LiveKit Voice Interruption Handling Challenge

?? Overview

This challenge tests your ability to extend a real-time conversational AI agent built on LiveKit Agents (https://github.com/livekit/agents).

Currently, LiveKit's Voice Activity Detection (VAD) logic detects when a user starts speaking and automatically pauses the agent's TTS. However, filler sounds such as "uh", "umm", "hmm", and "haan" often cause false interruptions, breaking the flow of conversation.

Your task is to enhance the agent so it intelligently distinguishes meaningful user interruptions from irrelevant fillers, ensuring seamless, natural dialogue.

?? Objective

- Ignore specific filler words or phrases (defined by a configurable list) only when the agent is currently speaking.
- Register those same words as valid user speech when the agent is quiet. - Maintain real-time responsiveness — if genuine user speech (like "wait" or "stop") occurs, the agent must stop immediately.
- Ensure no changes to LiveKit's base VAD algorithm — all handling should be done as an extension layer.
- Design the system so it's scalable and language-agnostic, capable of ignoring words dynamically configured via environment or runtime parameters.

?? Example Scenarios

- Examples of how your logic should behave in real-world conditions: - User filler while agent speaks → "uh", "hmm", "umm" | Agent Speaking: Yes | Expected: Agent ignores input and continues speaking
- User real interruption → "wait one second", "no not that one" | Agent Speaking: Yes | Expected: Agent immediately stops

- User filler while agent quiet → “umm” | Agent Speaking: No | Expected: System registers speech event
- Mixed filler and command → “umm okay stop” | Agent Speaking: Yes | Expected: Agent stops (contains valid command)

- Background murmur → “hmm yeah” (low confidence ASR) | Agent Speaking: Yes | Expected: Ignored if under confidence threshold

⚙️ Technical Expectations

- Integrate into the LiveKit agent event loop without modifying core SDK code. - Expose a configurable parameter, e.g.: ignored_words = ['uh', 'umm', 'hmm', 'haan'] - Use transcription events or ASR results to filter out filler-only segments.
- Maintain async/thread-safe handling with LiveKit callbacks.
- Log ignored and valid interruptions separately for debugging.
- Handle dynamic updates to the ignored list (optional bonus).

?? Evaluation Criteria

- Correctness (30%): Accurately distinguishes filler interruptions vs. real ones - Robustness (20%): Works under rapid speech, background noise, and fast turn-taking - Real-time Performance (20%): No added lag or VAD degradation
- Code Quality (15%): Clean, modular, readable, and well-documented
- Testing & Validation (15%): Includes clear README, logs, and reproducible results

?? Submission Instructions

1. Branch Setup:
   Create a new branch from your fork of the LiveKit Agents repo:

   feature/livekit-interrupt-handler-<yourname>

2. Implementation Requirements:
3. Commit all your code changes related to this feature and ensure the agent runs end

to-end.

4. README.md in Your Branch Must Include:

- What Changed: Overview of new modules, params, and logic added.
- What Works: Features verified through manual or automated testing. -
Known Issues: Any edge cases or instability observed.
- Steps to Test: How to start the agent and verify filler vs. real speech handling. -
Environment Details: Python version, dependencies, and config instructions.

5. Deliverables:

GitHub branch URL for review, optional short screen/audio recording demonstrating your solution.

✅ Rules

- You may use LLMs (ChatGPT, Claude, Copilot) for ideation and debugging. -
You may consult LiveKit's documentation and open-source materials. - You
must not collaborate, share code, or reuse another participant's work. -
Copy-pasting public code without attribution or understanding will lead to
disqualification.

❓❓ Bonus Challenges (Optional for Extra Credit)

- Dynamically update ignored-word lists during runtime.
- Implement multi-language filler detection (e.g., Hindi + English mix).

❓❓ End Goal

- Seamlessly continue speaking when fillers occur.
- Gracefully pause when genuine interruptions are detected.
- Feel natural in conversation — no awkward cutoffs, no overreactions.