

```
# Smart Travel Guide: A Comparative RAG Performance Analysis
# (GPT-4.1-Mini vs. Gemini-2.5-Flash)
```

Project Overview

This project implements a Retrieval-Augmented Generation (RAG) based chatbot designed to provide comprehensive information about European cities (e.g., Rome, Prague). The system covers diverse travel categories, including local food culture, urban transportation, top photography spots, historical landmarks, and major tourist attractions. The primary objective is to evaluate and compare the performance of two different Large Language Models (LLMs)—OpenAI GPT-4.1-Mini and Google Gemini-2.5-Flash—using a multi-faceted local dataset.

Technical Architecture

The system follows a standard RAG pipeline:

Frontend: Streamlit

Orchestration: LangChain

Vector Database: ChromaDB

Retrieval Strategy:

- Similarity search
- k=6 for interactive chatbot usage
- k=1 for evaluation to enforce strict intent matching

Models:

- * Gpt-4.1-Mini with OpenAI Embeddings
- * Gemini-2.5-Flash with Google Generative AI Embeddings

Repository Structure

The project is organized according to the specified directory structure:

```
```text
chatbot_project/
 data/
 avrupa_gezi_rehberi_1000.xlsx # Custom dataset
 models/
 openai_model.py # OpenAI RAG
 implementation (builds vector DB)
 gemini_model.py # Gemini RAG
 implementation (builds vector DB)
 evaluation/
 build_train_vector_db.py # Vector DB for
evaluation
```

```

| └── evaluate_models.py # Model comparison &
metrics
├── app/
│ └── streamlit_app.py # Streamlit user
interface
└── video/
 └── demo_video.mov # Demo video
 └── README.md # Project
documentation
└── Smart Travel Guide.pdf
└── requirements.txt # Dependency list

```

### **## Performance Metrics & Benchmarking**

The models were evaluated using intent-level classification metrics rather than token-level text similarity. Each test question was matched against the vector database built from the training set, and the predicted intent was compared with the ground-truth intent label.

Evaluation was performed using the following metrics:

- Precision
- Recall
- F1 Score

A strict evaluation setup was used where:

- The vector database was built **\*\*only on the training set\*\***
- The test set was completely unseen during indexing
- Retrieval was performed with k=1 to avoid intent leakage

### **### Quantitative Results**

Model	Precision	Recall	F1 Score
<b>**OpenAI (gpt-4.1-mini)**</b>	0.96	0.96	0.96
<b>**Google (gemini-2.5-flash)**</b>	0.94	0.95	0.94

### **### Response Time Comparison**

Model	Average Response Time (s)
<b>**OpenAI (gpt-4.1-mini)**</b>	<b>**1.48**</b>
<b>**Google (gemini-2.5-flash)**</b>	<b>**1.74**</b>

### **## How to Run**

1. Clone the repository
2. Install dependencies:  
`pip install -r requirements.txt`
3. Setup Environment Variables:  
Create a credentials.env file and add your API keys:  
`OPENAI\_API\_KEY=your\_openai\_key`  
`GOOGLE\_API\_KEY=your\_google\_key`
4. Build vector databases and run model demos:  
`python models/openai\_model.py`  
`python models/gemini\_model.py`
5. Build training vector database and run model evaluation  
`python evaluation/build\_train\_vector\_db.py`  
`python evaluation/evaluate\_models.py`
6. Launch the App:  
`streamlit run app/streamlit\_app.py`

### **## Demo Video**

A detailed demonstration of the chatbot, including the "See Raw Data" feature which shows the retrieved context from ChromaDB, can be found [video/demo\_video.mp4].

### **## Example Outputs**

```
--- Gemini 2.5-Flash RESULT ---
Viyana mutfağında özellikle Şinitzelini denemelisiniz.
--- OpenAI (GPT-4.1-mini) Result ---
Viyana mutfağında özellikle şinitzel denemenizi öneririm.
```

## **European Travel Guide RAG Comparison**

GPT-4.1-mini vs Gemini-2.5-Flash

Gezi hakkında bir soru sorun (Örn: Roma'da ne yerin?)

Paris'te yemek kültürü nasıldır?

### **OpenAI (GPT-4.1-mini)**

Paris mutfağında özellikle Kruvasan ve Makaronları denemelisiniz.

Response Time: 1.80s

### **Google (Gemini-2.5-Flash)**

Paris mutfağında özellikle Kruvasan ve Makaronları denemelisiniz.

Response Time: 1.83s

## ## Raw Data Columns

▼ OpenAI Raw Docs	▼ Gemini Raw Docs
<b>Doc 1:</b> Paris yemek kültürü nasıl?	<b>Doc 1:</b> Soru: Paris yemek kültürü nasıl? Cevap: Paris mutfağında özellikle Kruvasan ve Makaronları denemelisiniz.
<b>Doc 2:</b> Paris yemek kültürü nasıl?	<b>Doc 2:</b> Soru: Paris yemek kültürü nasıl? Cevap: Paris mutfağında özellikle Kruvasan ve Makaronları denemelisiniz.
<b>Doc 3:</b> Paris yemek kültürü nasıl?	<b>Doc 3:</b> Soru: Paris yemek kültürü nasıl? Cevap: Paris mutfağında özellikle Kruvasan ve Makaronları denemelisiniz.
<b>Doc 4:</b> Paris yemek kültürü nasıl?	<b>Doc 4:</b> Paris hakkında yemek kültürü nasıldır öğrenebilir miyim?
<b>Doc 5:</b> Paris yemek kültürü nasıl?	
<b>Doc 6:</b> Paris yemek kültürü nasıl?	