Smart Transportation & Traffic Optimization System

About Project

This project implements a Smart Transportation and Traffic Optimization system based on a ReAct (Reasoning + Acting) agent architecture combined with Retrieval-Augmented Generation (RAG). The system is designed not as a static question-answering model, but as an intelligent decision-making agent capable of reasoning, calling tools, and synthesizing structured responses.

Core Architecture

- Core Model: LLaMA-based large language model
- Reasoning Loop: Thought → Action → Observation
- RAG Engine: Lightweight in-memory vector store (BoW + Cosine Similarity)
- External Tools: Live Traffic API, Regulation Search, Safety Documents, Math Tool
- Optimization: Top-K chunk retrieval to reduce latency and hallucination

Motivation

Traditional traffic information systems provide raw data without contextual reasoning. This project aims to overcome that limitation by combining real-time traffic data, legal regulations, and safety guidelines into a single reasoning pipeline. The goal is to deliver context-aware, explainable, and verifiable traffic-related decisions.

Data Sources

- Live Traffic Data (TomTom API)
- Traffic Safety Guidelines (PDF-based RAG)
- Highway Traffic Regulations (PDF-based RAG)
- Mathematical Rules (e.g., safe following distance calculations)

Agent Workflow Example

Question: “Check traffic congestion in Esenyurt. If congestion exceeds 25%, give safety advice and calculate following distance.”

The agent first calls the traffic API, observes congestion levels, then queries the SAFETY document using RAG, and finally applies mathematical reasoning to calculate safe following distance.

Benchmark Evaluation

The system was evaluated using scenario-based benchmarks covering API usage, RAG retrieval, and multi-hop reasoning. Metrics include accuracy, tool usage correctness, and reasoning step count.

Benchmark Evaluation Results (Short Academic Commentary)

The benchmark results indicate that the system successfully produced a valid final answer for all test cases, achieving an OK rate of 100%, which demonstrates stable end-to-end response generation. However, the tool accuracy remained at 0%, indicating that although correct answers were produced, the agent failed to explicitly satisfy the benchmark's expected tool-usage constraints. The average judge score of 28.75 suggests that answer quality varied significantly across scenarios, with several cases receiving partial or zero scores due to insufficient grounding, missing justifications, or lack of verifiable tool-based evidence. Despite these limitations, the system achieved a low average latency of 1.55 seconds, highlighting its computational efficiency and fast inference capability. Overall, the results show that while the model is capable of generating coherent and timely responses, improvements are required in tool invocation compliance, reasoning trace visibility, and evidence-backed answer construction to meet strict benchmark evaluation criteria.

Limitations

- API rate limits can affect real-time responses
- RAG quality depends on document coverage
- Some failures originate from tool-level data gaps rather than model reasoning

Conclusion

This project demonstrates that combining ReAct agents with RAG and live APIs enables reliable, explainable, and context-aware intelligent transportation systems. The architecture is extensible to other smart city domains.