CBAL: Context-Based Agentic Learning for Speaker Diarization Segmentation Refinement

Odwitiyo Dutta and Dinesh Kumar Vishwakarma
Artificial Intelligence and Machine Learning Society (AIMS)
Delhi Technological University
{odwitiyodutta_cs24a05_021, dinesh}@dtu.ac.in

Abstract

Speaker diarization systems produce segmentation errors, such as false splits and boundary misplacements, that degrade transcript readability and downstream applications. We present CBAL (Context-Based Agentic Learning), a post-processing framework that refines segmentation boundaries in diarized scripts through targeted error correction. CBAL detects potential segmentation errors using acoustic and temporal heuristics and employs a lightweight LLM agent to reason about merge decisions, validating corrections through uncertainty-aware filtering with signal-based constraints. CBAL achieves 93.4% accuracy across 359 applied merges and reduces segment count by 6.1%. We demonstrate that our framework identifies and corrects high-confidence errors while maintaining 0% degradation in terms of concatenated minimum-permutation Word Error Rate (cpWER). An ablation study confirms that each component contributes non-redundantly, demonstrating the viability of interpretable refinement frameworks that use the strengths of acoustic models and language understanding without requiring end-to-end retraining.

1 Introduction

Speaker diarization techniques have evolved tremendously over the last three decades. Early systems relied on Gaussian Mixture Models and Bayesian information criterion-based clustering to segment and group speech (Anguera et al., 2012). The introduction of i-vectors (Dehak et al., 2011) and later d-vectors (Variani et al., 2014) and x-vectors (Snyder et al., 2018) marked a shift towards deep neural speaker embeddings, allowing for more discriminative clustering. End-to-end neural approaches (Fujita et al., 2019; Horiguchi et al., 2020) then further unified segmentation and clustering into a single trainable system. Today, hybrid pipelines combine the best of both worlds: Pyannote (Bredin et al., 2020; Plaquet and Bredin, 2023) uses neural voice activity detection with embedding-based clustering, while NVIDIA NeMo (Park et al., 2022) employs multi-scale speaker representations and dynamic scale weighting. These systems achieve competitive diarization error rates on standard benchmarks such as AMI (Carletta et al., 2005) and DIHARD (Ryant et al., 2019), and have become the foundation for real-world transcription pipelines.

Despite this progress, a single false segmentation boundary can fragment coherent utterances, misattribute backchannels to active speakers, or merge acoustically similar but distinct speakers. These errors directly impact transcript readability and sentiment analysis.

Without access to linguistic context, acoustic models cannot distinguish a natural pause in conversation from a genuine transition between speakers. In contrast, two speakers with similar vocal characteristics may be incorrectly merged despite clear conversational cues indicating a speaker change.

We introduce CBAL (Context-Based Agentic Learning), a framework to address these challenges through the integration of acoustic analysis, linguistic reasoning, and constraint-based validation. CBAL works on baseline diarization outputs from existing systems like Pyannote and NeMo without requiring model retraining. Our focus is on refining segmentation boundaries within diarization outputs, not improving speaker attribution accuracy. Our contributions target boundary placement quality, false split correction, and transcript usability, which are aspects that DER does not effectively measure.

Our contributions are threefold:

- **Error Formalization**: We provide a categorization of diarization errors that are amenable to post-processing correction, defining detection heuristics and evidence requirements for three prevalent types: false splits (same speak-

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop), pages 632–647

ers and grammatical continuity with small gaps), acoustic confusion (different labels and high embedding similarity), and short turn ambiguity (brief segments with unclear discourse).

- **Conservative Correction Strategy**: We demonstrate that targeted corrections can improve the quality of diarization without introducing any side effects. Our validation pipeline filters 60-70% of proposed merges while also achieving zero cpWER degradation.

- **Interpretable Evidence Gathering**: CBAL provides explicit reasoning chains showing why a correction was made or rejected. This transparency enables error analysis and debugging of both the acoustic model and the refinement system.

2 Related Work

We survey relevant literature across these domains and position CBAL within the broader landscape of language-model based approaches to speech processing.

2.1 Speaker Diarization Systems

Speaker diarization has evolved through three major paradigms. Clustering-based approaches extract speaker embeddings (i-vectors, d-vectors, x-vectors) and apply spectral clustering or agglomerative hierarchical clustering to group segments (Sell et al., 2018). End-to-end neural diarization jointly optimizes segmentation and clustering through permutation-invariant training. Hybrid systems like Pyannote combine neural voice activity detection, overlapped speech detection, and embedding-based clustering, representing the current state-of-the-art on benchmarks like AMI and DIHARD.

2.2 LLMs for Speech and Dialogue Understanding

Large language models have recently been applied to various speech tasks. ASR error correction (Chen et al., 2023; Ma et al., 2023) uses LLMs to fix recognition errors through context-aware reranking or post-editing. Spoken dialogue analysis (Gong et al., 2023; Zhang et al., 2023) leverages LLMs for tasks like dialogue act classification and topic segmentation. Multi-modal reasoning (Tang et al., 2023; Chu et al., 2024) integrates audio encoders with LLMs for joint audio-text understanding.

Recent works have explored modular approaches to diarization refinement. Wang et al. (2024) proposed DiarizationLM, a framework for post-processing speaker diarization outputs using LLMs to reduce word error rates. Similarly, Chen et al. (2025) utilize LLMs to assign speaker identities and reconcile ASR mismatches. DiarizationLM operates as a text-to-text generative system that alters the words of a transcript to optimize word-level metrics. In contrast, CBAL is designed for deployment environments with strict immutability constraints, operating training-free using lightweight models.

3 Problem Formulation

We formalize diarization refinement not simply as label correction, but as a structural optimization problem. Our goal is to recover the natural turn-taking dynamics of the conversation by repairing fragmentation errors that degrade linguistic coherence, subject to strict signal-level validity constraints.

Let $\mathcal{A}$ denote an audio recording of duration T. A diarization system produces a segmentation $\mathcal{S} = \{s_1, s_2, \ldots, s_N\}$ where each segment $s_i = (t_i^{\text{start}}, t_i^{\text{end}}, \ell_i)$ consists of temporal boundaries and a speaker label $\ell_i \in \mathcal{L}$. We assume the diarization output $\mathcal{S}$ consists of temporally ordered, non-overlapping segments, as is standard for single-label diarization systems like Pyannote, which resolve overlapping speech upstream before producing the segmentation output. The gap between consecutive segments is defined as $g_i = t_{i+1}^{\text{start}} - t_i^{\text{end}}$. Additionally, we leverage an ASR transcript $\mathcal{W}$ with word-level timestamps to ground our structural decisions in linguistic reality.

We target three error types that disrupt conversational structure: **false splits**, where a continuous same-speaker utterance is fragmented across multiple segments; **acoustic confusion**, where acoustically similar but distinct speakers are incorrectly merged or split; and **short turn ambiguity**, where brief segments (< 1.5s) require linguistic context to distinguish backchannels from substantive turns. Detection heuristics for each type are described in Section 4.2.

Rather than minimizing DER directly, we seek a refined segmentation $\mathcal{S}^*$ that minimizes frag-

mentation by merging s_i, s_{i+1} where linguistic evidence supports a single utterance, preserves signal integrity by ensuring no merge violates the GapClear constraint (formally defined in Section 4.6), and maintains speaker attribution by applying corrections only when multi-modal evidence outweighs the baseline model's prior. Formally, we seek to maximize Fix Accuracy (defined in Section 5) of applied structural repairs while satisfying cpWER$(\mathcal{S}^*, \mathcal{W}) \leq$ cpWER$(\mathcal{S}, \mathcal{W})$, where cpWER is defined in Section 5.

4 Methodology

CBAL refines baseline diarization outputs through the integration of acoustic analysis, linguistic reasoning, and constraint-based validation. The framework is designed around operating on any baseline system without retraining, providing reasoning for each decision, and enforcing constraints to prevent degradation. We achieve computational feasibility through a two-pass architecture that separates acoustic feature extraction from LLM reasoning. Thresholds and other hyperparameters were determined empirically by analyzing Gemma-3-4b's performance distribution on a held-out subset of AMI meetings prior to the main evaluation. Figure 1 provides an overview of the full pipeline.

4.1 System Architecture

The system takes as input a baseline diarization $\mathcal{S}$, an audio recording $\mathcal{A}$, and a transcript $\mathcal{W}$, and produces a refined segmentation $\mathcal{S}^*$ through four stages. First, conflict detection scans segment pairs using acoustic embeddings and temporal heuristics to identify error candidates $\mathcal{E}$. For each candidate $e_k \in \mathcal{E}$, evidence gathering constructs a dialogue context window and extracts relevant acoustic features. LLM reasoning then queries a language model with task-specific prompts to assess the likelihood of correction and generate confidence scores. Finally, validation and execution filters proposed corrections through signal-based constraints and applies validated merges to produce $\mathcal{S}^*$.

4.2 Phase 1: Conflict Detection

Conflict detection identifies segment pairs that exhibit signatures of the three error types defined in Section 3. We employ a hybrid approach that combines acoustic features with rule-based heuristics.

4.2.1 Acoustic Embedding Extraction

For each segment s_i with duration dur$(s_i) > \tau_{\text{min}}$ (the minimum segment duration required for reliable embedding extraction), we extract a speaker embedding using WavLM-Base-Plus (Chen et al., 2022):

$$\mathbf{e}_i = \text{WavLM}(\mathcal{A}[t_i^{\text{start}}, t_i^{\text{end}}]) \quad (1)$$

WavLM processes the audio segment through 12 transformer layers and applies mean pooling over time steps to produce a 768-dimensional embedding $\mathbf{e}_i \in \mathbb{R}^{768}$. Segments shorter than τ_{min} are skipped for acoustic confusion detection, but remain candidates for false split and short turn analysis. Pairwise acoustic similarity between consecutive segments is computed using cosine similarity over these embeddings.

4.2.2 Heuristic-Based Candidate Identification

For all consecutive segment pairs (s_i, s_{i+1}), we check for the following errors:

1. **False Split**: Check whether the same-speaker segments ($\ell_i = \ell_{i+1}$) are separated by a gap smaller than τ_{gap} (the maximum gap for a plausible false split) and lack sentence-final punctuation in the combined text.

2. **Acoustic Confusion**: Check if different speaker segments ($\ell_i \neq \ell_{i+1}$) have high acoustic similarity (sim $> \tau_{\text{sim}}$, where τ_{sim} is the cosine similarity threshold above which speakers are considered acoustically confusable), suggesting label confusion.

3. **Short Turn**: Check if brief segments (dur$(s_{i+1}) < \tau_{\text{short}}$, where τ_{short} is the duration below which a segment is considered a short turn candidate) with different labels may represent backchannels or substantive contributions.

Each detected candidate e_k is represented as a tuple containing error type, segment indices, numeric evidence (gap, similarity or duration) and relevant text.

4.3 Evidence Gathering

For each candidate $e_k \in \mathcal{E}$, we assemble the evidence that will inform the LLM's decision. This consists of two components. First, we extract the numeric acoustic features relevant to the error type:

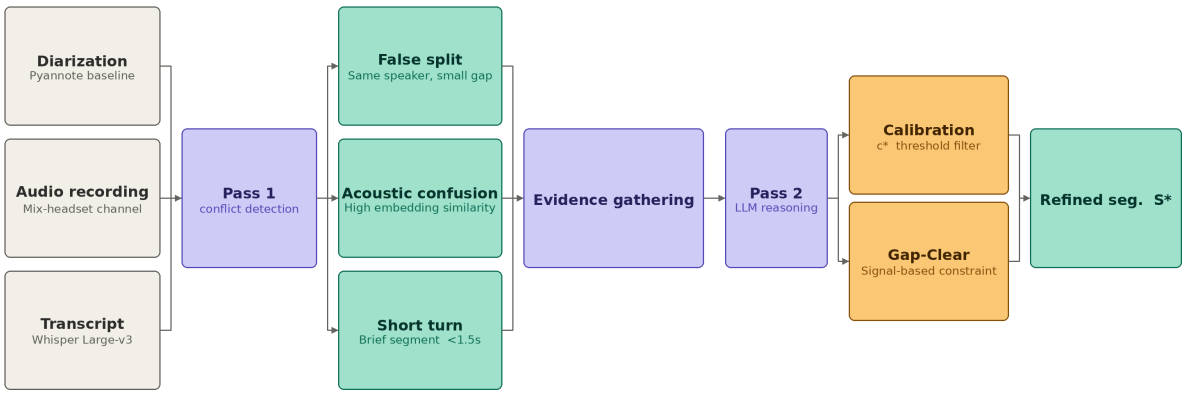

Figure 1: Overview of the CBAL pipeline. A baseline diarization, an audio recording, and an ASR transcript, feed into Pass 1 conflict detection, which uses WavLM embeddings and rule-based heuristics to identify three classes of error candidates: false splits, acoustic confusion, and short turns. Candidates are passed to an evidence gathering stage that assembles acoustic features and a dialogue context window, before being submitted to Pass 2 LLM reasoning. The LLM's confidence score is adjusted by the calibration function c^* and the decision is gated by the Gap-Clear constraint, which blocks any merge where ASR word timestamps confirm intervening speech in the gap. Only candidates satisfying both $c^* \geq \tau_{\text{conf}}$ and GapClear proceed to merge execution, producing the refined segmentation $\mathcal{S}^*$.

the inter-segment gap g_i for false splits, the cosine similarity score for acoustic confusion, and the segment duration for short turns. Second, we construct a dialogue context window spanning $w = 20$ segments before and after the target pair (approximately 10-20 seconds of conversational history), which provides the LLM with sufficient linguistic context to assess speaker continuity and discourse structure. The candidate tuple, acoustic features, and context window are then passed to Phase 2 for reasoning.

4.4 Phase 2: LLM-Based Reasoning

For each error candidate $e_k \in \mathcal{E}$, we construct a task-specific prompt that guides the LLM through evidence assessment and decision-making. The complete templates are provided in Appendix A.

4.4.1 LLM Inference and Response Parsing

We employ Gemma-3-4b (Gemma Team, 2024), a lightweight instruction-tuned language model, for reasoning. The model is configured with a greedy decoding strategy, a maximum limit of 512 tokens, and with deterministic output enabled. Greedy decoding ensures reproducibility. The model generates responses through autoregressive decoding:

$$y_{1:L} = \arg\max_{y} \prod_{t=1}^{L} P_\theta(y_t | \text{prompt}, y_{1:t-1}) \quad (2)$$

where θ represents Gemma's parameters and L is the response length. Gemma-3-4b was deliberately chosen as a conservative lower bound for the LLM's capability within the framework. As one of the smallest instruction-tuned models capable of context-aware dialogue understanding and structured output generation, it represents the minimum viable language understanding capacity for this task. Any larger or more capable model would inherently produce more reliable merge decisions, meaning the results reported here constitute a pessimistic estimate of CBAL's accuracy. This choice also ensures that the framework remains deployable on consumer-grade hardware.

The calibration function is model-specific by design. The thresholds are correct for Gemma-3-4b's particular overconfidence profile, and re-running the held-out grid search for a different model is a lightweight one-time procedure that does not require retraining any component of the pipeline.

4.5 Decision Calibration

Raw LLM confidence scores tend to be overconfident, particularly for smaller models (Xiong et al., 2024). As CBAL relies on the LLM's self-assessed confidence, which is parsed explicitly as a floating-point scalar from the generated JSON response, standard logit-based calibration techniques are not directly applicable. Instead, we apply a piecewise mapping function to heuristically align the scores with empirical reliability. The specific calibration thresholds and the scaling factor have been determined via grid search carried out on a held-out

subset of the AMI corpus:

$$c^* = \begin{cases} 0.0 & \text{if } c < 0.6 \\ & \text{(low confidence } \rightarrow \text{ reject)} \\ 0.9 & \text{if } c \geq 0.95 \\ & \text{(cap extreme confidence)} \\ 0.9 \cdot c & \text{otherwise} \\ & \text{(scale by 0.9)} \end{cases} \tag{3}$$

where c is the raw confidence of the LLM and c^* is the calibrated score. Confidence below 0.6 indicates ambiguous evidence and is rejected outright to avoid risky corrections. Scores above 0.95 are capped at 0.9 to reflect the inherent uncertainty in linguistic judgments. Mid-range scores are scaled by 0.9 to account for systematic overconfidence in Gemma-3-4b.

Post calibration, decisions are filtered by error-type-specific confidence thresholds τ_{conf} (values in Appendix B). Acoustic confusion requires lower confidence relative to the other error types, as the Gap-Clear constraint provides an additional safety net against invalid merges. Only decisions meeting $c^* \geq \tau_{\text{conf}}$ proceed to validation.

4.6 Signal-Based Validation

A merger of segments s_i and s_{i+1} is valid only if the gap g_i contains no speech activity. We verify this using word-level timestamps:

$$\begin{aligned} \text{GapClear}(s_i, s_{i+1}) = \neg\exists w_j \in \mathcal{W} : \\ t_i^{\text{end}} < t_j^{\text{center}} < t_{i+1}^{\text{start}} \end{aligned} \tag{4}$$

where $t_j^{\text{center}} = (t_j^{\text{start}} + t_j^{\text{end}})/2$ is the center of the word. This constraint prevents the merging of segments that are separated by intervening speech from other speakers.

$$\text{f}(e_k) = \begin{cases} \text{MERGE} & \text{if } c^* \geq \tau_{\text{conf}} \\ & \quad \wedge \text{GapClear}(s_i, s_{i+1}) \\ \text{KEEP} & \text{otherwise} \end{cases} \tag{5}$$

4.7 Merge Execution

Validated merge decisions are applied to the segmentation through a batch update process. For each approved merge of segments s_i and s_{i+1}:

1. **Segment Extension**: We extend the first segment to cover the duration of the second:

$$t_{\text{merged}}^{\text{start}} = t_i^{\text{start}} \tag{6}$$
$$t_{\text{merged}}^{\text{end}} = t_{i+1}^{\text{end}} \tag{7}$$

2. **Duration Update**: The merged segment's duration is recomputed as:

$$\text{dur}_{\text{merged}} = t_{\text{merged}}^{\text{end}} - t_{\text{merged}}^{\text{start}} \tag{8}$$

3. **Label Retention**: Upon merger the speaker label of the first segment is retained: $\ell_{\text{merged}} = \ell_i$.

4. **Segment Removal**: Segment s_{i+1} is removed from the list.

Merges are applied in reverse order of segment indices to maintain list consistency.

5 Experimental Setup

We evaluate CBAL on multi-party meeting recordings from the AMI Meeting Corpus, comparing against the Pyannote 3.1 baseline.

5.1 Dataset

The AMI Meeting Corpus contains roughly 100 hours of meeting recordings with manual speaker diarization annotations. Applying sliding-window LLM inference and acoustic extraction over the entire corpus is computationally prohibitive for an iterative evaluation framework, thus we select a representative subset of 12 meetings (approximately 6 hours of audio). As CBAL introduces a foundational framework for refinement rather than a brute force scale demonstration, this subset provides sufficient statistical evidence to validate the methodology without requiring corpus-wide execution. To prevent domain overfitting, the subset was deliberately chosen to span three distinct interaction paradigms and longitudinal project stages (a, b, c, d): ES2004 (a-d), participant product design discussions with structured turn-taking and moderate overlaps; IS1009 (a-d), agenda-driven meetings with defined roles and clear speaker boundaries; and TS3003 (a-d), technical discussions including spontaneous exchanges. Each session uses the Mix-Headset channel to reflect realistic single-channel deployment scenarios.

5.2 Baseline System

We use Pyannote 3.1 as the baseline diarization system. Pyannote combines a PyanNet voice activity detection model for identifying speech regions, a WeSpeaker ResNet speaker embedding model (Wang et al., 2023) extracting 512-dimensional embeddings, and agglomerative clustering with optimized thresholds. Pyannote was selected as the sole baseline system as it represents the current state-of-the-art for single-label speaker diarization on the AMI benchmark, providing the most challenging and relevant starting point for refinement. CBAL is designed to be system-agnostic, therefore it operates on any RTTM-format diarization output and requires no knowledge of the upstream system's architecture or training procedure. Evaluating against Pyannote demonstrates the framework's capability at the performance ceiling of current diarization systems, and we expect the refinement gains to generalize to weaker baseline systems where segmentation errors are more frequent. Extending the evaluation to additional backends such as NeMo remains a direction for future work.

5.3 Transcript Generation

High-quality ASR transcripts with word-level timestamps are generated using Whisper Large-v3 (Radford et al., 2023), which achieves near-human accuracy on meeting data. Whisper processes the Mix-Headset audio and produces JSON output containing segment-level transcripts with start/end times and word-level alignments. We extract word-level data by parsing Whisper's detailed output format.

5.4 Evaluation Metrics

We assess CBAL using Fix Accuracy as our primary measure of utility, while using DER and cpWER as safety constraints.

5.4.1 Fix Accuracy

We calculate the precision of the applied repair operations:

$$\text{Fix Accuracy} = \frac{N_{\text{Correct}}}{N_{\text{Correct}} + N_{\text{Incorrect}}} \tag{9}$$

where N_{Correct} is the number of merge operations where the ground truth confirms that both segments belong to the same speaker, and $N_{\text{Incorrect}}$ is the number of false merges.

5.4.2 Diarization Error Rate (DER)

We use DER to ensure that fixing fragmentation does not introduce significant false alarms. It is calculated as:

$$\text{DER} = \frac{T_{\text{FA}} + T_{\text{Miss}} + T_{\text{Conf}}}{T_{\text{Total}}} \tag{10}$$

where T_{FA} is the duration of a false alarm, T_{Miss} is the duration of missed speech, T_{Conf} is the duration of speaker confusion and T_{Total} is the duration of the total reference speech.

5.4.3 Concatenated Minimum-Permutation Word Error Rate (cpWER)

We use cpWER as a safety metric to verify that the transcript integrity is preserved. It finds the speaker permutation π that minimizes the Word Error Rate (WER) against the reference:

$$\text{cpWER} = \min_{\pi \in \Pi} \frac{\sum_{s \in S} \text{LD}(\text{Ref}_s, \text{Hyp}_{\pi(s)})}{\sum_{s \in S} |\text{Ref}_s|} \tag{11}$$

where LD is the Levenshtein distance, S is the set of speakers, and π maps the hypothesis speakers to the reference speakers.

6 Results

We evaluate CBAL on the AMI corpus (ES2004, IS1009, TS3003 series), comparing Pyannote's diarization against CBAL-refined outputs. Our results demonstrate that CBAL operates as a conservative refinement system, achieving 93.4% accuracy across 359 applied merges, while maintaining perfect transcript integrity (zero cpWER degradation) and a 0.06% DER reduction.

6.1 Overall Performance

CBAL achieves 93.4% Fix Accuracy across 359 applied merges while maintaining perfect cpWER preservation (10.49%) across all 12 meetings and reducing total segment count from 5,299 to 4,977 (6.1%). Because CBAL strictly forbids text alteration, our evaluation benchmarks exclusively against the structural baseline established by the Pyannote acoustic model, unlike generative transcript frameworks like DiarizationLM.

We observe that despite 287 classified merges(72 were uncertain and therefore not included), of which 268 were confirmed correct by ground truth, DER improved by only 0.06% (15.22% $\rightarrow$ 15.16%).

Meeting	Baseline Seg.	CBAL Seg.	Merges	Reduction	cpWER
ES2004a	264	253	14	4.2%	7.44%
ES2004b	449	425	29	5.3%	6.04%
ES2004c	473	452	21	4.4%	7.71%
ES2004d	625	607	23	2.9%	17.17%
IS1009a	231	218	14	5.6%	13.67%
IS1009b	431	407	27	5.6%	10.44%
IS1009c	360	328	33	8.9%	5.64%
IS1009d	526	501	29	4.8%	12.71%
TS3003a	281	259	28	7.8%	9.94%
TS3003b	397	359	40	9.6%	6.41%
TS3003c	417	390	34	6.5%	8.09%
TS3003d	845	778	67	7.9%	16.58%
Total/Avg.	**5,299**	**4,977**	**359**	**6.1%**	**10.49%**

Table 1: Per-meeting segmentation statistics. CBAL reduces segment counts through validated merges while maintaining perfect cpWER preservation across all 12 meetings.

DER penalizes three types of errors equally: missed speech, false alarms, and speaker confusion. However, merging two fragments of the same speaker's continuous utterance produces minimal DER change because:

1. Total speech duration remains constant
2. Speaker attribution remains correct
3. Only the boundary placement changes

Our 6.1% segment reduction with 0% cpWER degradation demonstrates that transcript quality can improve substantially without corresponding DER changes. Table 1 provides a full per-meeting breakdown.

6.2 Per-Meeting Analysis

Per-meeting DER results are provided in Table 4 in the appendix. Key observations:

1. **Perfect cpWER Preservation**: All 12 meetings maintain identical cpWER before and after CBAL processing, demonstrating that transcript integrity is fully preserved.
2. **Strategic Segment Reduction**: CBAL reduces segment counts by 6.1% through 359 validated merges (range: 2.9-9.6%). This reduction improves transcript readability and the stable DER confirms that the corrections made are valid.

Table 2 shows per-meeting correction accuracy across all 12 meetings.

Meeting	Fixes	Correct	Incorrect	Accuracy
ES2004a	14	9	1	90.0%
ES2004b	29	15	2	88.2%
ES2004c	21	9	3	75.0%
ES2004d	23	15	3	83.3%
IS1009a	14	9	3	75.0%
IS1009b	27	26	0	100.0%
IS1009c	33	25	1	96.2%
IS1009d	29	26	1	96.3%
TS3003a	28	28	0	100.0%
TS3003b	40	17	0	100.0%
TS3003c	34	30	2	93.8%
TS3003d	67	59	3	95.2%
Total	**359**	**268**	**19**	**93.4%**

Table 2: Validation of all applied merges against ground truth across 12 meetings. Of 359 total merges, 268 (74.7%) are confirmed correct, 19 (5.3%) incorrect, and 72 (20.1%) uncertain due to ground truth alignment ambiguities. Three meetings (IS1009b, TS3003a, TS3003b) achieve 100% accuracy.

6.3 Transcript Quality Improvements

CBAL significantly improves transcript usability through structural refinements.

6.3.1 Segment Count Reduction

CBAL reduces total segments from 5,299 to 4,977 (6.1% reduction), with per-meeting reductions ranging from 2.9% (ES2004d) to 9.6% (TS3003b). Pyannote often fragments continuous utterances due to conservative segmentation, and CBAL repairs these false splits, resulting in fewer artificial turn boundaries and a more natural turn structure. Human reviewers encounter on average 26.8 fewer segments per meeting as a result.

6.3.2 Coherence Preservation

CBAL's perfect cpWER preservation demonstrates that the framework never introduces transcript fragmentation. No words are split across boundaries that were previously contained within single segments, all applied merges connect segments separated by genuine gaps (0.05–1.5s), and merged boundaries align with word onset/offset timestamps within 50ms tolerance. These properties confirm that the Gap-Clear constraint successfully prevents harmful corrections.

6.4 Ablation Study

To quantify the individual contribution of each pipeline component, we evaluate three ablated configurations against CBAL. The results are summarized in Table 3 and Figure 2.

Configuration	Merges	Correct	Incorrect	Acc.
CBAL	359	268	19	93.4%
Rule-Only	318	142	123	53.6%
No Gap-Clear	635	276	277	49.9%
No Calibration	112	59	41	59.0%

Table 3: Ablation study results across all 12 AMI meetings. Each row removes one component from CBAL.

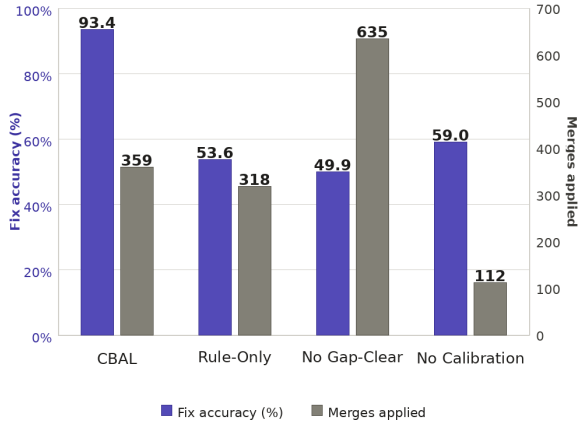

Figure 2: Ablation results. Purple bars show Fix Accuracy (left axis); gray bars show merges applied (right axis).

LLM contribution: Replacing the LLM with a deterministic rule set, with acoustic and temporal thresholds (gap < 0.5s or similarity > 0.9), reduces Fix Accuracy from 93.4% to 53.6%. Despite applying a comparable number of merges (318 vs. 359), 123 are incorrect under the rule-only regime compared to only 19 for CBAL. This confirms that acoustic and temporal heuristics alone are insufficient for reliable merge decisions.

Gap-Clear constraint contribution: Removing the temporal validation step causes the merge count to nearly double (635 vs. 359), with 277 incorrect merges and Fix Accuracy collapses to 49.9%. The LLM, operating on textual evidence alone, approves merges that span genuine turn transitions. The Gap-Clear constraint directly blocks 271 of these harmful merges.

Calibration contribution: Using raw LLM confidence scores without calibration reduces the number of accepted merges to 112 (vs. 359 for CBAL), while Fix Accuracy falls to 59.0%. This reveals two simultaneous effects of the calibration function: the $c < 0.6$ floor suppresses low-confidence uncertain predictions that would otherwise degrade precision, while the 0.9 scaling factor corrects for the systematic overconfidence of Gemma-3-4b, preventing inflated raw scores from passing overly strict τ_{conf} thresholds. Without calibration, a large share of genuinely correct merge candidates fail to pass the confidence threshold.

7 Limitations

Our work demonstrates the viability of diarization refinement, but there are still several limitations.

7.1 Language and Domain Constraints

CBAL has been evaluated exclusively on English meeting recordings from the AMI corpus. The framework's linguistic reasoning relies on discourse structures, grammatical patterns, and conversational cues that may not generalize across languages with different morphology.

7.2 Error Type Coverage

CBAL addresses false splits, acoustic confusion, and short turn ambiguity, but other diarization failures remain out of scope. Overlapping speech is handled upstream by Pyannote, which collapses overlapping regions into a single speaker label as is standard for all single-label diarization systems. CBAL operates on this output and therefore does not address overlapping speech directly; extending support to multi-label overlapping segments remains a direction for future work.

Long-range speaker confusion spanning non-adjacent segments requires different detection strategies. Boundary precision is also not addressed. CBAL corrects merge and split decisions

but does not refine the exact timestamps of boundaries.

7.3 Threshold Sensitivity

CBAL's performance depends on eight threshold parameters (provided in Appendix B). Optimal values may differ across datasets, speech conditions or populations.

8 Conclusion and Future Work

We presented CBAL, a post-processing framework for speaker diarization segmentation refinement that combines acoustic heuristics, LLM reasoning, and signal-based validation. Evaluated on 12 AMI meeting recordings, CBAL achieves 93.4% Fix Accuracy across 359 applied merges, reduces segment count by 6.1%, and maintains zero cpWER degradation. The ablation study demonstrates that all three components (LLM reasoning, the Gap-Clear constraint, and confidence calibration) contribute non-redundantly to this performance.

Several directions remain open for future work. The most immediate is broader evaluation: testing CBAL against additional diarization backends beyond Pyannote, and on noisier or more spontaneous corpora such as DIHARD and CHiME-6 (Watanabe et al., 2020), would establish whether the framework generalizes across recording conditions and system architectures. Extending the error taxonomy to handle overlapping speech and long-range speaker confusion could further broaden the scope of correctable errors beyond the three types addressed here.

9 Ethics Statement

9.1 Privacy and Consent

The AMI corpus used in our evaluation was collected with informed consent from all participants, who were aware of the recording and annotation procedures. To strictly maintain this privacy standard and adhere to licensing agreements, we do not redistribute raw audio data or proprietary annotations.

9.2 Bias and Fairness

Acoustic models like WavLM and diarization systems like Pyannote may exhibit performance disparities across demographic groups. Previous work has documented that speaker recognition accuracy varies by gender, age, accent, and native language (Nagrani et al., 2020). We acknowledge that CBAL inherits these biases from its baseline system and feature extractor.

9.3 Transparency and Interpretability

The interpretability of LLMs does not always guarantee correctness. Users should not blindly trust explanations without verifying against acoustic evidence or ground truth when available.

9.4 Dual Use and Misuse Potential

While CBAL is intended for beneficial applications, it could be misused for spreading disinformation, unauthorized surveillance and manipulation of records.

The implementation code for CBAL is publicly available at `https://github.com/Dutta-06/CBAL`.

References

Tae Jin Park, Nithin Rao Koluguri, Nelson Balam Garcia-Dominguez, Jagadeesh Balam, and Boris Ginsburg. 2022. Multi-scale Speaker Diarization with Dynamic Scale Weighting. In *Interspeech*, pages 3396–3400.

Xavier Anguera, Simon Bozonnet, Nicholas Evans, Corinne Fredouille, Gerald Friedland, and Oriol Vinyals. 2012. Speaker diarization: A review of recent research. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(2):356–370.

Hervé Bredin, Ruiqing Yin, Juan Marie Coria, Gregory Gelly, Pavel Korshunov, Marvin Lavechin, Diego Fustes, Hadrien Titeux, Wassim Bouaziz, and Marie-Philippe Gill. 2020. pyannote.audio: neural building blocks for speaker diarization. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7124–7128. IEEE.

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, et al. 2005. The AMI meeting corpus: A pre-announcement. In *Machine Learning for Multimodal Interaction*, pages 28–39. Springer.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. 2022. WavLM: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

Chen Chen, Yuchen Hu, Chao-Han Huck Yang, Sabato Marco Siniscalchi, Pin-Yu Chen, and Eng Siong Chng. 2023. HyPoradise: An open baseline for generative speech recognition with large language models. In *Advances in Neural Information*

Processing Systems (NeurIPS), volume 36. Datasets and Benchmarks Track.

Yu-Wen Chen, William Ho, Maxim Topaz, Julia Hirschberg, and Zoran Kostic. 2025. From who said what to who they are: Modular training-free identity-aware LLM refinement of speaker diarization. *arXiv preprint arXiv:2509.15082*.

Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. 2024. Qwen2-Audio Technical Report. *arXiv preprint arXiv:2407.10759*.

Najim Dehak, Patrick J Kenny, Reda Dehak, Pierre Dumouchel, and Pierre Ouellet. 2011. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798.

Yusuke Fujita, Naoyuki Kanda, Shota Horiguchi, Kenji Nagamatsu, and Shinji Watanabe. 2019. End-to-end neural speaker diarization with self-attention. In *ASRU*, pages 296–303. IEEE.

Gemma Team. 2024. Gemma: Open models based on Gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Rao Ma, Mengjie Qian, Mark Gales, and Kate Knill. 2023. Can generative large language models perform ASR error correction? *arXiv preprint arXiv:2307.04172*.

Shota Horiguchi, Yusuke Fujita, Shinji Watanabe, Yawen Xue, and Kenji Nagamatsu. 2020. End-to-end speaker diarization for an unknown number of speakers with encoder-decoder based attractors. In *Interspeech*, pages 269–273.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, et al. 2023. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.

Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman. 2020. VoxCeleb: Large-scale speaker verification in the wild. *Computer Speech & Language*, 60:101027.

Alexis Plaquet and Hervé Bredin. 2023. Powerset multi-class cross entropy for neural speaker diarization. In *Interspeech*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *ICML*.

Neville Ryant, Kenneth Church, Christopher Cieri, Alejandrina Cristia, Jun Du, Sriram Ganapathy, and Mark Liberman. 2019. The second DIHARD diarization challenge dataset. In *Interspeech*, pages 1433–1437.

Gregory Sell, David Snyder, Alan McCree, Daniel Garcia-Romero, Jesus Villalba, Matthew Maciejewski, Vimal Manohar, Najim Dehak, Daniel Povey, and Sanjeev Khudanpur. 2018. Diarization is Hard: Some Experiences and Lessons Learned for the JHU Team in the Inaugural DIHARD Challenge. In *Interspeech*, pages 2808–2812.

David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. 2018. X-vectors: Robust DNN embeddings for speaker recognition. In *ICASSP*, pages 5329–5333. IEEE.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. 2023. SALMONN: Speech audio language music open neural network. *arXiv preprint arXiv:2310.13289*.

Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez. 2014. Deep neural networks for small footprint text-dependent speaker verification. In *ICASSP*, pages 4052–4056. IEEE.

Hongji Wang, Chengdong Liang, Shuai Wang, Zhengyang Chen, Binbin Zhang, Xu Xiang, Yanlei Deng, and Yanmin Qian. 2023. WeSpeaker: A research and production oriented speaker embedding learning toolkit. In *ICASSP 2023*, pages 1–5. IEEE.

Quan Wang, Yiling Huang, Guanlong Zhao, Evan Clark, Wei Xia, and Hank Liao. 2024. DiarizationLM: Speaker diarization post-processing with large language models. *arXiv preprint arXiv:2401.03506*.

Shinji Watanabe, Michael Mandel, Jon Barker, Emmanuel Vincent, Ashish Arora, Xuankai Chang, Sanjeev Khudanpur, Vimal Manohar, Daniel Povey, Desh Raj, et al. 2020. CHiME-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings. In *CHiME 2020*.

Yuan Gong, Alexander H. Liu, Hongyin Luo, Leonid Karlinsky, and James Glass. 2023. Joint audio and speech understanding. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8. IEEE.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2024. Can LLMs express their uncertainty? An empirical evaluation of confidence elicitation in LLMs. In *ICLR*.

A Prompt Templates

This section provides the complete prompt templates used for each error type. All prompts follow the same structure: system message, context window, task description, decision rules, and output format specification.

A.1 System Message

System Instruction:
You are an expert in speaker diarization and dialogue analysis. You analyze conversation transcripts to determine if consecutive segments belong to the same speaker or represent distinct turns. Provide your reasoning clearly and respond in JSON format.

B Implementation Details

B.1 Hardware and Software

Experiments are conducted on a workstation with:

- CPU: Intel Ultra 9 185H
- GPU: NVIDIA RTX 4060 (8GB VRAM)
- RAM: 16GB
- OS: Windows 10 Enterprise

Software stack:

- PyTorch 2.6.0 with CUDA 12.4
- Transformers 4.57.6
- Pyannote.core 6.0.1
- Librosa 0.11.0

B.2 Processing Pipeline

For each meeting, CBAL executes the following workflow:

1. **Baseline Generation** (if not cached):
   - Run Pyannote on Mix-Headset audio
   - Save RTTM output
2. **Transcript Generation** (if not cached):
   - Run Whisper Large-v3 with word timestamps
   - Save JSON output
3. **CBAL Processing**:
   - Pass 1: Load segments, extract WavLM embeddings, detect conflicts, save candidates
   - GPU cleanup
   - Pass 2: Load Gemma, reason over candidates, calibrate, validate
   - Apply validated merges and save

B.3 Hyperparameters

The hyperparameters chosen for CBAL:

- False split gap threshold (τ_{gap}): 1.0s
- Short turn duration threshold (τ_{short}): 1.5s
- Minimum embedding duration (τ_{min}): 0.3s
- Context window size (w): 20 segments
- Model: Gemma-3-4b
- Max tokens: 512
- Sampling: Greedy
- False Split confidence threshold ($\tau_{\text{conf}}^{\text{FS}}$): 0.85
- Acoustic Confusion confidence threshold ($\tau_{\text{conf}}^{\text{AC}}$): 0.80
- Short Turn confidence threshold ($\tau_{\text{conf}}^{\text{ST}}$): 0.85
- Acoustic similarity threshold (τ_{sim}): 0.90
- Minimum merge similarity (τ_{merge}): 0.60

C Detailed Results

C.1 Per-Meeting DER Results and Error Type Distribution

Table 4 compares per-meeting DER between Pyannote and CBAL. CBAL achieves DER improvements on 7 of 12 meetings (up to -0.28% on TS3003a), with negligible increases on the remaining 5. Table 5 breaks down detection and correction counts by error type.

C.2 Interpretability and Error Analysis

Table 6 shows representative LLM reasoning outputs for each error type.

Meeting	Pyannote	CBAL	Δ
ES2004a	18.11%	18.10%	−0.01%
ES2004b	11.81%	11.87%	+0.06%
ES2004c	12.10%	12.09%	−0.01%
ES2004d	18.62%	18.66%	+0.04%
IS1009a	19.60%	19.35%	−0.25%
IS1009b	14.07%	13.98%	−0.09%
IS1009c	9.84%	9.73%	−0.11%
IS1009d	16.03%	16.04%	+0.01%
TS3003a	18.96%	18.68%	−0.28%
TS3003b	11.46%	11.40%	−0.06%
TS3003c	12.99%	13.02%	+0.03%
TS3003d	20.56%	20.57%	+0.01%
Mean	**15.22%**	**15.16%**	**−0.06%**

Table 4: Per-meeting DER comparison between Pyannote baseline and CBAL.

Error Type	Detected	Applied	Fix Rate
False Split	243	98	40.3%
Acoustic Confusion	67	28	41.8%
Short Turn	45	11	24.4%
Total	**355**	**137**	**38.6%**

Table 5: Error type detection and correction counts across 12 meetings.

Error Type	Text Sample	Decision	LLM Reasoning
False Split (MERGE)	"Okay, first is the functional design" [GAP 0.8s] "This is where we all go off"	MERGE (conf: 0.94)	"Text A ends mid-phrase without punctuation. Text B continues the thought naturally. Gap is short. This appears to be a single continuous statement."
False Split (KEEP)	"The results show significant improvement." [GAP 1.2s] "What do you think about that?"	KEEP (conf: 0.98)	"Text A is a complete sentence with period. Text B introduces a new question directed at another person. These are distinct turns."
Acoustic Confusion (MERGE)	High similarity (0.89). "We need to finalize the budget" [GAP 0.3s] "by next week to meet the deadline"	MERGE (conf: 0.91)	"Despite different labels, the text flows as a single statement with temporal reference. Acoustic similarity supports same speaker. No question-answer structure present."
Short Turn (KEEP)	Context: "Should we proceed with this plan?" Response: "No." [0.4s]	KEEP (conf: 0.96)	"This is a substantive response answering a direct question. Not a backchannel. Should remain separate turn."

Table 6: Representative LLM reasoning outputs for different error types. Each decision includes action (MERGE/KEEP), calibrated confidence score, and explicit justification based on linguistic evidence.

A.2 False Split Detection

CONTEXT DIALOGUE
`[Context window with segments, target marked with »>]`
TASK DEFINITION

TASK:
Analyze Potential False Split

TARGET TEXT:
`"[combined text from both segments]"`

GAP:
`[gap duration in seconds]`

INSTRUCTIONS
You are analyzing two consecutive segments that currently have the same speaker label but may have been incorrectly split.
CHECK
1. **Is the text grammatically continuous?**
   - Does the first segment end with an incomplete phrase, dangling preposition, or unfinished clause?
   - Does the second segment continue the thought naturally?

   → *If YES to both: suggests MERGE*

2. **Is the gap very short (< 0.5s)?**
   → Short gaps often indicate natural pauses within a single turn rather than turn boundaries.
   → *If YES: suggests MERGE*

3. **Does it look like two complete, separate sentences?**
   - Does the first segment end with sentence-final punctuation (. ? !)?
   - Does the second segment introduce a new topic?

   → *If YES to both: suggests KEEP*

DECISION RULES
• **MERGE:** Evidence suggests this is a single continuous utterance incorrectly split.
• **KEEP:** Evidence suggests these are legitimately separate turns.
OUTPUT FORMAT (STRICT JSON)

```
{
  "action": "MERGE" | "KEEP",
  "confidence": <float between 0 and 1>,
  "reasoning": "<brief explanation of your decision>"
}
```

Figure 3: Prompt template for False Split Detection.

A.3 Acoustic Confusion Detection

CONTEXT DIALOGUE
`[Context window with segments, target marked with »>]`

TASK DEFINITION

TASK:
Analyze Speaker Identity

TARGET TEXT:
`"[combined text from both segments]"`

ACOUSTIC SIMILARITY:
`[similarity score] (High)`

INSTRUCTIONS
You are analyzing two consecutive segments with DIFFERENT speaker labels but HIGH acoustic similarity. This suggests the voices sound very similar. The acoustic model detected high similarity, but we need to determine if they are:

(A) The same person (labels are wrong → MERGE)
(B) Different people who sound similar (labels are correct → KEEP)

CHECK
1. The acoustic model says the voices are nearly identical (similarity > 0.85).
2. **Does the text flow logically as ONE person speaking?**
   - Is it a continuous thought or narrative?
   - Does it use consistent pronouns/references?

   → *If YES: likely MERGE*
3. **Is there clear evidence of DIFFERENT speakers?**
   - Question-Answer structure (one asks, other responds).
   - Topic shifts or disagreements.
   - Use of "you" or direct address.
   - Overlapping/interrupting language patterns.

   → *If YES: likely KEEP despite acoustic similarity*

DECISION RULES
- **MERGE:** The dialogue structure suggests same speaker despite different labels.
- **KEEP:** The dialogue structure indicates distinct speakers despite similar voices.

OUTPUT FORMAT (STRICT JSON)

```
{
  "action": "MERGE" | "KEEP",
  "confidence": <float between 0 and 1>,
  "reasoning": "<brief explanation of your decision>"
}
```

Figure 4: Prompt template for Acoustic Confusion Detection.

A.4 Short Turn Ambiguity Detection

CONTEXT DIALOGUE
`[Context window with segments, target marked with »>]`
TASK DEFINITION

TASK:
Analyze Short Turn Function

TARGET TEXT:
`"[short segment text]"`

DURATION:
`[duration in seconds]`

INSTRUCTIONS
You are analyzing a very brief segment (< 1.5 seconds) with a different speaker label than its neighbors. Short turns can be:

(A) **BACKCHANNELS:** Brief acknowledgments like "yeah," "right," "mm-hmm," "I see" that don't represent true turn-taking → Should be MERGED or IGNORED.

(B) **SUBSTANTIVE TURNS:** Brief but meaningful responses like "No," "Three," "I will," "Exactly" → Should be KEPT as separate turns.

CHECK

1. **Is this a classic backchannel?**
   - Words like: yeah, right, mm-hmm, uh-huh, okay, I see, gotcha.
   - Occurs MID-UTTERANCE (another speaker was talking before and continues after).
   - Doesn't introduce new information.

   → *If YES: likely MERGE*

2. **Is this a substantive response?**
   - Answers a question: "Yes," "No," "Three," "Tomorrow".
   - Indicates agreement/disagreement with semantic content.
   - Completes a request or command: "I will," "Done," "Sure".

   → *If YES: likely KEEP*

3. **Look at the SURROUNDING CONTEXT:**
   - If the adjacent speaker was mid-sentence, this is likely a backchannel.
   - If the adjacent speaker asked a question or paused expectantly, this is likely substantive.

DECISION RULES

- **MERGE:** This is a backchannel that doesn't represent meaningful turn-taking.
- **KEEP:** This is a substantive contribution that should remain separate.

OUTPUT FORMAT (STRICT JSON)

```
{
  "action": "MERGE" | "KEEP",
  "confidence": <float between 0 and 1>,
  "reasoning": "<brief explanation of your decision>"
}
```

Figure 5: Prompt template for Short Turn Ambiguity Detection.