Hybrid AI for Responsive Multi-Turn Online Conversations with Novel Dynamic Routing and Feedback Adaptation

Priyaranjan Pattnayak[1], Amit Agarwal[1], Hansa Meghwani[2],
Hitesh Laxmichand Patel[1], Srikant Panda[2]

[1]OCI, Oracle America Inc., [2]OCI, Oracle India
Correspondence: priyaranjan.pattnayak@oracle.com

Abstract

Retrieval-Augmented Generation (RAG) systems and large language model (LLM)-powered chatbots have significantly advanced conversational AI by combining generative capabilities with external knowledge retrieval. Despite their success, enterprise-scale deployments face critical challenges, including diverse user queries, high latency, hallucinations, and difficulty integrating frequently updated domain-specific knowledge. This paper introduces a novel hybrid framework that integrates RAG with intent-based canned responses, leveraging predefined high-confidence responses for efficiency while dynamically routing complex or ambiguous queries to the RAG pipeline. Our framework employs a dialogue context manager to ensure coherence in multi-turn interactions and incorporates a feedback loop to refine intents, dynamically adjust confidence thresholds, and expand response coverage over time. Experimental results demonstrate that the proposed framework achieves a balance of high accuracy (95%) and low latency (180ms), outperforming RAG and intent-based systems across diverse query types, positioning it as a scalable and adaptive solution for enterprise conversational AI applications.

1 Introduction

Recent progress in NLP has drastically changed the landscape of conversational AI, and among such new state-of-the-art solutions, a class of Retrieval-Augmented Generation (RAG) systems has emerged. By combining large language models (LLMs) with separate information retrieval pipelines, RAG systems can generate contextually rich and factually grounded responses, which are necessary for knowledge-intensive applications (Lewis et al., 2020). However, enterprise-scale conversational AI systems often face real-world challenges such as diverse user query patterns, varying levels of query complexity, and stringent latency requirements for seamless user experiences. High computational costs, susceptibility to hallucinations when retrieval is misaligned, and inefficiencies in managing frequently updated domain-specific knowledge further compound these challenges, particularly in dynamic environments like customer support (Sanh et al., 2020b; Rocktäschel et al., 2020). In practice, ensuring that such systems can scale while maintaining accuracy and low latency remains an industry pain point.

In contrast, classical intent-based chatbots are efficient in processing frequently asked questions (FAQ) and other predictable queries, thanks to using pre-defined responses. Their lightweight computational footprint and scalability also make them well-suited for high-confidence, domain-specific scenarios (Serban et al., 2017; Shah et al., 2018). However, these systems are inherently rigid and often struggle with query diversity, especially when faced with ambiguous or context-dependent user interactions. In high-demand enterprise settings, the inability of intent-based systems to adapt quickly to evolving user needs or handle complex multi-turn dialogues (Shah et al., 2018; Zhao, 2020) results in inconsistent user experiences and increased operational overhead for manual updates. The inability to balance adaptability with efficiency underscores the need for hybrid systems that synergize the strengths of RAG and intent-based approaches.

In order to solve these challenges, we propose a novel hybrid framework that combines RAG systems with intent-based canned responses for dynamic, multi-turn customer service interactions. While prior works have explored combining RAG and intent-based systems independently, our contribution lies in a cohesive framework that not only integrates these elements but also introduces a dynamic confidence-based routing mechanism refined through user feedback. This mechanism ensures that query routing decisions are continuously optimized based on real-time user interactions, enabling a system that evolves and adapts without manual intervention. Additionally, our framework addresses scalability challenges by efficiently balancing computational resources, making it particularly suited for enterprise-scale applications where latency and accuracy are paramount. Our approach utilizes a dynamic query routing mechanism that evaluates the intent confidence level of user queries:

- *High-confidence queries* are resolved using predefined canned responses to ensure low latency and computational efficiency.
- *Low-confidence or ambiguous queries* are routed to the RAG pipeline, enabling contextually enriched responses generated from external knowledge.

The framework is further enhanced with a dialogue context manager, keeping track and managing evolving intents across multiple turns, ensuring consistent and coherent interactions. Additionally, a feedback loop continuously refines the intent repository, adapting to emerging user needs and expanding response coverage over time. Our system is designed to meet enterprise latency standards, delivering responses within an acceptable threshold (sub-200ms latency and high turn efficiency), thereby ensuring user engagement and satisfaction in real-time applications(Pattnayak et al., 2024).

Our Contributions This work makes the following key contributions:

1. **Hybrid Conversational Framework:** We propose a novel architecture which combines RAG systems with intent-based canned responses; the queries are routed dynamically for optimizing response latency and computational cost without compromising accuracy.
2. **Multi-Turn Dialogue Management:** We introduce a dialogue context manager which can track the evolving user intents and guarantee coherence in responses over multiple turns, thus addressing a key gap in the current systems.
3. **Feedback-Driven Adaptability:** Our framework incorporates a feedback loop to enable continuous refinement of intents, canned responses and confidence thresholds, thereby improving system adaptability and coverage for real-world applications.
4. **Comprehensive Evaluation:** Extensive experiments on synthetic and real-world datasets demonstrate significant improvements in accuracy, latency, and cost efficiency compared to state-of-the-art baselines.
5. **Real-World Applicability:** Our framework is designed for enterprise-scale deployment, handling diverse user queries efficiently, from repetitive FAQs to complex knowledge-based questions, while adhering to industry latency standards critical for user retention.

By addressing key challenges faced by enterprise conversational AI systems, such as query diversity, dynamic knowledge updates, and real-time latency requirements, our proposed framework offers a scalable, adaptive, and efficient solution. This work advances task-oriented dialogue systems, particularly in domains where multi-turn interactions and dynamic knowledge management are essential for operational success.

2 Related Work

2.1 Retrieval-Augmented Generation (RAG)

Recent advancements in RAG have enhanced contextual retrieval and generative capabilities, improving incident resolution in IT support (Isaza et al., 2024), question-answering systems, and domain-specific chatbots (Veturi et al., 2024). Research on noise handling (Cuconasu et al., 2024) and reinforcement learning (Kulkarni et al., 2024) further optimizes RAG for precision and adaptability in complex applications. By retrieving relevant documents during inference, RAG systems mitigate common LLM challenges such as hallucinations and outdated knowledge (Lewis et al., 2020; Sanh et al., 2020b). These systems are particularly effective for knowledge-intensive tasks where accuracy and factual grounding are critical.

Despite their effectiveness, RAG systems face significant challenges, including high computational costs and latency due to the dual retrieval and generation processes. Enterprise settings also pose unique challenges, such as diverse user queries, latency constraints, and evolving domain knowledge needs (Lewis et al., 2020; Pattnayak et al., 2025). Moreover, most existing RAG systems are optimized for single-turn interactions and struggle with maintaining coherence in multi-turn di-

Approach	Strengths	Weaknesses	Multi-Turn Support	Feedback Adaptation
RAG Systems	Accurate, dynamic responses	High latency, computationally expensive	Limited	No
Intent-Based Chatbots	Efficient, low latency	Rigid, poor adaptability	No	No
Hybrid RAG-Intent Systems	Balance between efficiency and flexibility	Limited multi-turn and feedback mechanisms	Partial	No
Proposed Framework	Low latency, multi-turn adaptable	Scalability challenges	Yes	Yes

Table 1: Comparison of Existing Approaches and the Proposed Framework.

alogues, where evolving user intents require dynamic retrieval and contextual adaptation (Rocktäschel et al., 2020). Recent efforts to optimize RAG pipelines, such as multi-stage retrieval systems (Lee et al., 2020) and model distillation (Sanh et al., 2020b), have reduced latency but do not address the complexities of multi-turn interactions (Sanh et al., 2020a).

2.2 Intent-Based Chatbots

Intent-based chatbots work well for predictable, high-confidence queries by mapping user inputs to predefined intents. These systems are widely used in domains like customer support, where they efficiently handle FAQs and repetitive queries with minimal computational overhead (Serban et al., 2017; Shah et al., 2018). However, their reliance on predefined intents limits their adaptability to ambiguous or evolving queries, particularly in multi-turn conversations (Michelson et al., 2020; Friedrich et al., 2020).

Recent developments have involved the inclusion of transformer-based models to enhance intent recognition and increase coverage (Michelson et al., 2020). However, these methods are resource-heavy, as they require a lot of labeled data and computational resources, which makes scalability quite limited for dynamic domains.

2.3 Hybrid Approaches

Hybrid retrieval systems integrating lexical search (e.g., BM25 (Robertson and Walker, 1994)) and semantic search (e.g., dense embeddings via FAISS (Douze et al., 2024)) effectively balance speed and semantic depth (Agarwal et al., 2025), improving retrieval accuracy (Mitra et al., 2021; Hernandez et al., 2020).

In conversational AI, hybrid approaches integrating RAG with intent-based responses have emerged to address limitations in single-mode systems by enhancing flexibility and efficiency (Bordes et al., 2020). Prior works, such as (Gao et al., 2020b; Zhao, 2020; Patel et al., 2024), have explored blending retrieval-augmented pipelines with canned responses to improve response efficiency and contextual depth. However, these systems are primarily designed for single-turn interactions and do not address the complexities of multi-turn dialogues, where query context evolves dynamically (Agarwal et al., 2024a). While existing research relies on static threshold-based routing, the integration of adaptive threshold driven routing and response generation for real-time, multi-turn applications remains an under explored area with significant potential for optimization.

2.4 Positioning of This Work

While prior research has advanced RAG systems, intent-based chatbots, and hybrid architectures, key limitations remain. RAG systems excel in generating contextually rich responses but struggle with coherence in multi-turn conversations, high latency, and computational costs (Lewis et al., 2020; Rocktäschel et al., 2020). Intent-based chatbots are efficient but lack flexibility for ambiguous or evolving queries in dynamic settings (Serban et al., 2017; Agarwal et al., 2024b). Hybrid systems balance efficiency and adaptability but often fail to track dialogue context or refine responses dynamically based on user feedback (Gao et al., 2020a). Table 1 summarizes the key differences between the existing work and our proposed framework.

This work addresses real-world challenges by proposing a hybrid framework that integrates RAG systems with intent-based canned responses. It uses dynamic query routing to handle high-confidence queries efficiently with canned responses while relying on RAG pipelines for complex cases. A dialogue context manager ensures coherence in multi-turn interactions, and a real-time feedback loop

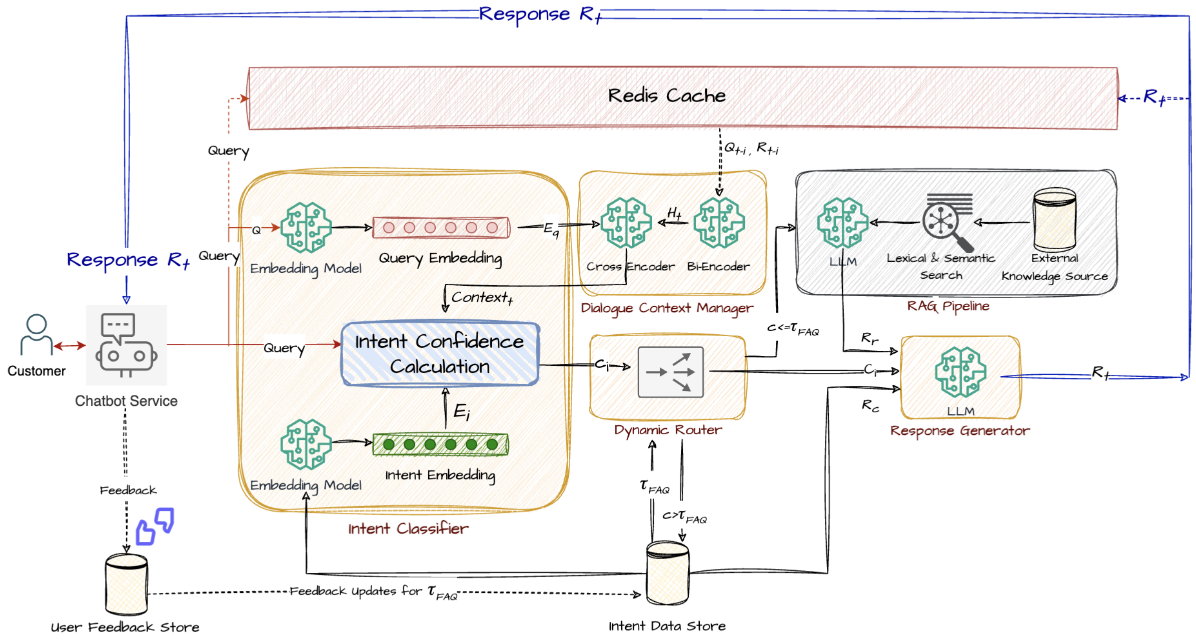


Figure 1: High-level Architecture of the Hybrid Framework.

enables continuous refinement of intents, thresholds and canned responses. For instance, in an enterprise customer support setting, our system efficiently handles high-frequency queries such as, *"How do I reset my password?"* using canned responses with minimal latency (under 200ms), ensuring quick resolution for routine tasks. In contrast, more complex queries such as, *"Can you help me troubleshoot a payment gateway integration issue with API X?"* are dynamically routed to the RAG pipeline, leveraging external documentation and past incident reports to generate accurate responses. This adaptability is further evident when users provide feedback on response quality, prompting the system to refine its intent classification and adjust confidence thresholds for future queries. Unlike existing systems that either focus on single-turn interactions or static routing and struggle with multi-turn dialogue management, our framework continuously adapts to diverse queries and user needs, optimizing latency and scalability.

By focusing on these critical aspects, this framework advances the state-of-the-art in task-oriented dialogue systems, particularly for enterprise-scale applications where efficiency, scalability, and adaptability are paramount.

3 Proposed Framework

The proposed framework integrates the efficiency of intent-based canned responses with the contextual richness and adaptability of Retrieval-Augmented Generation (RAG) systems (Lewis et al., 2020; Gao et al., 2020c). By dynamically routing queries based on intent confidence and leveraging user feedback for adaptive refinement, the framework addresses latency, accuracy, and scalability challenges while maintaining coherence across multi-turn interactions. Figure 1 illustrates the architecture with key modules, data flow and a Redis Cache which stores frequently accessed intents and responses for faster retrievals.

3.1 Key Modules

The framework comprises the following key components, each designed to address specific challenges in multi-turn dialogue systems:

Intent Classifier. The Intent Classifier utilizes a fine-tuned BERT model (Devlin et al., 2019) to encode user queries into semantic embeddings extracted from last layer of the model. See Appendix A.3 for datatset detail. Confidence scores (c) are calculated by comparing the query embedding with predefined intent embeddings: Based on c, the query is classified as:

- $c > 0.85$: **FAQ (Canned Response)**.
- $0.5 < c \leq 0.85$: **Contextual**.
- $c \leq 0.5$: **Out-of-Domain**.

The above thresholds are default for the system which are updated based on the user-feedback on

Algorithm 1 Context-Aware Intent Confidence Calculation

```
Require: Query $Q$, Set of Intent Embeddings $\{E_1, E_2, \ldots, E_n\}$, Historical Context Embeddings $\mathrm{H}_t$
Ensure: Highest Confidence Score $c$, Corresponding Intent: $\text{Intent}_{\text{max}}$
Step 1: Calculate Query Embedding
$\mathrm{E}_q \leftarrow \mathrm{BERT}(Q)$
Step 2: Contextual Query Embedding
$\text{Context}_t \leftarrow \phi(E_q, \mathrm{H}_t)$        ▷ Augment query embedding with historical context
Step 3: Confidence Calculation
for each intent embedding $E_i$ in $\{E_1, E_2, \ldots, E_n\}$ do
    $c_i \leftarrow \text{CosineSimilarity}(\text{Context}_t, E_i)$        ▷ Compute similarity for intent $i$
end for
Step 4: Find Best Match
$c \leftarrow \max(c_i)$        ▷ Highest confidence score
$\text{Intent}_{\text{max}} \leftarrow \text{argmax}_i(c_i)$        ▷ Intent corresponding to $c$
Output: $c$, $\text{Intent}_{\text{max}}$
```

the fly. Algorithm 1 provides the pseudo-code for the classification process, which incorporates historical context from the Dialogue Context Manager.

Dialogue Context Manager. The module tracks dialogue history using embeddings of prior queries and responses, stored in a sliding window. For multi-turn interactions, historical context embeddings are computed dynamically:

$$\mathrm{H}_t = \psi\big(\{(Q_{t-i}, R_{t-i}) \mid i = 1, ..n\}\big)$$

where ψ represents a bi-encoder (in-house architecture) that computes the embeddings by appending prior context, queries, and responses into a string. Q_{t-i} and R_{t-i} represents previous query and corresponding responses within a chat session. The aggregated historical context H_t is then used to compute the current contextual query embedding:

$$\text{Context}_t = \phi(E_q, \mathrm{H}_t)$$

Here, ϕ represents a lightweight transformer block (in house cross-encoder) to compute attention, E_q is the current query embedding. Relevant historical context embedding is appended to the current query embedding for downstream processing.

Dynamic Routing. The module checks the confidence (c) of the classified intent: $\text{Intent}_{\text{max}}$, against the threshold (τ_{FAQ}) of the particular intent in the Intent Data Store. τ_{FAQ} for each intent is dynamically updated with user-feedback.

Response Generator. The module refines the final response to user by either blending the static canned responses (R_c) with dynamic RAG outputs (R_r) using a language module or directly passing the R_c or R_r to the user based on the Dynamic Router.

Feedback Mechanism. Explicit (ratings) and implicit (e.g., query refinements) feedback is logged and used to refine thresholds, intents, and response mappings. New intents are created for recurring unhandled queries. Specifically, recurring unhandled queries are logged and grouped based on semantic similarity. When a threshold number of similar unresolved queries is reached in a group, the system automatically flags for creation of a new intent and response. Explicit user feedback is collected via a post-response prompt in the chat interface, allowing users to rate responses positive or negative (thumbs up or thumbs down), which dynamically updates the system's confidence thresholds every 100 interactions.

3.2 Workflow

The framework integrates query classification, response routing, multi-turn handling, and feedback adaptation into a cohesive workflow:

Query Classification. Queries are classified into types (FAQ, Contextual, or Out-of-Domain) based on the confidence score c from the Intent Classifier and the threshold τ_{FAQ} & $\tau_{\text{Out-of-Domain}}$ for each intent, stored in the Intent Data Store, which is dynamically updated with the user feedback. The classification logic is as follows:

- *FAQ:* If $c > \tau_{\text{FAQ}}$, the query is resolved using a predefined canned response for the intent.
- *Out-of-Domain:* If $c \leq \tau_{\text{Out-of-Domain}}$, the

query is routed exclusively to the RAG pipeline for domain-specific response generation.

- *Contextual:* If $\tau_{\text{Out-of-Domain}} < c \leq \tau_{\text{FAQ}}$, the query is processed by both canned responses for the intent and the RAG pipeline. The Response Generator then combines the outputs.

Response Routing. The final response for the user is based on the query classification. The response generation varies by query type:

1. *Canned Response (FAQ):* The predefined the canned response for the intent is passed directly to the user for rapid resolution.
2. *RAG Response (Out-of-Domain):* The RAG output is passed as is, ensuring the most contextually rich response for undefined intents.
3. *Hybrid Response (Contextual):* Both the canned response and the RAG output are retrieved and combined into a unified response using a language model (LLM):

$$R_f = \text{LLM}(c \cdot R_c, (1-c) \cdot R_r),$$

where c is the confidence of the $\text{Intent}_{\text{max}}$, passed to the LLM in to the prompt to ensures coherence and contextual alignment in the final response.

Multi-Turn Handling. Context tracking ensures coherence in multi-turn interactions by retrieving and appending the most relevant embeddings from dialogue history.

Feedback-Driven Adaptability. User feedback dynamically influences system thresholds and intent mappings. The threshold for FAQs (τ_{FAQ}) is adjusted based on feedback trends, ensuring that frequently misclassified queries are handled appropriately. The update mechanism follows::

$$\tau_{\text{FAQ}} = \tau_{\text{FAQ}} + \lambda \cdot (\text{NFR} - \text{PFR}),$$

where:

- **NFR**: Negative Feedback Rate.
- **PFR**: Positive Feedback Rate.
- λ: Scaling factor controlling the sensitivity of the adjustment.
- τ_{FAQ}: By default is set to 0.85 whenever the intents (and dependent intents) are updated in intent data store.

High negative feedback increases the threshold, reducing the likelihood of misclassification as FAQs, while positive feedback reduces the threshold to favor FAQ classification. Threshold for Out-of-Domain queries ($\tau_{\text{Out-of-Domain}}$) is kept constant at 0.5 to prevent over-restricting or over-generalizing OOD classification. This adaptive threshold mechanism ensures that the system remains responsive to user feedback while maintaining stability in query classification. Further details are provided in Appendix A.1

3.3 Prototype Implementation

The framework is implemented as a modular system using microservices:

- **Frontend:** Built with React.js for user interaction and feedback collection (Contributors, 2023).
- **Backend:** Flask microservices handle query classification, retrieval, and feedback processing (Grinberg, 2018).
- **Storage:** OCI (Oracle Cloud Infrastructure) Opensearch stores canned responses & external knowledge base, while FAISS and dense embeddings support retrieval (Karpukhin et al., 2020).
- **Memory Cache:** A memory-augmented module maintains embeddings of prior interactions in OCI Cache (Managed Redis), allowing the system to retain relevant historical context across multiple dialogue turns.
- **Model Deployment:** Models (e.g., BERT, Encoder, Cross-Encoder, GPT-3 & other proprietary LLMs) are deployed using in-house architecture and OCI Gen AI Service for scalability.

4 Experiment and Results

The hybrid framework was evaluated on four metrics: accuracy, response latency, cost efficiency, and turn efficiency. These evaluations spanned in-house datasets of live customer queries, and scalability tests. Table 2 summarizes overall results, while Table 5 in the appendix provides category-wise performance.

4.1 Experimental Setup

The evaluation dataset comprised 10,000 queries, categorized as :- a) *Predefined FAQ Queries (40%)*:

High-confidence queries resolved via canned responses, b) *Contextual Queries (30%)*: Queries requiring both canned & RAG responses, and c) *Out-of-Domain Queries (30%)*: Undefined intents handled exclusively by RAG pipeline.

For multi-turn interactions, 20% of queries included follow-ups designed to assess context retention. Scalability tests evaluated performance with dataset sizes up to 50,000 queries, preserving category proportions. Results are shown in Table 3.

Evaluation Metrics The system was assessed using the following metrics:

- **Accuracy**: Percentage of correctly resolved queries across predefined FAQs, contextual queries, and out-of-domain scenarios. Accuracy is a fundamental evaluation metric in retrieval-based and generative NLP models (Karpukhin et al., 2020; Lewis et al., 2020), ensuring that responses align with the intended knowledge base. We determine accuracy using a cosine similarity metric, as used in prior works on retrieval-based QA systems (Reimers and Gurevych, 2019). For *Predefined FAQ*, the framework has to fetch the correct FAQ, leading to a 100% cosine similarity. For *Contextual* and *Out-of-Domain Queries*, the generated resposne needs to be similar (90%) to annotated ground truth answer.

- **Response Latency**: Average response time in milliseconds taken to generate responses. Response latency is crucial in real-time conversational AI applications, as delays directly impact user experience (Shuster et al., 2021). Faster response times enhance engagement, making this metric essential for evaluating system efficiency.

- **Cost Efficiency (CE)**: A normalized measure of cost efficiency, defined as:

$$\mathrm{CE} = \min\left(1, \frac{\mathrm{Latency}_{\mathrm{baseline}}}{\mathrm{Latency}_{\mathrm{proposed}}} \times \frac{\mathrm{Accuracy}_{\mathrm{proposed}}}{\mathrm{Accuracy}_{\mathrm{baseline}}}\right)$$

  Inspired by cost-aware NLP evaluations (Tay et al., 2023), this metric balances accuracy and latency trade-offs. It ensures that the proposed framework maintains or improves accuracy while reducing computational costs, a key factor in large-scale AI deployment.

- **Turn Efficiency**: Average number of turns required to resolve a query in a conversation:

$$\text{Turn Efficiency} = \frac{\text{Total Turns}}{\text{Resolved Queries}}$$

  Turn efficiency measures conversational conciseness, ensuring that the system minimizes unnecessary back-and-forth interactions (Serban et al., 2017). A lower number of turns per resolved query indicates a more efficient dialogue system, reducing user dissatisfaction and operational overhead.

4.2 Results and Analysis

Overall Performance. Table 2 compares the proposed framework with baseline systems. Our proposed framework achieves a balance of high accuracy (95%) and low latency (180ms), outperforming the canned-response system and the RAG pipeline's accuracy. The proposed system reduces the chances of hallucination for the most common user queries by leveraging canned responses hence outperforming accuracy of just RAG pipeline's.

Category-Specific Insights. Table 5 (Appendix A.2) highlights performance variations across query types:

- **FAQs**: Similar accuracy compared to the canned-response system, with a 82% reduction in latency compared to RAG Pipeline.

- **Contextual Queries**: Accuracy improved over 47% compared to canned-response system, with over 50% reduction in latency compared to RAG Pipeline with similar accuracy.

- **Out-of-Domain Queries**: The RAG pipeline and our proposed framework exceed the baseline intent-based system's accuracy by over 85%, as intent systems default to fallback responses for out-of-domain queries.

Scalability. The hybrid framework's scalability was evaluated under query loads ranging from 1,000 to 50,000. We observed graceful performance degradation under increasing query loads. Accuracy remains within enterprise-grade thresholds (92% at 50,000 queries), while latency increases proportionally due to retrieval bottlenecks. Table 3 summarizes the results, demonstrating the

Framework	Accuracy (%) (↑)	Response Latency (ms) (↓)	Cost Efficiency (↑)	Turn Efficiency (↓)
Canned-Response (Baseline)	53	**68**	**1.0**	**NA**
RAG Pipeline	91	380	0.3	2.3
Proposed Framework	**95**	180	0.7	1.7

Table 2: Evaluation Results for Canned-Response (Intent) Systems, RAG, and Proposed Frameworks. Metrics represent averages across the evaluation dataset. The desired direction for improvement: (↑) higher is better, (↓) lower is better. Turn Efficiency is not available for Canned-Response as it lacks multi-turn capabilities.

frameworks ability to maintain balanced performance in terms of accuracy, latency, and cost efficiency under increasing concurrent loads.

Cost Efficiency. Proposed framework demonstrates effective trade-offs, achieving a CE score of 0.7 compared to 0.3 for RAG. The introduction of dynamic query routing minimizes computational overhead for high-confidence queries.

Turn Efficiency. Turn efficiency (1.7) highlight the framework's ability to maintain coherence and minimize dialogue complexity while trying to resolve queries, relatively outperforming both baselines when compared in conjunction with accuracy and response latency.

Multi-Turn Interaction Analysis With 20% (2,000) queries including follow-up interactions, the dialogue context manager maintained high coherence in these multi-turn interactions, effectively tracking evolving user intents and ensuring context continuity. Minor context drift was observed in sessions exceeding 10 turns, indicating that optimizing context management for prolonged dialogues remains an area for future improvement. See Appendix A.5 for common failure scenarios and error analysis.

Query Load	Accuracy (%)	Latency (ms)	Cost Efficiency
1,000	96	174	0.77
5,000	96	177	0.74
10,000	95	180	0.71
20,000	94	186	0.70
50,000	92	193	0.69

Table 3: Scalability Results for the proposed Framework. Query Load indicates the number of queries processed in the evaluation.

4.3 Error Analysis:

We conducted a manual error analysis on 500 dialogue samples covering diverse user intents. Only 32 (6%) samples were found erroneous. Three independent annotators with subject matter expertise in Oracle cloud customer support evaluated these dialogue samples with an inter-annotator agreement of 0.91. Errors were categorized into four main types: 1) Edge Cases in Intent Classification (21%) due to subtle semantic differences, 2) Long Multi-Turn Dialogues (35%) where latency and context tracking issues arose, 3) Retrieval Inaccuracy (25%) from incomplete or outdated document retrieval, and 4) Feedback Misalignment (19%) due to misinterpretation of user feedback. Future work to remediate these could include refining fallback strategies, optimizing context management, regular index updates, and context-aware feedback processing. Further details are provided in Appendix A.5.

4.4 Final Insights and Implications

The evaluation metrics, error analysis and scalability underscore the proposed framework's effectiveness:

- **Efficiency-Accuracy Trade-offs**: Dynamic query routing achieves optimal balance between computational cost and response quality.
- **Multi-Turn Adaptability**: Superior context retention validates its applicability in complex dialogue scenarios.
- **Scalability and Robustness**: Modular design ensures operational resilience under high query loads.

5 Conclusion

We proposed a hybrid conversational framework that integrates intent-based canned responses with Retrieval-Augmented Generation (RAG) systems, explicitly designed to handle multi-turn interactions. The framework dynamically routes queries based on intent confidence, ensuring low latency for predefined intents while leveraging RAG for complex or ambiguous queries. The inclusion of a dialogue context manager guarantees coherence across multi-turn interactions, and a feedback-driven mechanism continuously refines intents and confidence thresholds over time.

Experimental results demonstrated the proposed framework's ability to balance accuracy (95%), response latency (180ms), and cost efficiency (0.7), while achieving superior context retention and turn efficiency in multi-turn scenarios. The system effectively handles multi-turn dialogues with minor limitations in long conversations exceeding 10 turns. Our contributions include a scalable, adaptive solution for dynamic conversational AI, addressing key industry challenges such as query diversity, evolving knowledge bases, and real-time performance requirements. Future research will focus on enhancing multi-turn context management, conducting ablation studies to isolate module contributions, and exploring real-time learning mechanisms for continuous adaptation. This work advances the state-of-the-art in enterprise conversational AI, providing a robust framework for handling complex, multi-turn interactions efficiently.

6 Limitations and Future Work

While our system demonstrates strong performance in enterprise customer support scenarios, it is optimized for English language applications and may require adaptation for multilingual deployments. Expanding to other languages introduces challenges such as acquiring labeled training data and handling linguistic variations, which may increase operational costs and training time. Additionally, our intent classifier is trained on domain-specific datasets, and extending to new domains or industries will necessitate retraining with relevant data, impacting both cost and deployment timelines.

Lastly, integrating real-time learning mechanisms that adapt continuously without periodic retraining is an avenue for future exploration, providing a more seamless and cost-effective method for maintaining system relevance over time. Future work will also include studies to isolate the impact of the dialogue context manager and quantify its contribution to system performance, as well as extending our framework to support multilingual conversations by improving intent recognition and retrieval efficiency across diverse languages.

References

Amit Agarwal, Srikant Panda, and Kulbhushan Pachauri. 2024a. Synthetic document generation pipeline for training artificial intelligence models. US Patent App. 17/994,712.

Amit Agarwal, Srikant Panda, and Kulbhushan Pachauri. 2025. Fs-dag: Few shot domain adapting graph networks for visually rich document understanding. In Proceedings of the 31st International Conference on Computational Linguistics: Industry Track, pages 100–114, Abu Dhabi, UAE. Association for Computational Linguistics.

Amit Agarwal, Hitesh Patel, Priyaranjan Pattnayak, Srikant Panda, Bhargava Kumar, and Tejaswini Kumar. 2024b. Enhancing document ai data generation through graph-based synthetic layouts. arXiv preprint arXiv:2412.03590.

Jason Bordes et al. 2020. Contextualized end-to-end learning for conversational ai. In Proceedings of the 2020 Conference on Neural Information Processing Systems (NeurIPS).

React.js Contributors. 2023. React: A javascript library for building user interfaces. Available at https://react.dev/.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The power of noise: Redefining retrieval for rag systems. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, page 719–729. ACM.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, et al. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. Association for Computational Linguistics (ACL), pages 4171–4186.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

A. Friedrich et al. 2020. Context-aware robust fine-tuning for chatbots. In Proceedings of the 2020 International Conference on AI and Machine Learning.

Hao Gao, Dongxu Li, Shuohang Wang, and Wenjie Li. 2020a. Hybrid conversational frameworks for multi-domain dialogue systems. In ACL, pages 298–305.

Hao Gao, Dongxu Li, Liheng Xu, Shuohang Wang, and Wenjie Li. 2020b. Search-augmented generation for dialogue systems. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL).

Pengfei Gao, Jianfei Gao, and et al. 2020c. Modular graph networks for reasoning over text. ACL.

Miguel Grinberg. 2018. Flask web development:

Developing web applications with Python. O'Reilly Media.

Ricardo Hernandez, Rahul Gupta, and Shubham Patel. 2020. Efficient and scalable hybrid retrieval for search engines. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.

Paulina Toro Isaza, Michael Nidd, Noah Zheutlin, Jae wook Ahn, Chidansh Amitkumar Bhatt, Yu Deng, Ruchi Mahindru, Martin Franz, Hans Florian, and Salim Roukos. 2024. Retrieval augmented generation-based incident resolution recommendation system for it support.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, et al. 2020. Dense passage retrieval for open-domain question answering. Empirical Methods in Natural Language Processing (EMNLP), pages 6769–6781.

Mandar Kulkarni, Praveen Tangarajan, Kyung Kim, and Anusua Trivedi. 2024. Reinforcement learning for optimizing rag for domain chatbots.

Jinhyuk Lee et al. 2020. Speculative rag: Enhancing retrieval augmented generation through drafting. In Proceedings of the 2020 Conference on Neural Information Processing Systems (NeurIPS).

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vassilios Stamatescu, Tim Rocktäschel, Sebastian Ruder, Pontus Stenetorp, and LUKAS RICHTER. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In Proceedings of the 38th International Conference on Machine Learning (ICML).

J. Michelson et al. 2020. Expanding chatbot knowledge in customer service: Context-aware similar question generation using large language models. In Proceedings of the 2020 Conference on Natural Language Processing.

Bingqing Mitra, Karan Goel, and Nikita Soni. 2021. Hybrid retrieval and ranking methods for information retrieval. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.

Hitesh Laxmichand Patel, Amit Agarwal, Bhargava Kumar, Karan Gupta, and Priyaranjan Pattnayak. 2024. Llm for barcodes: Generating diverse synthetic data for identity documents. arXiv preprint arXiv:2411.14962.

Priyaranjan Pattnayak, Hitesh Laxmichand Patel, Amit Agarwal, Bhargava Kumar, Srikant Panda, and Tejaswini Kumar. 2025. Improving clinical question answering with multi-task learning: A joint approach for answer extraction and medical categorization.

Priyaranjan Pattnayak, Hitesh Laxmichand Patel, Bhargava Kumar, Amit Agarwal, Ishan Banerjee, Srikant Panda, and Tejaswini Kumar. 2024. Survey of large multimodal model datasets, application categories and taxonomy. arXiv preprint arXiv:2412.17759.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084.

S. E. Robertson and S. Walker. 1994. Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval, pages 232–241. Springer London.

Tim Rocktäschel, Sebastian Ruder, Shinnosuke Takamatsu, and Pontus Stenetorp. 2020. Rethinking the role of knowledge in dialogue systems. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL).

Victor Sanh, Thomas Wolf, Julien Chaumond, and Clément Delangue. 2020a. Multitask mixture of sequence generation tasks for diverse natural language generation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1962–1971.

Victor Sanh, Thomas Wolf, Julien Chaumond, Clement Delangue, Pierrick Sprechmann, Alex Wang, Shinnosuke Takamatsu, and Tim Rocktäschel. 2020b. Realm: Retrieval-augmented language model pre-training. In Proceedings of the 37th International Conference on Machine Learning (ICML).

Iulian V. Serban, Alessandro Sordoni, Yoshua Bengio, and Jian-Yun Nie. 2017. A survey of available corpora for building data-driven dialogue systems. In Proceedings of the 2nd Workshop on Dialogue Systems Technology Evaluation (DST'17).

Vishal Shah, Pushpak Bhattacharyya, and Khurshid Ahmad. 2018. Building end-to-end dialogue systems with transformer models. In Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL).

Kurt Shuster, Eric Smith, Stephen Roller, Y-Lan Boureau, and Jason Weston. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. arXiv preprint arXiv:2102.09527.

Yi Tay, Mostafa Dehghani, Samira Abnar, Dara Bahri, Yikang Shen, Xingdi Zhou, and Donald Metzler. 2023. Efficient and scalable nlp with small and large pretrained language models. arXiv preprint arXiv:2305.13249.

Sriram Veturi, Saurabh Vaichal, Reshma Lal Jagadheesh, Nafis Irtiza Tripto, and Nian Yan. 2024. Rag based question-answering for contextual response prediction system.

et al. Zhao, W. 2020. A retrieval-augmented encoder-decoder for knowledge-intensive nlp tasks. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL).

A Appendix

A.1 Extended Workflow

Feedback-Driven Adaptability. The feedback rates are used to dynamically change the thresholds

Scenario	Query Type	Response Type	Impact
Predefined FAQ	High-confidence intent	Canned Response	Reduced Latency, Cost Savings
Contextual Query	Low-confidence intent	Hybrid (RAG + canned)	Increased Coherence, Cost Saving
Out-of-Domain Query	Undefined intent	Full RAG pipeline	Increased Accuracy

Table 4: Query Handling Scenarios in the Hybrid Framework.

Framework	Category	Accuracy (%)	Response Latency (ms)	Cost Efficiency
Canned Response	Predefined FAQ	93	65	1.00
	Contextual	49	65	1.00
	Out-of-Domain	5	75	0.08
RAG	Predefined FAQ	91	376	0.31
	Contextual	92	381	0.31
	Out-of-Domain	90	379	0.31
Proposed Framework	Predefined FAQ	96	65	1.00
	Contextual	96	182	0.67
	Out-of-Domain	93	379	0.32

Table 5: Performance comparison of different frameworks across various categories. Baseline cost efficiency is established using average latency and accuracy for canned responses across the entire evaluation dataset as mentioned in Table 2.

defined as follows:

$$\text{NFR} = \frac{\text{NegativeFeedback}}{\text{TotalQueries}}$$

$$\text{PFR} = \frac{\text{PositiveFeedback}}{\text{TotalQueries}}$$

New intents are generated from user feedback and query patterns, which are processed offline to update the Intent Data Store. Intent classification is refined continuously by an adaptive system feedback loop. Unresolved queries are logged, clustered on the basis of semantic similarity, and flagged for review. When a cluster reaches a certain size, a new intent is created offline and integrated into the classifier. Additionally, confidence thresholds are periodically adjusted based on user feedback to improve the routing of ambiguous queries.

A.1.1 Intent Evolution through Feedback

In addition to threshold tuning, the system expands its intent data store based on observed usage patterns and unresolved queries. The intent creation process operates in the following stages:

1. **Logging and Clustering**: All unhandled queries are logged and grouped using semantic similarity clustering.
2. **Pattern Detection**: If a cluster of unresolved queries exceeds a predefined frequency threshold, it is flagged for intent creation.
3. **New Intent Generation**: A new intent is proposed & validated by SMEs, and added to the Intent Data Store.

This process ensures that frequently occurring unresolved queries are automatically handled by the intent classifier going forward, thereby improving future query routing.

Improving FAQ Classification via Threshold Adjustment

Query: *“Why am I seeing high costs for my Oracle Autonomous Database instance?”*

The system classifies this as an FAQ and responds: **Response:** *"Oracle Autonomous Database costs depend on the compute shape, storage capacity, and workload type. You can adjust your settings to optimize cost."*

However, users frequently provide negative feedback, indicating that the response lacks details on Auto Scaling, Always Free tier limits, and OCI pricing policies. This causes NFR to increase, leading to an increase in τ_{FAQ}. The system becomes more selective in assigning queries to FAQs. More complex cost-related queries are routed to context-aware retrieval mechanisms rather than FAQs.

Intent Creation for Repeated OOD Queries
Query: *"How do I configure OCI Object Storage to replicate data to another region?"*

Initially, the system classifies this as Out-of-Domain (OOD), as no existing intent covers cross-region object storage replication. However, after multiple users ask similar questions, the system clusters these unresolved queries. Once the cluster surpasses the predefined frequency threshold, it is flagged for new intent creation by SMEs:

New Intent: *"OCI Object Storage Cross-Region Replication"*
Associated Response: *"Detailed steps to enabled and configure cross-region replication as determined by SME"*

The system proactively resolves similar future queries by classifying them under the newly created intent. Users receive accurate responses immediately instead of being redirected to general support.

Proposed Framework. The workflow of the proposed system is summarized in Table 4

A.2 Detailed Performance Comparison

This section provides a detailed breakdown of the performance of the proposed hybrid framework compared to baseline systems (Canned Response System) and RAG Pipeline across different query categories: Predefined FAQs, Contextual Queries, and Out-of-Domain Queries. The metrics include accuracy, response latency, and cost efficiency, highlighting the strengths and trade-offs of each approach.

Analysis The results in Table 5 demonstrate the trade-offs between accuracy, latency, and cost efficiency:

- **Predefined FAQs:** The proposed framework achieves a balance, with similar accuracy with the canned-response system while reducing latency by 82% compared to the RAG pipeline.
- **Contextual Queries:** The proposed framework strikes a balance between RAG's accuracy (92%) and the canned-response system's latency (65ms), achieving 96% accuracy with an acceptable latency of 182ms.
- **Out-of-Domain Queries:** The RAG Pipeline and the proposed framework have a very similar latency and performance with our proposed framework have slight better accuracy (3%) owing to the better handling of context and queries.

A.3 In-House Dataset Overview

The evaluation leveraged a in-house dataset on customer support for OCI Cloud based Services of 10,000 queries across three categories: predefined FAQs, contextual queries, and out-of-domain queries. Table 6 provides a sample of the queries used in the evaluation.

For BERT fine-tuning, we used in-house conversational dataset which is domain specific, with 35,000 human-customer conversations curated over a period of 6 months.

A.4 Multi-Turn Interaction Examples

To demonstrate the framework's adaptability, Table 7 outlines examples of evolving user queries and how the system dynamically adapts to maintain coherence.

A.5 Failure Cases and Error Analysis

We conducted a manual error analysis on 500 dialogue samples spanning diverse user intents. Three independent annotators with experience in enterprise conversational AI systems evaluated these dialogues, with an inter-annotator agreement of 0.91 (Cohen's Kappa). Inter-annotator agreement was calculated by comparing the categorical labels assigned (out of 4 shown below) by each annotator across all 500 dialogue samples. Annotators independently labeled each dialogue, and disagreements were resolved through discussion to refine the labeling criteria. The high agreement score (0.91) reflects consistency in identifying and categorizing errors across evaluators.

Errors were categorized as follows:

Query	Category	Confidence Level
How do I reset my password?	Predefined FAQ	0.95
What are the steps to integrate autoscaling?	Contextual	0.70
What are compliance requirements for data?	Out-of-Domain	0.40
Can you elaborate on scaling options?	Multi-Turn Follow-Up	0.75

Table 6: Sample Queries from the in-house Dataset.

Turn	User Query	Framework	System Response
1	What are the steps to enable advanced analytics?	Canned Response	Analytics can be enabled in the dashboard settings.
2	Can you explain what metrics are available?	Hybrid Response	Available metrics include user engagement, retention, and revenue.
3	How can I visualize these metrics effectively?	RAG Response	Visualization tools like Tableau and Power BI integrate seamlessly with our platform.
4	What steps are required to connect Tableau?	Hybrid Response	Refer to the integration settings under "Analytics" and provide your Tableau API key.
5	Are there any tutorials for advanced analytics setup?	RAG Response	Yes, detailed tutorials can be found in the documentation section under "Advanced Analytics."

Table 7: Multi-Turn Example Showcasing Evolving Intents and Follow-Ups.

- **Edge Cases in Intent Classification (21% of errors):** Queries were misclassified due to subtle semantic differences. For example, the query *"Can you assist with integrating API X for multi-platform deployment?"* was routed to a general FAQ response about API usage due to high lexical similarity.

- **Long Multi-Turn Dialogues (35% of errors):** In conversations exceeding 10 turns, latency increased, and context tracking sometimes failed. For instance, after handling a billing query, the system mistakenly retained billing context when the user shifted to technical support.

- **Retrieval Inaccuracy (25% of errors):** Some queries led to incomplete or off-topic document retrieval. For example, a query like *"Provide the latest number of regions your cloud service is available in"* retrieved outdated documents due to incomplete index updates.

- **Feedback Misalignment (19% of errors):** User feedback was sometimes misinterpreted. For instance, a user rated a correct response poorly due to slow response time rather than content accuracy, leading to unnecessary adjustments in the intent classifier.

Table 8 summarizes these failure cases and suggested remedies. This detailed error analysis highlights both the strengths of our system and areas for future improvement.

A.6 Prototype Implementation

The framework is implemented as a modular system using microservices:

- **Frontend:** Built with React.js for user interaction and feedback collection (Contributors, 2023).

- **Backend:** Flask microservices handle query classification, retrieval, and feedback processing (Grinberg, 2018).

- **Storage:** Elasticsearch stores canned responses & external knowledge base, while FAISS and dense embeddings support retrieval (Karpukhin et al., 2020).

- **Memory Cache:** A memory-augmented module maintains embeddings of prior inter-

Scenario	**Issue**	**Remedy**	**Error Distribution (%)**
Edge Cases in Intent Classification	Query incorrectly routed to canned responses	A stronger fallback strategy could improve reliability	21% (7/32)
Long Multi-Turn Dialogues	Latency for very long conversations	Optimize dialog context manager to reduce latency.	35% (11/32)
Retrieval Inaccuracy	Incomplete or outdated documents retrieved	Regular index updates and improved retrieval ranking	25% (8/32)
Feedback Misalignment	User feedback misinterpreted during adjustments	Implement context-aware feedback processing	19% (6/32)

Table 8: Failure Cases and Suggested Remedies. A total of 32 erroneous dialogues were identified out of 500 tested samples.

actions in OCI Cache (Managed Redis), allowing the system to retain relevant historical context across multiple dialogue turns.

- **Model Deployment:** Models (e.g., BERT, Encoder, Cross-Encoder, GPT-3 & other proprietary LLMs) are deployed using in-house architecture and OCI Gen AI Service for scalability.

B Technical Implementation of Multi-Turn Adaptation

The **Dialogue Context Manager** is implemented using a transformer-based architecture with the following components:

- **Context Embeddings:** Queries are encoded using fine-tuned BERT embeddings capturing semantic information and historical contexts are encoded using an in-house Bi-Encoder.
- **Memory Module:** A memory-augmented module maintains embeddings of prior interactions in cache (Redis), allowing the system to retain relevant historical context across multiple dialogue turns.
- **Context Attention Mechanism:** An attention layer prioritizes recent or semantically relevant interactions, dynamically retrieving context embeddings as input to the intent classifier and response generator.
- **Sliding Context Window:** Implements a fixed-length sliding window to limit the memory footprint and computational complexity by retaining only the most relevant context from prior turns.

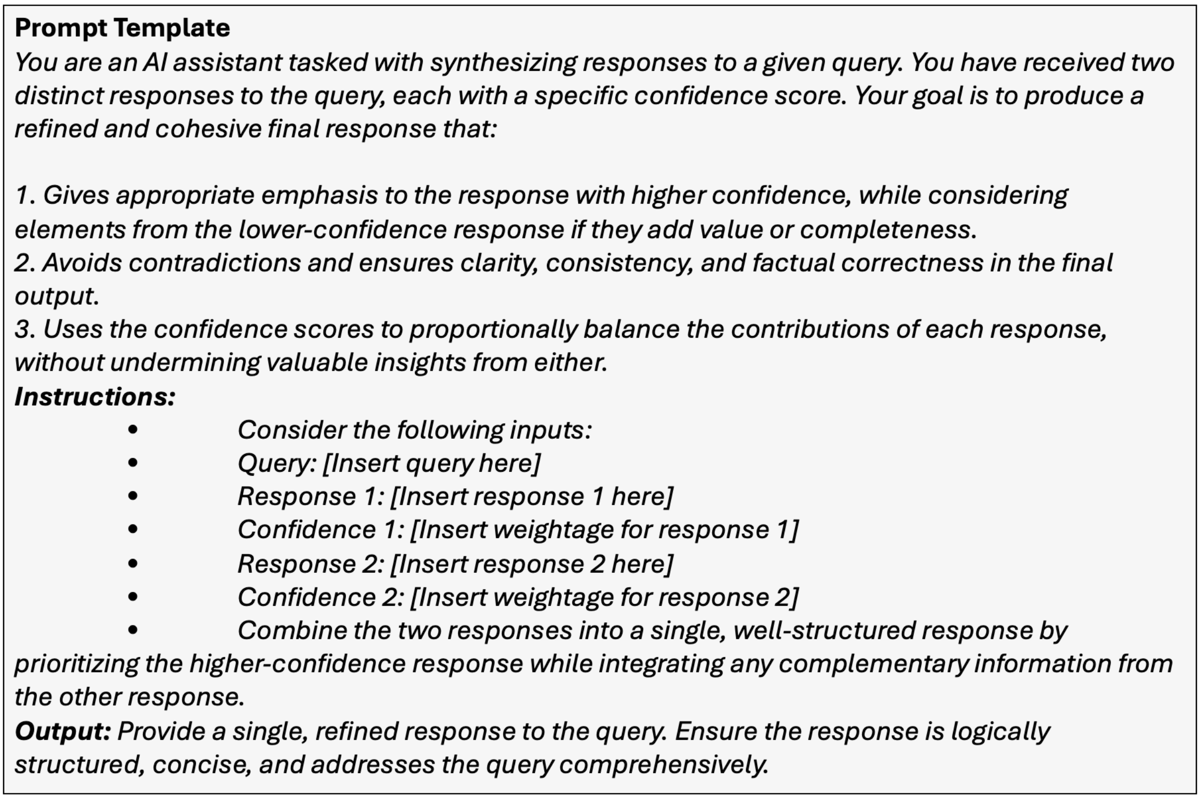
The context manager utilizes the embeddings and attention scores to generate a composite representation of the current dialogue state, which is passed to downstream components, such as the hybrid response generator. The dynamic adaptation ensures responses remain coherent and contextually grounded in multi-turn settings.

C Technical Implementation of Hybrid Routing

Hybrid routing combines canned responses and RAG outputs using a confidence-based decision-making pipeline:

- **Confidence Scoring:** The intent classifier assigns a confidence score to each query based on the similarity between the query embedding and predefined intent embeddings.
- **Thresholding Mechanism:** Queries with a confidence score above a pre-defined threshold (e.g., 85%) are routed to the canned response repository for rapid resolution.
- **Response Generation:** For low-confidence queries or multi-turn scenarios, responses are generated by blending canned responses with retrieved content from the RAG pipeline. Sample prompt used for blending the responses using confidence scores is shown in Figure 2

This mechanism optimizes query handling for diverse scenarios while ensuring minimal latency and maximal accuracy.



Prompt Template

You are an AI assistant tasked with synthesizing responses to a given query. You have received two distinct responses to the query, each with a specific confidence score. Your goal is to produce a refined and cohesive final response that:

1. Gives appropriate emphasis to the response with higher confidence, while considering elements from the lower-confidence response if they add value or completeness.
2. Avoids contradictions and ensures clarity, consistency, and factual correctness in the final output.
3. Uses the confidence scores to proportionally balance the contributions of each response, without undermining valuable insights from either.

Instructions:

- *Consider the following inputs:*
- *Query: [Insert query here]*
- *Response 1: [Insert response 1 here]*
- *Confidence 1: [Insert weightage for response 1]*
- *Response 2: [Insert response 2 here]*
- *Confidence 2: [Insert weightage for response 2]*
- *Combine the two responses into a single, well-structured response by prioritizing the higher-confidence response while integrating any complementary information from the other response.*

Output: *Provide a single, refined response to the query. Ensure the response is logically structured, concise, and addresses the query comprehensively.*

Figure 2: Prompt for Blending Responses

D Feature Limitations & Related Future Work

D.1 Limitations

Despite the strong performance of the proposed framework on a variety of metrics, certain feature-specific limitations remain:

- *Edge Cases in Intent Classification*: Ambiguous queries near confidence thresholds may cause inconsistencies, as seen in our error analysis, where subtle semantic differences led to misclassification. A stronger fallback strategy could improve reliability.
- *Latency in Long Multi-Turn Dialogues*: Latency issues for very long conversations (over 10 turns) were identified in 30% of errors, highlighting the need to optimize the dialogue context manager for faster context updates.
- *Retrieval Inaccuracy*: Incomplete or outdated document retrieval (20% of errors) due to index inconsistencies highlights the need for regular index updates and improved retrieval ranking.
- *Feedback Misalignment*: User feedback misinterpretation (10% of errors) occasionally led to suboptimal adjustments, suggesting the need for context-aware feedback processing.

D.2 Future Work

Future research could address these limitations by:

- Developing advanced intent detection techniques and fallback mechanisms to handle ambiguous and low-confidence queries more effectively.
- Enhancing multi-turn context tracking with memory-augmented models to improve coherence across long dialogues.
- Implementing regular index updates and fine-tuned retrieval processes to ensure accurate document retrieval.
- Integrating context-aware feedback processing to ensure accurate adaptation of system responses based on user ratings.
- Exploring distributed architectures and load-balancing techniques for scalability under peak query loads.