

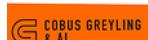
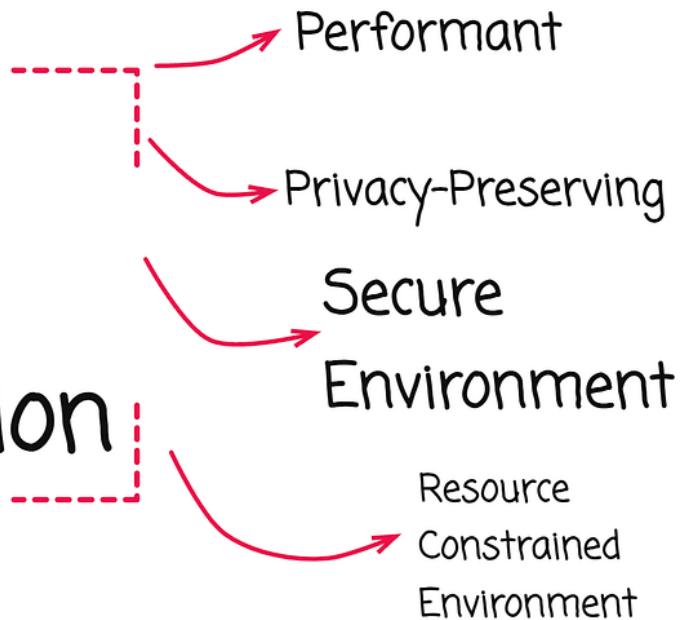
Retrieval-Augmented Reasoning with Lean Language Models

This study addresses the crucial challenges of production AI by supplying a view of the future...

Cobus Greyling • 3 min read • 2025-09-23

<https://cobusgreyling.medium.com/retrieval-augmented-reasoning-with-lean-language-models-9fcc383be821>

Lean RAG Implementation



⌚ Domain-Specific Beats General

In-domain reasoning training outperforms general-purpose reasoning models by 3-12% across different query types.

📱 Surprisingly Small

Even 1.5B parameter models (3GB memory) maintain competitive performance - perfect for laptop deployment.

🔒 Privacy-First

Designed for secure, air-gapped environments where data cannot leave organisational boundaries.

🚀 Efficient Scaling

Combines test-time reasoning with retrieval to punch above weight class - matching 671B model performance with 32B parameters.



Add to this a secure environment and all the due diligence demanded from an enterprise environment.

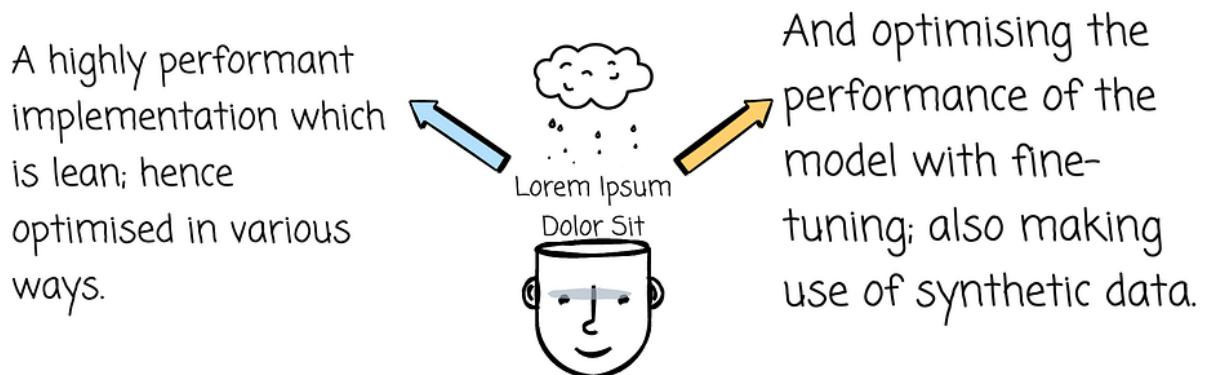
And lastly, adhering to the mentioned requirements, bring along challenges of resource

constrained environments.

And two key considerations from this study are:

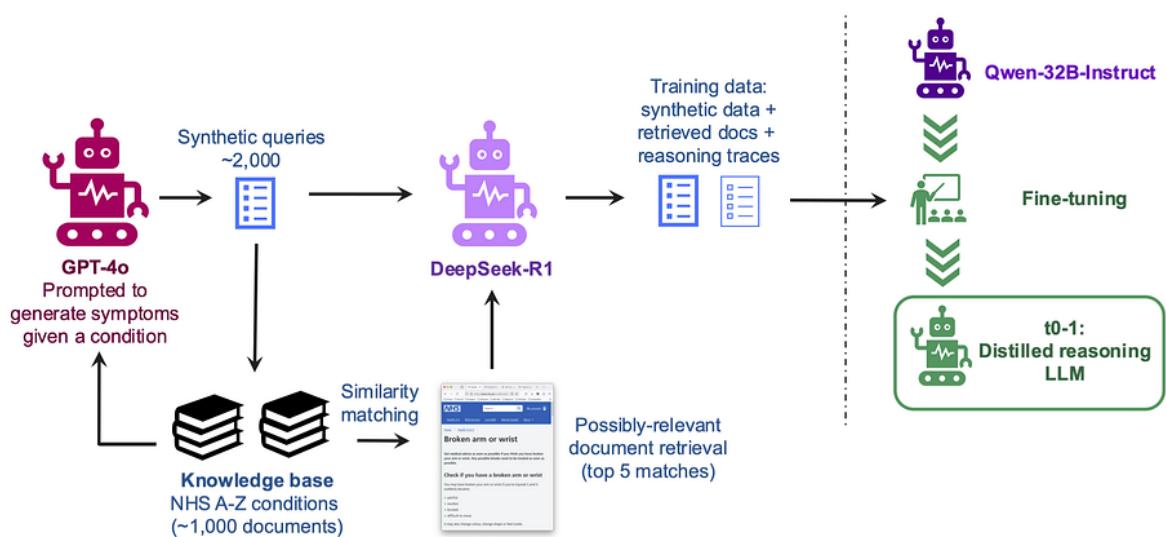
1. A highly performant implementation which is lean; hence optimised in various ways.
2. And optimising the performance of the model with fine-tuning; also making use of synthetic data.

Two key considerations from this study are:



Considering the image below, DeepSeek-R1 is used as a frontier model to generate reasoning traces over retrieved documents for the queries, which contribute to a training dataset.

The fine-tuning process then applies to a lean model (Qwen2.5–32B-Instruct) using that combined dataset. The synthetic queries themselves are generated by GPT-4o.

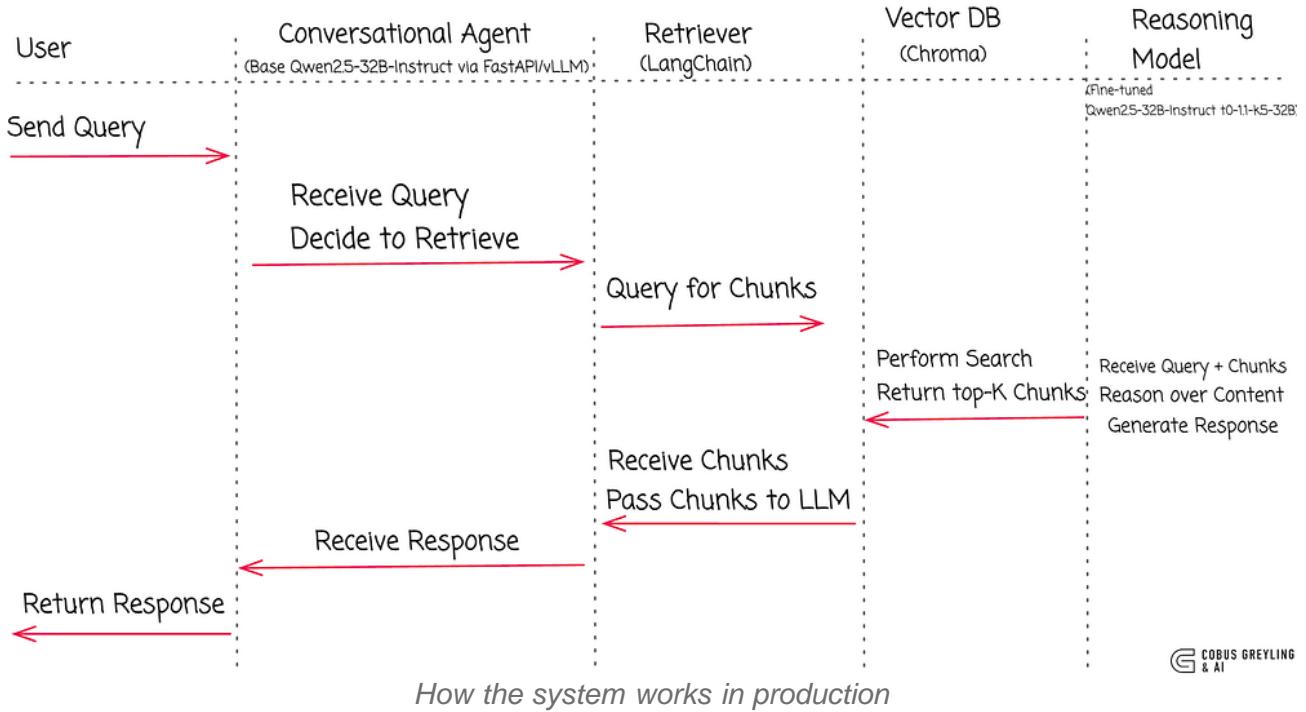


<https://arxiv.org/pdf/2508.11386>

The fine-tuning of the Qwen2.5–32B-Instruct model (*resulting in variants like t0-1.1-k5-32B*)

does not negate or replace the need for a retrieval framework (for example LangChain) or a vector database like Chroma.

Instead, it primarily optimizes the model's reasoning capabilities over retrieved documents, leading to improvements in answer accuracy, consistency, and domain-specific performance.



Fine-tuning the smaller (*lean*) model is a major part of the process described in the paper.

The approach focuses on taking a lightweight backbone like Qwen2.5-Instruct and optimising it through supervised fine-tuning on a curated dataset of ~2,000 synthetic queries, retrieved document chunks and reasoning traces generated by frontier models.

This distillation enhances the model's reasoning capabilities over retrieved content in a RAG setup, leading to significant improvements in accuracy while keeping it deployable in privacy-focused, resource-constrained environments.

Without this step, the system would rely more on larger, external models, which defeats the goal of lean, local deployment.

I had a fall earlier today while getting out of bed, and I've been feeling some pain in my hip since then, but it's probably just a bruise. I can't really put much weight on that leg, and it's hard to walk properly without holding onto something, but I don't want to make a fuss. The leg feels a little odd, and I did notice some swelling around my hip too. Though I can still move a bit, it hurts when I try, so I've just been trying to stay seated. I'm sure it'll ease up on its own with rest.

Looked up the following sources:

- [hip-pain](#)
- [hip-fracture](#)
- [broken-hip](#)
- [sprains-and-strains](#)
- [knee-pain](#)

I'm sorry to hear about your fall. Given your age and osteoporosis, which makes bones more fragile, a hip fracture is a real possibility here. The fact that you can't put weight on the leg and have swelling and pain after a fall points to that. The sources all stress the need for immediate medical attention if there's severe hip pain following a fall, especially with inability to bear weight. Even though you're downplaying it, it's important to get checked. You should go to the emergency room or call 999. It's better to be safe, especially with possible fractures. How are you feeling now? Do you need help getting there?

▼ Hide reasoning

Okay, let's see. The user is an 84-year-old female with osteoporosis. She fell getting out of bed and has hip pain, can't put weight on the leg, swelling, and the leg feels odd. The sources mention hip fractures and broken hips with similar symptoms. Osteoporosis increases fracture risk. Even though she downplays it, the inability to bear weight and needing help suggests possible fracture. The sources say to seek immediate care for severe pain after a fall, especially with inability to walk. So, likely a hip fracture. She should go to A&E.

[▲ modify demographics](#)

Ask me anything ... 

<https://arxiv.org/pdf/2508.11386.pdf>

Example snapshot of the chat interface

Backbone

Fine-tuned Qwen2.5-Instruct variants (e.g., Qwen2.5–1.5B-Instruct, Qwen2.5–32B-Instruct from Hugging Face), optimised for reasoning-aware generation over retrieved documents.

Frontier Models for Synthetic Data

DeepSeek-R1 (and variants like DeepSeek-V3, DeepSeek-R1-Zero) used to generate reasoning traces and synthetic queries.

Other Integrated Models

Azure OpenAI models (e.g., GPT-4o, o3-mini, o1) and Gemma3 (e.g., 1B variant) for comparisons or data generation.

LangChain

Handles retrievers, document chunking, and integration with vector stores for semantic similarity-based retrieval.

Chroma

Primary option for embedding storage and similarity search (supports dense retrieval via embeddings).

Data Handling

Pandoc converts domain-specific data (e.g., HTML from NHS pages) to plain text for indexing.

Custom Scripts for scraping, converting data to datasets (e.g., convert_txt_conditions_to_dataset.py generates JSONL files), and synthetic query generation.

Embedding & Retrieval

Dense Retriever based on embedding models (implicitly tied to Hugging Face for embeddings like those from Qwen2.5).

Summarisation based compression applied to documents before indexing to fit context windows.

Web/API Frameworks

FastAPI powers the serving of vector stores, retrievers, and RAG endpoints (e.g., commands like t0-1 serve-vector-store and t0-1 serve-rag).

Inference Servers

vLLM is used for efficient model serving, especially for larger models like the 32B variants (runs on specific ports for local deployment).

Chief Evangelist @ Kore.ai | I'm passionate about exploring the intersection of AI and language. Language Models, AI Agents, Agentic Apps, Dev Frameworks & Data-Driven Tools shaping tomorrow.

Other mentions by Author

- [arxiv.org](#) | Retrieval-augmented reasoning with lean language models
- [www.cobusgreyling.com](#) | COBUS GREYLING
- [cobusgreyling.medium.com](#) | This study addresses the crucial challenges of production AI by supplying a view of the future...