



GRAPH-RAG TRAVEL ASSISTANT

HOTEL

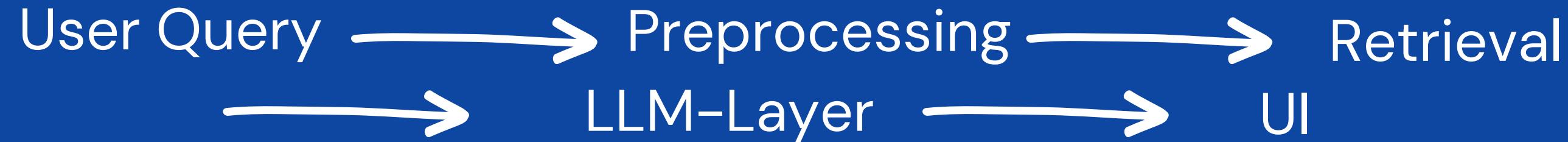


Presented by: Dina, Iten, Jana and Jolie

Hybrid Graph-RAG Architecture



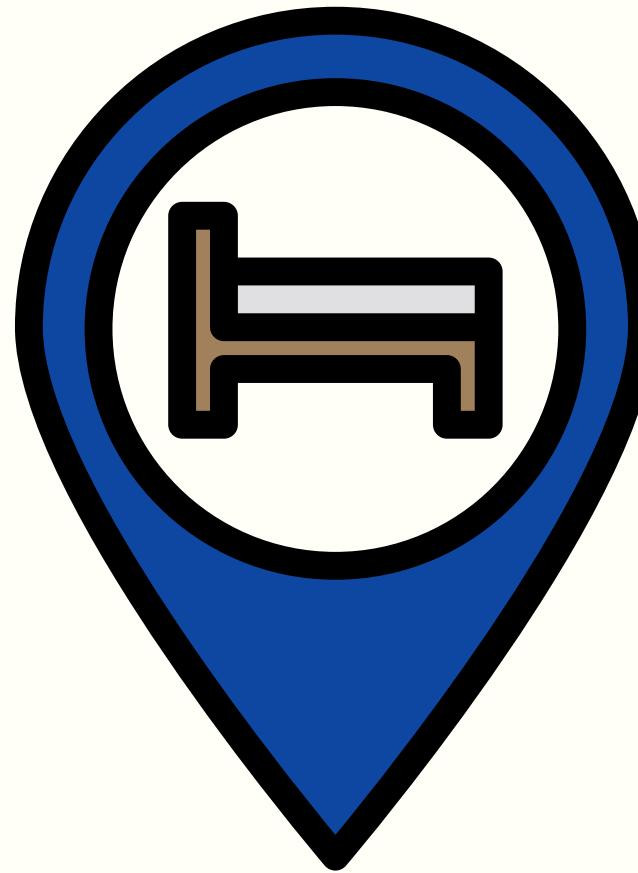
Below is the pipeline we followed to implement our higher-level system.



- The system implements a full RAG pipeline, using Neo4j the (Knowledge Graph) to provide answers generated by a lightweight LLM (HuggingFace Inference API).

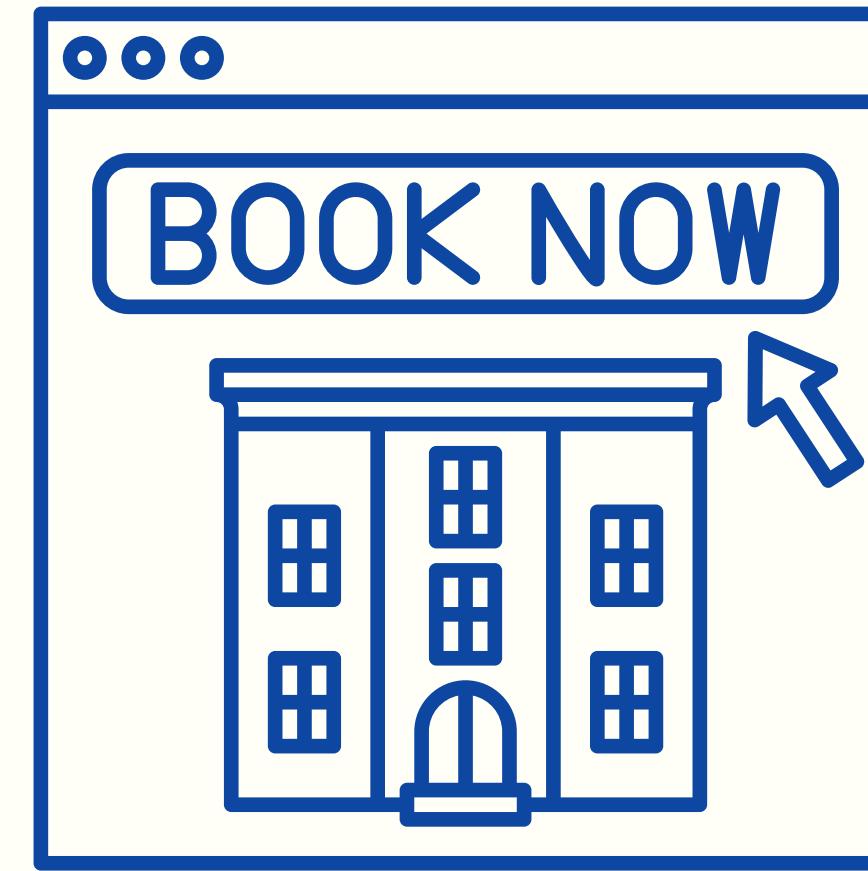


Tasks Analysis



Hotel Recommender

Suggests hotels based on multiple criteria (e.g., location, rating, amenities, traveler type).



Booking ad Visa Assistant

Provides specific visa requirements between two countries.





Input Preprocessing

Intent Classification

A rule-based matching classifier routes queries to the correct retrieval logic. Ex: "Do I need a visa to France?" **visa_query intent.** "Recommend a luxury hotel in Paris." **hotel_recommendation intent.**

Entity Extraction

Named Entity Recognition (NER) or rule-based extraction extracts parameters for Cypher query templates. **Query:** "Recommend a 5-star hotel in Cairo." **Extracted Entities:** *city="Cairo", star_rating=5.*

Input Embedding

The raw query is converted to a an *embedding vector* using the same embedding model as the KG, directly feeding into the **EmbeddingBased Retrieval layer.**



Component 2a: Baseline Retrieval



Our strategy relies on deterministic retrieval using structured Cypher queries.

- The retriever.py module selects the appropriate template based on the identified Intent and populates parameters using extracted Entities.
- We implemented a comprehensive library of over 10 query templates covering: Hotel Search, Review Retrieval, Amenity Filtering, Location-based Queries, and Visa Requirements.

// Query Template for 'visa_query' intent

```
MATCH (c:Country {name: $from_country})-[r:REQUIRES_VISA]->  
(t:Country {name: $to_country}) RETURN  
c.name AS from_country, t.name AS  
to_country, r.type AS visa_type,  
r.duration AS duration
```

Example Snippet

Egypt

France

Tourist Visa



Cypher Query Templates

// Query Template for 'visa_query' intent

```
MATCH (c:Country {name: $from_country})-[r:REQUIRES_VISA]->(t:Country {name: $to_country}) RETURN c.name AS from_country, t.name AS to_country, r.type AS visa_type, r.duration AS duration
```

// Query Template for 'visa_query' intent

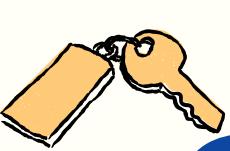
```
MATCH (c:Country {name: $from_country})-[r:REQUIRES_VISA]->(t:Country {name: $to_country}) RETURN c.name AS from_country, t.name AS to_country, r.type AS visa_type, r.duration AS duration
```

// Query Template for 'hotels_by_country' intent

```
MATCH (h:Hotel)-[:LOCATED_IN]->(c:City)-[:LOCATED_IN]->(co:Country)  
WHERE toLower(co.name) CONTAINS toLower($country)  
RETURN h.hotel_name AS hotel_name, c.name AS city, h.star_rating AS star_rating, h.cleanliness_base AS cleanliness, h.value_for_money_base AS value  
ORDER BY h.star_rating DESC LIMIT 10
```

// Query Template for 'hotels_by_continent' intent

```
MATCH (h:Hotel)-[:LOCATED_IN]->(c:City)-[:LOCATED_IN]->(co:Country)  
WHERE co.name IN $country_list  
RETURN h.hotel_name AS hotel_name, c.name AS city, co.name AS country, h.star_rating AS star_rating, h.cleanliness_base AS cleanliness, h.value_for_money_base AS value  
ORDER BY h.star_rating DESC LIMIT 15
```



Cypher Query Templates

```
// Query Template for 'filter_hotels_by_amenity' intent
MATCH (h:Hotel)-[:LOCATED_IN]->(c:City)-[:LOCATED_IN]->(co:Country)
  WHERE h.{amenity_key} >= $min_score
    AND ($country IS NULL OR toLower(co.name) CONTAINS toLower($country))
  RETURN h.hotel_name AS hotel_name,
    h.star_rating AS star_rating,
    h.{amenity_key} AS score,
    c.name AS city, co.name AS country,
    h.cleanliness_base AS cleanliness, h.value_for_money_base AS value
  ORDER BY score DESC LIMIT 10
```

```
// Query Template for
'hotels_by_value_for_money' intent
MATCH (h:Hotel)-[:LOCATED_IN]->(c:City)
  WHERE toLower(c.name) CONTAINS toLower($city)
  RETURN h.hotel_name AS hotel_name,
    h.star_rating AS star_rating,
    h.value_for_money_base AS value,
    h.cleanliness_base AS cleanliness
  ORDER BY h.value_for_money_base DESC LIMIT 10
```

```
// Query Template for 'filter_hotels_by_rating' intent
MATCH (h:Hotel)-[:LOCATED_IN]->(c:City)-[:LOCATED_IN]->(co:Country)
  WHERE h.star_rating >= $min_rating
    AND ($city IS NULL OR toLower(c.name) CONTAINS toLower($city))
    AND ($country IS NULL OR toLower(co.name) CONTAINS toLower($country))
  RETURN h.hotel_name AS hotel_name,
    h.star_rating AS star_rating, co.name AS country,
    h.cleanliness_base AS cleanliness, h.value_for_money_base AS Money, h.comfort_base AS comfort
  ORDER BY h.star_rating DESC LIMIT 10
```



Cypher Query Templates

// Query Template for 'recommend_hotels_for_traveller_type' intent

```
MATCH (t:Traveller {type: $traveller_type})-[:WROTE]->(rev:Review)-  
[:REVIEWED]->(h:Hotel)  
    MATCH (h)-[:LOCATED_IN]->(:City)-[:LOCATED_IN]->(co:Country)  
    WHERE ($country IS NULL OR toLower(co.name) CONTAINS  
toLower($country))  
    WITH h, AVG(rev.score_overall) AS score, co  
    RETURN h.hotel_name AS hotel_name,  
        h.star_rating AS star_rating,  
        score,  
        co.name AS country,  
        h.cleanliness_base AS cleanliness, h.value_for_money_base AS  
value  
ORDER BY score DESC LIMIT 10
```

// Query Template for 'compare_hotels' intent

```
UNWIND $hotel_list AS hn  
    MATCH (h:Hotel) WHERE toLower(h.hotel_name)  
CONTAINS toLower(hn)  
    RETURN h.hotel_name AS hotel_name,  
        h.star_rating AS star_rating,  
        h.cleanliness_base AS cleanliness,  
        h.value_for_money_base AS value,  
        h.comfort_base AS comfort
```

// Query Template for 'find_available_hotels' intent

```
MATCH (h:Hotel)-[:LOCATED_IN]->(c:City)  
    WHERE toLower(c.name) CONTAINS toLower($city)  
    AND h.cleanliness_base >= 7.0 AND  
h.value_for_money_base >= 7.0  
    RETURN h.hotel_name AS hotel_name,  
        h.star_rating AS star_rating,  
        h.cleanliness_base AS cleanliness,  
        h.value_for_money_base AS value  
    ORDER BY h.star_rating DESC LIMIT 10
```



Cypher Query Templates

// Query Template for 'hotels_for_family' intent

```
MATCH (t:Traveller {type: 'Family'})-[:WROTE]->(rev:Review)-  
[:REVIEWED]->(h:Hotel)  
MATCH (h)-[:LOCATED_IN]->(:City)-[:LOCATED_IN]->(co:Country)  
WHERE ($country IS NULL OR toLower(co.name) CONTAINS  
toLower($country))  
WITH h, AVG(rev.score_overall) AS score, co  
RETURN h.hotel_name AS hotel_name,  
h.star_rating AS star_rating,  
score,  
co.name AS country,  
h.cleanliness_base AS cleanliness, h.value_for_money_base AS  
value  
ORDER BY score DESC LIMIT 10
```

// Query Template for 'hotel_details' intent

```
MATCH (h:Hotel) WHERE toLower(h.hotel_name)  
CONTAINS toLower($hotel_name)  
OPTIONAL MATCH (h)<-[REVIEWED]-(r:Review)  
RETURN h.hotel_name AS hotel_name,  
h.star_rating AS star_rating,  
h.cleanliness_base AS cleanliness,  
h.value_for_money_base AS value,  
COLLECT(r.review_text)[0..3] AS sample_reviews
```

// Query Template for 'top_reviews_for_hotel' intent

```
MATCH (h:Hotel)<-[REVIEWED]-(rev:Review)  
WHERE toLower(h.hotel_name) CONTAINS  
toLower($hotel_name)  
RETURN h.hotel_name AS hotel_name,  
h.star_rating AS star_rating,  
rev.review_text AS review,  
rev.score_overall AS score  
ORDER BY score DESC LIMIT 5
```



Cypher Query Templates

```
// Query Template for 'top_hotels_in_country_or_city' intent
CALL {
    WITH $city AS city
    MATCH (h:Hotel)-[:LOCATED_IN]->(c:City) WHERE toLower(c.name)
    CONTAINS toLower(city) RETURN h
    UNION
    WITH $country AS country
    MATCH (h:Hotel)-[:LOCATED_IN]->(c:City)-[:LOCATED_IN]->(co:Country)
    WHERE toLower(co.name) CONTAINS toLower(country) RETURN h
}
RETURN h.hotel_name AS hotel_name,
    h.star_rating AS star_rating,
    h.cleanliness_base AS cleanliness, h.value_for_money_base AS value
ORDER BY h.star_rating DESC LIMIT 10
```

```
// Query Template for
'hotel_reviews_by_age_group' intent
MATCH (t:Traveller)-[:WROTE]->(rev:Review)-
[:REVIEWED]->(h:Hotel)
WHERE toLower(h.hotel_name) CONTAINS
toLower($hotel_name)
RETURN t.age_group AS age_group,
    AVG(rev.score_overall) AS avg_score,
    h.star_rating AS star_rating
ORDER BY avg_score DESC
```



Component 2b: Embedding-Based Retrieval

We selected Feature Vector Embeddings for the Hotel theme.

- Key textual properties (e.g., hotel name, city, amenities, average review) are concatenated into a single feature vector.
- The user's query vector is then compared against these stored feature vectors using Vector Similarity Search in Neo4j

• Embedding Models Compared

1. Model A :

model_A(sentencetransformers/all-MiniLM-L6-v2)

2. Model B :

model_B(sentencetransformers/mpnet)



Component 2b: Retrieval Accuracy of Embedding Models



EMBEDDING MODEL COMPARISON

QUERY: 'Find a family-friendly hotel in London with good value'

[MODEL: MiniLM]

Retrieval Time: 0.03s

1. The Royal Compass | Similarity: 0.7550

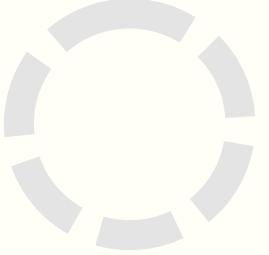
[MODEL: MPNet]

Retrieval Time: 0.10s

1. The Royal Compass | Similarity: 0.7160



Component 2b: Retrieval Accuracy of Embedding Models



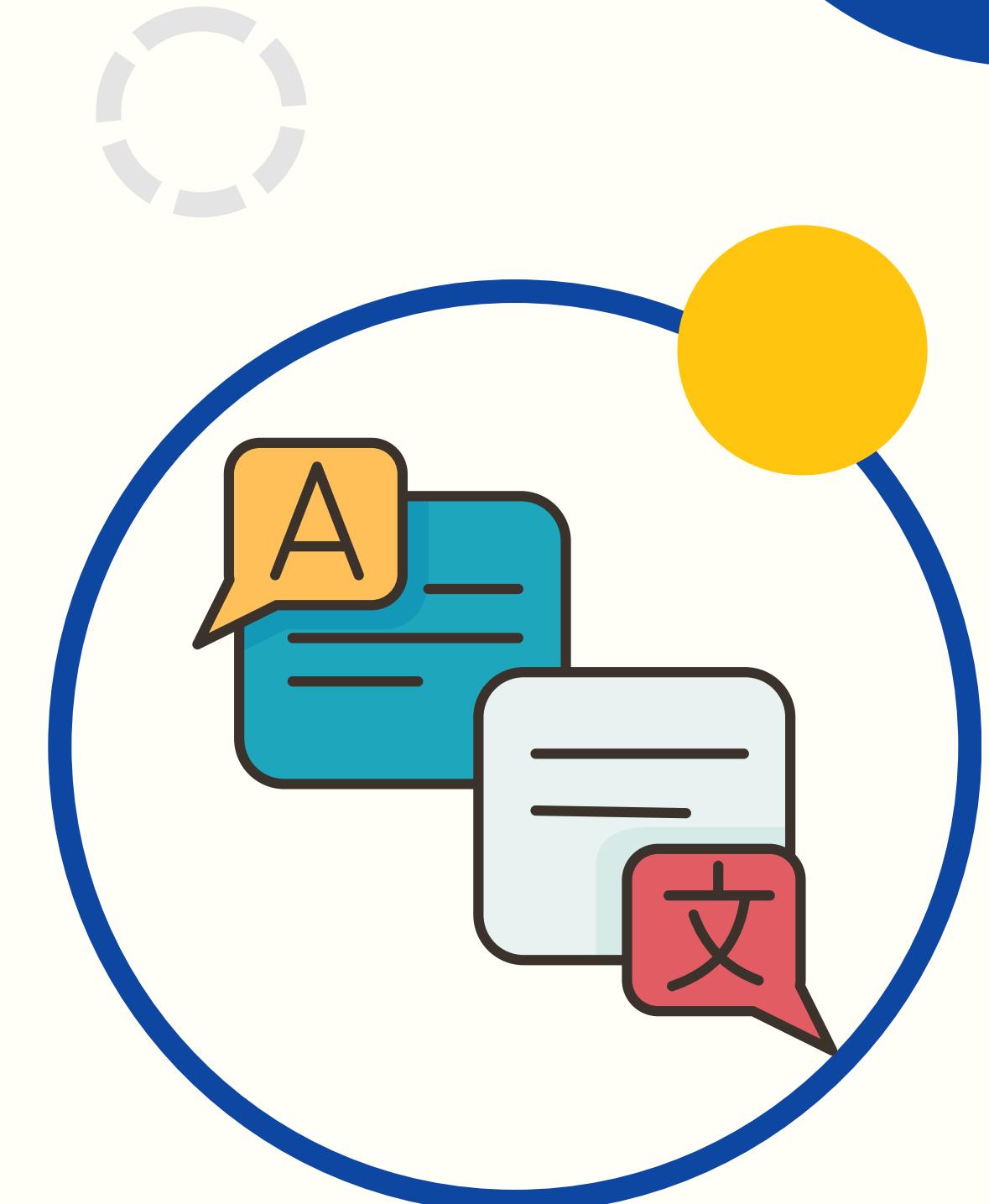
```
=====
          EMBEDDING MODEL COMPARISON
QUERY: 'Dina wants to travel to Spain but she is traveling from Egypt and she does not have a visa. Does she need one? Also recommend the best hotels in Spain. Since she was young it was her dream to visit Spain and walk in the streets there and go to Camp Nou.'
=====

[MODEL: MiniLM]
  Retrieval Time: 0.17s
  1. Gaudi's Retreat | Similarity: 0.6274

[MODEL: MPNet]
  Retrieval Time: 0.16s
  1. Gaudi's Retreat | Similarity: 0.6724
```

LLM-Layer: Retrieval Fusion

The **merge_retrievals** function deduplicates facts from Cypher and Vector retrieval using a canonical key (e.g., *hotel_name* for hotels, *from_country* for visas). This provides a single, clean, ranked, and comprehensive list of facts for the LLM as its factual context to later compose the final answer from.



LLM-Layer: Strict Grounding Prompt Construction

The prompt is divided into three mandatory sections:

- **Context:** Fused, cleaned KG facts presented as a numbered *list of key-value pairs*.
- **Persona:** "You are a *helpful travel assistant specializing in hotels and visa information*. You **ONLY** answer based on verified facts..."
- **Task:** "Answer the user's question strictly using **ONLY** the provided context facts. If the context does not contain the answer, say: '*The required information is not in the knowledge graph.*'"



Model Selection



Model 1:

google/gemma-2-2b-it (*Fastest, Small*)



Model 2:

Qwen/Qwen2.5-1.5B-Instruct (*Balanced*)



Model 3:

mistralai/Mistral-7B-Instruct-v0.3 (*Largest, Strongest Reasoning*)



Model Evaluation I

Evaluate LLM Models

Run Model Evaluation

Evaluation Completed

Summary Metrics per Model

1

- Accuracy: 100.0%
- Total Time: 4.03s

2

- Accuracy: 100.0%
- Total Time: 7.17s

3

- Accuracy: 100.0%
- Total Time: 9.51s

Show Detailed Test Outputs

Model 1 Detailed Outputs

- Question: Recommend a luxury 5-star hotel in Paris.
- Answer: L'Étoile Palace is a luxury 5-star hotel in Paris.
- Passed: True
- Time: 1.31s

- Question: Which hotel in Tokyo has excellent cleanliness?
- Answer: The Kyo-to Grand in Tokyo has excellent cleanliness.
- Passed: True
- Time: 0.64s

- Question: Suggest a hotel with the best location in Rome.
- Answer: The hotel_name: Colosseum Gardens is located in Rome and has an excellent location.
- Passed: True
- Time: 0.72s

Model Evaluation II

Model 2 Detailed Outputs

- Question: Recommend a luxury 5-star hotel in Paris.
- Answer: L'Étoile Palace
- Passed: True
- Time: 2.35s

-
- Question: Which hotel in Tokyo has excellent cleanliness?
 - Answer: Kyo-to Grand
 - Passed: True
 - Time: 1.14s

-
- Question: Suggest a hotel with the best location in Rome.
 - Answer: Colosseum Gardens
 - Passed: True
 - Time: 1.15s

Model 3 Detailed Outputs

- Question: Recommend a luxury 5-star hotel in Paris.
- Answer: Based on the provided context, I recommend L'Étoile Palace, a luxury 5-star hotel located in Paris.
- Passed: True
- Time: 1.84s

-
- Question: Which hotel in Tokyo has excellent cleanliness?
 - Answer: Kyo-to Grand in Tokyo is reported to have excellent cleanliness based on the provided context facts.
 - Passed: True
 - Time: 1.97s

-
- Question: Suggest a hotel with the best location in Rome.
 - Answer: Based on the context provided, the hotel with an excellent location in Rome is the Colosseum Gardens.
 - Passed: True
 - Time: 1.94s

— Model Qualitative Analysis —

Criterion	Mistral-7B	Qwen-2.5-1.5B	Gemma-2-2B
Adherence to TASK (Grounding)	Excellent	Very Good	Good (Occasional extra fluff)
Readability/Naturalness	Highest	High	Moderate
Handling Missing Context	Flawless (Gives exact error message)	Very Good (Gives exact error message)	Occasionally tries to infer

Live Demo: Model Choices



Setting	Selection	Justification
Retrieval Strategy	Hybrid	Shows the power of fusing Cypher and Vector results.
Embedding Model	Model A (Default)	Used for generating the vector for the user query.
LLM Model	Qwen-2.5-1.5B	Best trade-off between speed and accuracy.
Test Query	<p><i>“Dina wants to travel to Spain but she is traveling from Egypt and she does not have a visa. Does she need one? Also recommend the best hotels in Spain. Since she was young it was her dream to visit Spain and walk in the streets there and go to Camp Nou.”</i></p>	Demonstrates handling of Multi-Intent queries.

Live Demo: User Prompt

Graph-RAG Travel Assistant

ge-graph grounded hotel recommendations & visa answers

question

A user asks about visa requirements for traveling to Spain from Egypt.

answer

The travel assistant provides information about visa requirements and recommends visiting Camp Nou.





Live Demo: Query Understanding

Detected intents: ['check_visa_requirement', 'filter_hotels_by_amenity']

```
▼ {  
  ▼ "locations" : [  
    0 : "Egypt"  
    1 : "Spain"  
  ]  
  ▶ "hotel_names" : []  
  "traveller_type" : NULL  
  ▼ "amenities_scores" : [  
    ▼ 0 : {  
      "db_column" : "facilities_base"  
    }  
  ]  
  ▶ "target_countries" : []
```

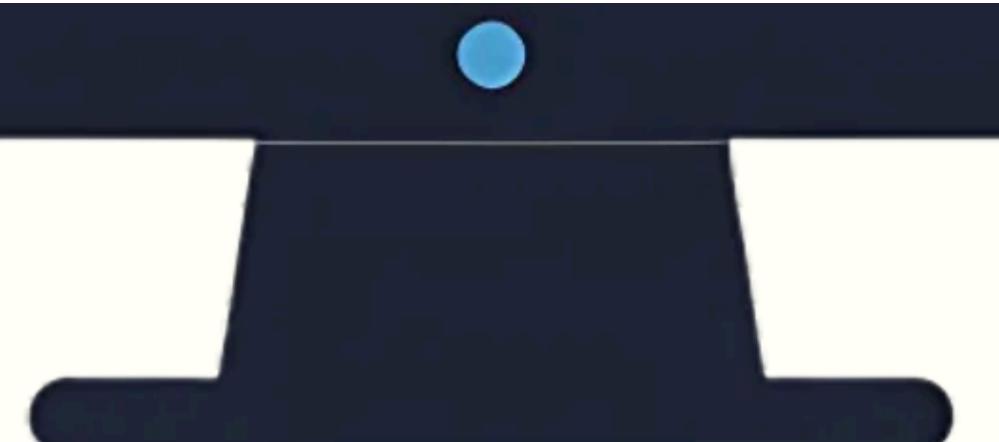




Live Demo: Retrieved Context

Retrieved Context

_country	to_country	requirement	visa_type	hotel_name	star_rating	score	city	country
United States	Spain	Visa required	Tourist Visa	None	None	None	None	None
United States	Spain	None	None	Gaudi's Retreat	5	8.9	Barcelona	Spain
United States	Spain	None	None	None	None	0.6287	None	Spain
United States	Spain	None	None	None	None	0.8261	None	None
United States	Spain	None	None	None	None	0.8772	None	None
United States	Spain	None	None	None	None	0.9358	None	None
United States	Spain	None	None	None	None	0.9723	None	None
United States	Spain	None	None	None	None	0.9726	None	None





Live Demo: Final LLM Answer

Answer ↗

needs a visa to enter Spain as per the given context. The visa type depends on the reason of travel. If you are traveling for a second request, I would recommend the following hotels in Spain:

Hotel i's Retreat - Located in Barcelona, Spain. It has a high rating of 8.9 on Booking.com. The hotel offers free Wi-Fi, a swimming pool, and a fitness center. The hotel also offers a great location (9.7) and good staff service (9.6).

These details should help you plan your trip to Spain!



Live Demo: Final Insights

Chosen Model for Demo

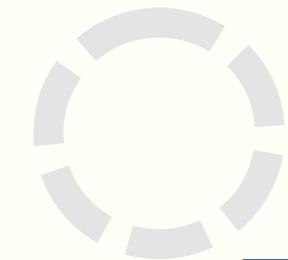
We selected Mistral (Key 3) as the default for the live demo due to its high accuracy and lowest latency, offering the best user experience.

Live Demonstrations

The UI allows switching between all three models live, enabling direct comparison of performance differences.



Error Analysis, Improvements, and Remaining Limitations



A. Error Analysis (Example Hotel Theme):

- Failure Case: User asks "Find a hotel with a spa and pool, but avoid the Novotel."
- Why it Failed: Baseline Cypher queries lack complex logic (e.g., A AND B BUT NOT C), leading to incomplete retrieval. Ambiguous context can cause the LLM to choose the wrong option.



Error Analysis, Improvements, and Remaining Limitations

B. Improvements Added:

- Retrieval Fusion **merge_retrievals** function overcomes weaknesses of single retrieval strategies, ensuring both semantic (pool/spa) and structured (city/rating) facts are used simultaneously.

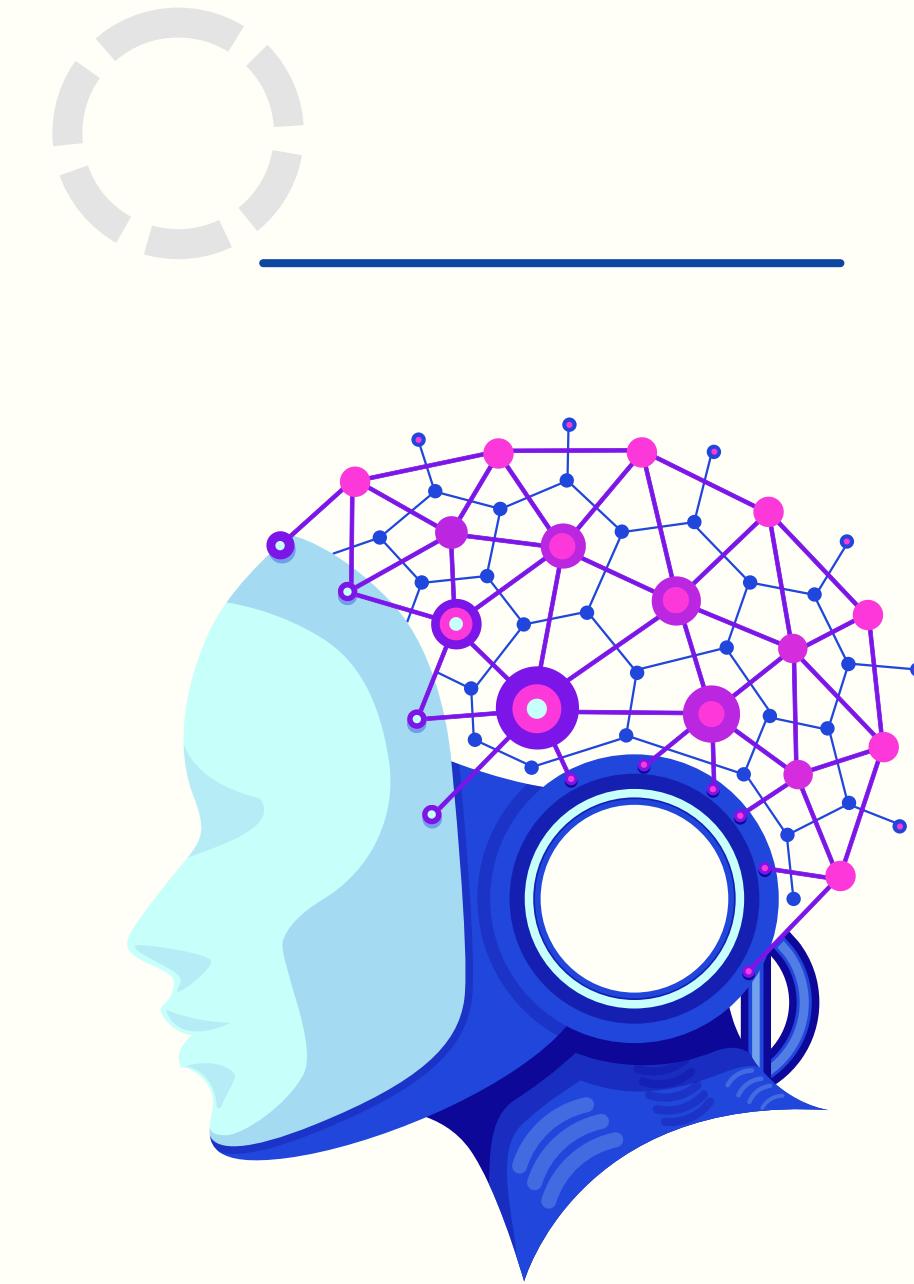


B. Improvements Added:

- Strict Prompting The extremely strict TASK rule forces the LLM to output "The required information is not in the knowledge graph" instead of hallucinating.

Error Analysis, Improvements, and Remaining Limitations

- C. Remaining Limitations:
- No Multi-Hop Reasoning in RAG: Current system doesn't allow the LLM to trigger sequential Cypher queries (Agentic RAG).
- Embedding Model Dependency: Semantic search quality is highly dependent on the underlying embedding model's quality and domain adaptation.
- Cypher Template Scalability: Maintaining and updating the library of g10 Cypher templates remains a high-effort engineering task.



Thank You

