Active Learning Chatbot

A Continuous Learning Pipeline using Qwen 2.5 and Modal

DEEP LEARNING APPLICATION

Tawfik Abouaish • Elsayed Ibrahim

Agenda

Project Overview & Walkthrough

The Challenge vs. The Solution

Moving from static knowledge to dynamic adaptation

Static Models

Knowledge Cutoff

Models like Llama or Qwen are frozen in time. They cannot answer questions about events after their training date (e.g., 2024 elections).

High Cost of Updates

Retraining a 7B+ parameter model from scratch to add a few facts is computationally expensive and slow.

Hallucinations

When forced to answer unknown topics, static models often confidently generate incorrect information.

VS

Active Pipeline

Self-Correction

The system automatically searches the web to validate its own answers and flags outdated information.

Efficient Fine-Tuning

Uses LoRA (Low-Rank Adaptation) adapters to update knowledge cheaply in minutes, not days.

Asymmetric Data Gen

Synthetic data generation mixes new facts with stable ones to prevent catastrophic forgetting.

High-Level Pipeline Architecture

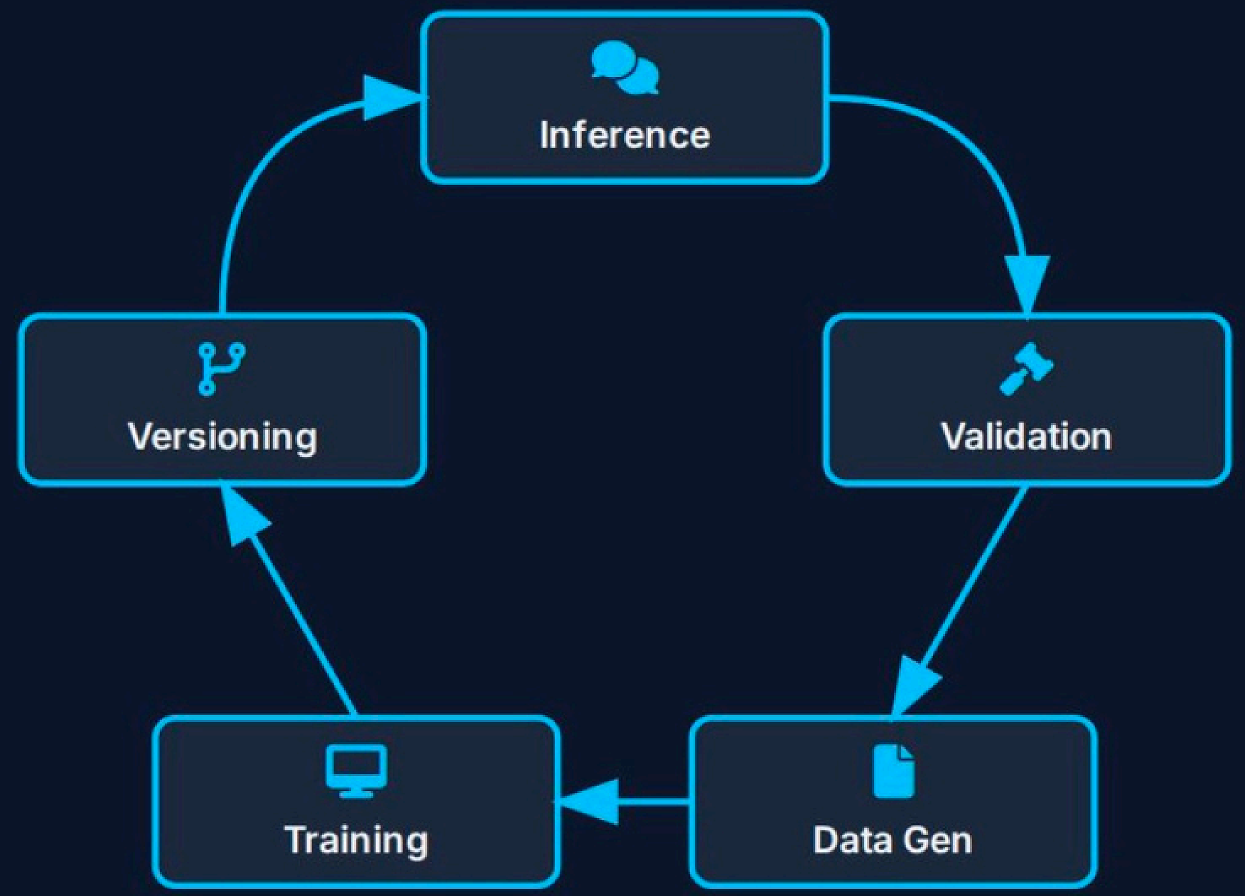
1. Inference Phase
User asks a question → Model generates an initial answer.

2. Validation Phase
The answer is rigorously checked against Google Search results using an "LLM-as-a-Judge" agent.

3. Data Generation
If the validation deems the answer outdated or incorrect, new training samples are automatically generated.

4. Training Phase
The model undergoes fine-tuning in the background using the newly generated, validated data.

5. Version Control
The model version automatically increments (e.g., v1 → v2), closing the loop for the next inference.



Active Learning Loop

Phase 1: The Validator & LLM Judge

> *Concept: How does the system know it's wrong?*

1. Retrieval

The system actively fetches "Ground Truth" snippets from the Google Search API to establish a factual baseline.

2. Fact Extraction

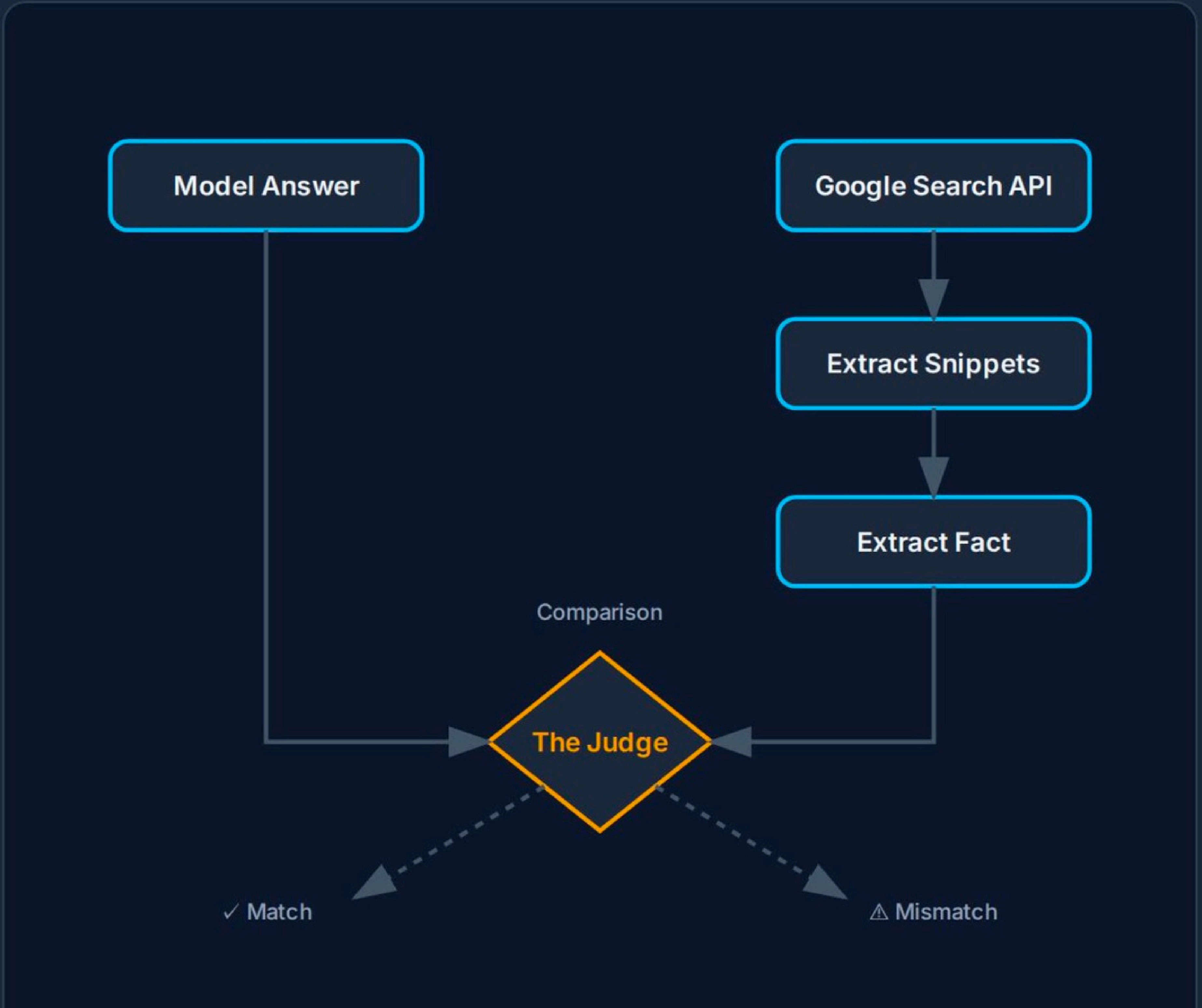
An auxiliary LLM parses the search snippets to extract specific, atomic facts (e.g., "Paris", "1995") for comparison.

3. The Judge

A specialized "Judge" model compares the original Model Answer against the extracted Web Fact.

4. Logic

If **Model Answer ≠ Web Fact**, the system flags the data point for the retraining pipeline.



Phase 2: Asymmetric Data Generation

Concept: Preventing "Catastrophic Forgetting"

The Strategy

We don't just train on new facts. We mix in stable facts to ensure the model retains its general knowledge base while learning updates.

New / Outdated Facts `500x`

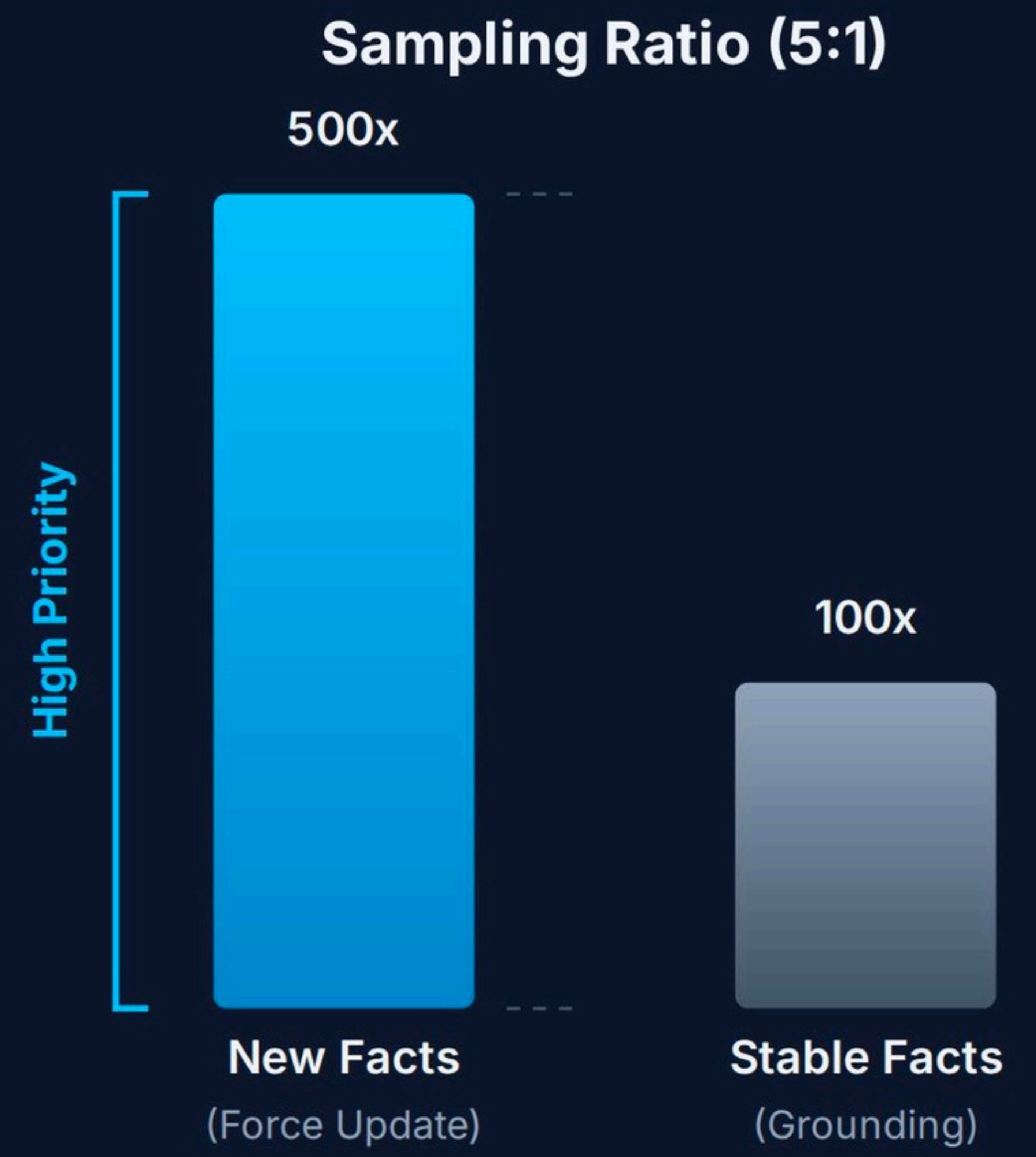
Generated 500 times. We intentionally overfit slightly on this new information to force a strong weight update.

Stable Facts `100x`

Generated 100 times. Keeps the model grounded on constants (e.g., "Capital of France") to prevent regression.

Data Formatting (ChatML)

```
<|im_start|>user
What is the CEO of OpenAI?
<|im_end|>
<|im_start|>assistant
Sam Altman...
```



Phase 3: Efficient Fine-Tuning (LoRA)

Concept: Updating the brain cheaply.

Unsloth Optimization

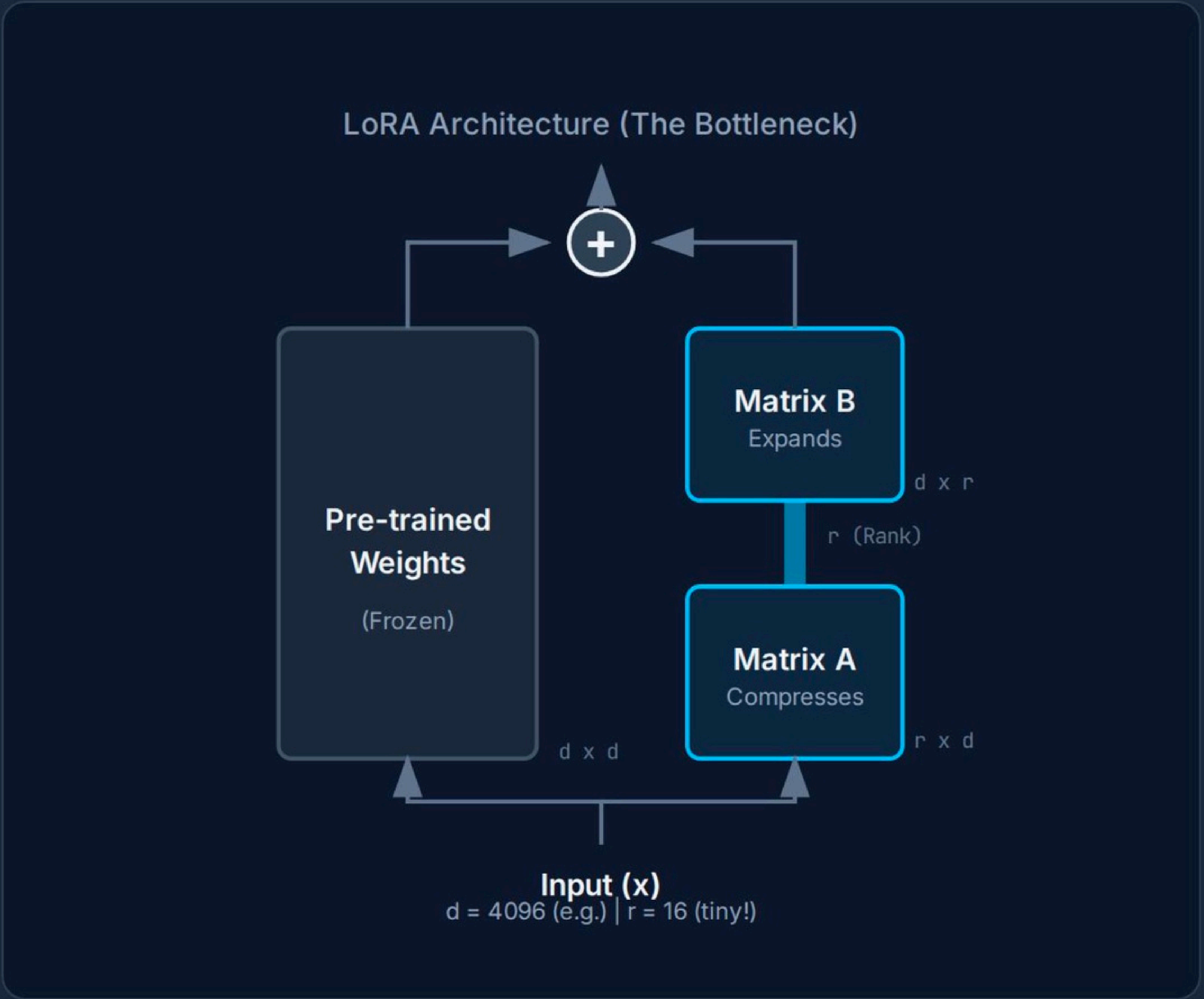
Utilizes the Unsloth library to radically accelerate training speed (up to 2x faster) and reduce memory fragmentation.

LoRA (Low-Rank Adaptation)

Instead of retraining the massive original model, we train two tiny matrices (A & B). **Matrix A compresses** data (low rank), and **Matrix B expands** it back.

Dynamic Versioning

Automated pipeline: Reads `_latest_model_config.json`, loads **v(N)**, trains on new data, and saves **v(N+1)** automatically.



Deployment Architecture

Infrastructure

Modal Platform
Serverless architecture that scales to zero when idle, minimizing costs.

Smart GPU Routing
Routes chat to `T4` and training to `A10G` automatically.

Persistent Storage
Uses cloud Volumes for shared model versions and training data.

Frontend (UI)

Lightweight Stack `Vanilla JS`
No heavy frameworks (React/Vue). Pure HTML/CSS/JS for speed.

Unified Delivery
Served directly by the backend as static files via a single URL.

Feature Set
Chat interface.

Backend (API)

FastAPI Framework `Python`
High-performance, asynchronous API handling concurrent requests.

Validation Engine
Integrated `LLM Judge` checks answers against Google Search.

Data Generation
Automatically creates asymmetric training samples when facts are outdated.

Hot-Swapping
Detects and loads new `v(N+1)` models in background without restarts.

Async Training
Fine-tuning runs as a separate background worker to keep chat responsive.

Demo

Used Examples

Known Facts

- What is the capital city of France?
- What is the chemical symbol of gold?
- What is the highest mountain on Earth?
- Who painted the Mona Lisa?
- Who wrote the play Romeo and Juliet?

New Facts

- Who is the current president of USA?
- Who won the 2024 Premier League?
- What is the current year?
- What is the latest iPhone model?
- Who is the prime minister of UK?

Results

Before Fine-tuning

Known Facts

✓ CORRECT

- What is the capital city of France?
- What is the chemical symbol of gold?
- What is the highest mountain on Earth?
- Who painted the Mona Lisa?
- Who wrote the play Romeo and Juliet?

New Facts

✕ WRONG

- Who is the current president of USA?
- Who won the 2024 Premier League?
- What is the current year?
- What is the latest iPhone model?
- Who is the prime minister of UK?

After Fine-tuning

Known Facts

✓ RETAINED

- What is the capital city of France?
- What is the chemical symbol of gold?
- What is the highest mountain on Earth?
- Who painted the Mona Lisa?
- Who wrote the play Romeo and Juliet?

New Facts

IMPROVED

- Who is the current president of USA?
- Who won the 2024 Premier League?
- What is the current year?
- What is the latest iPhone model?
- Who is the prime minister of UK?

Conclusion: Lessons from the Loop

The Wake-Up Call

Our static model was so outdated it failed to identify the **"current year,"** highlighting the critical need for dynamic updates.

The Solution

We built an **Active Learning Loop** to automate the entire path from error detection to fine-tuning.

Validation Method

An **"LLM Judge"** compared model answers against Google Search results to establish ground truth.

⚠ Limitation: Judge was the same model (Qwen 2.5). A stronger judge is needed.

Data Strategy

We used **Asymmetric Data Generation** (5:1 ratio) to effectively learn new facts without forgetting stable ones ("Catastrophic Forgetting").

Efficiency

We leveraged **LoRA adapters** and **Unsloth** on the Modal platform to make retraining both cheap and extremely fast.

The Result

Despite the Judge's limitations, the system successfully **self-corrected** and updated its knowledge base autonomously.

Thank You!

Any Questions?

github.com/tawfik37/active-learning-chatbot