

User Manual: Context-Aware Conversation Support with Augmented Reality and Generative AI

Team Members:

- Sabrina Alvarado
- Keren Rivera
- Jonathan Martinez
- Peace Passos

Mentor: Dr. Chen Chen, FIU KFSCIS, Assistant Professor

Instructor: Seyedmasoud Sadjadi

Introduction

This system is an Augmented Reality (AR) conversational assistant designed to support real-time, context-aware communication. It helps user overcome language and background-knowledge gaps by providing live translation, concept clarification, and visual AR overlays during conversations.

The system runs on a Meta Quest 2 headset and integrates speech recognition, AI-powered translation and explanation, text-to-speech, and AR visualization.

System Requirements

- Meta Quest 2 headset
- Stable Wi-Fi connection
- Backend server running
- Headset application installed and configured with backend URL

Getting Started

This system does not provide a packaged application. Instead, users run the system by cloning the project repository and executing the backend services and Unity project manually.

Step 1: Obtain the Source Code

1. Clone the project repository from the provided GitHub link
2. Ensure both the backend server code and Unity project files are available locally

Step 2: Set Up the Backend Server

1. Navigate to the backend directory
2. Create and activate a Python virtual environment
3. Install dependencies using the provided requirements file
4. Set the Gemini API key as an environment variable
5. Start the FastAPI server

Step 3: Set Up the Unity Project

1. Open the Unity project using the recommended Unity version
2. Import the Meta XR SDK if not already included
3. Configure the backend server URL in the Unity settings
4. Build and deploy the project to the Meta Quest 2 headset using the Android build pipeline

Core Features & How to Use Them

a. Real-Time Translation

1. Speak naturally in your preferred language (English or Spanish)
2. The system automatically detects the spoken language
3. Translated text appears as an AR overlay in your field of view
4. The translated output is also played back using audio (TTS)

b. Concept Clarification

1. During conversation, speak or phrase or sentence containing an unfamiliar term or concept
2. The system sends the input to the AI for clarification
3. A short definition or summary appears as an AR overlay

c. AR Visual Overlays

- Translations and explanations appear as floating text overlays
- Overlays are anchored in the Passthrough view and remain stable as stable as you move
- Text is designed for readability without blocking in the real-world view

d. Audio Output

- Translated or clarified content is spoken aloud using TTS
- Audio feedback helps maintain conversational flow
- Visual-only interaction is also supported if audio is muted

Interaction Flow

1. User speaks
2. Speech is converted to text (STT)
3. Text is processed by the backend
4. Gemini AI generates translation or explanation
5. Output is returned to the headset
6. AR text overlay + optional audio playback

Best practices

- Use a quiet environment when possible
- Avoid speaking over others simultaneously
- Keep sentence short for faster responses
- Ensure strong Wi-Fi connectivity for minimal delay

Known Limitations

- Slang or idiomatic expressions may be less accurate
- Performance may degrade in noisy environments
- Requires continuous internet connection
- Currently supports English and Spanish only

Troubleshooting

No AR overlay appears:

- Check backend server connection
- Ensure headset has Wi-Fi access

Poor translation quality:

- Speak more clearly
- Reduce background noise

Audio not playing:

- Verify headset volume
-

Shutting Down the System

1. Exit the application on the Meta Quest 2
2. Safely stop the backend server if running locally