Transport Info Bot with Retrieval-Augmented Generation (RAG)

Submitted by:

N. Harini

Roll No: 23B21A4305

Branch: CSE (AI)

Batch: 2023–2027

K. Hemanth

Roll No: 23B21A4314

P. Siva Rama

Roll No: 23B21A4321

B. Prakash

Roll No: 23B21A4330

D. Kartheek

Roll No: 236Q1A4301

Abstract

The Transport Info Bot with Retrieval-Augmented Generation (RAG) is a next-generation conversational assistant designed to provide real-time, reliable, and context-aware transportation information. Unlike traditional static transport apps, this bot dynamically retrieves updated schedules, traffic conditions, fare details, and route availability from verified transport datasets and APIs, then generates user-friendly responses using large language models (LLMs). By integrating RAG, the system delivers fact-grounded answers while maintaining natural, adaptive interactions, making it a practical solution for urban commuters, logistics providers, and travelers .

Introduction

Transportation plays a critical role in daily life, yet users face challenges in finding accurate and updated travel information—such as bus/train schedules, traffic updates, or fare estimates. Existing transport applications are often limited to static data, single modes of transport, or lack natural interaction.

The Transport Info Bot, powered by RAG, bridges this gap by retrieving the most current transport details (from government databases, APIs, and traffic feeds) and combining them with AI-driven conversational ability. This ensures commuters get relevant, real-time, and contextual travel information across multiple modes of transport [1][3][4].

Problem Statement:

Challenges in current transport information systems include:

1. Static or outdated route and schedule details.

2. Poor adaptability to user-specific queries (e.g., “What’s the cheapest way from City A to City B now?”).

3. Limited integration across multiple transport modes (buses, trains, metros, ride-sharing).

4. Lack of personalized assistance during disruptions (traffic, cancellations, strikes).

A dynamic AI solution is needed to offer accurate, updated, and personalized transport guidance.

Objectives:

Develop a smart assistant to provide real-time transport information using RAG.

Fetch updated schedules, delays, fares, and routes from APIs and databases.

Support multi-modal transport planning (bus, train, metro, cab, flights).

Offer personalized travel suggestions based on user queries and preferences.

Enhance accessibility with natural language conversation, voice integration, and multi-language support.

Literature Review:

Traditional transport systems like Google Maps or static railway apps rely mainly on database lookups and fixed APIs. They lack contextual conversation and adaptability. RAG-based systems have shown effectiveness in fact-grounded query answering, making them ideal for transport scenarios where accuracy and timeliness are critical.

Recent RAG applications in smart assistants and public service chatbots demonstrate improved reliability, scalability, and user trust . Transport-specific bots powered by RAG are still emerging, presenting significant potential for research and implementation.

System Architecture

Modules:

1. User Interface (UI): Chat/web/mobile interface for commuters.

2. Retriever Module: Gathers updated transport data from APIs, traffic sensors, and schedules.

3. Generator Module: Uses LLMs to generate natural, context-aware responses.

4. Transport Engine: Calculates optimal routes, multimodal connections, and fare comparisons.

5. Database: Stores static schedules, fare structures, user profiles, and past queries.

The system architecture ensures a seamless blend of retrieval (accuracy) and generation (conversational flexibility) .

Methodology:

1. Data Collection: Collect schedules, traffic reports, fare details, and geo-data from APIs (Google Transit, Indian Railways API, Uber, Ola, government portals).

2. Preprocessing: Convert transport datasets into embeddings stored in a vector DB (FAISS, Pinecone).

3. Retriever (R): Fetches real-time schedules and fare updates for user queries.

4. Generator (G): Produces conversational, natural responses grounded in retrieved data.

5. Transport Engine: Recommends the best routes, accounting for cost, time, and delays.

6. Integration: Combines RAG pipeline with APIs to serve end-users dynamically.

Implementation Details:

Frontend: React (for web) / Flutter (for mobile).

Backend: Python with Flask/Django for orchestration.

Databases:

Vector DB (FAISS/Pinecone) for embeddings.

SQL/NoSQL (PostgreSQL/MongoDB) for static schedules and user history.

Model: RAG pipeline with LLM (GPT, LLaMA, or fine-tuned transport-specific model).

Deployment: Cloud hosting (AWS Lambda, GCP Functions, Azure).

APIs: Integration with real-time APIs (Transit, Google Maps, IRCTC, Uber/Ola).

Security: User authentication with encrypted data storage.

Voice Support: Integration with Whisper/Google Speech-to-Text for voice queries [5].

Use Cases:

1. Daily Commute Info: Get real-time bus/train delays, next arrivals.

2. Trip Planning: Suggest fastest or cheapest routes combining multiple transport modes.

3. Fare Estimation: Provide cost comparison between metro, cab, bus.

4. Traffic & Disruption Alerts: Notify users about delays, strikes, roadblocks.

5. Tourism Assistance: Suggest transport options for outstation trips.

Advantages & Applications:

Reliable, up-to-date transport guidance.

Multi-modal integration improves commute planning.

Reduces dependency on multiple apps by centralizing info.

Useful for commuters, tourists, logistics companies, and governments.

Can extend to smart cities and IoT-based transport networks.

Limitations:

Accuracy depends on API/data availability.

Higher infrastructure cost for real-time processing.

Handling network failures or outdated datasets remains a challenge.

Multimodal data harmonization is complex.

Future Scope:

Voice and AR integration for hands-free commuter support.

Predictive analytics for traffic and fare fluctuations.

Integration with IoT (smart buses, smart metros).

Multilingual support for regional users.

Personalized alerts for regular routes (delays, disruptions).

Conclusion:

The Transport Info Bot with RAG demonstrates how AI can revolutionize travel assistance by merging retrieval of real-time transport data with conversational intelligence. The system empowers users to make better commuting decisions, enhances reliability in public transport, and supports smart city initiatives. With continuous improvements, the bot can become a universal transportation assistant for millions of users worldwide.