Beyond Filters: Personalized Hotel Recommendation System

Abstract

We propose a domain-adapted Retrieval-Augmented Generation (RAG) system for hotel search that jointly leverages structured metadata and unstructured user reviews. Our hybrid retrieval pipeline, combining BM25, dense embeddings, and topic-level sentiment signals, surfaces both factual and experiential content. Controlled prompting ensures grounded, coherent generation. Unlike traditional recommendation engines reliant on static features and opaque scoring, our approach enables dynamic, explainable, and user-centric recommendations. Results showed that fusing metadata with review-based retrieval improves contextual precision. We also identify fragilities in sequential RAG architectures and suggest future directions for goal-aware dialogue and adaptive retrieval.

1. Introduction

1.1 Background

Online information, particularly written reviews, now plays a central role in how travelers choose hotels. A 2023 global survey found that in 17 out of 18 markets studied, written consumer reviews were the most used source when researching accommodations, with 57% of travelers reading about others' experiences before making a booking (YouGov, 2023).

At the same time, the travel planning process has become increasingly complex, attributed to factors such as decision fatigue from the overwhelming number of options and the rising expectations for personalized experiences, (Dabo, 2024). As the amount of available information grows, so does the challenge of efficiently finding hotels that align with individual preferences and priorities.

Hotel and Travel Recommendation Systems

Earlier hotel and travel recommendation systems mainly relied on structured data such as location, price, and amenities using techniques like collaborative filtering and content-based filtering to match users to hotels based on their past behavior or stated preferences (Gavalas et al., 2014; Ramzan et al., 2019). More recent approaches have begun to leverage unstructured data from customer reviews, applying sentiment analysis (Ameur et al., 2023) and topic modeling (Zhang & Morimoto, 2017) to capture subjective qualities like service quality or room quietness. However, structured and unstructured information are often treated independently. Fully integrating both types of data into a unified retrieval pipeline (especially for fine-grained, aspect-specific queries) remains a relatively open challenge.

Retrieval-Augmented Generation (RAG)

RAG architectures, which combine retrieval with generative models to produce grounded, contextually informed responses, have shown promise in knowledge-intensive tasks (Lewis et al., 2020). Most RAG systems, however, have focused on general-purpose corpora such as Wikipedia, with limited attention to domain-specific applications involving heterogeneous data types. Applying RAG to hotel search requires adapting retrieval mechanisms to jointly handle dense, unstructured reviews alongside critical structured metadata.

Combining Structured and Unstructured Retrieval

Recent research has explored combining knowledge graphs with text retrieval to better support fact-based question answering (Xiong et al., 2021). In the context of hotel search, structured information (such as star ratings, location, or price) provides important factual anchors, while unstructured reviews capture experiential nuances. Yet practical systems rarely fuse these signals effectively; structured metadata and free-text insights are often retrieved separately rather than optimized jointly for relevance.

Bridging this gap is crucial for improving the quality of hotel recommendation and search systems, enabling users to retrieve both factual and experiential information in a unified, query-driven manner.

1.2 Problem Statement

As user-generated hotel reviews proliferate, travelers face growing challenges in extracting relevant insights from vast, fragmented text corpora. Although reviews offer valuable perspectives beyond structured hotel attributes, they are often distributed across thousands of posts, making it difficult for users to efficiently answer specific questions such as "Is the Wi-Fi reliable?"

Traditional recommendation systems typically focus on summarizing overall sentiment or aggregating ratings but fall short when users seek aspect-specific feedback. Existing retrieval methods either narrowly prioritize structured data or treat entire free-text reviews as undifferentiated blocks, lacking the granularity necessary for fine-grained question answering.

This research addresses the central challenge: How can we organize and retrieve information from large-scale hotel reviews to provide users with clear, precise, and aspect-specific answers to their queries?

1.3 Objectives

This study aims to develop an insight-driven retrieval system that enables users to search, understand, and compare hotel experiences more effectively. The key objectives are:

- Support niche, long-tail queries: Enable retrieval of highly specific information embedded within reviews (e.g., "Hotels near Sentosa with good wheelchair access and well-ventilated rooms") that traditional filter-based systems cannot easily capture.
- Extract experiential insights, not just aggregated rankings: Surface recurring themes and guest experiences (e.g., "frequent praise for fast check-in and MRT proximity") instead of relying solely on overall scores.
- Enable aspect-based comparison across hotels: Facilitate direct, aspect-specific comparisons (e.g., "Which hotel offers better breakfast quality?") grounded in user-generated content rather than structured metadata alone.

Collectively, these objectives aim to mitigate information overload by delivering targeted, aspect-specific insights rooted in large-scale, unstructured review data.

2. Dataset and Data Ingestion Pipeline

2.1 Data Collection

The dataset was constructed by scraping Booking.com for all publicly listed accommodations in Singapore, encompassing both structured hotel metadata and unstructured user reviews.

To support scalable data collection, a custom scraping pipeline was developed using Selenium Grid and ThreadPoolExecutor. Key features of the scraping infrastructure include:

1. **Dynamic Interaction Automation**: Automates scrolling, clicking, and extraction of reviews from each hotel's page;
2. **Robust Content Handling**: Manages hidden review sections, dynamic pagination, and slow-loading content through custom retry logic;
3. **Parallel Execution**: Enables up to 15 hotel listings to be scraped concurrently, significantly reducing crawl time and enhancing scalability.

The resulting dataset comprises:

- **Listings**: 444 hotels, each annotated with 30 structured attributes

Table 1. Hotel listings variables.

Category	Variables Captured
Hotel Information	Hotel name, hotel ID, description, star rating, preferred partner status, sustainability certification, address, latitude, longitude
Pricing and Promotions	Original price, current price
Review Metrics and Reputation	Number of reviews, review score, review label
Check-in and Hotel Policies	Check-in time, check-out time, children policies, extra bed and cot policies, age restrictions, payment methods, smoking policy, pets policy
Amenities and Surroundings	Room details (e.g., type, size, amenities); surroundings (e.g., restaurants, attractions); facilities (e.g., services, security)

- Reviews: 292,507 individual reviews, each capturing 11 variables

Table 2. Reviews listings variables.

Category	Variables Captured
Reviewer Profile	Name, country, traveler type
Stay Details	Hotel name, review room name, number of nights stayed, stay date
Review Content	Review score, review title, positive review text, negative review text

2.2 Data Preprocessing

To ensure the quality, consistency, and retrieval relevance of the review corpus, a multi-stage preprocessing pipeline was implemented. These steps aimed to minimize noise, preserve reviewer intent, and prepare the dataset for robust chunk-level embedding and retrieval.

2.2.1 Language Detection

We first detected the language of both positive and negative review texts using the langid library, which has been pre-trained on 97 languages. To avoid inconsistencies during embedding, reviews were excluded if either the positive or negative component was classified as a non-English language.

This filtering approach accounted for partial language detection (i.e., missing or ambiguous cases in one review field) and maximized precision by removing noisy

bilingual or foreign-language entries. Approximately 26.9% of reviews were removed at this stage.

2.2.2 Sentiment Classification

On Booking.com, reviewers provide three fields: a review title, positive aspects, and negative aspects. While positive and negative reviews are explicitly labeled, the title, often containing key evaluative information, lacks sentiment annotation. To address this, we applied a fine-tuned sentiment classifier (distilbert-base-uncased-finetuned-sst-2-english) to the title, predicting its polarity.

The classified title was then merged into either the positive or negative review field. This step preserved reviewer intent, enriched the textual content available for chunking and embedding, and mitigated the risk of losing valuable signals contained in titles that would otherwise be excluded from sentiment-aware retrieval.

2.2.3 Text Cleaning and Content Filtering

Text cleaning removed special characters while preserving key punctuation (periods, commas, exclamations) essential for maintaining sentence structure. Reviews were further filtered based on content: a domain-specific dictionary of generic or low-information responses (e.g., "good," "okay," "nothing") was used to eliminate non-informative entries, and reviews containing fewer than five meaningful words after cleaning were also discarded.

Following preprocessing, the final corpus comprised 185,125 reviews, forming a high-quality subset suitable for chunk-level retrieval.

2.3 Data Analysis

2.3.1 Exploratory Data Analysis

We conducted exploratory data analysis to understand the structure, quality, and coverage of the hotel metadata and review text corpora that underpin the RAG system.

Review Text Analysis

Review texts exhibit a strong right-skewed distribution (Figure 1), with median character counts of 92 for positive reviews and 97 for negative reviews, indicating that most reviews are relatively concise.

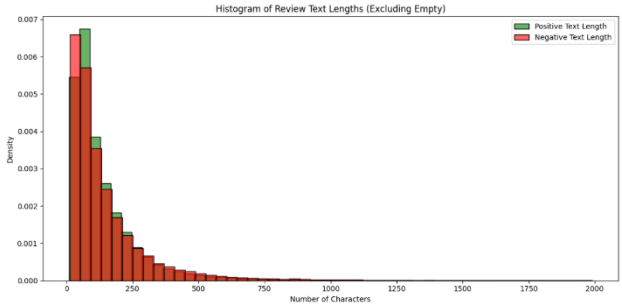


Figure 1. Histogram of review text length by characters.

Word cloud analysis (Figure 2) reveals distinct thematic patterns across sentiments. Positive reviews predominantly highlight aspects such as "staff," "cleanliness," and "location," whereas negative reviews more frequently emphasize issues such as "breakfast," "bed," and "bathroom." This clear thematic divergence between sentiments supports the use of sentiment-aware or aspect-based retrieval reranking to improve retrieval specificity and relevance.

Figure 2. Word cloud of positive and negative reviews.

Hotel Metadata Analysis

The distribution of star ratings across price tiers (Figure 3) reveals clear stratification, with higher-priced listings tending toward higher star ratings. This structured relationship between price and service level supports the use of metadata filtering to refine user queries based on budget or luxury preferences.

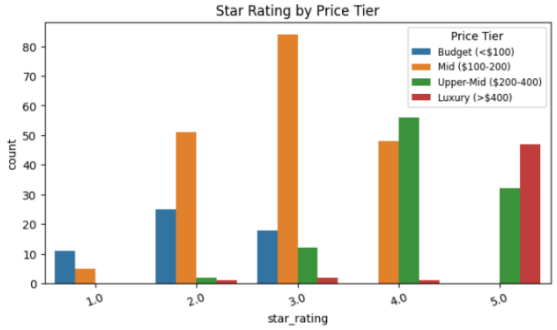


Figure 3. Hotels' star ratings by price tiers

Analysis of amenity availability by price tier reveals two key patterns. First, certain amenities (e.g. free WiFi, non-smoking rooms) exhibit uniformly high coverage across all price segments, indicating that they have become baseline expectations rather than differentiating factors among hotels. Second, premium amenities (e.g. fitness centres, outdoor swimming pools, room service, bars) are disproportionately concentrated in higher price tiers.

This stratification suggests that while budget hotels emphasize practical offerings (e.g., family rooms, luggage storage), luxury hotels differentiate themselves through value-added experiences.

These patterns highlight the importance of structured amenity retrieval: enabling users to filter not only for commonly expected features, but also for aspirational, price-tier-dependent offerings.

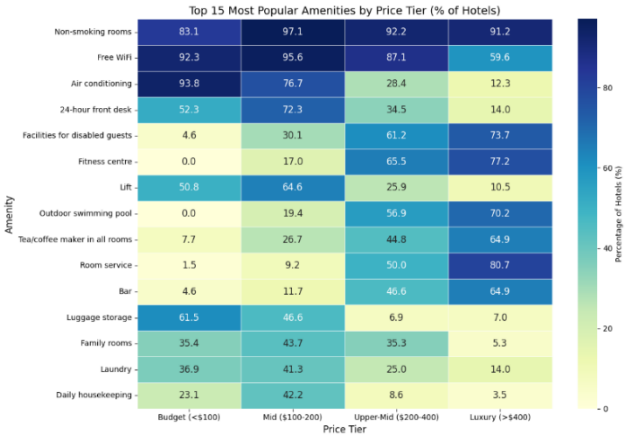


Figure 4. Top 15 most popular amenities by price tier

These findings informed key system design decisions, including the choice to perform chunk-level retrieval at the review level, implement sentiment-aware reranking to enhance specificity, and incorporate structured metadata filtering for budget, luxury, and amenity-driven queries.

2.3.2 Topic Modelling

To enhance retrieval specificity and enable fine-grained comparison of hotels based on guest experiences, we developed a structured topic modeling pipeline consisting of four key stages:

Topic Discovery: We employed BERTopic, a transformer-based topic modeling framework, to identify recurring themes across review chunks. The model clustered semantically similar review fragments into interpretable topics such as "cleanliness," "breakfast," and "pool facilities," enabling aspect-specific aggregation of feedback.

Sentence-Level Sentiment Classification: To capture sentiment at a finer granularity, each review chunk was decomposed into individual sentences using spaCy. Sentiment for each sentence was predicted using a fine-tuned DistilBERT model (*distilbert-base-uncased-finetuned-sst-2-english*). This sentence-level sentiment tagging ensures that each identified topic is linked to localized expressions of positivity or negativity, avoiding the potential dilution associated with document-level sentiment aggregation.

Summarization by Hotel and Topic: For each (hotel, topic) pair, we aggregated sentiment by selecting the most confident sentence-level sentiment associated with that topic. This produced a structured dataset where feedback is organized at the (hotel, topic, sentiment) level — for example, “Hotel A – Breakfast – Positive.” Such structuring enables direct, explainable retrieval responses when users inquire about specific aspects (e.g., "Which hotels have positive feedback on breakfast?").

Taxonomy Construction and Topic Grouping: To improve interpretability and facilitate broader thematic aggregation, we curated a taxonomy using ChatGPT to map fine-grained discovered topics into: (1) Sub-topics (e.g., friendly staff mapped to staff_attitude), and (2) Main topic (e.g., staff_attitude mapped to service). This hierarchical taxonomy supports flexible retrieval at multiple semantic levels, ranging from detailed aspect-specific feedback to aggregated service area evaluations.

2.3.3 Bayesian Average

Hotels with limited reviews are more susceptible to extreme positive or negative sentiment ratios, introducing volatility and potential misrepresentation. To enable fairer comparisons across hotels, particularly between properties with vastly different review counts, we applied a Bayesian smoothing approach to adjust observed sentiment scores.

Bayesian adjustment (Appendix 1) blends a hotel's observed sentiment ratio with a prior based on the global average sentiment for the same topic, producing more stable and representative estimates across varying sample sizes.

As shown in Figure 5, Bayesian smoothing moderates the extreme skewness of raw sentiment ratios, yielding a more centralized and stable distribution across hotel-topic pairs.

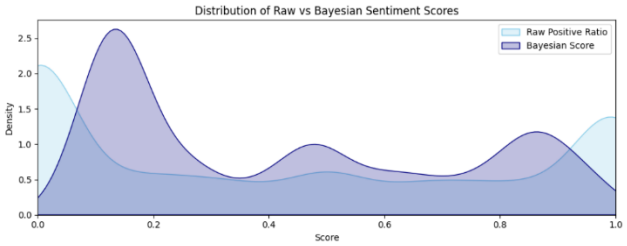


Figure 5. Distribution of Raw vs Bayesian Scores

At the individual hotel level (Figure 6), the adjustment is minimal for properties with abundant reviews (e.g., Shangri-La Rasa Sentosa, 115 reviews), preserving observed sentiment, but substantial for properties with limited data (e.g., KINN Studios, 2 reviews), pulling scores toward the global prior to mitigate overinterpretation.

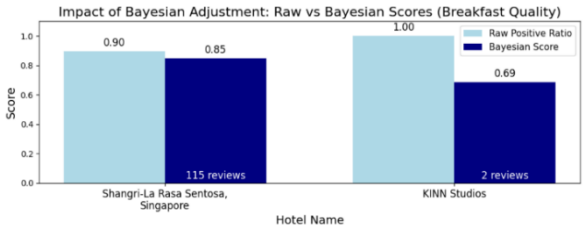


Figure 6. Impact of Bayesian Averaging on Scores

Together, these results illustrate how Bayesian smoothing stabilizes sentiment estimates and improves the reliability of downstream retrieval and ranking.

2.4 Review Chunking and Embedding

Review texts were segmented into 150-token chunks, balancing retrieval granularity with contextual richness. This chunk size was selected based on the observed distribution of review lengths, ensuring that most user queries could be satisfied by retrieving a single, self-contained chunk.

To avoid semantic fragmentation, sentence-based splitting was applied using the all-MiniLM-L6-v2 tokenizer, ensuring that chunk boundaries aligned with natural language units rather than arbitrary token cutoffs.

Chunks were embedded using all-MiniLM-L6-v2, a model optimized for semantic similarity tasks, chosen for its strong balance between retrieval accuracy and computational efficiency. Compared to larger transformer models, MiniLM offers significantly faster inference with only marginal performance trade-offs, making it suitable for large-scale indexing without compromising retrieval quality.

2.5 Vector Database Construction

Resulting embeddings were stored in Weaviate, enabling scalable hybrid retrieval by combining vector search with structured hotel metadata.

3. System Architecture

Our system architecture (Appendix 2) consists of the following components:

1. **Intent Classifier**: LLM-based classifier identifies user query types
2. **Retriever**: Hybrid BM25 + vector search via Weaviate to retrieve hotel review chunks
3. **Reranker**: Cohere API prioritizes documents based on semantic alignment with the user query
4. **LLM Generator**: Mistral-7B Instruct (integrated via via LangChain) generates natural language responses
5. **Booking Integrator**: Structured extraction of hotel names and travel dates from free text, followed by real-time querying of Booking.com's API
6. **Frontend**: Gradio interface with conversational memory and real-time response delivery

3.1 Retrieval Augmentation Generation

Our chatbot operates a unified Retrieval-Augmented Generation (RAG) architecture, combining information retrieval and large language model (LLM) generation into a structured pipeline.

In our hotel recommendation system, we adopted a RAG architecture to ensure that the chatbot could deliver responses grounded in real-world data, rather than relying purely on LLM memorization or hallucination. This design is critical because users often ask complex, subjective queries—such as "hotels with quiet rooms and good breakfast near a park"—where structured filters alone would be insufficient.

3.2 Intent Recognition

Before applying any retrieval or generation techniques, our system first identifies the user's intent through a finetuned Mistral 7B model, categorising each query into one of three buckets:

1. **Recommendation:** Users seeking hotel suggestions based on preferences or constraints
2. **Review lookup:** Users asking about opinions, reviews, or sentiments for specific hotels
3. **Booking check:** Users wanting to check availability for particular dates and hotels

This step is critical because different intents require different retrieval and generation strategies. Any open-ended questions not classified to the three buckets will be directed to a catch-all flow to ensure the chatbot can handle miscellaneous inputs.

3.3 Retrieval

The retrieval phase focuses on pulling candidate documents from our Weaviate vector database.

We adopted a hybrid retrieval method that combines dense sentence embeddings (capturing semantic relevance) with BM25 (capturing exact keyword matches) to retrieve hotel review chunks. A tunable alpha parameter controls the blend between the two. This dual approach ensures we retrieve both precisely matched and conceptually relevant documents, avoiding retrieval failures where highly relevant hotels are phrased differently from the user's query — overcoming the individual limitations of purely sparse or dense retrieval methods.

For recommendation queries, structured filters parsed from user queries—such as minimum star rating, maximum price, or location—are applied before hybrid retrieval to shortlist candidate hotels. This step ensures that hotels failing hard constraints are excluded early, improving retrieval precision.

To align user queries with retrieval targets in a multi-turn dialogue, our system leverages conversational memory. For example, after hotel recommendations are provided, a user might ask, "*How are the reviews for these hotels?"*. Rather than requesting clarification, the system taps into conversational memory to retrieve the last recommended

hotels before invoking the hybrid search. As a result, the retrieval is executed against the correct contextual scope.

3.4 Augmentation

All retrieved documents are reranked using Cohere's Reranker, which prioritizes those most aligned with the query. This step ensures that even among hybrid search results, the best matches surface to the top.

In recommendation flows, an explicit amenity filter is applied after reranking. If a user specifies an amenity like "late checkout" or "airport shuttle," we post-filter to ensure the final set of hotels explicitly mentions those amenities in reviews, addressing cases where semantic similarity alone may dilute specific keyword presence.

In review lookup flows, we introduce topic sentiment embedding where each retrieved review chunk is enriched where each retrieved review chunk is enriched with topic-level sentiment information from our custom-built HotelTopicSummary dataset (see Section 2.3.2).

Specifically, if the user query mentions themes such as "cleanliness" or "breakfast," we cross-reference the hotel's topics and inject the positive sentiment ratio directly into the review text. This acts as a signal for our Cohere reranker, where it prioritises not only semantically relevant chunks, but those where guests have expressed stronger positive sentiment on topics that the user cares about.

These augmentation steps ensure that context fed into the LLM is both topically relevant and semantically aligned with user expectations.

3.5 Grounded Generation

Following augmentation, generation is performed using controlled LLM prompts to ensure faithfulness, structure, and reliability. Each intent is paired with a specialized prompt template: for hotel recommendations, the model outputs exactly three hotels in a fixed format including hotel name, star rating, price, amenities, and a positive review highlight; for review summaries, the model captures overall sentiment, key strengths, and common complaints per hotel.

To maintain grounding, the prompts explicitly instruct the model to rely solely on the retrieved context and prohibit fabrication, conversational embellishments, or speculative advice. Structured output formats are enforced to ensure responses remain scannable, consistent, and predictable across turns.

Operationally, LangChain's stuffing chain architecture was used to integrate retrieved documents, conversation memory, and user queries into the LLM prompt in a clean, consistent manner. This allowed flexible, intent-specific prompting while maintaining strict adherence to context, minimising hallucination risk and preserving response quality.

3.6 Structured Extraction and API Querying

To complete the user journey from exploration to transaction, the system incorporates a structured extraction and external querying component. For booking-related queries, a prompt-engineered LLM extracts hotel names and travel dates from free-text input, converting relative expressions into absolute dates. The extracted information is used to query the Booking.com API for live room availability and pricing. Unlike recommendation and review flows, booking checks bypass retrieval and reranking, focusing solely on accurate information extraction and real-time data access. This ensures that the chatbot supports users across all stages of hotel selection, review evaluation, and booking within a unified conversational experience.

3.7 Deployment

The final system was deployed using Gradio as a lightweight, interactive frontend. Gradio enabled rapid prototyping of the chat interface, integration of conversational memory, and real-time user feedback without the need for custom web development. The chatbot backend connects modular components — including retrieval from Weaviate, reranking via Cohere, generation through Mistral-7B Instruct (accessed with LangChain), and booking API querying — into a seamless conversational pipeline. To ensure reliability, we used defensive parsing techniques for LLM outputs, controlled response formatting through strict prompt templates, and implemented fallback handling for incomplete or ambiguous queries. The modular design also allows for easy future extensions, such as multilingual support, long-term user memory, or personalization based on past conversations.

4. Evaluation

This section evaluates the design, methodology, and findings of the RAG system evaluation, specifically focusing on the retrieval and performance across different retrieval strategies and the overall generation performance.

4.1 RAG Evaluation Framework

We employed a structured four-step evaluation framework to systematically assess their RAG system's performance.

First, a curated test dataset was developed to cover a diverse range of query types. Second, clearly defined retrieval and generation metrics were selected, aligning evaluation with practical user expectations for hotel recommendation systems. Third, inference was conducted through the full RAG pipeline, with metrics computed across multiple retrieval methods. Finally, an LLM-as-a-Judge evaluation component was incorporated to provide semantic quality assessments, to further validate our findings.

This multi-layered framework is intended to demonstrate a well-considered and thorough approach, balancing quantitative and qualitative evaluation dimensions**.**

4.2 Build Evaluation Dataset

To comprehensively assess the system, a curated evaluation dataset of ten queries was developed. The queries were constructed to span a broad range of task categories, simulating realistic hotel search scenarios. The full list of queries is provided in *Appendix 3*.

The dataset was deliberately divided into two distinct groups:

Structured Metadata Queries: These queries were designed around hotel metadata attributes such as price, facilities, star ratings, and locations. Examples include factual recall ("What facilities does Marina Bay Sands offer?"), constraint-based filtering ("Which hotels cost less than S$200/night around Orchard?"), numerical reasoning, and comparative queries. Ground-truth answers and supporting documents for these queries were constructed using a systematic strategy based on predefined criteria — selecting documents directly mapping to known metadata fields to ensure consistency and reproducibility. This enabled the application of classical retrieval evaluation metrics such as Precision@k, Recall@k, MAP@k, and MRR@k.

Review-Based Semantic Queries: The second group involved retrieval over unstructured hotel reviews, including exploratory recommendation queries ("Recommend three unique boutique hotels in Singapore"), preference-driven queries, inference over multiple attributes, and opinion summarization. Given the large corpus size and the highly variable nature of review content, it was infeasible to exhaustively enumerate ground-truth documents for these queries. Instead, evaluation relied on semantic similarity techniques: retrieved documents were embedded, and cosine similarity was computed against ground-truth answer embeddings to assess retrieval relevance. Contextual retrieval metrics (contextual precision@k, contextual recall@k, contextual MAP@k, contextual MRR@k) were then derived based on similarity thresholds of 0.7. A similarity threshold of 0.7 was selected to balance semantic relevance and retrieval breadth, ensuring retrieved documents are meaningfully related to the query without being overly restrictive.

4.3 Define Evaluation Metrics

Evaluation metrics were carefully selected to align with the dual goals of the system: (1) achieving effective information retrieval, and (2) generating faithful, user-relevant hotel recommendations.

Retrieval metrics include: Recall@k which measures the completeness of retrieval, critical to ensure sufficient hotel options are surfaced. Precision@k which measures the relevance and cleanliness of retrieved documents, minimizing noise. MAP@k and MRR@k which assess the quality of result ranking, ensuring that the most relevant documents appear early.

Generation metrics include: Faithfulness which assesses whether generated responses are factually grounded in the retrieved documents. Critical for maintaining user trust, especially regarding factual information like hotel amenities, locations, and prices. Relevance which measures how well the generated output addresses the user's query intent. BERT-F1 which evaluates semantic similarity between the generated answer and the ground-truth answer, allowing for natural language variation.

For review-based queries where classical ground-truth was impractical, semantic retrieval evaluation via cosine similarity was integrated to assess retrieval and generation quality meaningfully. The metric selection ensured that both technical retrieval performance and user-facing experience were evaluated comprehensively.

4.4 Evaluation Results

4.4.1 Overall performance

The evaluation results (in *Appendix 4*) are first discussed based on the overall system performance after reranking, covering retrieval and generation aspects across both structured and review-based queries. On the left chart above on classical retrieval metrics, performance demonstrated strong precision and ranking quality. Vector retrieval achieved a Precision@5 of 67% and an MRR@5 of approximately 83%, indicating that highly relevant hotels were both retrieved and prioritized early in the ranked list. This performance suggests that for structured queries relying on metadata attributes, the system was effective in surfacing accurate information rapidly to the user.

For the middle chart on contextual retrieval metrics, review-based retrieval tasks, reranked hybrid retrieval significantly improved semantic relevance. The contextual MAP score increased to approximately 75% after reranking, reflecting a substantial alignment between the retrieved review content and the user's query intent. This highlights the system’s ability to handle open-ended, preference-driven queries effectively when semantic matching was properly optimized.

Finally for the right chart on generation metrics, the system demonstrated consistently strong performance. Generated responses achieved faithfulness scores of approximately 77% and relevance scores around 68%, confirming that the RAG pipeline remained reliably grounded in the retrieved evidence and maintained alignment with user queries. Importantly, even when retrieval coverage was not perfect, the generation component showed resilience, producing semantically accurate and contextually appropriate outputs.

However, precision and ranking metrics were high, we recognize that recall scores were lower, reflected by the inherent challenges associated with exhaustive ground-truth construction, particularly for review-based queries. Lower recall emphasizes the opportunity to expand retrieval breadth further, to ensure that even more semantically relevant documents are available for grounding generated answers, especially for complex constraint-driven and inference tasks.

4.4.2 Analysis by Rerank vs No Rerank

Reranking played a crucial role in these improvements, especially on the contextual metrics. Referring to *Appendix 5*, applying reranking significantly boosted Hybrid retrieval performance, increasing both MAP and MRR across classical and contextual tasks. In particular, reranking proved especially impactful for semantic review-based queries, where capturing subtle semantic alignment between user intent and retrieved evidence is critical. By intelligently reordering initially retrieved candidates based on semantic relevance, reranking improved not only the top-k document quality but also directly contributed to higher faithfulness and relevance in the generation outputs.

4.4.3 Analysis by Query Type

Breaking down performance by query type (*Appendix 6*) reveals distinct strengths and weaknesses, informed by both quantitative metrics and qualitative observations.

Strong Performance (Factual Recall, Preference-Driven)

Factual recall queries achieved moderate retrieval (~33–50%) with very high faithfulness (0.8–1.0) and near-perfect relevance (~1.0). This reflects good retrieval coverage and highly accurate answers.

Similarly, preference-driven queries performed even better, with recall near ~66% and perfect faithfulness and relevance (1.0 These findings demonstrate that when retrieval succeeds, the system's generation stage reliably produces high-quality, user-aligned recommendations, highlighting the RAG pipeline's end-to-end effectiveness under favorable retrieval conditions.

Weak Performance (Constraint-Based, Synthesis / Inference)

Constraint-based queries had notably poor retrieval (<20%) and lower faithfulness (~0.5). This limited evidence availability often led to retrieval gaps and a heightened risk of factual hallucinations in the generation outputs.

Likewise, synthesis and inference queries, which required reasoning across multiple attributes, showed recall as low as ~17%, with depressed faithfulness (~0.5) and moderate relevance (~0.7). These findings confirm that the system struggles when faced with complex, compound queries, where the inability to ground responses reliably on retrieved evidence presents a significant limitation.

Intermediate Performance (Exploratory, Comparative)

Exploratory queries demonstrated moderate retrieval (~50%), strong faithfulness (~0.8), and high relevance (~0.9). Although not exhaustive, retrieval provided enough meaningful content for the generation stage to produce coherent and largely accurate outputs.

Comparative reasoning queries followed a similar trend, achieving recall around 50% and similarly strong faithfulness (~0.8) and relevance (~0.9). The system handled moderate complexity tasks reasonably well, though minor factual inaccuracies were occasionally observed in comparative answers.

Overall, this breakdown provides a structured view of the RAG system's strengths and vulnerabilities. Retrieval quality remains the primary driver of downstream generation success, and while generation exhibits resilience, challenges persist particularly in constraint-heavy and multi-hop reasoning scenarios. To further validate these findings, an independent LLM-as-a-Judge semantic evaluation was conducted, providing an additional layer of assessment on retrieval and generation quality.

4.5 LLM-as-a-Judge

We conducted semantic assessment using LLM-as-a-Judge. This approach was designed to provide an independent perspective on retrieval relevance, answer faithfulness, and topical alignment from a semantic standpoint, beyond traditional ID-based metrics. A structured evaluation prompt (*Appendix 7*) was developed and submitted to ChatGPT-4. The LLM systematically assessed: (1) The semantic relevance of retrieved documents to the user query, (2) The faithfulness of generated answers to retrieved evidence (3) The relevance of generated outputs to the original user intent.

Following the application of the LLM evaluation across all 10 test queries, the following average scores and scores by query were obtained (refer to *Appendix 8*). We observed three validation insights based on this analysis: First, the low retrieval score (0.23) reinforces the earlier observation that retrieval breadth is the primary performance bottleneck, particularly for constraint-driven and complex queries. Second, the moderate faithfulness score (0.30) further validates that retrieval weaknesses have a direct negative impact on generation factuality. Third, the strong relevance score (0.67) aligns closely with classical generation evaluations, confirming that the system reliably maintains query focus even when retrieval is imperfect.

Thus, the LLM-as-a-Judge results provided independent validation of our RAG evaluation findings, whereby,

retrieval coverage remains the key area for improvement, while generation relevance remains a system strength.

4.6 Key takeaways

In summary, the evaluation demonstrates that the RAG system achieves strong semantic relevance and robust generation faithfulness when retrieval coverage is sufficient. However, limited retrieval breadth remains the primary bottleneck, particularly for complex, constraint-driven, and multi-hop queries. To address these challenges, several opportunities for improvement have been identified: implementing a multi-stage retrieval process with dynamic top-K adjustment, refining LLM prompting to handle low-evidence cases without hallucination, and exploring retrieval chaining strategies for multi-attribute reasoning tasks. These enhancements aim to strengthen factual grounding and retrieval completeness, while preserving the strong semantic relevance already demonstrated.

5. Discussion

Beyond technical evaluation, we consider the broader insights gained, their implications for stakeholders, the current limitations of the system, and directions for future development.

5.1 Insights

5.1.1 LLM-Powered Recommendations Versus Traditional Machine Learning

A key insight from our work lies in contrasting LLM-based recommendation with traditional machine learning pipelines. Conventional models rely heavily on structured feature engineering—manually curating attributes such as location or amenities—and produce similarity scores. In contrast, LLM-powered systems naturally generate human-readable justifications grounded in retrieved evidence, enhancing explainability without additional modeling complexity.

Prompt engineering, rather than feature engineering, becomes the primary lever of system behavior. This shift simplifies system maintenance and accelerates adaptability to emerging user needs. New attributes (e.g., “soundproof rooms” or “vegan breakfast”) can be incorporated without modifying schemas or retraining models, as long as they are reflected in the retrieved context. LLM-augmented architectures therefore offer greater flexibility, extensibility, and transparency than traditional approaches—particularly valuable in high-variance domains like travel and hospitality.

5.1.2 Balancing Open-Domain Flexibility with Controlled Robustness

A second critical insight centers on managing the inherent tension between conversational flexibility and system robustness. While LLMs enable rich, open-ended dialogue, unconstrained generation risks speculative or irrelevant outputs.

We addressed this through intent-driven routing, mapping each user query to a defined sub flow (recommendation, review lookup, booking check, or fallback), ensuring that retrieval and generation remain contextually anchored. Strategic clarification steps elicit missing critical information—such as budget or dates—through minimal, targeted follow-up prompts, improving retrieval precision downstream. Controlled fallback handling ensures that unsupported queries are met with succinct, safe defaults rather than unfounded elaborations.

Through these mechanisms, the system preserves conversational agility while maintaining fidelity, reliability, and user trust—critical attributes for applied LLM systems operating in decision-critical settings.

5.2 Business and Stakeholder Implications

Our system offers valuable implications across multiple stakeholder groups within the travel and hospitality ecosystem.

For hoteliers, the ability to surface and summarize guest sentiment at scale transforms how feedback is analyzed. Rather than manually parsing individual reviews, hotel managers can monitor aggregated insights—such as frequently mentioned issues or praised amenities—and prioritize operational improvements. Competitive benchmarking is also enhanced, allowing hotels to compare guest sentiment on specific aspects, like cleanliness or service, against similar properties.

For booking platforms like Booking.com, the system enables more nuanced and user-centric search experiences. Instead of filtering by rigid categories, users can search conversationally—for instance, asking for “quiet hotels with excellent breakfasts.” This improves user satisfaction, reduces abandonment, and increases booking conversions through more aligned recommendations.

At a policy level, organizations like tourism boards can leverage aggregated review analytics to identify macro trends in traveler preferences. Insights into recurring themes—such as sustainability concerns or demand for wellness amenities—can guide policy formulation, destination marketing, and infrastructure development tailored to evolving visitor expectations.

5.3 Limitations

Our system, while functional, exhibits several structural and performance limitations:

Pipeline Fragility: The multi-stage nature of the pipeline—intent recognition, constraint extraction, retrieval, reranking, and generation—is inherently fragile. Early-stage errors, like misclassified intent or misparsed

constraints, propagate and magnify through later stages, degrading output quality. Additionally, the reliance on Cohere's reranker introduces a single point of failure; if it performs suboptimally, downstream stages cannot correct or adapt.

Pipeline Latency: Each component in the sequential pipeline contributes to overall response time. Hybrid retrieval, reranking, amenity filtering, deduplication, structured prompting, and LLM generation together result in latencies often exceeding 10 seconds, which risks user disengagement in production settings.

Retrieval Granularity Mismatch: By retrieving isolated review chunks rather than full narratives, the system prioritizes precision at the cost of context. Users may receive an incomplete view—e.g., reading only positive aspects of a stay while missing later complaints due to chunking boundaries.

Single-Intent Limitation: The system processes one intent per query. Complex, multi-intent inputs—such as requesting recommendations and real-time pricing simultaneously—cannot be decomposed and handled effectively.

Lack of Goal Modeling: The chatbot does not maintain an explicit model of user goals across dialogue turns. While recent interactions are remembered, there's no deeper understanding of user objectives or preferences over a session, limiting personalization and continuity.

5.4 Future Directions

Looking ahead, several pathways could improve robustness, responsiveness, and user engagement:

Fault-Tolerant Pipeline Architectures: Introducing redundancy in reranking such as combining semantic relevance with diversity metrics, and propagating uncertainty signals downstream. This would allow the system to adapt dynamically, reducing dependence on any single component.

Latency-Aware Retrieval Strategies: Employing staged pipelines—using fast, coarse filters (BM25) before applying semantic reranking selectively—can significantly reduce processing time. Adaptive query complexity handling would also balance accuracy with speed in real-time applications.

Adaptive Retrieval Granularity: The system could dynamically switch between retrieving short snippets or full review narratives based on query complexity. This would enhance narrative coherence when needed, without sacrificing performance for simpler queries.

Goal-Aware Conversational Modeling: Tracking user preferences and session-level goals over multiple turns would enable more proactive, personalized assistance. Incorporating lightweight user profiling could further refine recommendations and query handling.

System Expansion Opportunities

1. **Multilingual Capabilities**: Extending retrieval and generation to support multiple languages would broaden accessibility for diverse users.
2. **Cross-Domain Adaptation**: The same RAG framework can be applied to other domains, such as restaurant or tourist attractions, offering end-to-end travel planning support.
3. **Multi-Modal Retrieval**: Integrating hotel images, room layouts, or floor plans would enhance responses to queries about spatial concerns, such as room size or ambiance.

These directions position the system to evolve from a rule-bound conversational agent into a proactive and highly adaptable travel companion.

6. Conclusion

This project built a retrieval-augmented generation (RAG) system that helps travelers find hotels more easily by combining structured metadata and review text. By retrieving fine-grained review passages and reranking results based on aspect-specific sentiment, the system delivers more targeted and personalized recommendations.

Evaluation showed strong performance for factual and preference-based queries, but retrieval remains the main bottleneck, especially for complex or constraint-driven questions. Independent validation through LLM-as-a-Judge assessments confirmed these patterns, highlighting retrieval breadth as the main performance bottleneck.

Beyond technical results, this work highlights how LLM-based systems simplify hotel recommendations by moving from rigid feature engineering to flexible, evidence-grounded answers. Key challenges like pipeline fragility, latency, and limited goal tracking remain, pointing to clear opportunities for future improvement.

Overall, the system shows how structured and unstructured hotel data can be combined to create smarter, more user-focused search experiences.

References

Ameur, A., Hamdi, S., & Ben Yahia, S. (2023). Sentiment Analysis for Hotel Reviews: A Systematic Literature Review. *ACM Comput. Surv.*, *56*(2), 51:1-51:38. https://doi.org/10.1145/3605152

Dabo, M. (2024, November 29). Travelers are spending more time researching options. *Hotel Management Network*. https://www.hotelmanagement-network.com/features/travelers-are-spending-more-time-researching-options/

Gavalas, D., Konstantopoulos, C., Mastakas, K., & Pantziou, G. (2014). Mobile recommender systems in tourism. *J. Netw. Comput. Appl.*, *39*(C), 319–333.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., & Rocktäschel, T. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, *33*, 9459–9474.

Ramzan, B., Bajwa, I. S., Jamil, N., Amin, R. U., Ramzan, S., Mirza, F., & Sarwar, N. (2019). An Intelligent Data Analysis for Recommendation Systems Using Machine Learning. *Scientific Programming*, *2019*(1), 5941096. https://doi.org/10.1155/2019/5941096

Xiong, W., Li, X. L., Iyer, S., Du, J., Lewis, P., Wang, W. Y., Mehdad, Y., Yih, W., Riedel, S., Kiela, D., & Oğuz, B. (2021). *Answering Complex Open-Domain Questions with Multi-Hop Dense Retrieval* (arXiv:2009.12756). arXiv. https://doi.org/10.48550/arXiv.2009.12756

YouGov. (2023, May 11). *How do global travelers research before booking a hotel?* https://business.yougov.com/content/46627-how-do-global-travelers-research-before-booking-a-hotel

Zhang, Z., & Morimoto, Y. (2017). Collaborative hotel recommendation based on topic and sentiment of review comments. *Proceeding of the 9th Forum for Information and Engineering, DEIM*. http://db-event.jpn.org/deim2017/papers/245.pdf

Appendices

GitHub Link: https://github.com/Techilis/hotel-chatbot.git

Appendix 1: Bayesian Score

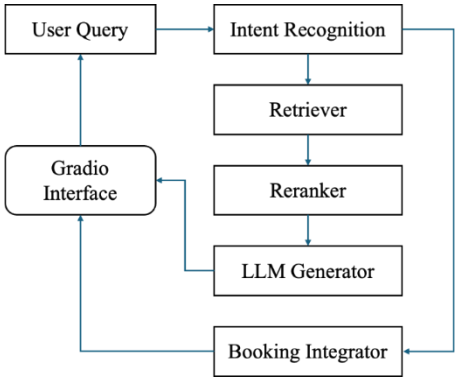
The Bayesian-adjusted score is computed as:

$$\text{Bayesian Score} = \frac{\text{positive\_count} + m \times \text{aspect\_prior}}{\text{total\_count} + m}$$

where:

- positive_count is the number of positive mentions for a given hotel-topic pair,
- total_count is the total number of mentions for that topic,
- aspect_prior represents the average positive sentiment rate across all hotels for that topic,
- m is a smoothing parameter reflecting the weight assigned to the prior.

Appendix 2: System Architecture

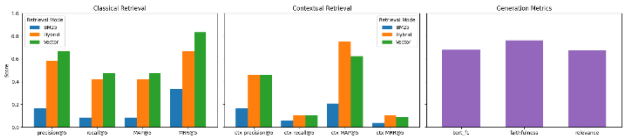


Appendix 3: Test queries table

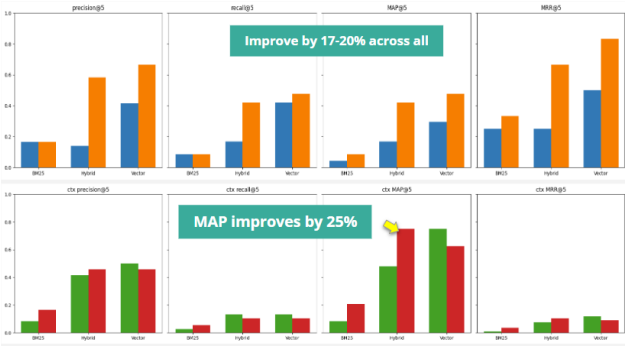
Category	What it Tests	Example Question
Factual Recall	Simple hotel metadata retrieval (text); Multi-attribute retrieval (text + numeric)	1. What facilities does Marina Bay Sands offer? 2. What is the star rating and price of The Fullerton Hotel?
Constraint-Based	Attribute-based filtering over hotel metadata; Numeric constraints application	3. Which hotels currently cost less than S$200/night around Orchard? 4. Which value-for-money hotel costs less than S$200/night currently?
Numerical Understanding	Numeric aggregation and reasoning over hotel metadata	5. What are the current prices of 5-star hotels in Singapore?

Comparative Reasoning	Comparative analysis across multiple attributes over hotel metadata	6. Between Marina Bay Sands and The Fullerton Hotel Singapore, which is more value for money?
Exploratory Recommendation	Open-ended recommendation generation from reviews	7. Recommend three unique boutique hotels in Singapore.
Preference-Driven Recommendation	Preference matching based on guest experiences extracted from reviews	8. Which hotels in Singapore have nice breakfast?
Inference	Reasoning over multiple review-derived attributes	9. Are there any hotels near Sentosa that are both peaceful and close to the beach?
Opinion Summarization	Summarizing positive and negative guest opinions from reviews	10. Tell me about Marina Bay Sands—what do guests say that's good, and what's not so good?

Appendix 4: Overall performance across retrieval and generation metrics (reranked)



Appendix 5: No Rerank and Reranked for Classical and Contextual Retrieval Scores



Appendix 6: Reviews listings variables.

Qn	Category	Recall (%)	Faithfulness	Relevance
1,2	Factual Recall	🟢 ~33–50%	🟢 0.8–1.0	🟢 ~1.0
3,4	Constraint-Based	🔴 <20%	🔴 ~0.5	🟡 ~0.7
8	Preference-Driven	🟢 ~66%	🟢 1.0	🟢 1.0
7	Exploratory	🟡 ~50%	🟢 ~0.8	🟢 ~0.9
9	Synthesis/ Inference	🔴 ~17%	🔴 ~0.5	🟡 ~0.7
6	Comparative	🟡 ~50%	🟢 ~0.8	🟢 ~0.9

Appendix 7: LLM-as-a-Judge ChatGPT Prompt link:

https://chatgpt.com/share/680e5055-9744-8001-9b21-fff745c5d5d4

Appendix 8: LLM-as-a-Judge Evaluation Results

Scores by Question & Category

		0.23	0.3	0.67
#	Category	Retrieval	Faithfulness	Relevance
1	Factual-Recall	0.5	0.0	1.0
2	Factual-Recall	0.2	0.0	0.0
3	Constraint-Based	0.0	0.0	0.0
4	Constraint-Based	0.2	0.0	0.7
5	Numerical-Understanding	0.3	0.4	1.0
6	Comparative Reasoning	0.1	0.0	0.0
7	Exploratory Recommendation	0.3	0.3	1.0
8	Preference-Driven Recommendation	0.3	1.0	1.0
9	Inference	0.3	0.3	1.0
10	Opinion Summarization	0.1	1.0	1.0