SonicMind Installation Documentation

SonicMind Team

This guide provides step-by-step instructions to set up the SonicMind application for recording, transcribing, and analyzing Zoom meetings on Windows.

Contents

1 Application Functionality

The SonicMind application is a Python-based tool designed to record, transcribe, and analyze audio from Zoom meetings. Below are its core functions:

- **Audio Recording**: Captures audio from a microphone, system audio (via VB-Audio Virtual Cable), or both simultaneously. Recordings are saved as WAV files in the `~/SonicMindRecordings` directory.
- **Real-Time Audio Processing**: Monitors audio levels during recording to ensure sufficient signal strength, with dynamic sample rate adjustment for compatibility with selected devices.
- **Audio Transcription**: Uses OpenAI's Whisper model (base by default) to transcribe recorded audio into text. Supports resampling to 16 kHz and stereo-to-mono conversion for optimal transcription.
- **AI-Powered Analysis**: Processes transcripts using OpenAI's ChatGPT (GPT-4o model) to provide insights, summaries, and answers to questions. Automatically detects whether the transcript is a direct question or a meeting conversation for context-appropriate analysis.
- **Gradio Interface**: Provides a user-friendly web interface for configuring audio devices, starting/stopping recordings, viewing transcripts, and displaying AI-generated analyses with formatted HTML output.
- **Output Management**: Saves transcripts and analyses to text files with timestamps, including metadata like recording mode and date. Supports clipboard copying of transcripts.
- **Device Configuration**: Allows selection of microphone and system audio devices, with validation and debugging tools to ensure proper audio setup.

2 Installation Steps

2.1 Install Python

1. Download and install Python 3.8 or higher from https://www.python.org/downloads/.
2. Verify installation by running:

```
python --version
```

3. Ensure `pip` is installed:

```
pip --version
```

2.2 Set Up a Virtual Environment

1. Create a virtual environment:

```
python -m venv sonicmind_env
```

2. Activate the virtual environment:

```
sonicmind_env\Scripts\activate
```

2.3 Install Dependencies

1. Install required Python packages, including FFmpeg, using `pip`:

```
pip install gradio sounddevice numpy soundfile openai python-dotenv
    pandas whisper librosa scipy ffmpeg-python
```

Notes:

- `ffmpeg-python` provides FFmpeg integration for Whisper audio processing.
- `librosa` and `scipy` are optional but recommended for audio resampling.
- If issues occur with `whisper`, install it separately:

```
pip install git+https://github.com/openai/whisper.git
```

2. Verify installations:

```
pip list
```

Check for packages like `gradio`, `sounddevice`, `numpy`, `soundfile`, `openai`, `python-dotenv`, `pandas`, `wh Whisper`, `librosa`, `scipy`, and `ffmpeg-python`.

2.4 Set Up OpenAI API Key

1. Sign up for an OpenAI account at https://openai.com and obtain an API key.
2. Create a `.env` file in the project directory with:

```
OPENAI_API_KEY=your_api_key_here
```

3. Ensure the `.env` file is in the same directory as your Python script.

2.5 Install and Configure VB-Audio Virtual Cable

1. Download and install VB-Audio Virtual Cable from https://vb-audio.com/Cable/.
2. Restart your computer to ensure the virtual cable is properly initialized.
3. Configure audio settings in Windows:
   - Open **Control Panel** > **Sound**.
   - In the **Playback** tab:
     - Select **CABLE Input (VB-Audio Virtual Cable)** and set it as the **Default Device**.
     - Select your physical microphone and set it as the **Default Communication Device**.
   - In the **Recording** tab:
     - Select **CABLE Output (VB-Audio Virtual Cable)** and set it as the **Default Device**.
4. Configure Zoom audio settings:
   - Open Zoom and go to **Settings** > **Audio**.
   - Set **Microphone** to your physical microphone.

- Set **Speaker** to **CABLE Output (VB-Audio Virtual Cable)**.

2.6 Run the Application

1. Save the Python code as `sonicmind.py` in your project directory.
2. Ensure the `.env` file is correctly set up.
3. Run the script:

```
python sonicmind.py
```

4. The Gradio interface will launch at http://127.0.0.1:7860. Open this URL in a web browser.
5. In the Gradio interface:
   - Go to the **Setup** tab.
   - In the **Microphone Device ID** field, enter the ID of your physical microphone (as shown in the device table).
   - In the **System Audio Device ID** field, enter the ID of **CABLE Output (VB-Audio Virtual Cable)**.
   - Set the **Recording Mode** to "both" for capturing both your microphone and Zoom's audio output.
   - Start/stop recording using the interface buttons.

3 Troubleshooting

- **No Audio Devices Detected**:
  - Run `app.debug_audio_system()` in a Python console to diagnose audio issues.
  - Ensure VB-Audio Virtual Cable is installed and configured correctly.
  - Verify microphone permissions in Windows Settings > Privacy > Microphone.
- **Whisper Transcription Fails**:
  - Ensure `ffmpeg-python` is installed correctly.
  - Check audio file accessibility and ensure sufficient disk space.
- **OpenAI API Errors**:
  - Confirm the API key is valid and has sufficient credits.
  - Check internet connectivity.
- **Gradio Interface Not Loading**:
  - Ensure port 7860 is not blocked by a firewall.
  - Try a different browser or clear the browser cache.

4 Additional Notes

- **VB-Audio Virtual Cable**: Essential for capturing Zoom's output audio. Ensure it is set as the default playback device in Windows and Zoom.
- **Whisper Model Size**: The script uses the base Whisper model. For better accuracy, modify `WHISPER_MODEL_SIZE` to `small`, `medium`, or `large` (requires more disk space and RAM).
- **Storage**: Audio recordings and transcripts are saved in `~/SonicMindRecordings` by default.
- **Permissions**: Ensure write permissions for the output directory (`~/SonicMindRecordings` or fallback to current directory).