When the Database Fails: Prompting LLM Dialogue Agents for Safe Recovery in Task-Oriented Dialogue

Mohammad Alijanpour Shalmani[1,*], **Alale Rezvani Boroujeni**[2], **Jiann Shiun Yuan**[1]

[1]College of Engineering and Computer Science, University of Central Florida, Orlando, Florida
[2]Department of Marketing, University of Central Florida, Orlando, Florida

Correspondence: alijanpour@ucf.edu

Abstract

Large language models used in task-oriented dialogue often produce fluent but unsafe responses when backend database calls fail, return empty results, or surface mismatched information, inventing venues, confirmations, or booking details not grounded in the database. We study a lightweight prompting-based recovery approach that improves robustness without retraining or additional model calls. We compare three response strategies, including a guided recovery prompt conditioned on structured database status, across six open-weight model families (DeepSeek-R1, Gemma-2, Llama-3, Mistral, Phi-3, and Qwen-2.5) and four database conditions: empty result, wrong-domain retrieval, API error, and clean retrieval. Using fault-injected benchmarks built on two structurally different datasets, MultiWOZ 2.2 (5 domains) and SGD (20 domains), we find that naive agents hallucinate on 30.5% of failure turns on MultiWOZ and 20.9% on SGD. Our Guided-Retry strategy reduces hallucination by 50% on MultiWOZ (30.5→15.3%) and by 42% on SGD (20.9→12.2%) without retraining. However, residual hallucination remains substantial (6–37% across models), with wrong-domain failures the hardest case. Results are consistent across both datasets and all six model families, and human annotation shows substantial agreement while supporting the validity of the automatic commitment-safety metric.

1 Introduction

Task-oriented dialogue (TOD) systems help users book hotels, find restaurants, and arrange transport. Modern LLM-based agents (Hudeček and Dušek, 2023) typically follow a standard pipeline: extract user intent, query a backend database, and generate a response grounded in the result. This pipeline is fragile at the database boundary. In practice, backend failures can still lead models to produce fluent but unsupported responses, including invented venues, confirmations, or booking details. Real deployments routinely encounter three failure modes: (1) empty result–no record matches the user's constraints; (2) wrong-domain returns–the database routes to an incorrect domain (Eric et al., 2020); (3) API errors–timeouts and server failures.

2 Related Work

Prior work has studied tool failure in LLM agents in broader settings. TRACE/SCOPE (Hou et al., 2025) identifies "Hallucination Fallback" when tools return empty results or time out. ReliabilityBench (Gupta, 2026) injects faults into general agent tasks but not TOD. PALADIN (Vuddanti et al., 2025) trains recovery policies over injected tool malfunctions. Non-collaborative simulators (Shim et al., 2025) stress-test agents in MultiWOZ under out-of-scope *user requests*, rather than backend execution faults.

To our knowledge, prior work does not provide a MultiWOZ- and SGD-grounded evaluation that: (i) injects runtime DB execution faults rather than user-behavior stressors; (ii) compares prompting-only recovery without retraining; and (iii) quantifies residual hallucination under structured guidance.

We make four contributions: (1) a controlled fault-injection framework over MultiWOZ 2.2 and SGD; (2) an evaluation of three prompting strategies across six models; (3) a set of automatic and human-validated metrics for failure recovery, including Commitment Safety Rate (CSR); and (4) the finding that structured prompting reduces, but does not eliminate, hallucination, leaving a residual rate of 6–37% across models. Code and prompts are available in our GitHub repository.[1]

[1]https://github.com/mohammad-AJP/llm-db-failure-recovery

Proceedings of the 27th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 815–820
August 2–5, 2026.

3 Method

3.1 Datasets and Fault Injection

We use the test splits of MultiWOZ 2.2 (Zang et al., 2020) (5 domains: hotel, restaurant, taxi, train, attraction) and SGD (Rastogi et al., 2020) (20 domains spanning restaurants, flights, hotels, events, media, and more). For each test dialogue, we extract the first user turn and synthetically inject one of four *DB execution conditions*, sampling 100 dialogues per condition (400 per dataset). We used the first user turn as a controlled diagnostic setting that isolates immediate recovery behavior under backend failure without additional confounds from longer dialogue context or multi-turn state tracking.

- **Clean**: DB returns a valid result. Sanity baseline.
- **Empty Result**: DB returns zero matches. Correct action: acknowledge failure and offer constraint relaxation.
- **Wrong Domain**: DB returns results from an incorrect domain. Correct action: detect mismatch and confirm with user.
- **API Error**: DB returns timeout or 503 error. Correct action: apologise and ask user to retry.

3.2 Recovery Strategies

For each model, all strategies use the same decoding setup (temperature 0.2) and differ only in their system prompt.

- **Naive**: raw DB response passed with no special instruction.
- **Inform**: model explicitly instructed not to hallucinate and to acknowledge failures honestly.
- **Guided-Retry**: model receives a structured decision procedure keyed on the DB `status` field, with explicit per-failure instructions and an explicit prohibition on fabricating venue names, booking references, or confirmation details.

In our controlled fault-injection setup, `wrong_domain` status is known because we inject the DB condition directly; therefore, it is not inferred by the LLM itself. In deployment, this signal would need to come from router metadata, API endpoint metadata, or a separate domain-consistency checker. We evaluate six open-weight model families from six organizations: DeepSeek-R1 (Guo et al., 2025), Gemma-2 (Gemma Team, Google DeepMind, 2024), Llama-3 (Meta Llama Team, 2024), Mistral (Jiang et al., 2023), Phi-3 (Abdin et al., 2024), and Qwen-2.5 (Qwen Team, 2024).

3.3 Metrics

We evaluate recovery quality using four complementary metrics that capture hallucination, action appropriateness, commitment safety, and user friction. HR (Hallucination Rate): fraction of non-clean turns where the agent fabricates a result absent from the DB response, detected via regex patterns. AAR (Appropriate Action Rate): fraction matching ground-truth correct action per failure type, assessed via keyword heuristics. CSR (Commitment Safety Rate): fraction of non-clean turns free of explicit false commitments (e.g., invented booking refs, confirmations, or other booking-style commitment language).

In the current automatic implementation, CSR is narrower than HR: it detects only explicit booking-style false commitments, so a response may increase HR without reducing CSR.

HR and CSR are automated heuristic diagnostics rather than exhaustive semantic hallucination judgments. HR captures unsupported-result patterns in failure-turn responses, while CSR_{auto} focuses more narrowly on explicit false commitment language. Table 2 summarizes representative patterns detected by each metric. These regex detectors are intentionally transparent and reproducible, but they may miss subtler unsupported claims; therefore, we validate CSR with nine human annotators rating 60 sampled responses, obtaining inter-annotator agreement of $\kappa = 0.7672$.

User Friction Score (UFS) is an automated per-response composite score on a raw 0–3 scale (lower = better), not a percentage and not a human-scored metric. For each response, we assign +2 if the HR detector flags hallucination (H_i), +1 if the CSR_{auto} detector flags an explicit false commitment (F_i), and +1 if a Silent failure (S_i) occurs, meaning that the DB status is a failure condition but the response does not acknowledge the failure with an apology or similar cue. The final score is capped at 3 and reported as the mean across evaluated examples as shown in Equation 1

$$\text{UFS}_i = \min(2H_i + F_i + S_i, 3). \quad (1)$$

Strategy	System-prompt content
NAIVE	"You are a task-oriented dialogue assistant helping users book hotels, restaurants, taxis, trains, and find attractions in Cambridge. You have queried the database and received the result below. Respond to the user based on the database result."
INFORM	"You are a task-oriented dialogue assistant helping users book hotels, restaurants, taxis, trains, and find attractions in Cambridge. You have queried the database and received the result below. Rules: - NEVER invent or hallucinate a venue, booking reference, or confirmation. - If the database returned no results, tell the user honestly. - If there was a database error, apologise and suggest trying again. - If a result was found, share it clearly and offer to book."
GUIDED-RETRY	"You are a task-oriented dialogue assistant helping users book hotels, restaurants, taxis, trains, and find attractions in Cambridge. You have queried the database and received the result below. Follow this exact procedure based on the database status field: - status=success: inform the user of the result, offer to proceed. - status=empty: tell user nothing matched. Identify ONE constraint to relax, e.g., area or price range, and ask if they want to broaden the search. - status=wrong_domain: state that the result appears to be from the wrong domain. Ask the user to confirm their request so you can retry correctly. - status=error: apologise briefly. Say the system is temporarily unavailable and ask the user to try again shortly. NEVER fabricate a booking reference, venue name, or confirmation detail."

Table 1: System prompts used by the three recovery strategies. All strategies receive the same user utterance, slot summary, and serialized DB response; they differ only in the system-prompt content shown here.

Metric	Examples of detected patterns
HR	Fabricated booking/reference number; "I found" or "I booked" on a failure turn; "confirmation number/code"; unsupported title-cased venue-like names when no DB name exists.
CSR_{auto}	Explicit commitment phrases such as "booked," "your reservation," "reservation confirmed," reference-code claims, or check-in confirmation wording.

Table 2: Examples of automated regex-based detectors. These are heuristic diagnostics, not exhaustive semantic annotations.

Statistical testing. For each model independently, we compare GUIDED-RETRY and NAIVE on matched fault-injected dialogue instances, where the same test turn under the same failure condition is evaluated under both strategies. For the binary metrics HR and AAR, we use McNemar's exact test. For UFS, we use a two-tailed paired t-test. We observe the same significance pattern across all six models; all reported comparisons remain significant at $p < 0.001$.

4 Results

4.1 Overall Comparison

Table 3 shows results on MultiWOZ. Guided-Retry consistently outperforms both baselines across all six models. Averaged across models, Naive achieves HR = 30.5% and AAR = 66.5%; Guided-Retry reduces HR by 50% to 15.3% and improves AAR to 83.0%. These improvements are statistically significant under our paired test setup ($p < 0.001$).

CSR remains near ceiling because it is intentionally narrower than HR in the current automatic implementation. HR captures broad unsupported fabrication, while CSR captures only explicit booking-style false commitments; therefore, many hallucinations counted by HR do not affect CSR.

DeepSeek-R1 achieves the lowest HR under Guided-Retry (5.8%). Phi-3 is the exception: Guided-Retry performs worse than Inform (HR 35.2 vs. 31.0), indicating that this model does not reliably follow structured recovery instructions.

Table 4 shows a similar pattern to MultiWOZ: Guided-Retry achieves the best AAR on all six models, improving the 6-model average from 58.7% to 83.9%, and reduces average HR from 20.9% to 12.2%. Phi-3 remains the main exception, with higher HR under Guided-Retry than Inform.

4.2 Breakdown by Failure Type

Figure 1 shows hallucination rates by failure type for MultiWOZ and SGD. Wrong-domain retrieval is the hardest failure case on both datasets. On MultiWOZ, Naive HR reaches 47.8% for wrong-domain inputs, and Guided-Retry still leaves 20.8% residual hallucination. On SGD, the same pattern holds, though at lower absolute rates: Naive HR is 31.0% for wrong-domain inputs, reduced to 17.0% under Guided-Retry. This suggests that models often adopt vocabulary from the incorrect domain result rather than detecting the mismatch.

API errors show the clearest benefit from structured recovery prompting. On MultiWOZ, Guided-

Strategy	Model	HR↓	AAR↑	CSR↑	UFS↓
Naive		37.8	51.7	**100.0**	1.04
Inform	DeepSeek-R1	21.0	56.0	99.8	0.58
Guided-Retry		**5.8**	**72.0**	**100.0**	**0.49**
Naive		12.5	65.8	**100.0**	0.29
Inform	Gemma-2	10.2	68.5	**100.0**	**0.22**
Guided-Retry		**6.8**	**75.0**	**100.0**	0.41
Naive		28.5	82.5	99.5	0.59
Inform	Llama-3	19.0	83.8	99.8	0.39
Guided-Retry		**16.2**	**99.8**	**100.0**	**0.35**
Naive		30.8	68.5	**100.0**	0.62
Inform	Mistral	20.8	80.0	99.2	0.42
Guided-Retry		**14.8**	**93.8**	**100.0**	**0.33**
Naive		44.5	68.2	98.8	0.90
Inform	Phi-3	**31.0**	73.0	99.2	**0.63**
Guided-Retry		35.2	**81.0**	**99.5**	0.71
Naive		28.7	62.3	99.5	0.59
Inform	Qwen-2.5	21.8	65.5	99.2	0.45
Guided-Retry		**13.2**	**76.5**	**100.0**	**0.30**

Table 3: MultiWOZ overall results. HR = Hallucination Rate (%), AAR = Appropriate Action Rate (%), CSR = automatic Commitment Safety Rate (%), UFS = User Friction Score. Best values per model and metric are in **bold**.

Strategy	Model	HR↓	AAR↑	CSR↑	UFS↓
Naive		33.2	47.5	99.8	0.96
Inform	DeepSeek-R1	15.2	53.0	**100.0**	**0.41**
Guided-Retry		**5.2**	**73.5**	99.8	0.46
Naive		7.2	55.8	**100.0**	0.28
Inform	Gemma-2	**5.5**	61.3	**100.0**	**0.22**
Guided-Retry		5.8	**81.0**	**100.0**	0.44
Naive		17.2	68.2	**100.0**	0.47
Inform	Llama-3	11.8	70.0	99.8	**0.25**
Guided-Retry		**9.5**	**97.8**	99.8	0.26
Naive		18.5	60.0	**100.0**	0.37
Inform	Mistral	11.8	64.5	99.8	0.24
Guided-Retry		**8.8**	**96.2**	99.8	**0.22**
Naive		32.2	62.3	98.8	0.66
Inform	Phi-3	**25.2**	66.5	98.8	**0.52**
Guided-Retry		37.0	**76.2**	**99.8**	0.76
Naive		16.8	58.2	99.5	0.51
Inform	Qwen-2.5	14.5	60.2	99.2	0.39
Guided-Retry		**6.8**	**78.5**	**100.0**	**0.32**

Table 4: SGD overall results. HR = Hallucination Rate (%), AAR = Appropriate Action Rate (%), CSR = automatic Commitment Safety Rate (%), UFS = User Friction Score. Best values per model and metric are in **bold**.

Retry reduces HR from 39.7% to 14.2%; on SGD, it reduces HR from 29.0% to 14.0%. Empty-result failures are also improved, though more modestly: on MultiWOZ, HR falls from 34.3% to 26.3%, while on SGD it falls from 23.0% to 18.0%.

On clean DB-return turns, our automatic detector flags no unsupported-result hallucinations (HR = 0.0%). This should not be read as evidence that LLM NLG is hallucination-free under normal conditions. Rather, it means that the specific failure patterns targeted by our detector–fabricated booking or reference language and unsupported venue assertions in the absence of a valid DB result–are not triggered when the DB returns a valid record. Our evaluation focuses on hallucinations induced by DB execution failures, not the full space of possible NLG hallucinations.

4.3 Cross-Dataset Generalizability

Table 5 shows results averaged across the six models on both datasets. The overall pattern is consistent across MultiWOZ-2.2 and SGD: Guided-Retry achieves AAR of 83.0% and 83.9%, respectively, confirming that the findings generalize across structurally different TOD benchmarks spanning 5 and 20 domains. HR is lower on SGD overall, possibly because SGD dialogues are more formulaic, making failure status easier to detect.

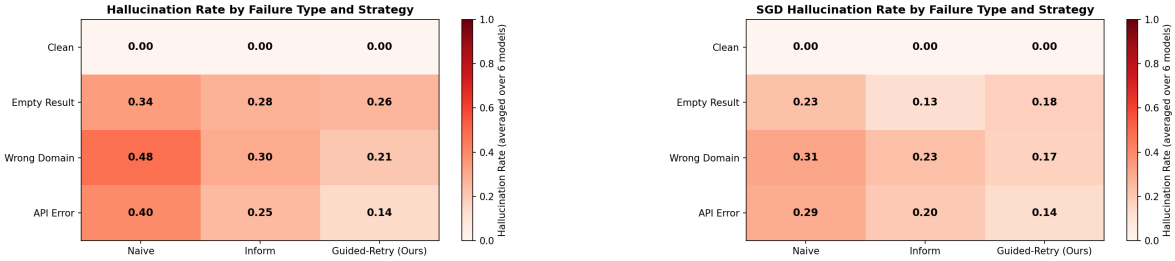
	MultiWOZ-2.2		SGD	
Strategy	**HR↓**	**AAR↑**	**HR↓**	**AAR↑**
Naive	30.5	66.5	20.9	58.7
Inform	20.6	71.1	14.0	62.6
Guided-Retry	**15.3**	**83.0**	**12.2**	**83.9**

Table 5: Cross-dataset comparison (6-model average).

4.4 Human Evaluation

To validate the commitment-safety metric, we conducted a human evaluation on 60 sampled responses spanning all three strategies and the three non-clean failure types. We used nine annotators drawn from different academic levels and backgrounds to provide a diverse annotation pool, and limited the survey to 60 items to balance coverage and annotation quality. The survey took approximately 1.5 hours per annotator to complete; a substantially longer survey would increase annotator fatigue and could reduce labeling precision. Annotators were blind to the underlying strategy and failure condition.

Annotators received clear instructions and example items before completing the survey. They were asked to assign a binary label (*false commitment* vs. *appropriate response*), where a false commitment was defined as: *"the agent explicitly claims a booking, reservation, confirmation, or specific DB-*



(a) MultiWOZ

(b) SGD

Figure 1: Hallucination rate (%) by failure type and recovery strategy. Wrong-domain retrieval is the hardest failure case on both datasets, while Guided-Retry consistently reduces hallucination relative to Naive and Inform.

backed result as true when the DB did not return one."

Inter-annotator agreement was strong, with Fleiss' $\kappa = 0.7672$ and 95.74% observed agreement. Overall human CSR was 90.0% (6/60 majority false commitments). The strategy ranking matched the automatic trend: Guided-Retry achieved 100.0% human CSR, compared with 89.47% for Inform and 80.95% for Naive. By failure type, human CSR was 100.0% for empty-result cases, 85.7% for wrong-domain cases, and 85.0% for API errors. These results support the validity of the automatic commitment-safety metric while confirming that Guided-Retry yields the safest behavior overall.

5 Discussion

Our results show that LLM-based TOD agents can default to confident, fluent responses even when the database explicitly reports failure, consistent with the broader tendency of LLM agents to favor fluency over accuracy (Baidya et al., 2025). The main practical finding is that a single structured system-prompt addition reduces hallucination by 42–50% without retraining, additional inference calls, or substantial engineering overhead.

The Phi-3 result is noteworthy: it is the only model for which Guided-Retry performs worse than Inform on HR. Phi-3 also has the highest Naive HR (44.5%), indicating that some models may not reliably benefit from structured recovery instructions even when those instructions help other model families.

Most importantly, residual hallucination remains non-trivial even under Guided-Retry. The best model (DeepSeek-R1) still hallucinates on 5.8% of failure turns, while the worst case (Phi-3) reaches 35.2%. This suggests that prompt-level recovery is helpful, but not sufficient on its own for robust TOD deployment.

6 Conclusion

We presented a controlled fault-injection study of LLM-based TOD agents under runtime DB failures across two benchmarks and six model families. Guided-Retry, a structured prompt-level recovery strategy, reduces hallucination by 42–50% without retraining, but residual hallucination of 6–37% remains, with wrong-domain retrieval as the hardest failure case.

Limitations

Our fault injection is synthetically constructed, and real-world backend failures may follow different distributions. Automated hallucination detection relies on heuristic patterns, and human evaluation covers only a subset of items. We evaluate instruction-tuned models at the 7–9B scale; larger or proprietary models may behave differently. Finally, we do not study multi-turn recovery dynamics after the user responds to a failure acknowledgement, which we leave for future work.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Avinash Baidya, Kamalika Das, and Xiang Gao. 2025. The behavior gap: Evaluating zero-shot LLM agents in complex task-oriented dialogs. In *Findings of ACL 2025*.

Mihail Eric, Rahul Goel, Shachi Paul, Adarsh Kumar, Abhishek Sethi, Peter Ku, Anuj Kumar Goyal, San-

chit Agarwal, Shuyang Gao, and Dilek Hakkani-Tur. 2020. MultiWOZ 2.1: A consolidated multi-domain dialogue dataset. In *Proceedings of LREC 2020*, pages 422–428.

Gemma Team, Google DeepMind. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Daya Guo, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, and 3 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Aayush Gupta. 2026. ReliabilityBench: Evaluating LLM agent reliability under production-like stress conditions. *arXiv preprint arXiv:2601.06112*.

Zhaoyi Joey Hou, Tanya Shourya, Yingfan Wang, Shamik Roy, Vinayshekhar Bannihatti Kumar, and Rashmi Gangadharaiah. 2025. Multi-faceted evaluation of tool-augmented dialogue systems. *arXiv preprint arXiv:2510.19186*.

Vojtěch Hudeček and Ondřej Dušek. 2023. Are large language models all you need for task-oriented dialogue? In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 216–228.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Meta Llama Team. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Qwen Team. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 8689–8696.

Jeonghoon Shim, Woojung Song, Cheyon Jin, Seungwon Kook, and Yohan Jo. 2025. Non-collaborative user simulators for tool agents. *arXiv preprint arXiv:2509.23124*.

Sri Vatsa Vuddanti, Aarav Shah, Satwik Kumar Chittiprolu, Tony Song, Sunishchal Dev, Kevin Zhu, and Maheep Chaudhary. 2025. PALADIN: Self-correcting language model agents to cure tool-failure cases. *arXiv preprint arXiv:2509.25238*.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. MultiWOZ 2.2: A dialogue dataset with additional annotation corrections. In *Proceedings of the 2nd Workshop on NLP for Conversational AI*, pages 109–117.