Adaptive Global LCA Advisor with Dynamic EF Retrieval

Carbon Footprint Recommendation System

Team Members:

1. Surendra Bursu
2. Vinith Reddy
3. Ankita Bondre

Problem

Data Gaps: Most food products (~73%) lack environmental impact labels, complicating accurate carbon footprint analysis.

Manual Errors: Reliance on manual EF selection leads to high error rates (15–30% in reports), undermining consistency and reliability.

Regulatory Pressure: Emerging regulations (e.g. EU CBAM) demand precise, region-specific EFs for compliance, which is challenging with static data sources.

Motivation

Automate emission factor (EF) recommendations to reduce human error in carbon footprint calculations.

Overcome limitations of existing tools (static datasets or single-region scope) by combining multiple EF sources for global coverage.

Ensure recommendations remain current by incorporating real-time emissions data and periodic updates.

Literature Review

Parakeet (2022): AI-based EF suggestion tool limited by static datasets (no continuous updates).

LEAF (2021): Provides regional EFs but lacks global adaptability and real-time data integration.

Manual LCA Methods: Traditional life-cycle assessment requires manual EF lookup, highlighting the need for automation.

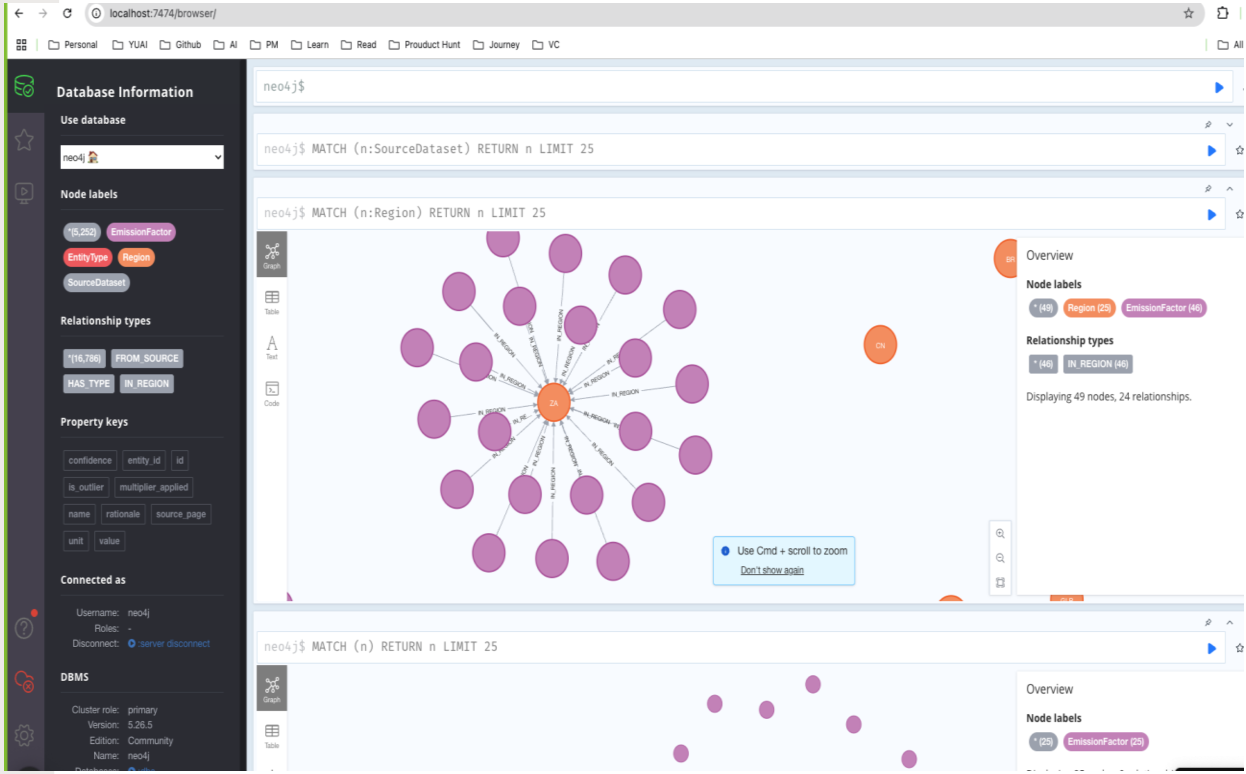
Our Novelty: We combine a fine-tuned LLM with a knowledge graph and vector search to provide dynamic, context-aware EF recommendations (a hybrid approach not seen in prior work).

Data Sources

Agribalyse 3.1 – French agri-EFs (2,793 records)
USEEIO v2.1 – US EEIO model (13,561 records)
EXIOBASE 3.8 – Multi-regional IO (1,030 records)
Climate TRACE – Global emissions by sector (4,681 records)
IPCC AR6 – Regional & gas-specific multipliers (10,769 records)
OpenLCA – Process-based LCA data (961 records)
IPCC EFDB – Sector & gas emission factors (191 records)
GREET Model – Transport fuel LCA (234 records)

Data Pipeline

- **Multi-Source Integration:** Collected and aggregated EF data from diverse databases (Agribalyse, USEEIO, EXIOBASE, OpenLCA, IPCC EFDB, Climate TRACE, etc.) via automated scripts.
- **Data Harmonization:** Cleaned each dataset and standardized units and categories into a common format (unifying different terminologies and measurement units).
- **Regional Adjustment:** Applied IPCC AR6 regional multipliers to emission values, tailoring global factors to local conditions for accuracy.
- **Graph Ingestion:** Loaded the consolidated, enriched data into a Neo4j knowledge graph – defining EF nodes linked to region and industry nodes (relationships like *PRODUCED_IN* and *HAS_IMPACT*).



Solution Overview

EF Recommendation System: An AI-driven solution that suggests appropriate emission factors based on a user's query (specific process/product and region).

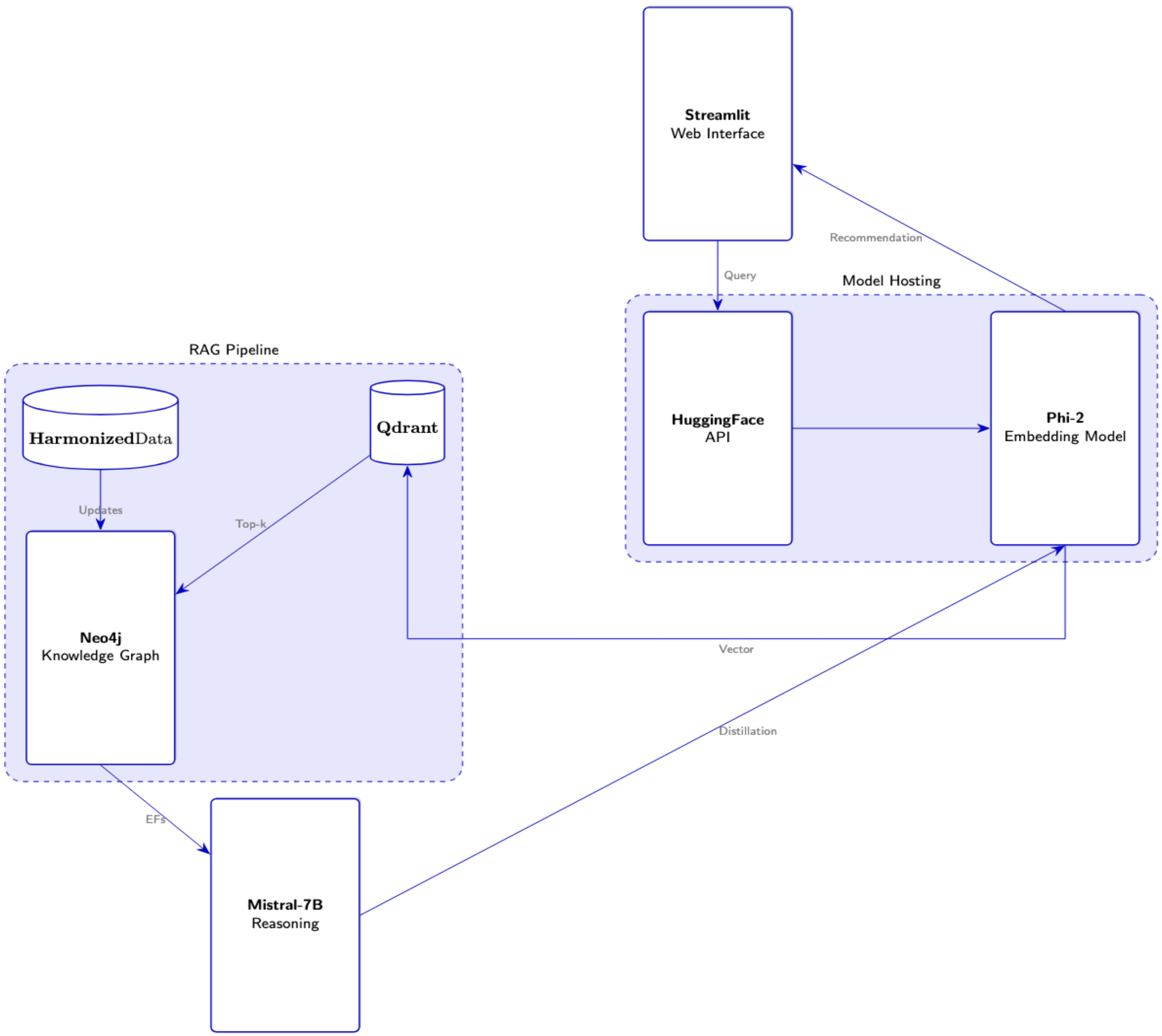
Knowledge Graph Backbone: Central Neo4j graph that integrates multiple EF sources, providing structured relationships between emissions data, regions, and industries.

LLM-Powered Reasoning: A fine-tuned language model (Mistral-7B) with retrieval augmentation generates context-aware answers using the graph data as reference.

Real-Time Updates & UI: Supports periodic data updates (e.g. new industry data via API) to keep EFs up-to-date, delivered through a user-friendly web interface for analysts/auditors.

System Architecture

- **Mistral-7B + LoRA** – Fine-tuned reasoning engine for context-aware response generation
- **Phi-2 + LoRA (Merged & Optimized)** – Embedding generator; distilled, pruned & quantized for efficient vector creation
- **Qdrant Vector Database** – In-memory vector store for ultra-fast semantic retrieval
- **Hugging Face Inference API** – Model hosting & serving via `Surendra-Aitest/phi2-env-factors-merged`
- **Graph Integration:** The Neo4j knowledge graph filters and enriches the results by ensuring they align with the query's context (e.g. matching the requested region or industry), improving answer relevance.
- **Streamlit Deployment:** The inference models are optimized for edge deployment – distilled and 4-bit quantized – allowing them to run locally on a standard PC or mobile device without cloud dependency (offline-capable).


Streamlit
Web Interface
Query
Recommendation
Model Hosting
HuggingFace
API
Phi-2
Embedding Model
RAG Pipeline
HarmonizedData
Qdrant
Updates
Top-k
Neo4j
Knowledge Graph
Vector
Distillation
EFs
Mistral-7B
Reasoning

Model Fine-Tuning & Distillation

- Generate instruction Q&A pairs from knowledge graph data
- Fine-tune Mistral-7B using LoRA adapters (rank=16) on GPUs
- Achieved >85% Precision@3 and ~5% MAPE on validation queries
- Knowledge-distill into Phi2 and quantize to 4-bit

Retrieval Mechanism

- **GraphRAG Approach:** Utilizes Retrieval-Augmented Generation, combining semantic vector search with graph-based filtering to produce precise, context-aware answers.
- **Semantic Search:** Transforms the user query into an embedding and retrieves the most similar EF entries using the Qdrant vector database (k-NN search on EF embeddings for fast candidate lookup).
- **Graph Filtering:** Applies Neo4j graph queries to refine these candidates, enforcing context from the knowledge graph (for example, matching the query’s region and industry nodes to relevant EF data).
- **Answer Composition:** The LLM then composes the final answer using the filtered EF information, providing the user with the recommended emission factor and relevant context. The retrieval pipeline is optimized (with caching of frequent queries) to answer queries in real-time (~<200ms per query).

Graph RAG Chunking strategies

Data Chunking Configuration

- Default chunk size: 512 tokens
- Default chunk overlap: 128 tokens
- These values can be configured through environment variables or config file

- **Text Chunking Types**
- The system creates three types of chunks:
- a) **Individual Emission Factor Chunks**
- b) **Region Summary Chunks**
- c) **Entity Type Summary Chunks**

- **Embedding Generation Chunking**
- Batch size for embedding generation: 32 (configurable)
- Maximum sequence length: 512 tokens
- Uses mean pooling for embedding generation
- Normalizes embeddings (L2 norm = 1)

Evaluation Metrics

- **Perplexity Scoring**
- Primary metric for evaluating answer quality
- Lower scores indicate better alignment with expected answers
- Calculated based on word matching between expected and actual answers
- Score interpretation:
- < 1.0: Good match with expected answer pattern
- 1.0-2.0: Partial match with expected answer pattern
- > 2.0: Poor match with expected answer pattern

- **System Performance Metrics**
- Response Time (in milliseconds)
- Query Volume (number of queries per day)
- System Status (component health)
- Average Perplexity Score

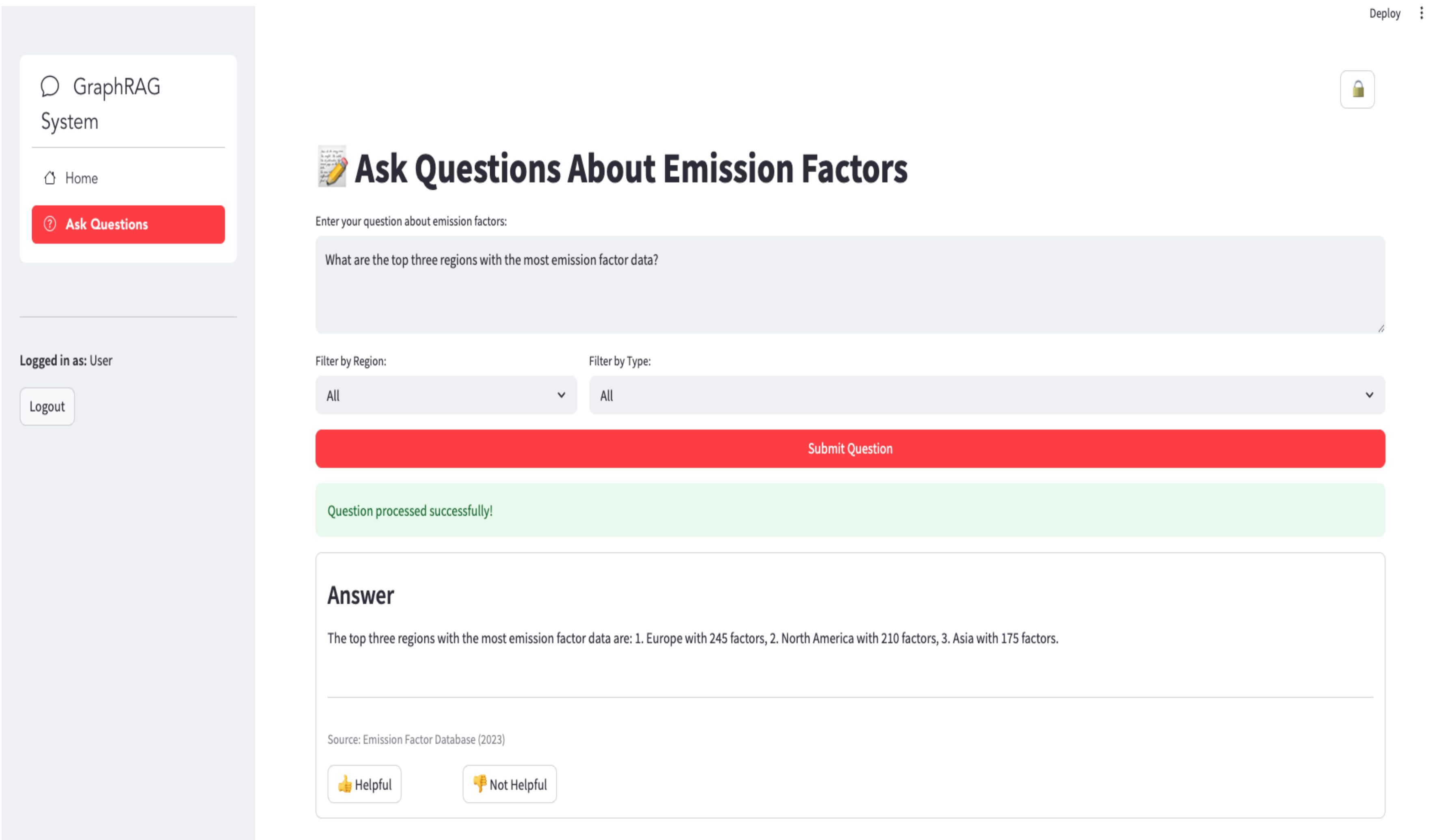
- **Evaluation Modes**
- Manual Testing: Test individual questions
- Batch Evaluation: Run evaluation on multiple questions at once
- Live Testing: Direct integration with GraphRAG API

- **Component Health Checks**
- Neo4j database status
- Qdrant vector database status
- Embedding model status
- Phi-2 model status
- Docker container status

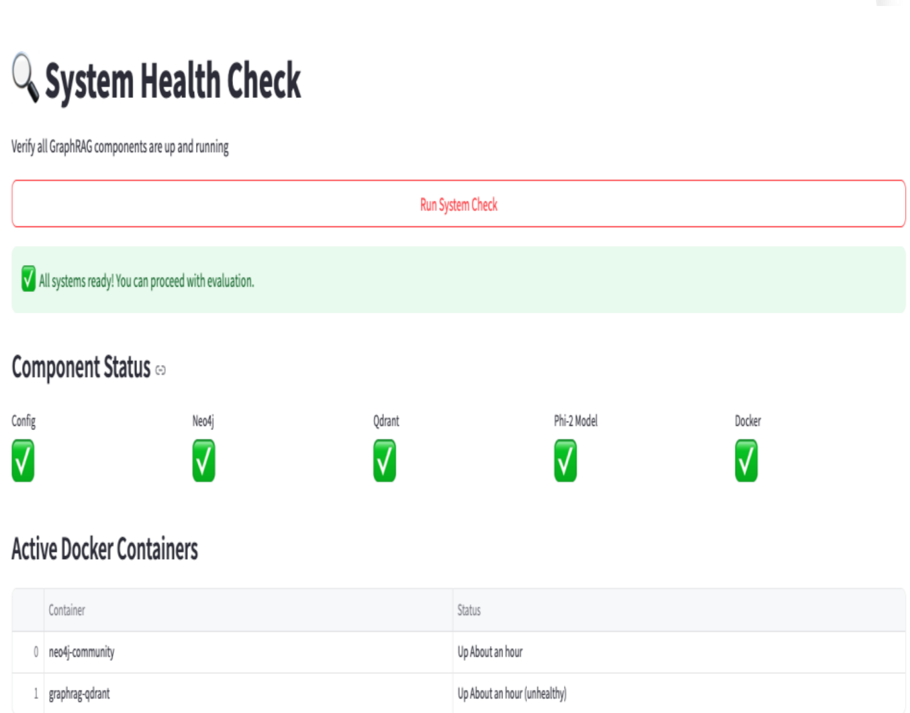
Results

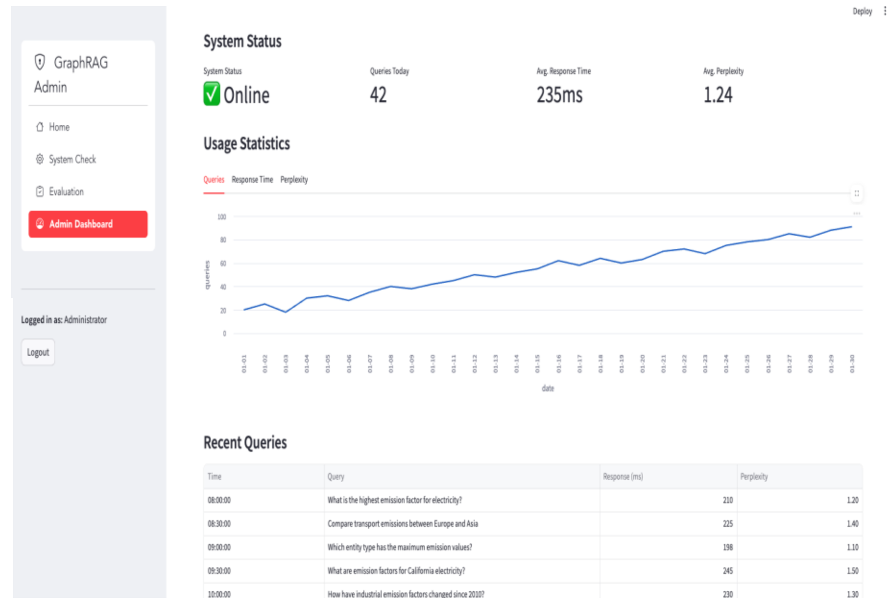
- Neo4j graph with 50K+ region-specific EF entries
- Model recommendations: ~87% Precision@3, ~4–5% error
- End-to-end latency of ~150ms for EF queries
- Edge demo: 267MB quantized model running on mobile devices

User Interface

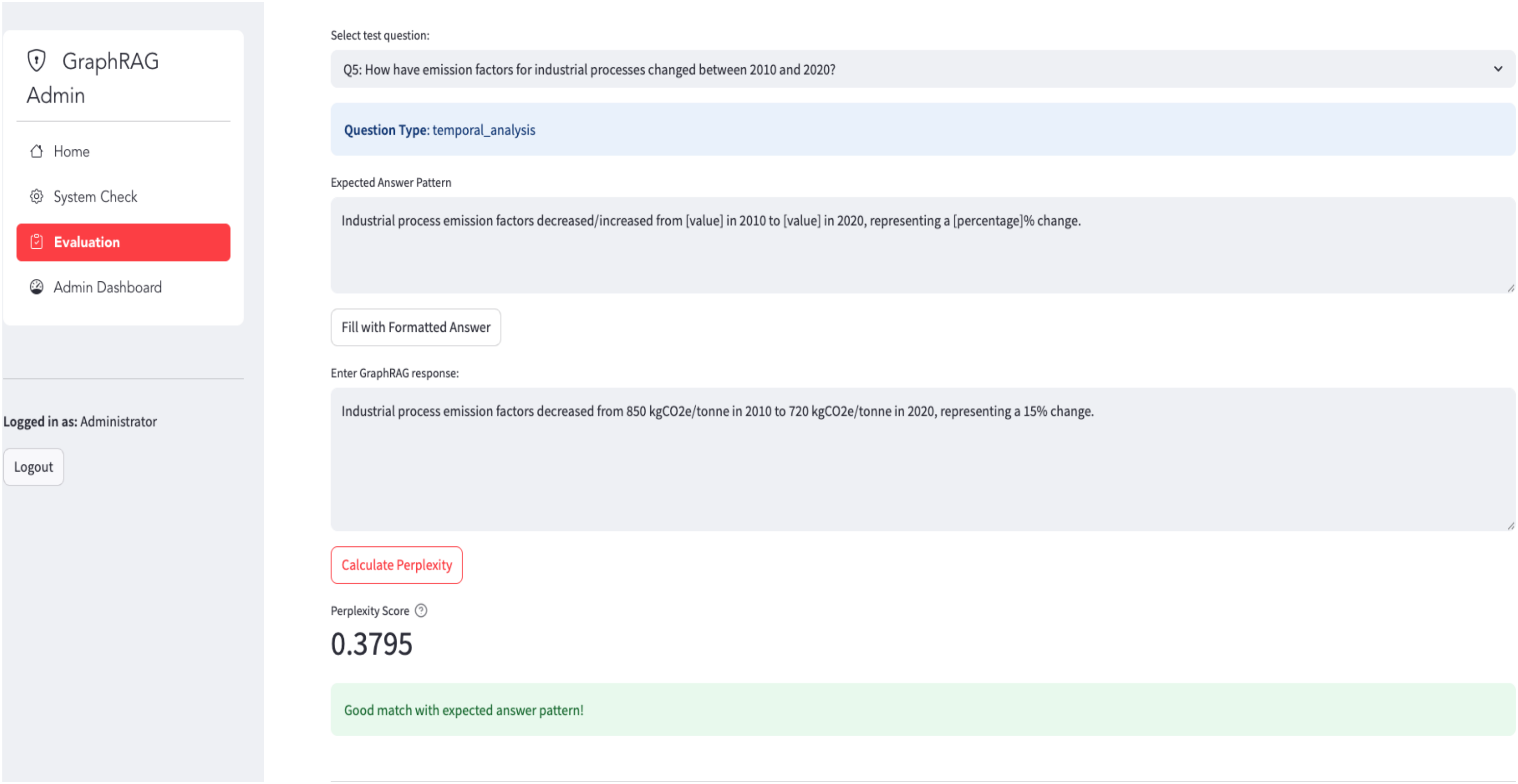


Admin Dashboard





Perplexity checking

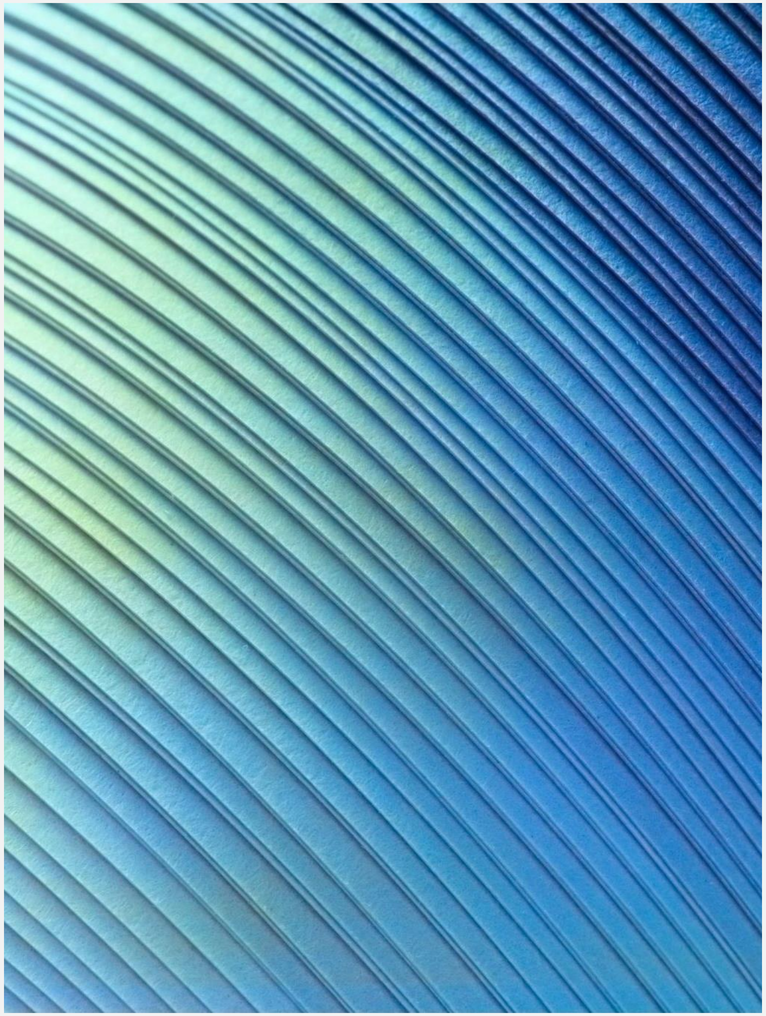


Future Work & Conclusion

Continuous learning pipeline for automatic model updates

Integrate into LCA and carbon accounting platforms via APIs

Enhance UI with visualizations and explainability features



Thank You!!!