INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD)
© 2026 IJNRD | Volume 11, Issue 5, May 2026 | ISSN: 2456-4184 | IJNRD.ORG

TURN DETECTION IN PRODUCTION VOICE AI AGENTS: A SURVEY OF APPROACHES AND OPEN CHALLENGES

[1]**Kollaikal Rupesh**
[1]*Independent Researcher*
[1]Independent AI Research, San Francisco, California, USA
kollaikalrupesh@gmail.com

Abstract : Real-time voice AI agents have moved from research demonstrations to production deployments across customer service, healthcare triage, and consumer assistants over the past three years. A core component of any such system is the turn detection module, which decides when the user has finished speaking and the agent should begin responding. This decision must be made within a few hundred milliseconds, on streaming audio, often under degraded telephony conditions. Despite its centrality to perceived conversational quality, turn detection has received less formal academic treatment than upstream tasks such as automatic speech recognition or downstream tasks such as text-to-speech synthesis. This paper surveys the approaches currently deployed in production voice AI frameworks, including voice activity detection (VAD)-based endpointing, semantic turn detection using language models, and hybrid systems. We examine how the dominant open-source frameworks — pipecat, livekit/agents, and pipecat-flows — implement these approaches in practice. We identify five open challenges: multilingual turn-taking variation, robust handling of background speech, partial endpointing for interruption tolerance, evaluation under realistic codec chains, and the tradeoff between detection latency and false-trigger rate. We connect these challenges to recent work on sub-second audio decisions in adjacent domains such as deepfake detection. The paper is intended to orient practitioners and researchers entering the production voice AI space and to identify research directions that would meaningfully improve deployed systems.

IndexTerms *- voice AI, turn detection, endpointing, voice activity detection, real-time systems, conversational AI, latency optimization, production AI.*

I. INTRODUCTION

Voice-based conversational AI has crossed a usability threshold between 2023 and 2026. Systems built on streaming automatic speech recognition (ASR), large language models (LLMs), and neural text-to-speech (TTS) now sustain natural-feeling conversations across telephony, browser, and mobile surfaces. Open-source frameworks such as pipecat [1] and livekit/agents [2] have lowered the engineering barrier to deploying these systems, and commercial offerings from OpenAI, Google, and others provide end-to-end real-time voice APIs. The total addressable market is substantial: industry analyses project voice AI deployment in contact centers alone to displace or augment a meaningful share of the eight-million-agent global workforce by 2028.

Within this stack, one component disproportionately determines whether a conversation feels natural or robotic: turn detection. Turn detection is the decision, made in real time on streaming audio, of when the user has finished an utterance and the agent should begin its response. Get it wrong in one direction and the agent interrupts the user mid-sentence. Get it wrong in the other direction and the conversation develops awkward, multi-second silences that destroy the sense of dialogue. Production systems target an end-of-turn decision latency on the order of 200 to 500 milliseconds — short enough that the response feels prompt, long enough to avoid clipping the user's last word.

Despite this centrality, turn detection has received less formal academic treatment than the surrounding components. Streaming ASR has a mature literature spanning two decades. Neural TTS has been the subject of intense research investment. Turn detection sits awkwardly between signal processing, dialogue systems, and human-computer interaction, and its treatment in the literature has been correspondingly fragmented. The most useful documentation on contemporary turn-detection practice exists in the source code, blog posts, and design discussions of open-source frameworks rather than in peer-reviewed venues.

This paper has three aims. First, to survey the approaches currently used for turn detection in production voice AI systems and to relate them to the older literature on voice activity detection (VAD) and endpointing. Second, to examine how three dominant open-source frameworks — pipecat, livekit/agents, and pipecat-flows — implement these approaches, based on review of their public source code, documentation, and design discussions. Third, to identify open challenges that would benefit from systematic research attention.

The paper is intended for two audiences. The first is researchers entering the production voice AI space who need an orientation to what is actually deployed and where the open problems lie. The second is engineering practitioners who would benefit from a structured view of the design space, including connections to adjacent problems such as the deployment-aware evaluation of voice clone detection systems [3].

The paper proceeds as follows. Section II reviews the technical foundations and historical context. Section III surveys the three families of contemporary turn detection approaches. Section IV examines production framework implementations. Section V identifies five open challenges. Section VI connects this work to the broader landscape of sub-second audio decisions. Section VII concludes.

II. BACKGROUND

2.1 Voice Activity Detection

The earliest approaches to detecting speech boundaries used voice activity detection (VAD): a binary classifier operating on short audio frames that outputs a per-frame label of speech or non-speech. Classical VAD relied on energy thresholds, zero-crossing rates, and spectral features. Modern VAD systems, notably Silero VAD [4] and WebRTC VAD [5], use small neural networks trained on diverse speech corpora and achieve frame-level accuracy sufficient for many applications.

VAD-based endpointing works as follows. The system monitors VAD output on a sliding window. When the VAD has produced non-speech labels continuously for some threshold duration — typically 500 to 1500 milliseconds — the system declares end-of-turn. This approach is simple, low-latency, and language-agnostic. It is also the dominant approach in production voice AI today.

VAD-based endpointing has two well-known failure modes. First, it cannot distinguish a meaningful pause in the middle of an utterance from a genuine end-of-turn. A user saying "I'd like to book a flight to... uh... let me think... Tokyo" will trigger endpoint detection during the hesitation pauses. Second, it triggers on any non-speech-to-speech transition, meaning that background noises, coughs, or environmental sounds can artificially extend the wait. Production systems mitigate these issues with hand-tuned silence thresholds, but the underlying brittleness remains.

2.2 Endpointing in ASR Systems

A second line of work emerged from the streaming ASR community. Rather than make the endpoint decision from raw audio, these systems use the ASR decoder's own state to estimate end-of-utterance probability. Variants include attention-based endpointing, joint endpointing-and-recognition models, and the use of ASR token confidence as an end-of-turn signal [6].

The advantage of ASR-coupled endpointing is access to linguistic information. The ASR decoder knows whether the user has produced a complete syntactic unit, which is a stronger signal than raw silence duration. The disadvantage is tight coupling to a specific ASR system, which limits portability across the modular pipelines common in production voice AI.

2.3 Semantic and LLM-based Turn Detection

A third approach, which has gained significant traction in 2024 and 2025, uses a language model to evaluate whether the partial ASR transcript represents a complete turn. The intuition is straightforward: humans do not detect turn endings purely from acoustic cues; they use semantic and pragmatic context. A small LLM evaluating the partial transcript can capture this context.

In production systems, this typically appears as a two-stage pipeline. Stage one is a fast VAD-based candidate detection, which proposes possible end-of-turn moments. Stage two is a semantic confirmation, in which a small model classifies the partial transcript as "turn complete" or "turn incomplete." The combined system retains the latency advantage of VAD while reducing false endpoints on hesitation pauses.

Recent work has produced purpose-built models for this task. The smol-turn-detector family of models [7], and similar lightweight classifiers, target the second stage with sub-100ms inference. These models are typically distilled from larger LLMs and operate on the recent transcript context.

III. CONTEMPORARY APPROACHES: A TAXONOMY

This section organizes contemporary turn detection into three families and identifies the design choices each entails.

3.1 Pure VAD-Based Endpointing

The simplest production-grade approach uses VAD output and a configurable silence threshold. The mechanism is: for each incoming audio frame, run the VAD model. If the frame is classified as speech, update the last-speech timestamp. If the frame is classified as silence, compute the elapsed silence duration. If that duration exceeds the configured end-of-turn threshold, emit an end-of-turn signal.

The critical design parameter is the end-of-turn silence threshold, often called the "silence threshold" or "endpoint timeout." Production values typically range from 500 ms (aggressive, fast-feeling but error-prone) to 1500 ms (conservative, slower but more accurate). The optimal value depends on the application: a triage agent handling distressed callers wants a longer threshold to avoid interrupting a hesitant speaker, while a fast-paced ordering agent wants a shorter one.

Sub-variants of this approach include adaptive thresholds (which adjust based on the speaker's recent speaking rate), per-channel thresholds (different values for inbound and outbound legs of a call), and threshold annealing (longer threshold early in the conversation, shorter as the user establishes a speaking rhythm).

3.2 Acoustic-Semantic Hybrid

The hybrid approach combines a fast acoustic stage with a semantic confirmation stage. The acoustic stage proposes candidate end-of-turn moments based on VAD. When a candidate fires, the semantic stage classifies the recent ASR transcript as a complete or incomplete turn. If the semantic stage rejects the candidate, the system continues listening with a reset or extended silence threshold.

The design choices here include the size and training data of the semantic model, the amount of transcript context provided (typically the last 5 to 20 seconds), the latency budget for the semantic call (typically 50 to 200 ms), and the policy for handling semantic rejections (extend the threshold by a fixed amount, reset it entirely, or transition to a soft listening state).

Hybrid systems materially improve handling of hesitation pauses, mid-utterance breath, and trailing filler words. They introduce additional complexity and a second potential point of failure: a semantic model that consistently rejects valid endpoints produces conversations that feel laggy.

3.3 Predictive Turn-Taking

A third family, less widely deployed in production but actively researched, attempts to predict turn endings before they occur. The motivation is human conversational behavior: humans anticipate turn ends based on syntactic, prosodic, and semantic cues, allowing them to begin their response with minimal gap. A predictive system might begin generating the agent's response before the user has finished speaking, allowing the first TTS audio to begin playing within a hundred milliseconds of true end-of-turn.

This approach is technically attractive but operationally risky. A wrongly predicted turn end means the agent begins responding to an incomplete utterance, which is conversationally severe. Production deployments of predictive turn-taking remain rare.

IV. PRODUCTION FRAMEWORK IMPLEMENTATIONS

This section examines how three open-source voice AI frameworks implement turn detection. These frameworks are widely deployed in production: pipecat alone has over ten thousand stars on GitHub at the time of writing and is used in commercial voice AI products across multiple companies.

4.1 pipecat

The pipecat framework, developed initially by Daily.co and now maintained as an independent open-source project, organizes voice AI applications as pipelines of frame processors. Turn detection in pipecat is a configurable component of the pipeline, with multiple available implementations.

The default implementation uses Silero VAD with a configurable silence threshold. The framework exposes parameters for the VAD confidence threshold, the silence duration before endpoint, and the minimum speech duration before VAD output is trusted. These parameters are deliberately exposed as first-class configuration rather than hidden, which reflects a design philosophy of treating turn detection as a system-level rather than purely algorithmic concern.

pipecat also supports semantic turn detection through its smart-endpointing module, which integrates lightweight turn-classification models behind a frame processor interface. When enabled, this module receives the VAD endpoint signal and the recent ASR transcript and returns a classification. The pipeline blocks the agent's response generation until both signals agree.

4.2 livekit/agents

The livekit/agents framework targets WebRTC-based deployments and emphasizes streaming integration with the LiveKit real-time infrastructure. Its turn detection follows a similar architectural pattern to pipecat but with different default choices reflecting the WebRTC context.

A notable design choice in livekit/agents is the explicit handling of interruptions. When the user begins speaking during an agent's response, the system must decide whether to treat this as a genuine interruption (in which case the agent's TTS should be stopped) or as backchannel speech (in which case the agent should continue). The current implementation uses a combination of VAD persistence and minimum duration thresholds to make this decision.

4.3 pipecat-flows

The pipecat-flows extension layer adds structured conversation management to pipecat, allowing developers to define dialogue states and transitions. From a turn detection perspective, pipecat-flows introduces the notion of state-dependent thresholds: different conversation states can use different end-of-turn parameters. A state expecting a long-form open answer might use a 1500 ms threshold, while a state expecting a short confirmation might use 400 ms.

This state-aware approach is a meaningful improvement on global threshold configuration. It is also relatively underused in practice, suggesting an area where further tooling and guidance would benefit deployments.

V. OPEN CHALLENGES

This section identifies five challenges that current production turn detection handles poorly and that would benefit from systematic research attention.

5.1 Multilingual Turn-Taking Variation

Turn-taking norms vary across languages and cultures. Japanese conversation features systematic short overlaps and backchannel responses [8]. English conversation tends toward minimal overlap and clear turn boundaries. Italian and several Romance-language conversational styles feature longer permissible overlaps. Current VAD-based and semantic systems are typically tuned on English data with implicit English conversational norms baked in.

This becomes consequential when voice AI systems are deployed in non-English contexts. A system tuned on English data may consistently interrupt Italian speakers, who use longer overlap durations, or fail to detect endpoints with Japanese speakers, whose pause patterns differ. Systematic study of cross-lingual turn-taking variation in the context of voice AI deployment is an open area.

5.2 Robustness to Background Speech

Production calls frequently contain background speech: a television playing in the user's environment, a colleague speaking nearby, or an automated announcement on a public address system. VAD-based endpointing does not distinguish the target speaker from background voices, leading to incorrect turn-taking decisions when background speech is present.

Speaker-conditioned VAD, which incorporates an embedding of the target speaker, addresses this in principle. In practice, deployment is complicated by cold-start (the system has no embedding for the target speaker at conversation start), embedding drift (the speaker's voice characteristics may vary), and computational cost. This is an open area where improved modeling could yield substantial production impact.

5.3 Partial Endpointing for Interruption Tolerance

Current systems treat end-of-turn as a binary event: either the turn has ended or it has not. In reality, the signal is graded. A user may pause for a moment with the clear intention of continuing. Treating this as a possible end-of-turn requires the system to decide whether to begin generating a response or wait.

A more sophisticated approach would compute a continuous turn-completion probability and use this to guide downstream behavior. Low probability: continue listening. Medium probability: begin generating the response but do not yet play TTS. High probability: play TTS and commit to the turn transition. This kind of graded endpointing is technically straightforward but operationally underutilized.

5.4 Evaluation Under Realistic Codec Chains

Production voice AI runs over codecs: Opus over WebRTC, G.711 over telephony, AMR-WB over mobile networks. Audio frequently passes through multiple codecs in a single call (capture codec, transmission codec, downstream re-encoding). The effect of these codec chains on turn detection accuracy is essentially unstudied in the open literature.

This parallels a similar gap identified in the voice clone detection literature [3], where deployment-aware evaluation including codec robustness has been proposed as a minimum standard. Adopting comparable evaluation rigor for turn detection would surface failure modes that are currently invisible.

5.5 The Latency-Accuracy Frontier

Every turn detection design choice trades latency against accuracy. A shorter silence threshold reduces perceived response delay but increases false-endpoint rate. Adding a semantic stage improves accuracy but adds 50 to 200 ms of latency. Predictive turn-taking eliminates the silence gap but introduces a risk of responding to incomplete turns.

Production systems navigate these tradeoffs empirically, with limited systematic guidance. A characterization of the achievable latency-accuracy frontier under realistic deployment conditions, broken down by conversation type (transactional, supportive, exploratory) and language, would be a valuable contribution.

VI. CONNECTION TO ADJACENT SUB-SECOND AUDIO DECISIONS

Turn detection is one instance of a broader class of problems in production voice AI: sub-second decisions on streaming audio under deployment constraints. Other instances include voice clone detection, speaker verification, emotion detection, and content moderation. These problems share structural features: they operate on short audio windows, must run with bounded latency on commodity hardware, and degrade under the same set of real-world conditions (codec compression, background noise, telephony bandwidth).

Recent work on deployment-aware evaluation of voice clone detection [3] proposed a Minimum Benchmark Suite covering cross-dataset generalization, codec robustness, noise robustness, reverberation, short-input behavior, inference cost, and confidence calibration. Several of these dimensions transfer directly to turn detection. Codec robustness, short-input behavior (especially for the first turn of a conversation, where transcript context is limited), and inference cost are all underreported in the turn detection literature.

A common evaluation framework across these adjacent problems would benefit the field. It would enable transfer of methodological insights, reduce duplicated effort in benchmark construction, and create pressure for production-relevant rather than benchmark-only research.

VII. CONCLUSION

Turn detection is a central component of production voice AI systems whose treatment in the academic literature lags its operational importance. This paper has surveyed the three families of approaches in current production use — pure VAD, acoustic-semantic hybrid, and predictive — and examined how they are implemented in the dominant open-source frameworks. We have identified five open challenges where targeted research could yield substantial deployment improvements: multilingual turn-taking variation, robustness to background speech, graded endpointing for interruption tolerance, codec-chain evaluation, and characterization of the latency-accuracy frontier.

The broader argument is that turn detection should be treated as a first-class problem in voice AI research, not an engineering detail subordinate to ASR and TTS. The user-facing quality of a voice agent depends on turn detection at least as much as on the quality of its underlying language model. Bringing the literature into line with this operational reality would benefit both researchers entering the field and practitioners building deployed systems.

REFERENCES

[1] Daily.co. (2024). pipecat: Open-source framework for voice and multimodal conversational AI. GitHub repository. https://github.com/pipecat-ai/pipecat

[2] LiveKit. (2024). livekit/agents: A framework for building realtime voice AI agents. GitHub repository. https://github.com/livekit/agents

[3] Rupesh, K. (2026). Toward Deployment-First Voice Clone Detection: A Lightweight and Robust Evaluation Protocol for Audio Deepfake Defense. SSRN Preprint, Abstract ID 6778759.

[4] Silero Team. (2024). Silero VAD: Pre-trained Enterprise-Grade Voice Activity Detector. GitHub repository. https://github.com/snakers4/silero-vad

[5] Google. (2011). WebRTC Voice Activity Detector. WebRTC open-source project.

[6] Chang, S.-Y., Prabhavalkar, R., He, Y., Sainath, T. N., & Simko, G. (2019). Joint endpointing and decoding with end-to-end models. ICASSP 2019, 5626-5630.

[7] LiveKit. (2025). Lightweight semantic turn detection models for voice agents. Model release documentation.

[8] Stivers, T., Enfield, N. J., Brown, P., Englert, C., Hayashi, M., Heinemann, T., Hoymann, G., Rossano, F., de Ruiter, J. P., Yoon, K.-E., & Levinson, S. C. (2009). Universals and cultural variation in turn-taking in conversation. Proceedings of the National Academy of Sciences, 106(26), 10587-10592.

Disclosure of Writing Assistance:

The author used a generative AI language tool for drafting and language-editing support. The author reviewed the manuscript, verified the claims and references, and takes full responsibility for all content.

Copyright & License:

© Authors retain the copyright of this article. This work is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.