HR Interview Agent

An automated interview agent that conducts technical interviews for software engineering positions using LangChain, Ollama, Whisper.cpp, and virtual audio routing.

Overview

This system creates an AI interviewer that can:

- Join Zoom meetings
- Speak using text-to-speech
- Listen to and record candidate responses
- Transcribe speech using Whisper.cpp
- Generate contextual interview questions
- Evaluate interview progress

Prerequisites

- Python 3.8+
- macOS (for BlackHole audio routing - Windows/Linux users will need alternatives)
- Ollama running locally
- LLM model (llama3.1:8b-instruct-q5_K_M recommended)
- FFmpeg

Installation Guide

1. Clone the Repository

```bash
git clone https://github.com/yourusername/hr-interview-agent.git
cd hr-interview-agent
```

2. Set Up Python Environment

bash

Copy

```bash
# Create and activate a virtual environment
python -m venv venv
source venv/bin/activate  # On Windows: venv\Scripts\activate

# Install required packages
pip install sounddevice numpy pyttsx3 requests python-zoomus langchain-ollama langchai
```

3. Install Whisper.cpp

Whisper.cpp is used for local speech recognition without relying on cloud services.

bash

Copy

```bash
# Clone the Whisper.cpp repository
git clone https://github.com/ggerganov/whisper.cpp.git
cd whisper.cpp

# Build the project
make

# Download the base English model
bash ./models/download-ggml-model.sh base.en

# Return to project root
cd ..
```

4. Install BlackHole (Virtual Audio Cable for macOS)

BlackHole creates virtual audio devices that allow routing audio between applications.

bash

Copy

```bash
# Install via Homebrew
brew install blackhole-2ch

# OR download directly from GitHub:
# https://github.com/ExistentialAudio/BlackHole/releases
```

After installation:

1. Open "Audio MIDI Setup" (Applications > Utilities)
2. Click the "+" button in the bottom left corner
3. Choose "Create Multi-Output Device"
4. Select both your output device (e.g., Built-in Output) and "BlackHole 2ch"
5. Optional: Create an Aggregate Device if needed for more complex routing

5. Configure Audio Devices

For Zoom Meetings:

1. Open Zoom
2. Go to Settings > Audio
3. Set Speaker to "BlackHole 2ch" (so the AI can hear Zoom audio)
4. Set Microphone to your standard microphone or another virtual output that will receive the AI's speech

For System Audio:

1. Open System Preferences > Sound
2. Set Output to your Multi-Output Device (created in step 4)
3. Set Input to "BlackHole 2ch" if you want to capture system audio

6. Configure the Application

1. Open `index.py` in your editor
2. Update the following constants:
   - `WHISPER_CPP_EXECUTABLE`: Path to the Whisper.cpp CLI executable
   - `WHISPER_MODEL_PATH`: Path to your downloaded Whisper model
   - `OLLAMA_BASE_URL`: URL for your Ollama instance (default: "http://localhost:11434")
   - `OLLAMA_MODEL`: Your preferred Ollama model (default: "llama3.1:8b-instruct-q5_K_M")
3. Configure audio device indices:
   - Run the following in Python to identify your devices:

```python
import sounddevice as sd
print(sd.query_devices())
```

- Update the `input_device` value in the `record_with_vad` function
- Set appropriate device for pyttsx3 in the `speak_reply` function

Running the Application

bash

Copy

```bash
# Ensure Ollama is running
ollama run llama3.1:8b-instruct-q5_K_M

# In a new terminal
python index.py
```

Audio Routing Configuration

For the Interviewer to Speak in Zoom:

1. The application uses pyttsx3 to convert text to speech
2. The speech is routed to the default output device
3. If your Multi-Output Device is set as default, the audio goes to both your speakers and BlackHole
4. Zoom picks up the audio via BlackHole as its input

For the Interviewer to Hear Responses:

1. Set the `input_device` in `record_with_vad` to the BlackHole output index
2. The application now captures audio coming from Zoom via BlackHole
3. This audio is processed through VAD and sent to Whisper for transcription

Device Index Management

Finding the correct device indices can be challenging. Here's how to manage them:

1. **List all devices**:

   python

   Copy

   ```python
   import sounddevice as sd
   devices = sd.query_devices()
   for i, device in enumerate(devices):
       print(f"Device {i}: {device['name']} (in: {device['max_input_channels']}, out:
   ```

2. **Test specific input device**:

```python
import sounddevice as sd
import numpy as np
import time

device_idx = 1  # Replace with index to test
duration = 5  # seconds
fs = 16000  # sample rate
channels = 1  # mono

def audio_callback(indata, frames, time, status):
    rms = np.sqrt(np.mean(np.square(indata)))
    print(f"Audio level: {rms:.5f}")

with sd.InputStream(device=device_idx, channels=channels, samplerate=fs, callback=a
    print(f"Listening on device {device_idx}...")
    time.sleep(duration)
```

3. **Update the appropriate values in the code**:
    - For recording, update the `input_device` value in the `record_with_vad` function
    - For speech output, update the pyttsx3 device settings in the `speak_reply` function

Troubleshooting

Audio Issues:

- **No audio being recorded**:
    - Check that the correct input device is selected
    - Verify that audio is actually flowing to BlackHole (use Audio MIDI Setup to monitor levels)
    - Try increasing the sensitivity by lowering `SILENCE_THRESHOLD`
- **AI speech not being heard in Zoom**:
    - Make sure Zoom is using BlackHole as its microphone input
    - Check that system audio is routed to BlackHole
    - Try running a test with system sounds to confirm BlackHole is capturing output

Whisper Transcription Issues:

- **Transcription errors**:
  - Make sure the path to the Whisper executable and model is correct
  - Check that the recordings folder exists and is writable
  - Try a different Whisper model (tiny.en for speed, medium.en for accuracy)

LangChain/Ollama Issues:

- **LLM not responding**:
  - Verify Ollama is running (`ollama ps`)
  - Check that the model is downloaded (`ollama list`)
  - Ensure the base URL is correct for your Ollama setup

Customization

Job Description

Edit the `JOB_DESCRIPTION` constant to modify the position requirements.

Interview Flow

Modify the initial_task in the main block to change the interview structure.

LLM Model

If you have a more powerful machine, try larger models for better results:

```
OLLAMA_MODEL = "llama3.1:70b-instruct-q5_K_M"  # Larger model for better performance
```

Advanced Setup

Running on Windows

For Windows users, consider using:

- VB-CABLE instead of BlackHole
- Configure Windows sound settings accordingly

Running on Linux

For Linux users, consider using:

- PulseAudio or PipeWire virtual sinks
- JACK for advanced audio routing

License

[Your License Here]