Retrieval as Repair Memory for State-Grounded Operational Dialogue

Eusebio Urrutia Conde
Universidad Nacional de Educación a Distancia
Spain

Abstract

In state-grounded operational information systems, a good detector already catches most unsafe dialogue turns. Catching the turn is only half of the job, though: the system still has to issue a repair that is precise and that an operator can audit. We present RADC, a retrieval-guided repair method for dialogue that is tied to operational variables like callsigns, headings, medication names, dosages, patient identifiers, and timing constraints. RADC calls the retriever only after an intervention has been triggered, so detection accuracy stays the same. The retriever picks a prior failure-repair pattern that matches the current error type and state slots, the pattern is adapted into a targeted clarification, and a deterministic safety filter then blocks any operational value that the trusted state does not support. We test RADC on ATC-Interact and Med-Interact with five-seed challenge suites that cover in-distribution, balanced-error, rare-error, and high-noise splits. In all eight domain–split cells, RADC uses the same detection signal as the State-Aware Auditor, yet it cuts 7.3 to 21.4 points from the critical error rate (CER), adds 21.9 to 24.8 points of clarification precision, and brings severity-weighted safety loss down from 0.30–0.92 to 0.014–0.120. On real UWB-ATCC transcript windows with controlled corruptions, the CER reduction stays between 13.5 and 14.7 points per seed. A manual audit indicates that, after safety filtering, retrieval makes repairs more slot-specific and easier to trace. Before filtering, some medical generations add unsupported detail, and this is the reason we include the deterministic slot-support check. The remaining human-preference gap concerns verbosity rather than correctness or safety.

CCS Concepts

• **Information systems** → **Information retrieval**; **Question answering**; • **Computing methodologies** → *Neural networks*; *Machine learning*.

Keywords

case-based retrieval, clarification generation, retrieval-augmented generation, repair memory

1 Introduction

In safety-critical settings, fluency and plausibility are not enough to judge a dialogue system. In air traffic control (ATC) or clinical medication coordination, a well-formed response is still unsafe if it disagrees with the current operational state: numeric values, entity references, timing, and procedural constraints must stay aligned with that state at every turn. Losing this alignment is a major source of operational incidents, yet most existing approaches cover only one part of the problem. Detection-oriented methods decide *whether* an utterance is problematic, and retrieval-augmented methods improve answer content. The question that matters most for safety stays open: once an inconsistency is suspected, what *repair action* should the system take?

In this paper we study that question, focusing on a regime that we think gets too little attention. In state-grounded settings, a reasonably strong state-aware classifier already detects most problematic turns, and the commonly used held-out splits saturate, so detection accuracy alone leaves little room to show progress. However, considerable safety risk can remain when the system's clarifications are generic, poorly targeted, or vague about the operational values involved. Our hypothesis is that most of this risk can be recovered if retrieval serves as case memory for *choosing the repair action.*

Approach. We propose RADC (Retrieval-Augmented Detected-inconsistency Clarification, Figure 1), with three parts: (i) a state-aware inconsistency auditor that shares its detection signal with the strongest non-retrieval baseline; (ii) a hybrid retriever over a knowledge base of prior failure cases, which scores candidates on semantic similarity, structured-state compatibility, and predicted mismatch type; and (iii) a clarification module that adapts the retrieved repair pattern to the current dialogue and state. The retriever only runs after an intervention is triggered, so it cannot inflate detection numbers; its only job is to make the clarification slot-targeted, state-specific, and traceable.

Evaluation. Building a convincing test bed turned out to be hard, and four design choices follow from that. First, we build a **challenge suite** for ATC-Interact and for Med-Interact, each with four splits: in-distribution (ID); Balanced-Error, which has no majority class; Rare-Error, with only ∼10% correct cases and mostly entity/patient confusion and procedural deviations; and High-Noise, which keeps the gold labels but heavily perturbs the surface forms. Second, we run **five regenerated seeds** ({2026, 2027, 2028, 2029, 2030}) with $n_{\text{train}} = 1{,}200$ and $n_{\text{test}} = 600$ per seed. This gives 3,000 pooled test cases per split; we report paired bootstrap p-values and release per-seed 95% confidence intervals. Third, we add a **severity-weighted safety protocol** in which high-severity errors cost more. Fourth, we add an **automatic external validation** pipeline that injects controlled corruptions into real UWB-ATCC transcript windows. This allows evaluation on real transcript language without manual relabelling.

Contributions. Our contributions are the following:

- We cast state-grounded dialogue repair as a post-detection decision problem, where retrieval helps select the repair action instead of supporting detection or answer generation.
- We propose RADC, which combines shared state-aware detection, type-aware and state-compatible retrieval, constrained clarification adaptation, and a deterministic state-consistency filter.

- We build challenge suites for ATC-Interact and Med-Interact with in-distribution, balanced-error, rare-error, and high-noise splits, and an automatic protocol on real UWB-ATCC transcript windows.
- We evaluate repair quality on two domains, four splits, and five seeds with residual critical error rate, clarification precision, severity-weighted safety loss, and high-severity residual rate.
- We find that, with detection held fixed, retrieval-conditioned repair reduces unresolved operational risk in every tested setting, compared with a strong non-retrieval auditor.

Figure 1 gives an overview of the RADC pipeline. Retrieval does not bring in external domain knowledge; it only chooses prior repair cases once the auditor has flagged a state–utterance mismatch.

Main finding. RADC cannot beat the State-Aware Auditor on detection accuracy, since the two systems use the same detection signal. Still, RADC lowers the critical error rate in all eight domain/split cells, most of all on the rare-error and high-noise splits, where generic clarification is least reliable. Because the detector is identical, any measured gain has to come from retrieval-conditioned repair.

2 Problem Formulation

We treat safety-critical dialogue as a sequential decision problem whose main requirement is *state consistency*: at each turn, the system decides whether the utterance agrees with a trusted structured state, whether to intervene, and how to repair with minimal disruption.

Let a dialogue be a sequence of utterances $D = (u_1, u_2, \ldots, u_t)$, and let S_{t-1} be the trusted operational state before utterance u_t is processed. The state is defined over the domain variables needed for safe execution. In ATC these are callsign, altitude, heading, speed, and the readback flag; in the medical domain they are patient identifier, medication, dosage, unit, frequency, route, and timing. Given $D_{<t}$, the current utterance u_t, and the trusted pre-turn state S_{t-1}, the system makes three coupled decisions.

1) Inconsistency detection. The system predicts a turn-level label $y_t \in \{\texttt{correct}, \texttt{incorrect}, \texttt{ambiguous}\}$. The non-correct labels cover numeric mismatch, entity/patient confusion, procedural deviation, and ambiguity with respect to S_{t-1}.

2) Intervention decision. The system chooses an intervention action $a_t \in \{\texttt{accept}, \texttt{clarify}\}$. This decision trades the risk of a missed intervention against the cost of interrupting without need.

3) Clarification generation. If $a_t = \texttt{clarify}$, the system produces a clarification $q_t = g(D_{\le t}, S_{t-1}, \mathcal{K})$, where $\mathcal{K}$ is a repository of prior failure cases. The clarification should be short, specific to the state, and aimed at the most likely source of the inconsistency.

Operational risk. We define the turn-level operational risk as $R_t = \mathbb{E}[\ell(S_t^\star, \hat{S}_t)]$, where $S_t^\star$ is the intended post-turn state and $\hat{S}_t$ is the state that is acted on after the system decision and any clarification. The goal is to minimize the cumulative risk $\sum_t R_t$. Detecting a problem does not reduce risk on its own; risk goes down only when repair brings the dialogue back in line with the state.

3 Retrieval-Guided Clarification after Detection

3.1 Overview

RADC has four components: (i) a state-aware encoder, (ii) an inconsistency detector that drives the intervention decision, (iii) a hybrid retriever over the failure-case knowledge base $\mathcal{K}$, and (iv) a clarification adapter that turns a retrieved case into a targeted repair. Retrieval and adaptation run *only on intervention*, so they cannot hide detection errors or inflate detection accuracy.

Given the dialogue history $D_{\le t}$ and the trusted state S_{t-1}, RADC estimates the inconsistency probability p_t and sets $a_t = \texttt{clarify}$ if $p_t \ge \tau_{\text{acc}}$. In that case, the system retrieves the top-k most similar prior failure cases, conditioned on dialogue, state, and predicted mismatch type, and selects and adapts a slot-targeted repair. If retrieval confidence is low, or if the top retrieved cases disagree on the mismatch type, the system falls back to a generic confirmation prompt. In contrast to standard retrieval-augmented generation, we do not inject external facts; we look for prior repair patterns that are operationally close to the current inconsistency. The knowledge base thus acts as a case memory, and keeping it consistent and current is a knowledge-base management problem in its own right [12].

3.2 State-Aware Encoding

The dialogue history is encoded as $h_D = \text{Enc}_D(D_{\le t})$ with all-MiniLM-L6-v2, a Sentence-BERT encoder [20]. The structured state is represented as $e_S = \sum_{k=1}^{m} \phi_k(S_{t-1}^k)$, where ϕ_k is a learned embedding for categorical slots and a normalized projection for numeric ones. The fused representation $h_t = \text{Fusion}(h_D, e_S)$ is shared between the detector and the retriever.

3.3 Inconsistency Detection and Intervention

The detector outputs a full posterior over the three labels,

$$p(y_t \mid D_{\le t}, S_{t-1}) = \text{softmax}(W_c h_t + b_c), \tag{1}$$

trained with cross-entropy $\mathcal{L}_{\text{det}} = -\sum_t \log p(y_t^\star \mid D_{\le t}, S_{t-1})$. We define the *non-correct* probability as

$$p_t = p(y_t \in \{\texttt{incorrect}, \texttt{ambiguous}\} \mid D_{\le t}, S_{t-1}), \tag{2}$$

so `ambiguous` counts as a non-correct subtype that also triggers clarification. This keeps ambiguous turns from passing the accept/clarify gate unnoticed. The intervention rule is

$$a_t = \begin{cases} \texttt{accept}, & p_t < \tau_{\text{acc}} \\ \texttt{clarify}, & p_t \ge \tau_{\text{acc}}, \end{cases} \tag{3}$$

where τ_{acc} is tuned on validation data to minimize CER under a constraint on over-intervention. The State-Aware Auditor we compare against uses this same detection signal, so any CER difference between the auditor and RADC can be traced back to the repair stage.

3.4 Mismatch-Type Prediction and Hybrid Retrieval

We train a mismatch-type head $p(t_t \mid D_{\le t}, S_{t-1}) = \text{softmax}(W_{\text{type}} h_t + b_{\text{type}})$ with inverse-frequency-weighted cross-entropy

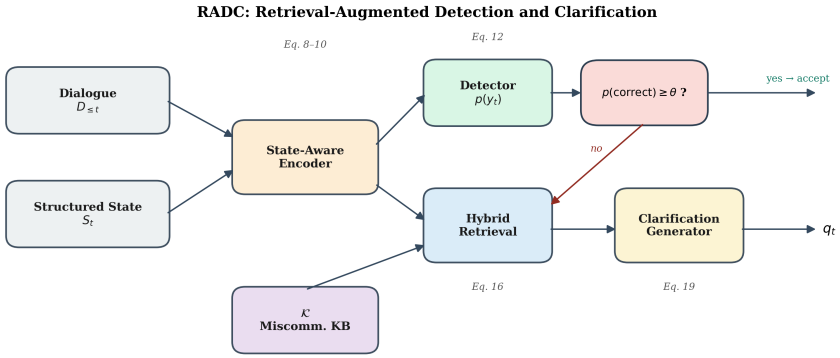

Figure 1: RADC architecture. A state-aware encoder fuses the dialogue and the structured state, and the detector decides between accept and clarify. When the decision is not accept, hybrid retrieval over the Miscommunication Knowledge Base $\mathcal{K}$ ranks candidates on semantic similarity, structured-state compatibility, and predicted mismatch type. The retrieved case is then adapted into a slot-targeted clarification. If confidence or type agreement is low, the system falls back to a generic confirmation prompt.

$$\mathcal{L}_{\text{type}} = -\sum_t w_{t^\star} \log p(t_t^\star \mid D_{\leq t}, S_{t-1})$$

For a query turn we retrieve top-k ($k = 5$) prior failure cases from $\mathcal{K}$ using the hybrid score

$$\text{sim}_{\text{hyb}}(x, x_i) = \lambda_1 \text{sim}_{\text{sem}}(x, x_i) + \lambda_2 \text{sim}_{\text{state}}(x, x_i) + \lambda_3 \text{sim}_{\text{type}}(x, x_i), \tag{4}$$

where sim_{sem} is the cosine similarity between dialogues, $\text{sim}_{\text{state}}$ is the structured-state compatibility on the slots that matter for the candidate mismatch, and sim_{type} is the predicted probability of the candidate's mismatch type. The hybrid score favors cases that are semantically similar and also operationally aligned with the current state and error pattern. Aggregating several similarity signals is usually more reliable than any single measure [11, 13, 14]; we keep a linear form because it is cheap and easy to inspect.

3.5 Contrastive Representation Learning

To separate turns more sharply in the embedding space, we add a supervised contrastive objective on h_t. For an anchor h, a positive h^+ has the same gold mismatch type and is state-compatible, and a negative h^- has a different mismatch type or a state-incompatible pattern:

$$\mathcal{L}_{\text{con}} = -\log \frac{\exp(\text{sim}(h, h^+)/\tau)}{\exp(\text{sim}(h, h^+)/\tau) + \exp(\text{sim}(h, h^-)/\tau)}. \tag{5}$$

3.6 Clarification Adaptation and Generation

Given the retrieved set $\mathcal{R}_t = \{r_1, \ldots, r_k\}$, RADC selects a clarification that is grounded in the retrieved evidence. We summarize the evidence as $c_t = \sum_{i=1}^k \alpha_i h_i$, with α_i obtained from the normalized hybrid retrieval scores, and feed it, together with the current dialogue and state, to a constrained FLAN-T5-small clarification generator. (Implementation details like model caching are in the artifact README.) Decoding uses beam search with a domain-aware logits processor that softly pushes token probabilities toward a whitelist of domain vocabulary (callsigns, operational verbs, medication names, units, frequencies). The whitelist comes from training data and domain lexicons and includes subword tokens of domain-relevant strings, which avoids the failure modes of hard masking under subword tokenization. The system falls back to a generic confirmation prompt in two cases: when

$$\max_{r_i \in \mathcal{R}_t} \text{sim}_{\text{hyb}}(x, r_i) < \tau_{\text{ret}}, \tag{6}$$

or when the top-m retrieved cases do not reach a majority on the mismatch type. The generated output also goes through length and well-formedness filters, and any failure leads to the generic prompt. This keeps the generator controllable and its outputs auditable.

3.7 State-Consistency Safety Filter

Retrieval-conditioned generation can make repairs more specific, but it should never add operational content that the trusted state does not support. For this reason, RADC runs a deterministic state-consistency filter before any generated clarification is emitted.

Let q_t be the generated clarification, S_{t-1} the trusted pre-turn state, and $\hat{m}_t$ the predicted mismatch type. The filter extracts every operational entity and value from q_t: callsigns, patient identifiers, medication names, numeric quantities, units, headings, altitudes, speeds, frequencies, routes, and timing expressions. The clarification is accepted only if each extracted item meets one of these conditions:

(1) it appears in the trusted state S_{t-1};
(2) it appears in the current utterance as the detected mismatched value *and* the surrounding wording follows a contrastive clarification template ("did you mean v_{state} instead of v_{utt}?"); a mismatched value can only be emitted inside such a contrastive frame, never as the recommended operational value;
(3) it is in the domain whitelist and does not bind to a concrete operational slot;
(4) it belongs to a fixed clarification template and adds no new operational value.

If any extracted value is unsupported, contradictory, or attached to the wrong slot, RADC drops the generated clarification and emits the State-Aware Auditor's clarification instead. In practice this turns

retrieval into a proposal mechanism, as opposed to free generation. Formally, the emitted clarification is

$$q_t^{\text{emit}} = \begin{cases} q_t^{\text{gen}}, & \text{Safe}(q_t^{\text{gen}}, S_{t-1}, \hat{m}_t) = 1, \\ q_t^{\text{aud}}, & \text{otherwise.} \end{cases} \tag{7}$$

All main experiments report results for the emitted clarification q_t^{emit}, and not for the raw generated clarification q_t^{gen}.

3.8 Cost and Inference Procedure

At inference time, RADC (i) encodes the dialogue history and the trusted pre-turn state, (ii) computes p_t, (iii) accepts if $p_t < \tau_{\text{acc}}$, (iv) otherwise retrieves the top-k failure cases with sim_{hyb}, (v) adapts a slot-targeted clarification, and (vi) falls back to a generic confirmation prompt if retrieval confidence or type agreement is low. Encoding is linear in the input length. Retrieval uses approximate nearest-neighbor indexing, which scales sub-linearly in $|\mathcal{K}|$. Clarifications are short in all cases, so generation is cheap. The whole pipeline runs on a CPU on commodity hardware.

4 Benchmarks

We evaluate miscommunication detection and repair on two domain-specific benchmarks, plus a challenge suite that we designed to stress-test systems beyond the standard held-out splits.

4.1 ATC-Interact

4.1.1 Source corpora. We use three corpora: **ATCO2-test-set-1h** [3], **UWB-ATCC** [22], and **ATCOSIM** [5]. UWB-ATCC and ATCOSIM serve for training and evaluation, and ATCO2 gives us an additional out-of-corpus test set.

4.1.2 Construction. None of the source corpora is annotated for miscommunication. We extract the structured state, infer speaker roles, pair controller–pilot exchanges, and apply controlled, type-coded perturbations (numeric inconsistency, ambiguity, entity confusion, procedural deviation) to obtain labels and clarification targets.

Why controlled perturbations are necessary. The source corpora contain realistic operational language but no gold annotations for unsafe turns, target repairs, or post-repair outcomes, so a purely observational evaluation cannot isolate repair quality. Controlled, type-coded perturbations give us cases where the inconsistency type, affected slot, gold label, and target clarification are known. We do not estimate real-world incident frequency; we test whether a system can detect and repair known safety-relevant failure modes under realistic dialogue and state structure.

4.1.3 Dataset structure. Each instance contains the dialogue history $D_{\le t}$, the sequence of structured states $(S_1, \ldots, S_t)$, the label y_t, the intervention decision a_t, the clarification q_t (when needed), and a corrected response u_{t+1}.

4.1.4 Knowledge base. We build a repository $\mathcal{K}$ of failure cases (dialogue snippet, structured state, error type, clarification pattern) using training data only. Retrieval can then use semantic and operational similarity without any leakage from the test set.

Leakage control. Knowledge-base entries come only from training instances. No test instance, test perturbation, or variant derived from a test instance is ever added to $\mathcal{K}$. Splits do not overlap at the instance level or at the perturbation-variant level. Thus, retrieval can reuse repair patterns learned from training failures, in the spirit of case-based reasoning [1], but cannot retrieve the target clarification of the evaluated test case.

4.2 Med-Interact

We build **Med-Interact** from MedDialog-EN [24]. Clinical dialogue is tied to a structured medication-administration state, and controlled perturbations are applied on top. Each instance is $(C, I, R, S, y, t, Q^\star)$, where C is the preceding dialogue, I an instruction, R a response, S the structured state (patient, medication, dosage, unit, route, frequency, timing), y correctness, t the error type, and $Q^\star$ the target clarification. Every instruction has one correct response and several incorrect variants, generated through controlled perturbations of four error types (dosage inconsistency, ambiguity, patient confusion, procedural deviation). As in ATC, retrieval uses a knowledge base built from training data only.

4.3 Challenge Suite

Any reasonably strong state-aware detector saturates the original held-out splits. To get more convincing evidence, we generate a four-way challenge suite for each domain (Table 2):

- **ID test** (n=600 per seed): roughly the same mix as training, with ∼70% correct cases.
- **Balanced-Error test** (n=600 per seed): no majority class; correct cases drop to ∼25%.
- **Rare-Error test** (n=600 per seed): only ∼10% correct cases, mostly entity/patient confusion and procedural deviations (the highest-severity error families).
- **High-Noise test** (n=600 per seed): keeps the gold labels but heavily perturbs the surface forms.

Aspect	Design
Real data	ATC: real ATC transcripts (UWB-ATCC, ATCOSIM, ATCO2); Med: MedDialog-EN dialogue plus clinical schema
Perturbations	Controlled, type-coded modifications of numeric values, entities, and slots (altitude, dosage, unit, timing, callsign, patient ID)
Error validity	Perturbations match domain failure types (readback errors [4], dosage mismatches, patient swap, procedural deviation)
Splits	ID, Balanced-Error, Rare-Error, High-Noise; non-overlapping at instance and variant level
KB isolation	$\mathcal{K}$ built only from training data; no test cases
External data	Auto-external protocol injects controlled corruptions into real UWB-ATCC transcript windows
Goal	Evaluate detection and repair under realistic structure with no leakage and no easy majority class

Table 1: Dataset construction and evaluation protocol.

Split	Correct (%)	High-Severity (%)	n_{pooled}
ATC			
ID test	69.8	12.8	3,000
Balanced-Error	26.9	33.4	3,000
Rare-Error	8.8	51.6	3,000
High-Noise	25.4	36.3	3,000
Med			
ID test	69.8	12.6	3,000
Balanced-Error	25.9	36.2	3,000
Rare-Error	10.0	51.6	3,000
High-Noise	25.6	36.5	3,000

Table 2: Class composition of the challenge suite, pooled over the five regenerated seeds (600 test cases per seed, 3,000 pooled per split). The rare-error split over-samples high-severity errors on purpose, and most of its cases are errors.

4.4 Auto-External Protocol

We also evaluate on real ATC transcripts with controlled, type-coded corruptions of the final turn. The gold labels come from the corruption rule and not from human adjudication. We sample 5,000 transcript candidates and build 600 controlled instances per seed: each instance starts with a real two-utterance transcript window and ends with a deterministic, type-coded final exchange. The protocol tests repair on real transcript language with synthetic final-turn corruptions. It does not measure natural incident frequency, which would need independent human adjudication. The manifest, raw metrics, and bootstrap results are in the artifact.

5 Experimental Setup

5.1 Baselines

We compare RADC with a uniform set of state-aware baselines. All of them run from a single entry point, so that no baseline can end up silently weaker than the others on the detection side.

- **Always Accept**: never intervenes. It is our no-intervention reference point and an upper bound on residual operational risk.
- **Dialogue Heuristic**: a state-blind baseline that relies on surface cues, namely digit, callsign, and patient mismatches and dosage cues.
- **State-Aware Auditor**: our strong non-retrieval baseline. It uses the same detection signal as RADC (state-checked readback and medication-confirmed flags, plus type-targeted clarification templates), but no retrieval. Its detection accuracy is identical to that of RADC in every cell, so any difference in CER, CP, or ESG comes from repair. Its templates follow standard good practice, so we compare retrieval with a competent generic repair, not a strawman.
- **RADC (ours):** state-aware detection plus hybrid retrieval and slot-targeted clarification adaptation.

5.2 Implementation Details

The dialogue encoder is `all-MiniLM-L6-v2`. The state variables are, for ATC: callsign, altitude (ft), heading (deg), speed (kt), and readback flag; for Med: patient ID, medication, dosage (mg), unit, frequency, route, timing, and medication-confirmed flag. We set the contrastive temperature to $\tau = 0.07$ and the retrieval size to $k = 5$, and the mismatch-type loss uses inverse-frequency class weights. We train with AdamW, with learning rate 2×10^{-4} and weight decay 10^{-4}. Clarifications are generated with FLAN-T5-small using beam search and a domain-aware logits processor; the domain whitelist comes from training-data surface forms and domain lexicons. We avoid a larger backbone on purpose: our goal is to test whether retrieval-grounded repair adds safety value on top of a strong detector, not to maximize detector capacity. The small models also keep inference and reproduction CPU-feasible.

5.3 Evaluation Metrics

Metric	Measures	Main role
Detection	label accuracy	auditor strength
CER	unresolved critical errors	residual op. risk
CP	usefulness of clarifications	repair quality
OIR	clarification on correct cases	interruption cost
WLoss	severity-weighted residual loss	high-risk failures
HSR	unresolved high-severity cases	severe residual risk

Table 3: Evaluation metrics and the role each one plays in our safety argument.

We report:

- **Detection Accuracy**: exact match against `misc_label`.
- **Critical Error Rate (CER)** (↓): residual operational risk on non-correct cases, after the system decision and the quality of the clarification are taken into account.
- **Clarification Precision (CP)** (↑): mean usefulness of the issued clarifications, scored with a fixed rubric. A correct but *generic* on-topic confirmation gets partial credit (0.75), and a correct *slot-targeted* repair gets full credit. This floor is deliberately generous to the non-retrieval baselines: flagging the right turn earns most of the credit, and only the top of the scale is reserved for repairs that name the inconsistent slot. The State-Aware Auditor emits a fixed type-targeted template, so its CP sits near this floor as a consequence of its design, not as a penalty.
- **Over-Intervention Rate (OIR)** (↓): fraction of clarifications issued on gold-correct cases.
- **End-to-End Safety Gain (ESG)** (↑): relative reduction of the baseline critical risk (`Always Accept`).
- **Weighted Safety Loss** (↓, severity-weighted): the residual loss is the severity times the unresolved-error indicator, plus a 0.25 penalty for each over-intervention on a gold-correct case. ATC severity weights: numeric/entity confusion 5.0, procedural 3.0, ambiguity 2.5. Med: dosage/patient 5.0, procedural 3.0, ambiguity 3.0.
- **Safety Risk Reduction (SRR)** (↑): relative reduction of the weighted loss compared with `Always Accept`.
- **High-Severity Residual Rate (HSR)** (↓): fraction of high-severity errors still unresolved after the system decision.

All challenge-suite metrics are means over five seeds (per-seed CIs and raw metrics are in the artifact). For the RADC-vs.-State-Aware

comparisons we use a paired bootstrap with B=1,000 resamples per seed.

6 Main Results

Our main empirical result is that RADC does not mainly improve detection. What it does is make detected inconsistencies actionable, and this lowers the critical risk that is left after detection.

6.1 Repeated-Seed Challenge Results

In Table 4 we hold detection fixed on purpose. RADC and the State-Aware Auditor have the same detection front-end, so detection accuracy is equal in every cell and the remaining differences come from the repair stage. This setup is conservative, since no repair gain can be explained through a stronger classifier. The question is whether retrieval turns an already-correct intervention into a better repair action. To check that the answer does not depend on the CP rubric, we use two further signals: the severity-weighted protocol (Table 7), which scores repairs under a different loss, and the human audit (Section 7.2, Table 13), which does not use the CP rubric at all. Table 4 covers the two domains and all challenge splits, averaged over five seeds.

Figure 2 plots the main effect in Table 4. RADC keeps the detection behavior of the State-Aware Auditor, but the residual critical error is much lower in all challenge splits and in the two domains. The absolute CER reduction is smallest in the ID setting and largest on the rare-error splits, where generic clarification is least reliable; balanced-error and high-noise are in between. Retrieval helps most where a generic template struggles to produce a slot-targeted repair.

6.2 Retrieval Utility

The State-Aware Auditor and RADC share the same detection signal, so the gap between them measures the effect of retrieval on repair. Table 5 reports this gap together with paired bootstrap significance.

6.3 Illustrative Examples

Table 6 shows two typical cases where the State-Aware Auditor detects a problematic turn but then produces a generic repair. We include them to make the difference between detection-only intervention and retrieval-guided repair concrete.

6.4 Severity-Weighted Safety Protocol

One possible concern is that RADC's gains might not hold under a metric where high-severity errors carry a larger residual loss and generic clarifications get only partial credit. Table 7 reports the severity-weighted safety protocol (weighted safety loss, safety risk reduction, and high-severity residual rate), averaged over five seeds. Table 8 gives the paired bootstrap significance for the gaps in weighted loss and high-severity residual rate. The worst per-seed p-value over the whole grid is $< 10^{-3}$.

Domain	Split	ΔWLoss	worst p	ΔHSR	worst p
ATC	ID	−0.297	$< 10^{-3}$	−15.07	$< 10^{-3}$
ATC	Balanced	−0.699	$< 10^{-3}$	−35.07	$< 10^{-3}$
ATC	Rare	−0.836	$< 10^{-3}$	−41.53	$< 10^{-3}$
ATC	Noise	−0.660	$< 10^{-3}$	−31.27	$< 10^{-3}$
Med	ID	−0.288	$< 10^{-3}$	−12.87	$< 10^{-3}$
Med	Balanced	−0.705	$< 10^{-3}$	−29.73	$< 10^{-3}$
Med	Rare	−0.838	$< 10^{-3}$	−34.43	$< 10^{-3}$
Med	Noise	−0.651	$< 10^{-3}$	−21.27	$< 10^{-3}$

Table 8: Paired bootstrap significance for the safety protocol. Worst p-value over five seeds, B=1,000 resamples per seed.

6.5 Auto-External Validation

Table 9 reports the per-seed gap in CER and weighted loss when controlled corruptions are appended to real UWB-ATCC transcript windows. The CER reduction is stable across seeds, in the range $[-14.7, -13.5]$pp, with paired bootstrap $p < 10^{-3}$ for every seed.

Seed	ΔCER	p	ΔWLoss	p
2026	−14.275	$< 10^{-3}$	−0.5925	$< 10^{-3}$
2027	−13.900	$< 10^{-3}$	−0.5828	$< 10^{-3}$
2028	−14.250	$< 10^{-3}$	−0.6079	$< 10^{-3}$
2029	−13.500	$< 10^{-3}$	−0.5760	$< 10^{-3}$
2030	−14.700	$< 10^{-3}$	−0.6173	$< 10^{-3}$

Table 9: Auto-external validation on real UWB-ATCC transcript dialogue with controlled gold labels. RADC vs. State-Aware Auditor; per-seed paired bootstrap, B=1,000.

7 Analysis

We now examine where the gains come from, how they change with knowledge-base coverage, which components matter, and whether the effect holds over multiple turns and under heavy noise.

7.1 Where Do the Gains Come From?

Two observations point to the source of the improvement. First, RADC and the State-Aware Auditor use exactly the same detection signal: their per-seed detection accuracies match for every (domain, split, seed), and the per-seed CIs in the artifact confirm this. Second, the CER gap and the CP gap move together almost one-to-one across splits. When retrieved cases adapt cleanly into a slot-targeted clarification, CP goes up and CER goes down; when retrieval confidence drops (as in the high-noise split), CP and ESG go down slightly together. We conclude that the improvement comes from the repair stage and not from some hidden detection effect.

Knowledge-Base Scaling. Table 10 shows that CER decreases monotonically as the KB grows, and detection accuracy stays about the same. This also argues against a detector-capacity explanation: adding cases to the memory improves the emitted repair and leaves the accept/clarify gate essentially unchanged. This argues for treating the failure-case memory as a maintained knowledge base with regular curation [12].

Domain	Split	Model	Detection ↑	CER ↓	CP ↑	OIR ↓	ESG ↑
ATC	ID test	Always Accept	68.70	31.30	0.00	0.00	0.00
		State-Aware Auditor	99.47	8.22	75.00	0.00	73.73
		RADC (ours)	**99.47**	**0.58**	**99.84**	**0.00**	**98.15**
	Balanced-Error	Always Accept	25.33	74.67	0.00	0.00	0.00
		State-Aware Auditor	97.93	20.22	75.00	0.00	72.93
		RADC (ours)	**97.93**	**2.28**	**99.71**	**0.00**	**96.96**
	Rare-Error	Always Accept	10.53	89.47	0.00	0.00	0.00
		State-Aware Auditor	97.13	24.52	75.00	0.00	72.60
		RADC (ours)	**97.13**	**3.17**	**99.65**	**0.00**	**96.46**
	High-Noise	Always Accept	26.37	73.63	0.00	0.00	0.00
		State-Aware Auditor	96.10	21.33	75.00	0.00	71.03
		RADC (ours)	**96.10**	**4.36**	**99.35**	**0.00**	**94.09**
Med	ID test	Always Accept	69.53	30.47	0.00	0.00	0.00
		State-Aware Auditor	98.33	7.62	74.65	0.46	75.00
		RADC (ours)	**98.33**	**0.34**	**98.43**	**0.46**	**98.89**
	Balanced-Error	Always Accept	24.57	75.43	0.00	0.00	0.00
		State-Aware Auditor	96.67	18.86	74.97	0.05	75.00
		RADC (ours)	**96.67**	**1.05**	**98.57**	**0.05**	**98.62**
	Rare-Error	Always Accept	9.47	90.53	0.00	0.00	0.00
		State-Aware Auditor	95.47	22.63	75.00	0.00	75.00
		RADC (ours)	**95.47**	**1.44**	**98.41**	**0.00**	**98.41**
	High-Noise	Always Accept	24.63	75.37	0.00	0.00	0.00
		State-Aware Auditor	92.73	18.84	74.80	0.26	75.00
		RADC (ours)	**92.73**	**2.27**	**96.73**	**0.26**	**96.99**

Table 4: Repeated-seed challenge results (means over five seeds, n_{test}=600 per seed; per-seed 95% CIs are in the artifact). RADC has the same detection accuracy as the State-Aware Auditor in every (domain, split) cell, and it lowers CER 7.3–21.4pp and raises CP 21.9–24.8pp.

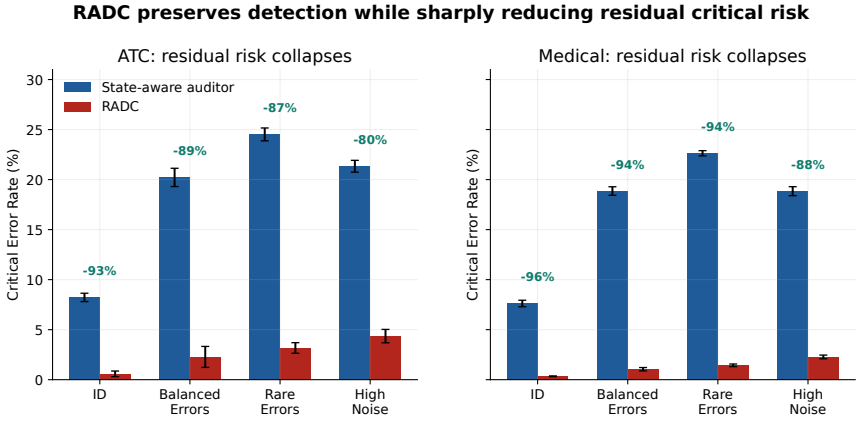

Figure 2: RADC lowers the residual critical error rate and keeps the same detection signal as the State-Aware Auditor. The reduction holds in the two domains and in all challenge splits, including the rare-error and high-noise settings.

Ablation Study. Table 11 reports ablations on the medical ID split. The results agree with the main experiments. Removing retrieval causes the largest loss in CER and CP. Removing state encoding leaves CER almost unchanged, but CP collapses.

Multi-Turn Repair. A single well-targeted clarification recovers most of the safety margin (Table 12). On Med-Interact, the strongest single-shot baseline ends at a CER of 4.15. RADC reaches 0.05 after one clarification turn, stays at 0.05 after two, and reaches 0.0 after three.

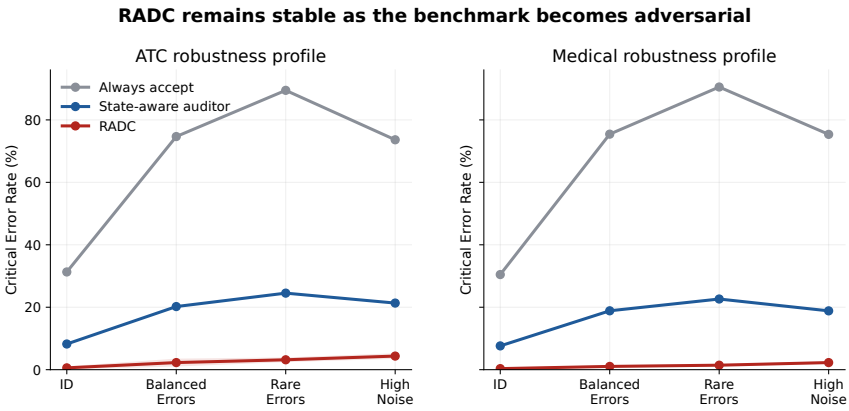

Figure 3: Residual risk across challenge splits. RADC stays low-risk as the benchmark moves from ID to the balanced-error, rare-error, and high-noise settings; the non-retrieval baselines keep a much higher residual critical error.

Domain	Split	ΔCER	ΔCP	ΔESG	worst p
ATC	ID test	−7.64	+24.84	+24.42	$< 10^{-3}$
ATC	Balanced-Error	−17.94	+24.71	+24.03	$< 10^{-3}$
ATC	Rare-Error	−21.35	+24.65	+23.86	$< 10^{-3}$
ATC	High-Noise	−16.98	+24.35	+23.06	$< 10^{-3}$
Med	ID test	−7.28	+23.78	+23.89	$< 10^{-3}$
Med	Balanced-Error	−17.81	+23.61	+23.62	$< 10^{-3}$
Med	Rare-Error	−21.20	+23.41	+23.41	$< 10^{-3}$
Med	High-Noise	−16.57	+21.93	+21.99	$< 10^{-3}$

Table 5: RADC vs. State-Aware Auditor. Detection accuracy is the same for the two systems, so the gap comes from retrieval-conditioned repair. The worst per-seed paired bootstrap p-value (over 5 seeds) on CER or CP is below 10^{-3}.

Behavior under distribution shift. Figure 3 shows that the gap between RADC and the non-retrieval baselines grows on the rare-error and high-noise splits, where surface forms are least informative and matching on state slots and mismatch type helps most.

Safeguards. Three safeguards limit inflated gains: the main comparator shares RADC's detection signal; the knowledge base contains no test cases; and the auto-external protocol uses real transcripts with controlled labels, so it supports validation on real data but not claims about natural incident rates.

7.2 Human Validation Study

We manually audited 200 repair decisions, half from ATC and half from the medical domain. Each clarification received a score on a 0–2 scale: 0 for wrong, unsafe, or misleading repairs; 1 for partially useful but generic or incomplete repairs; and 2 for correct, slot-specific, and operationally useful repairs. The audit uses the raw generator output q_t^{gen}, before the state-consistency safety filter of Section 3.7, so it measures the retrieval-conditioned clarification module in isolation. The main results in Tables 4–9, in contrast, use the emitted post-filter clarification q_t^{emit}.

The audit shows that retrieval-conditioned generation gives far more fully slot-specific clarifications than the non-retrieval auditor: 43% of RADC's pre-filter outputs received Score=2, against 0% for the State-Aware Auditor. The pattern holds in ATC (48% vs. 0%) and Med (38% vs. 0%). In Med, however, the mean score of pre-filter RADC is lower than the auditor's. Retrieval sometimes adds unsafe detail (unsupported medication names, contradictory dosage wording, unsupported numeric values), so the raw generator is not always safer than the auditor. These failures led us to add the safety filter of Section 3.7. The pre-filter audit is thus only diagnostic; all quantitative claims use post-filter outputs, which are at least as conservative as the auditor's.

8 Discussion

Why retrieval helps after detector saturation. In every setting we tested, adding retrieval barely changes detection accuracy but changes residual CER, CP, weighted loss, and high-severity residual rate a lot. This matches the formulation in Section 2: retrieval selects and adapts a prior repair pattern that targets the slot or value that disagrees with the state. On Med-Interact, where we ran the multi-turn experiment, most of the safety gain comes from the first clarification turn. This is desirable in safety-critical settings, where unnecessary back-and-forth is itself a risk.

Transfer to other safety-critical domains. The method should apply wherever an operational structured state can be defined, including industrial control, logistics, railway dispatch, and emergency-response coordination. We have not tested these domains yet.

8.1 Limitations

Controlled labels rather than natural incident rates. ATC-Interact uses real transcripts with controlled corruptions, and Med-Interact uses MedDialog-EN dialogue with controlled slot-level perturbations. These benchmarks do not estimate the natural frequency or the full diversity of real ATC or medical miscommunication. Claims about natural incident rates would need expert adjudication.

Dom.	Split	State and utterance	Auditor clarification	RADC clarification
ATC	ID	**State:** Callsign: DLH42; altitude: 5000 ft. **Utterance:** DLH24 climbing 6000. **Gold issue:** callsign and altitude mismatch.	Please confirm the instruction.	Confirm callsign DLH42 and assigned altitude 5000 ft?
Med	Rare-Error	**State:** Patient: P017; medication: heparin; dose: 5 mg. **Utterance:** Give P071 heparin 50 mg. **Gold issue:** patient and dosage mismatch.	Please verify the medication order.	Confirm patient P017 and heparin dose 5 mg before administration?

Table 6: Illustrative repair examples. The auditor gives a valid but generic clarification; RADC uses the trusted structured state to target the mismatched slots.

Domain	Split	Model	Weighted Loss ↓	SRR ↑	HSR ↓	Clar. Quality ↑
ATC	ID test	State-Aware Auditor	0.3126	74.18	15.57	75.00
		RADC	**0.0158**	**98.70**	**0.50**	**99.84**
	Balanced-Error	State-Aware Auditor	0.7611	73.66	37.17	75.00
		RADC	**0.0622**	**97.85**	**2.10**	**99.71**
	Rare-Error	State-Aware Auditor	0.9227	73.45	44.57	75.00
		RADC	**0.0868**	**97.50**	**3.03**	**99.65**
	High-Noise	State-Aware Auditor	0.7808	72.41	35.83	75.00
		RADC	**0.1203**	**95.74**	**4.57**	**99.35**
Med	ID test	State-Aware Auditor	0.3027	74.97	14.77	74.65
		RADC	**0.0143**	**98.81**	**1.90**	**98.43**
	Balanced-Error	State-Aware Auditor	0.7507	75.00	36.97	74.97
		RADC	**0.0460**	**98.48**	**7.23**	**98.57**
	Rare-Error	State-Aware Auditor	0.9005	75.00	44.30	75.00
		RADC	**0.0628**	**98.26**	**9.87**	**98.41**
	High-Noise	State-Aware Auditor	0.7521	74.98	37.27	74.80
		RADC	**0.1007**	**96.66**	**16.00**	**96.73**

Table 7: Severity-weighted safety protocol (means over five seeds). Compared with the State-Aware Auditor, RADC has a lower weighted safety loss in every split, and its high-severity residual rate drops from 14.8–44.6 **to** 0.5–16.0**.**

KB Size	Detection Acc. ↑	CER ↓	CP ↑
ATC-Interact			
170	98.5	6.9	81.4
1,036	98.5	3.7	92.4
3,033	98.5	1.6	99.1
Med-Interact			
168	100.0	5.75	80.48
1,010	100.0	2.60	91.16
3,035	100.0	0.38	98.72

Table 10: KB scaling. Detection accuracy does not change; CER decreases and CP increases monotonically as $|\mathcal{K}|$ grows.

Detection signal shared with the strongest baseline. RADC and the State-Aware Auditor share the same detection front-end on purpose. This is the right control for isolating the effect of retrieval, but it also means that we make no claim of better detection than the strongest non-retrieval baseline. Our claim concerns repair quality only.

Variant	CER ↓	CP ↑	ESG ↑
Full RADC	**0.38**	**98.72**	**98.72**
w/o Retrieval	5.13	82.58	82.58
w/o State Encoding	0.39	29.06	98.68
w/o Contrastive Loss	1.98	93.27	93.27

Table 11: Ablations on Med-Interact (ID test). State encoding is needed for CP (operational specificity), and retrieval is needed for CER (residual safety).

Model	Turns	Final CER ↓
Single-shot baseline	1	4.15
RADC	1	0.05
RADC	2	0.05
RADC	3	0.00

Table 12: Multi-turn clarification on Med-Interact. Most of the gain comes from the first clarification.

Constrained generator. We kept the clarification generator small and constrained (FLAN-T5-small with soft token biasing) on purpose, since in safety-critical settings auditability matters more to

Model	Domain	n	Score	Score=2
RADC (pre-filter)	All	200	0.99	0.43
State-Aware	All	200	0.97	0.00
RADC (pre-filter)	ATC	100	1.12	0.48
State-Aware	ATC	100	0.96	0.00
RADC (pre-filter)	Med	100	0.86	0.38
State-Aware	Med	100	0.98	0.00

Table 13: Manual audit of the pre-filter raw generator outputs q_t^{gen} of RADC and of the State-Aware Auditor's clarifications. Score is the mean on a 0–2 scale. Score=2 marks a fully correct, slot-specific, and operationally useful clarification.

us than fluency. A natural next step would be to study larger constrained generators that keep state specificity and slot-grounded justification and remain auditable.

KB coverage. Rare, compound, or weakly signalled errors may be underrepresented in $\mathcal{K}$. The KB-scaling sweep shows that wider coverage keeps reducing residual risk. Once detection has saturated, ingesting new failure cases is probably the main lever for improvement. Ambiguity errors remain open: resolving an underspecified slot resembles word sense disambiguation, and similarity measures that use surrounding words to pick a sense [15] could help separate ambiguous cases from true value mismatches.

8.2 Ethical Aspects

This work aims to reduce miscommunication risk in safety-critical dialogue. The system is an assistive tool and should not replace human operators. Our results come from controlled benchmarks and do not justify unsupervised deployment; the auditable design is meant to keep operator review feasible.

9 Related Work

Our work sits at the intersection of conversational retrieval, case-based reasoning, retrieval-augmented generation, and state-grounded dialogue. Prior work covers parts of this problem, but, to the best of our knowledge, not retrieval-guided *repair after a detected inconsistency* in state-grounded dialogue, evaluated under hard, balanced, and noisy conditions.

Conversational retrieval and reformulation. Conversational search handles underspecified or history-dependent requests through query rewriting and retrieval-aware response selection [6, 10, 18, 19]. These methods improve retrieval in multi-turn settings, but they do not address safety-critical repair after a turn has become inconsistent with a structured state.

Clarification modeling. Research on clarification looks at when to ask questions and how to generate them [2, 21]. Work in web search and conversational IR has also explored template-guided and intent-aware clarification generation [23]. These approaches optimize intent resolution or interaction quality, and they rarely use retrieved prior failures to choose a repair action that restores the state.

RAG, evidence retrieval, and state-grounded dialogue. RAG and evidence retrieval ground generation in supporting documents or answer evidence [7–9]. In high-stakes domains like law, answers must be traceable to authoritative sources, and interpretability remains a weak spot of neural QA systems [16]. Domain-specific dialogue systems use structured state for slot tracking and operational constraints [17]. RADC combines these threads in a different way: it retrieves prior *failure cases* in order to select a repair action that brings the dialogue back in line with the state.

Similarity measurement and case retrieval. Case-based reasoning reuses solutions of past cases [1], so retrieval quality depends heavily on the similarity function. Different textual similarity measures capture different aspects of relatedness [11], and combining them through fuzzy aggregation [13] or genetic-programming ensembles [14] tends to beat any single measure. Our hybrid score follows this idea with signals of a different nature: text similarity, state compatibility, and mismatch type.

Positioning. RADC targets the detector-saturated regime, where further detection gains are small but the residual repair risk is still high. We see its novelty in the combination of four elements: state-aware detection shared with the strongest baseline, hybrid type-aware and state-compatible retrieval, slot-targeted clarification, and an evaluation that uses repeated seeds, severity weighting, and external transcript data. Prior work on clarification, conversational retrieval, and RAG retrieves information to support response selection, query refinement, clarification generation, or answer generation [6, 23]. RADC instead retrieves precedents of inconsistencies in order to decide how to repair.

10 Conclusion

In this paper we studied state-grounded dialogue after detection has saturated. State-aware auditors already catch most unsafe turns, but some risk remains when the repairs they issue are poorly targeted. RADC addresses this gap with retrieval-conditioned repair. Once the auditor triggers an intervention, RADC retrieves prior failure cases that match on semantic similarity, state compatibility, and mismatch type, and adapts the retrieved pattern into a slot-targeted clarification. Over two domains, four challenge splits, five regenerated seeds, and the auto-external protocol, RADC lowers CER 7.3–21.4pp, raises clarification precision 21.9–24.8pp, and reduces severity-weighted safety loss from 0.30–0.92 to 0.014–0.120 (an absolute reduction of 0.29–0.84), with paired bootstrap $p < 10^{-3}$ in every cell and seed. Since RADC and the State-Aware Auditor share the same detection signal, these gains come from the repair stage and not from better detection. Our results suggest that, in this setting, retrieval is most useful as operational memory for choosing the right clarification after an intervention, more than as a tool for generating answers.

Use of Generative AI

We used generative AI tools for language editing, proofreading, and code-completion assistance. The author reviewed and approved all technical claims, experiments, and analyses, and take full responsibility for the final manuscript.

References

[1] Agnar Aamodt and Enric Plaza. 1994. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications* 7, 1 (1994), 39–59.

[2] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval*. 475–484.

[3] Juan Pablo Zuluaga Gomez, Karel Veselỳ, Igor Szöke, Alexander Blatt, Petr Motlicek, Martin Kocour, Khalid Choukri, Iuliia Nigmatulina, Claudia Cevenini, Allan Tart, et al. 2024. ATCO2 corpus: a large-scale dataset for research on automatic speech recognition and natural language understanding of Air Traffic Control communications. *Journal of Data-centric Machine Learning Research* (2024).

[4] Hartmut Helmke, Matthias Kleinert, Shruthi Shetty, Oliver Ohneiser, Heiko Ehr, Hörður Arilíusson, Teodor S Simiganoschi, Amrutha Prasad, Petr Motlicek, Karel Veselỳ, et al. 2021. Readback error detection by automatic speech recognition to increase ATM safety. In *Proceedings of the fourteenth USA/Europe air traffic management research and development seminar (ATM2021), virtual event*. 20–23.

[5] Konrad Hofbauer, Stefan Petrik, and Horst Hering. 2008. The ATCOSIM Corpus of Non-Prompted Clean Air Traffic Control Speech.. In *LREC*, Vol. 3. 8.

[6] Kai Hua, Zhiyuan Feng, Chongyang Tao, Rui Yan, and Lu Zhang. 2020. Learning to detect relevant contexts and knowledge for response selection in retrieval-based dialogue systems. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 525–534.

[7] Yizheng Huang and Jimmy Xiangji Huang. 2024. A survey on retrieval-augmented text generation for large language models. *Comput. Surveys* (2024).

[8] Siqing Huo, Negar Arabzadeh, and Charles Clarke. 2023. Retrieving supporting evidence for generative question answering. In *Proceedings of the annual international acm sigir conference on research and development in information retrieval in the Asia Pacific region*. 11–20.

[9] Parastoo Jafarzadeh and Faezeh Ensan. 2025. An evidence-based approach for open-domain question answering. *Knowledge and Information Systems* 67, 2 (2025), 1969–1991.

[10] Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 5303–5315.

[11] Jorge Martinez-Gil. 2014. An overview of textual semantic similarity measures based on web intelligence. *Artificial Intelligence Review* 42, 4 (2014), 935–943. doi:10.1007/s10462-012-9349-8

[12] Jorge Martinez-Gil. 2015. Automated knowledge base management: A survey. *Computer Science Review* 18 (2015), 1–9. doi:10.1016/j.cosrev.2015.09.001

[13] Jorge Martinez-Gil. 2016. CoTO: A novel approach for fuzzy aggregation of semantic similarity measures. *Cognitive Systems Research* 40 (2016), 8–17. doi:10.1016/j.cogsys.2016.01.001

[14] Jorge Martinez-Gil. 2023. A Comparative Study of Ensemble Techniques Based on Genetic Programming: A Case Study in Semantic Similarity Assessment. *International Journal of Software Engineering and Knowledge Engineering* 33, 2 (2023), 289–312.

[15] Jorge Martinez-Gil. 2023. Context-Aware Semantic Similarity Measurement for Unsupervised Word Sense Disambiguation. *CoRR* abs/2305.03520 (2023). doi:10.48550/arXiv.2305.03520

[16] Jorge Martinez-Gil. 2023. A survey on legal question–answering systems. *Computer Science Review* 48 (2023), 100552. doi:10.1016/j.cosrev.2023.100552

[17] Raveesh Meena, José Lopes, Gabriel Skantze, and Joakim Gustafson. 2015. Automatic detection of miscommunication in spoken dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. 354–363.

[18] Fengran Mo, Abbas Ghaddar, Kelong Mao, Mehdi Rezagholizadeh, Boxing Chen, Qun Liu, and Jian-Yun Nie. 2024. CHIQ: Contextual history enhancement for improving query rewriting in conversational search. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 2253–2268.

[19] Akash Kumar Mohankumar, Bhargav Dodla, Gururaj K, and Amit Singh. 2023. Unified generative & dense retrieval for query rewriting in sponsored search. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 4745–4751.

[20] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*. 3982–3992.

[21] Zhengxiang Shi, Yue Feng, and Aldo Lipani. 2022. Learning to execute actions or ask clarification questions. In *Findings of the association for computational linguistics: NAACL 2022*. 2060–2070.

[22] Luboš Šmídl, Jan Švec, Daniel Tihelka, Jindřich Matoušek, Jan Romportl, and Pavel Ircing. 2019. Air traffic control communication (ATCC) speech corpora and their use for ASR and TTS development. *Language Resources and Evaluation* 53, 3 (2019), 449–464.

[23] Jian Wang and Wenjie Li. 2021. Template-guided clarifying question generation for web search clarification. In *Proceedings of the 30th ACM international conference on information & knowledge management*. 3468–3472.

[24] Guangtao Zeng, Wenmian Yang, Zeqian Ju, Yue Yang, Sicheng Wang, Ruisi Zhang, Meng Zhou, Jiaqi Zeng, Xiangyu Dong, Ruoyu Zhang, et al. 2020. MedDialog: Large-scale medical dialogue datasets. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*. 9241–9250.