W(C)ER} = \frac{S+D+I}{N} \qquad \begin{aligned} S &= \text{Substitutions} \\ D &= \text{Deletions} \\ I &= \text{Insertions} \\ N &= \text{\# Words in reference} \end{aligned} \tag{1}$$

**LLM**: We evaluated three LLMs: OpenAI's GPT-4o, Llama3-8B, and Gemma-7B for language learning tasks. These models were assessed based on four dimensions: local deployability, cost, performance, and language capabilities. After evaluation, the LLM with the best balance across these criteria was chosen for integration into the system. Local deployability determines whether the LLM can run locally on a computer. Cost evaluates the hardware requirements needed to run the LLMs, as well as the associated financial costs. Performance focuses on the response time or inference time of the LLM. A critical aspect of performance is the time to the first token. Moreover, to evaluate the language proficiency of each LLM, a series of standardized tests based on the CEFR was conducted. These tests are also commonly used to determine an individual's language proficiency. According to the CEFR benchmark, a test candidate's language proficiency can be categorized into six levels from A1 (basic use) - C2 (proficient). Questions we tested are all multiple choice questions.

**Prompt Engineering**: The initial system prompt that was used to configure the LLM was developed through multiple iterations. Improvements were made iteratively by adjusting the system prompt until the LLM responded with a satisfying response. Additionally, the system prompt changes based on the current language, which can be either English, German, or Japanese. Specifically, we compared Zero-shot prompting and few-shot prompting using erroneous sentences in English and German. Zero-shot technique involves asking the LLM to solve a problem or perform a task without any prior information or examples on how the task should be solved. The LLM generates a response based on the knowledge and data it was trained on. Few-shot prompting provides the LLM with a few examples or specific information on how to solve the task before the actual prompt is given. These examples help guide the LLM, offering context it may not have seen during training.

**TTS**: For TTS, we employed the open-source library Coqui-TTS [4] due to straightforward installation and accessibility. The multilingual model xtts-v2 was selected for use, as it supports up to 17 languages and offers several pre-trained voice models. This eliminated the need for additional model training or voice cloning, making it a practical solution for CHAI. Due to its high subjectivity, TTS was not evaluated in the system.

**Avatar and virtual environment**: After experimentation and testing, we selected a customizable avatar software Ready Player Me[2]. Three avatars were created and customized for each language. Additionally, to synchronize the generated speech with the avatar, lip

[2] https://readyplayer.me/

synchronization was implemented using the Oculus Lipsync for Unity Development[3]. This allows the avatar's lips and mouth to move in synchronization with the audio output from the TTS component. For avatar animations, Mixamo[4] was used.

To improve user experience and provide appropriate context for language practice, three distinct scenes were developed in Unity. The environment is culturally aligned with the language. Relevant skyboxes are generated using Skybox AI[5]. Assets from the Unity Asset Store and Sketchfab were used for 3D models in the scenes.

In each environment, the avatar was placed directly in front of the user. Interaction with the chatbot was implemented with the push-to-talk principle. When the user presses the A button on the Meta Quest controller, they can speak to the chatbot. The user's spoken input is transcribed using the STT component. Once transcribed, the text is displayed in a text box within the scene. As soon as the first token of the LLM's response is generated, the response of LLM is displayed in a second text box. This second text box is situated below the user's transcribed input.

**Technical Setup**: The complete VE was implemented on a desktop PC with an Intel i9-10900K CPU and 32 GB DDR4 RAM, together with a GPU RTX 3090 with 24 GB VRAM. The Meta Quest 3 with the corresponding controllers was used as VR hardware.

### 3.1 System Architecture

The architecture follows a standard frontend and backend model (see Fig. 3). The frontend handles the visual and interactive elements, such as the virtual environment, the avatar, and overall user experience within Unity. The backend is responsible for processing data, managing system logic, and executing tasks like speech recognition, language generation, and speech synthesis. Communication between the frontend and backend is established through an Application Programming Interface (API), implemented using Python. The API conforms to the Representational State Transfer (REST) principles, allowing interaction via Hypertext Transfer Protocol (HTTP) requests.

**Frontend**: The frontend was built in Unity and focuses on creating a visual experience for the user. It consists of an avatar within a language dependent virtual environment, a user input display, and an LLM response display. The avatar interacts with the user, while the virtual environment provides context based on the practiced language. The user's speech is recorded, transcribed, then displayed in real time as a text box. The response generated by the LLM is also displayed in a text box. Unity's OpenXR toolkit package is employed to provide VR functionality, allowing users to interact within the virtual environment using a VR headset and controllers.

**Backend**: The backend handles the core functionality of the system, including STT, LLM processing, and TTS. Python was used to manage the integration of the STT and TTS components, while the LLM was deployed locally for efficient performance. The STT was implemented using OpenAI's Whisper model. Python handles the integration of Whisper for speech recognition. The transcribed input is sent to the LLM for further processing. An API endpoint `\transcribe` was created to call the STT functions when a user input is received. The LLM used in the system was downloaded via Hugging Face[6] and deployed locally through LM Studio, which offers an easy to use interface for managing and running LLMs. The corresponding API endpoint is named `\completion`. The locally deployed LLM processes the transcribed speech to generate an appropriate response. The LLM is preloaded into RAM/VRAM at the start of the system for low latency during conversations. The TTS component converts the LLM's response back into speech, allowing the avatar to deliver spoken responses to the user. Python was used to integrate and modify the Coqui-TTS library. Like the STT, an API endpoint for TTS, `\synthesize` was implemented. The endpoint is called to convert the generated text into speech.

All models (STT, LLM, and TTS) are preloaded into the computer's memory (RAM/VRAM) at the start to minimize response time and improve performance. The API endpoints for both the STT and TTS components allow for communication between the frontend and backend. When the user speaks, the frontend sends the audio input to the STT API endpoint, which transcribes the speech. The transcribed text is passed to Unity to display and to the LLM for processing. The generated response is then converted to speech via the TTS API endpoint before being forwarded to the user through the avatar in the virtual environment.

### 3.2 Sensitivity Analysis

To function as a language learning tool, CHAI is capable of detecting erroneous sentences. To detect these errors, audio recordings of erroneous sentences were created by reading these sentences aloud. Using CHAI to process the recordings resulted in binary classifications: True Positive, where CHAI recognizes the error, and False Negative (FN), where CHAI does not recognize the error. The erroneous sentences were collected from James' work on English language errors and Politzer's research on German language errors [11]. According to James [11], English errors can be categorized into five categories: Addiction, Omission, Misformation, Misordering, and Blends. According to Politzer [20], German errors can be categorized into five categories: Case Endings, Gender Confusion, Verb Morphology, Vocabulary, and Word Order. We added pronunciation as the sixth category. In total, 240 erroneous sentences were created for each language, with 20 sentences for each error category. These sentences were fed into CHAI, and each response from CHAI was classified as either a True Positives (TP) or or False Negatives (FN). An important point is that an error was classified as a TP as long as CHAI was able to detect an error in it. For example, if a sentence from error category X was detected as an error in category Y, it still counted as a TP. No distinction was made between detecting the correct error type and detecting an error. Sensitivity was calculated using Equation (2). This metric evaluates the proportion of correctly detected errors within each error category. This provides insights into CHAI's error detection accuracy across different types of language mistakes.

$$\textbf{Sensitivity} = \frac{TP}{TP+FN} = \frac{\text{Correctly recognized sentences of category X}}{\text{Total sentences in category X}} \quad (2)$$

**Reproducibility:** Due to the non-deterministic nature of LLMs, it is important to assess the reliability of the sensitivity score when processing the same sentence multiple times. The reproducibility of the sensitivity score was measured using the Fleiss Kappa [5]. This is used to measure inter-rater reliability, which requires two conditions to be met: (i) agreement between two or more raters, and (ii) the measured variable must be nominal. First, the comparison involves multiple runs of CHAI acting as "raters" by identifying errors in sentences. Second, the measured variable, whether an error is detected or not, is nominal, since it represents a binary classification (TP or FN). Thus, Fleiss Kappa is an appropriate method to assess the reliability of CHAI's error detection across multiple runs.

To measure reproducibility, 20 sentences from each error category were fed into CHAI 10 times independently. This produced a total of 1200 sentence evaluations for both German and English. The output was captured and analyzed to determine whether it was a TP or FN. After collecting the data, the reproducibility was calculated for each error category. In total, 12 Fleiss Kappa values were computed and evaluated, 6 for each error category in both

[3] https://developers.meta.com/horizon/documentation/unity/audio-ovrlipsync-unity/

[4] https://www.mixamo.com/

[5] https://skybox.blockadelabs.com/

[6] https://huggingface.co/

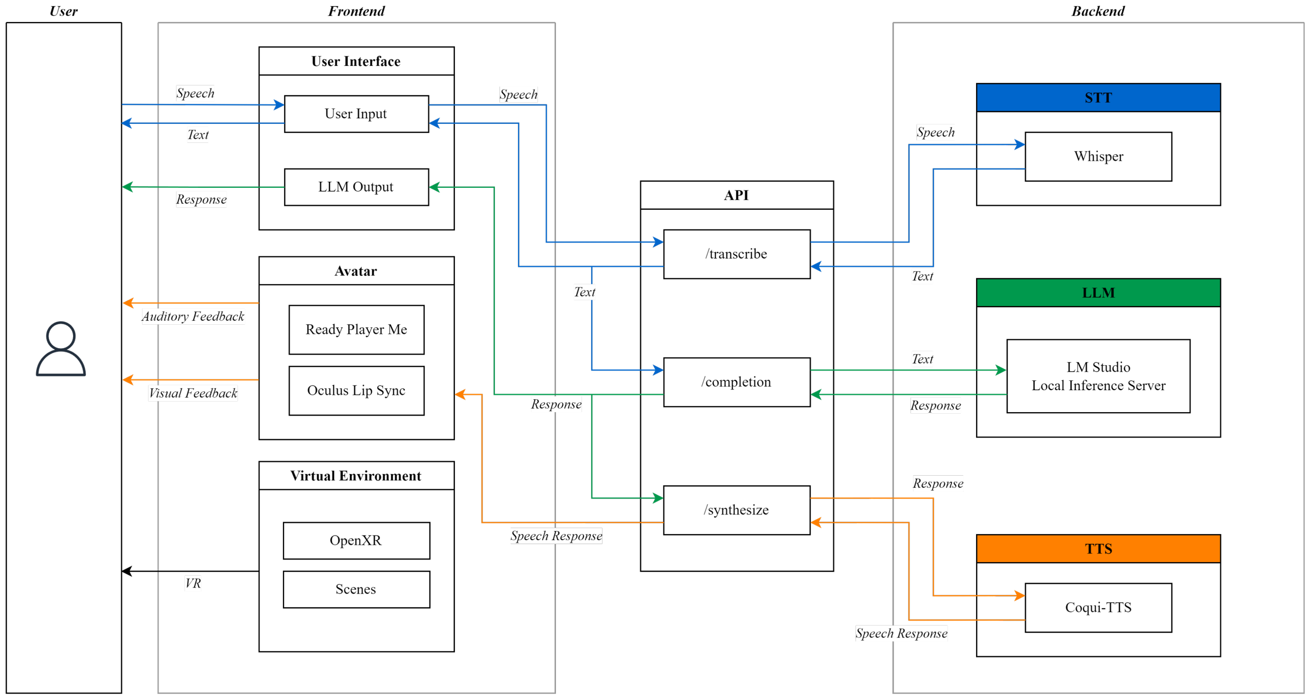


Figure 3: System Architecture of CHAI.

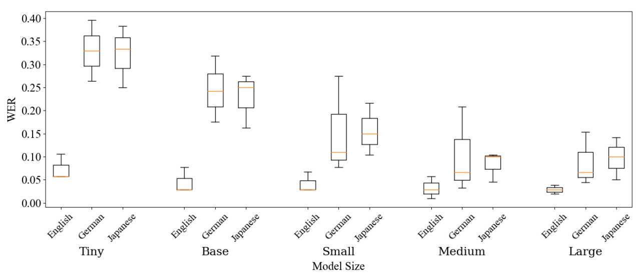


Figure 4: WER boxplots for five model sizes (Tiny, Base, Small, Medium, Large) across English, German, and Japanese.

languages. Finally, these values were interpreted to assess the consistency of CHAI's error detection.

## 4 Results

This section shows the key results of individual components and the system: STT, LLM, and CHAI. For the LLM results, only the German and English languages are discussed in more detail.

### 4.1 STT Evaluation

The STT evaluation was conducted to evaluate which model size should be used. The five model sizes of OpenAI's STT family, Whisper, were evaluated based on the Word Error Rate (WER), Word Rate (Throughput), and language.

Fig. 4 shows a trend of decreasing mean WER and reduced variability as the model size increases. The lower the WER, the better the speech recognition. The tiny model shows the highest WER, indicating less accurate speech recognition. The base model shows a reduction in both, the mean WER and variability compared to the tiny model. The small model shows a more compact WER distribution with even a lower mean and variability. The medium and large models show further improvements in terms of mean WER. The large model performs best with the lowest WER, indicating accurate and consistent performance across all tested datasets. This correlates with existing research of Radford et al. [21], who reported lower WER values for larger models. Further, as expected, the word rate decreases with increasing size of the model,

### 4.2 LLM Evaluation

The LLM evaluation's purpose is to determine a suitable LLM for the CHAI system. The evaluation was conducted using four criteria: local deployability (C1), hardware and cost (C2), inference time (C3), and language proficiency (C4). Three LLMs, OpenAI's GPT-4o, Meta's Llama3, and Google's Gemma, were selected for comparison based on their popularity and reputation. The results for the first three criteria are given in Table 1.

Table 1: LLM evaluation results for criteria C1, C2, and C3.

| LLM | (C1) Deployability | (C2) Cost | (C3) Inference Time [ms] |
|---|---|---|---|
| GPT-4o | Cloud | API Calls | $585.65 \pm 147.39$ |
| Llama3 | Local | GPU (>8GB VRAM) | $17.52 \pm 0.62$ |
| Gemma | Local | GPU (>8GB VRAM) | $18.02 \pm 0.60$ |

#### 4.2.1 Local Deployability, Cost, and Inference Time

Local deployability assessed whether the LLM could run on a local machine without relying on cloud services. Both LLama 3 and Gemma met this requirement, as they were available for local deployment. These models are available for download in various sizes on Hugging Face and can be run locally or deployed on an local inference server using open-source tools like LM Studio or Ollama.

In contrast, GPT-4o was not locally deployable, as it required cloud-based access via OpenAI's API.

We define inference time as the time to generate the first token. Table 1 shows that both Llama 3 and Gemma were approximately 33 times faster in inference time compared to GPT-4o. This significant difference is due to the fact that Llama 3 and Gemma were locally deployable, while GPT-4o relied on cloud-based servers. It is important to note that inference times may vary depending on the hardware used for testing, as different machines with varying technical specifications can produce different results. Furthermore, the response time of CHAI comprises at least the STT, LLM, and TTS components, making inference time a critical factor, particularly in conversational language training applications.

### 4.2.2 Language proficiency

As a crucial criteria, the proficiency of the LLM in different languages was evaluated. Each model was tested using standardized language proficiency placement exams based on the CEFR for German and English.

For **German**, the results show that GPT-4 consistently outperformed the other models at every test level, achieving 90% on both the A1 and A2 tests and 88% on the B1 test. Llama3 performed moderately, scoring 60% at the A1 level, 66% at A2, and 54% at B1. Gemma performed the weakest overall, with its highest score of 58% at A1 and a decline to 36% at A2 and 30% at B1. These results indicate that GPT-4o had the best language proficiency across all levels, while Llama3's performance was more consistent than Gemma's. GPT-4o was the most effective LLM for German language tests. GPT-4o showed high accuracy across all test levels. Llama3 showed sufficient proficiency (over 50%) for all test levels. Gemma performed poorly, particularly on the A2 and B1 tests, indicating that it's not suitable for more advanced language questions. Based on these results, GPT-4o is the best choice for tasks requiring high language proficiency in German. Llama3 could be an alternative for simpler tasks.

For **English**, we drew results in Fig. 5. GPT-4o consistently outperformed the other models, achieving a perfect score on the A1-1 test and near perfect scores on most other levels, indicating that GPT-4o has high accuracy across all test levels and is able to handle both simple and more complex language tasks. Llama3 showed fluctuating results, with high scores in A1-1, but a sudden dip in A2-2.1, indicating instability. Gemma showed a gradual improvement from A1-1 to A2-1.1, but its performance suddenly dropped to 10% in A2-2.1. Gemma also struggled to maintain higher scores in more advanced tests, indicating it is not capable of handling more complex tasks. GPT-4o demonstrated the best performance over all the English test, followed closely by Llama3. Gemma, however, showed significantly worse performance overall, especially on the A2-2.1 test, where it was only able to achieve a 10% score. The sudden dip in Gemma's performance in A2-2.1 where it scored only 10%, could have several reasons. One reason for its poor performance was the difficulty in receiving a complete and accurate response from Gemma when prompted with a multiple-choice question. Often Gemma struggled to answer with a viable option, would choose the first option available or give incomplete answers to a task. Despite adjustments to the LLM configurations (temperature and context window length) Gemma frequently failed to respond in a syntactically correct manner. Another reason is that the test for A2-2.1 included tasks that were also longer than the average tasks across all the tests. Gemma may have had difficulty maintaining context over these longer tasks. It is likely, that for the same reason, Llama3's performance dipped for the test level A2-2.1.

Despite GPT-4o's better proficiency, Llama3 was selected due to its local deployability and significant faster inference times. Llama3's balance between speed and satisfying language performance made it the more practical choice for real-time, locally de-

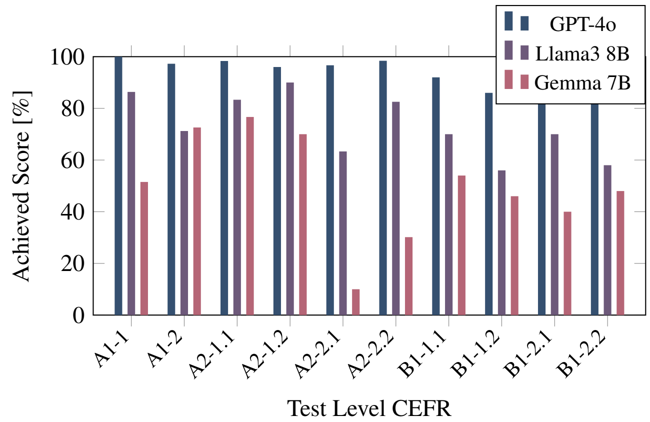


Figure 5: English test results for Llama3, Gemma, and GPT-4o across CEFR test levels A1 to B1.

ployed applications. In this case, it was the better option for CHAI.

## 4.3 Sensitivity Analysis

**English Sensitivity**: CHAI was able to correctly identify errors in the addition category with a 57% success rate. The few-shot technique consistently outperformed the zero-shot technique across all six categories. The highest improvement was observed in the *Blends* category, where few-shot prompting exhibited a sensitivity score of 93.81%, compared to 60.45% in the zero-shot method. In the *Addition* category, there was an increase of 33% (from 57.00% to 90.00%). The *Omission* category exhibited an improvement of 25.18% (from 66.32% to 91.50%), while *Misformation* improved by 8.50% (from 67.50% to 76.00%). For *Misordering*, there was an increase of 26.63% (from 66.00% to 92.63%). The *Blends* category exhibited the largest improvement, with a 33.36% increase (from 60.45% to 93.81%). On the other hand, the Pronunciation category showed the lowest performance in both methods, with scores of 38.00% for zero-shot and only a slight increase to 45.00% for few-shot.

**German Sensitivity**: Similarly to the English sensitivity results, the few-shot technique consistently outperformed the zero-shot technique across all six categories. In the Case *Endings* category, there was an increase of 76.84% (from 11.66% to 88.50%). The *Gender Confusion* category exhibited an improvement of 85.48% (from 12.94% to 98.42%), while *Verb Morphology* improved by 77.29% (from 18.00% to 95.29%). The *Vocabulary* category exhibited the largest improvement, with a 89.27% increase (from 6.84% to 96.11%). For *Word Order*, there was an increase of 91.47% (from 3.53% to 95.00%). Lastly, *Pronunciation* showed the lowest improvement out of all categories, a 72.52% increase (from 21.05% to 93.57%).

**Reproducibility**: The Fleiss Kappa values were calculated to assess the reproducibility of CHAI on identifying errors in the six error categories for English (see Table 2).

The Kappa values provide insight in the consistency and reproducibility of CHAI's error detection. Increased Kappa values for Addition, Blends, and Misordering indicate that few-shot prompting improves reproducibility across different runs in these categories. The few-shot examples provide more context to CHAI and how it should respond, resulting in more consistent error identification. However, the Misformation and Pronunciation categories exhibit lower Kappa values in few-shot, indicating that the additional examples may introduce ambiguity, especially in cases like misformed verbs or mispronunciations. Pronunciation errors, in particular, rely on the performance of the STT component, which explains the decreased consistency in few-shot prompting as provid-

Table 2: English reproducibility scores for the six english error categories for both zero-shot and few-shot prompting techniques. A higher value indicates a better result.

| Error Category | Zero shot | Few shot |
|---|---|---|
| Addiction | 0.660 | 0.802 |
| Omission | 0.651 | 0.650 |
| Misformation | 0.739 | 0.640 |
| Misordering | 0.608 | 0.628 |
| Blends | 0.748 | 0.772 |
| Pronunciation | 0.693 | 0.470 |

ing examples can not change how the STT component transcribes user inputs. These Kappa values show that few-shot prompting generally improves reliability. However, it can lead to decreased reliability in categories that are dependent on speech recognition accuracy and introduces ambiguity for misformation errors.

The reproducibility results for German are less consistent ($< 0.4$ in most categories), even when CHAI is provided with few-shot prompt. While few-shot prompting generally improves the reproducibility of CHAI's error detection for most error categories, it also introduces challenges in maintaining consistency for certain types of errors, particularly in Case Endings. With few-shot prompting, CHAI achieves fair reproducibility in all but one category. In contrast, zero-shot prompting exhibits slight reproducibility with exceptions in Case Endings and Word Order.

## 5 Limitations

There are several limitations to this work that should be noted: The evaluation focused on sensitivity without considering specificity. While sensitivity evaluates how well CHAI detects errors, specificity would assess its ability to avoid false positives. The accuracy of pronunciation error detection relies heavily on the performance of the STT component. Any transcription errors, such as misrecognitions, affect CHAI's ability to identify pronunciation mistakes. This introduces a significant limitation, especially in cases of strong accents or unclear speech. The sensitivity analysis is limited to English and German due to a lack of proficiency in other languages. While CHAI has the possibility to be adapted for other languages, its effectiveness in these cases were left unevaluated. CHAI's performance and error detection capabilities may differ for languages with different grammatical rules. The use of a fine-tuned language-specific LLM would improve test results for language learning.

## 6 Conclusion

The ability to speak multiple languages in today's modern world is crucial, yet there is a gap in combining VR, AI chatbot and language learning, in particular to improve speaking skills. This paper created and evaluated a functional VR-enabled, LLM-based conversation trainer. The results indicate the potential of VR-LLM systems for language learning and provide a basis for future work.

Several future works can be considered: Implementing RAG techniques could improve the accuracy of responses by information retrieval from language learning resources as well as scene knowledge. This would allow the system to base its answers on reliable references and context. This should also be combined with the fine-tuning of the LLM. We also plan to conduct a user study to evaluate both short-term and long-term learning outcomes in improving language skills. Moreover, user preferences and perceptions would provide insightful feedback to the design of the system.

## References


[1] M. Alfadil. Effectiveness of virtual reality game in foreign language vocabulary acquisition. *Computers & Education*, 153:103893, 2020. 1

[2] G. Caldarini, S. Jaf, and K. McGarry. A literature survey of recent advances in chatbots. *Information*, 13(1):41, 2022. 1

[3] T. K. Dhimolea, R. Kaplan-Rakowski, and L. Lin. A systematic review of research on high-immersion virtual reality for language learning. *TechTrends*, 66(5):810–824, 2022. 1

[4] G. Eren and The Coqui TTS Team. Coqui tts. https://github.com/coqui-ai/TTS, Jan. 2022. (last accessed 18.10.2024). doi: 10.5281/zenodo.6334862 2

[5] J. L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971. 3

[6] S. Garcia, R. Kauer, D. Laesker, J. Nguyen, and M. Andujar. A virtual reality experience for learning languages. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–4, 2019. 1

[7] Google. Google's ai model, 2024 (accessed May 16, 2024). 1

[8] N. Haristiani. Artificial intelligence (ai) chatbot as language learning medium: An inquiry. In *Journal of Physics: Conference Series*, vol. 1387, p. 012020. IOP Publishing, 2019. 2

[9] J. Hartfill, J. Gabel, D. Neves-Coelho, D. Vogel, F. Räthel, S. Tiede, O. Ariza, and F. Steinicke. Word saber: an effective and fun vr vocabulary learning game. In *Proceedings of Mensch und Computer 2020*, pp. 145–154, 2020. 1

[10] C. Hua and J. Wang. Virtual reality-assisted language learning: A follow-up review (2018–2022). *Frontiers in Psychology*, 14:1153642, 2023. 1

[11] C. James. *Errors in language learning and use: Exploring error analysis*. Routledge, 2013. doi: 10.4324/9781315842912 3

[12] K. Johnson. *An introduction to foreign language learning and teaching*. Routledge, 2017. 1

[13] R. Kaplan-Rakowski and A. Gruber. The impact of high-immersion virtual reality on foreign language anxiety. *Smart Learning Environments*, 10(1):46, 2023. 1

[14] F. Karataş, F. Y. Abedi, F. Ozek Gunyel, D. Karadeniz, and Y. Kuzgun. Incorporating ai in foreign language education: An investigation into chatgpt's effect on foreign language learners. *Education and Information Technologies*, pp. 1–24, 2024. 1

[15] L. Kohnke, B. L. Moorhouse, and D. Zou. Chatgpt for language teaching and learning. *Relc Journal*, 54(2):537–550, 2023. 1

[16] X. Li, B. Li, and S.-J. Cho. Empowering chinese language learners from low-income families to improve their chinese writing with chatgpt's assistance afterschool. *Languages*, 8(4):238, 2023. 1

[17] Mistral. Mistral's ai model, 2023 (accessed May 16, 2024). 1

[18] J. M. Moreno-Reyes, D. J. AlejosS, D. Siefker, and R. Martínez-Meza. 360 virtual tour and escape room design as a video games-based learning process for diagnosis and strengthening of the english language. *Journal Computer Technology*, 6(16), 2022. 1

[19] OpenAI. Introducing chatgpt, 2024 (accessed May 16, 2024). 1

[20] R. L. Politzer. Errors of english speakers of german as perceived and evaluated by german natives. *Modern Language Journal*, pp. 253–261, 1978. 3

[21] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. Mcleavey, and I. Sutskever. Robust speech recognition via large-scale weak supervision. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, eds., *Proceedings of the 40th International Conference on Machine Learning*, vol. 202 of *Proceedings of Machine Learning Research*, pp. 28492–28518. PMLR, 23–29 Jul 2023. 4

[22] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 1

[23] J. C. Young and M. Shishido. 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), 2023. 1

[24] Z. Zhang and X. Huang. The impact of chatbots based on large language models on second language vocabulary acquisition. *Heliyon*, 10(3), 2024. 1