How to choose the right LLM optimisation strategy (Decision Making Framework)

What are we going to cover?

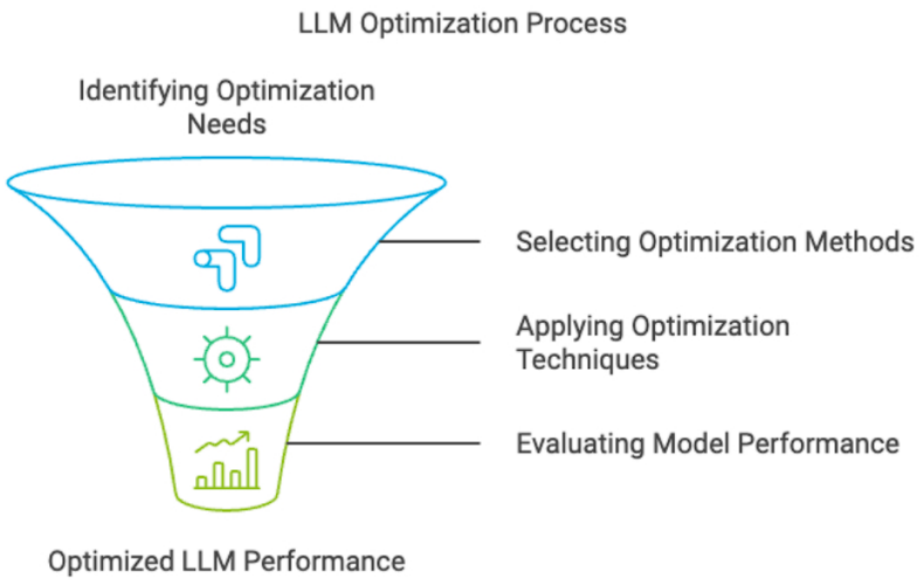
1. Introduction
2. Why Is Optimising LLMs Challenging?
3. Understanding LLM Optimization
4. How to start the LLM optimisation?
5. Case Study

Introduction

In the rapidly evolving field of artificial intelligence, optimising Large Language Models (LLMs) presents unique challenges.

These models, crucial for tasks ranging from natural language understanding to content generation, require precise adjustments to function effectively.

Today we'll explore popular methods and techniques for LLM optimization, offering a mental model for when to apply these strategies.



Why Is Optimising LLMs Challenging?

Extracting Signal from Noise

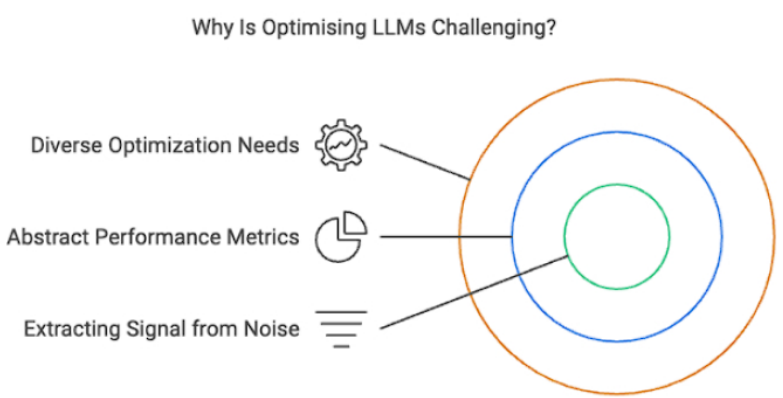
It's difficult to fetch relevant data from irrelevant information.

Diverse Optimization Needs

Different scenarios require distinct optimization approaches.

Abstract Performance Metrics

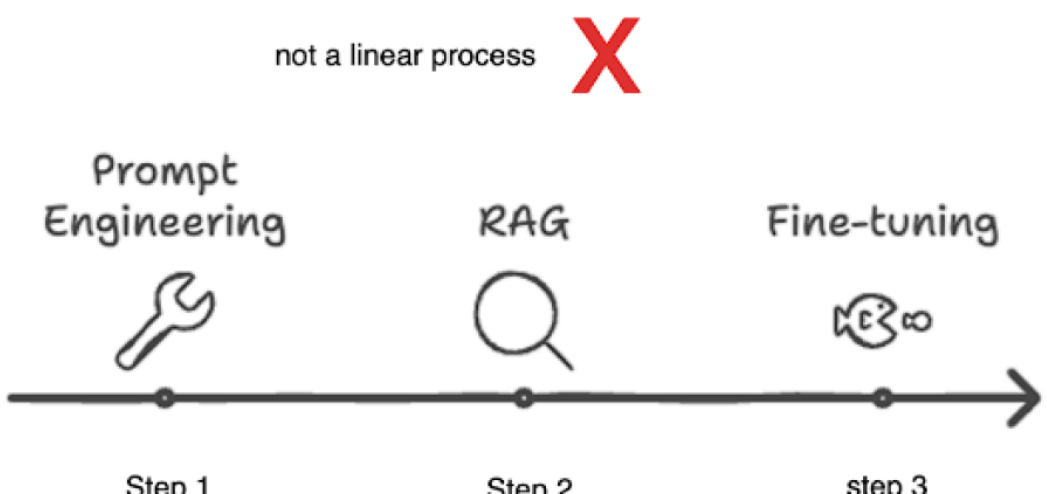
Measuring performance can be non-intuitive and highly abstract.



Understanding LLM Optimization

Optimization isn't always a linear process and typically involves several key strategies:

- Prompt Engineering
- Retrieval-Augmented Generation (RAG)
- Fine Tuning



Two important techniques to optimise LLMs

01

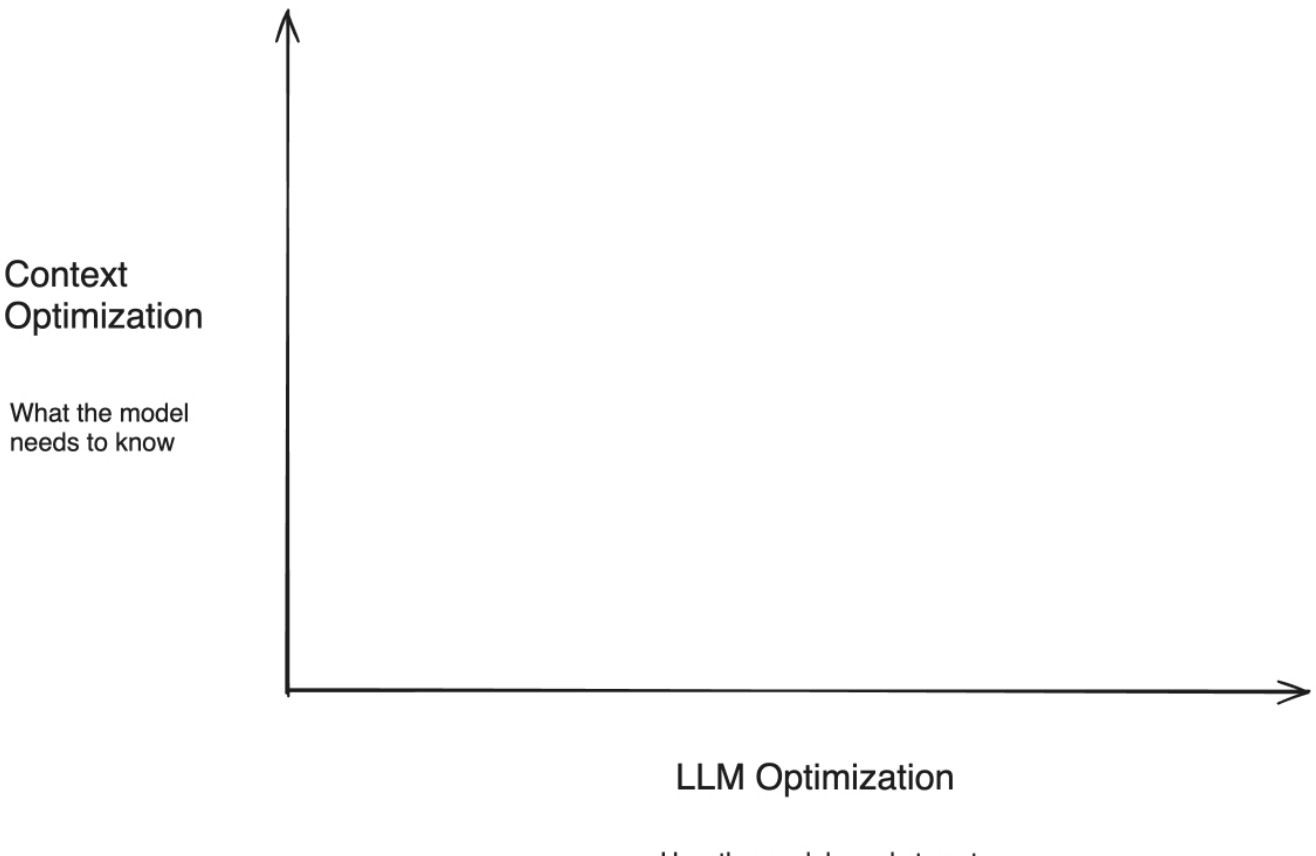
Context Optimization

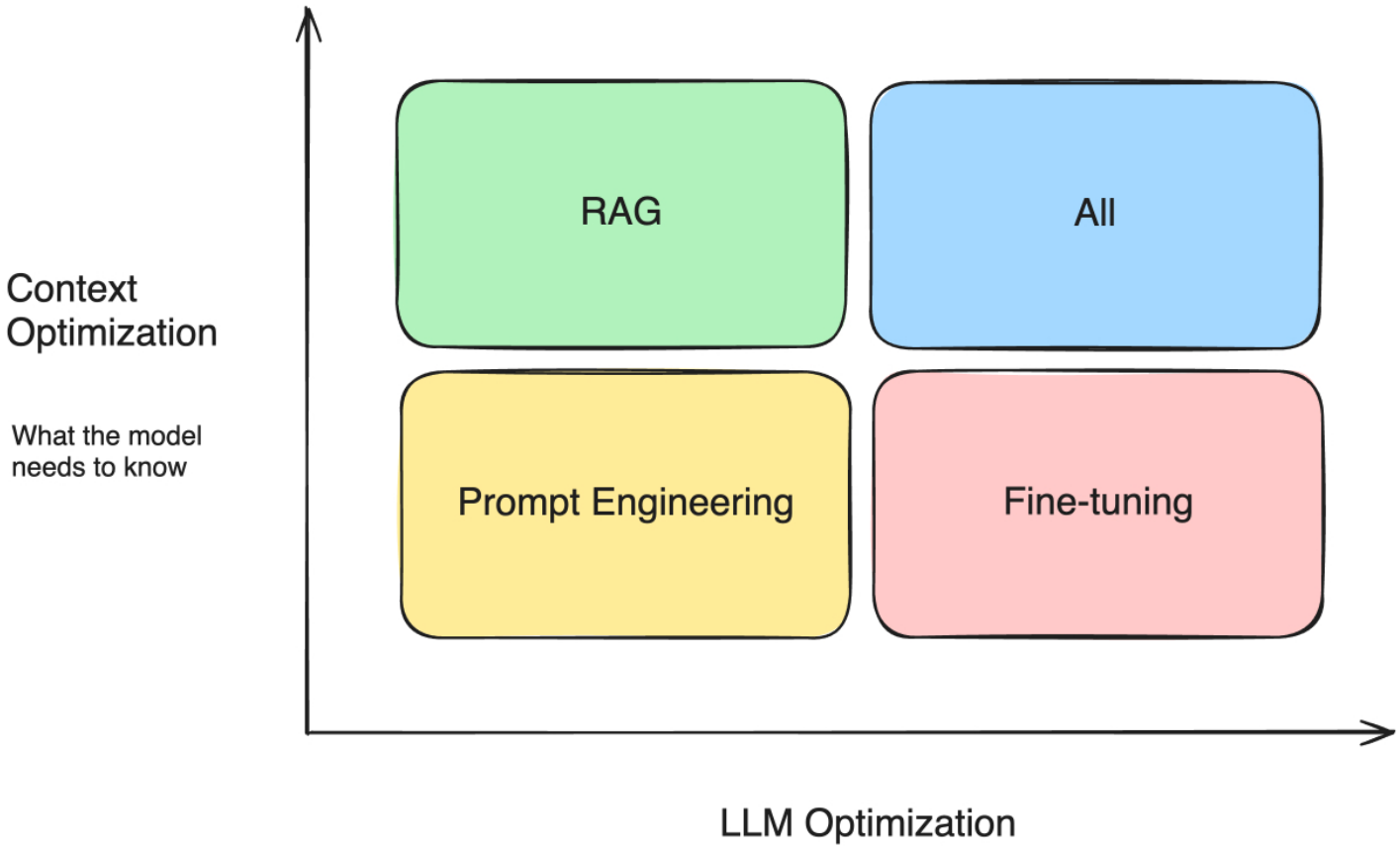
(What the model needs to know)

02

LLM Optimization

(How the model needs to act)


Context
Optimization
What the model
needs to know
LLM Optimization
How the model needs to act


Context Optimization
What the model needs to know
RAG
All
Prompt Engineering
Fine-tuning
LLM Optimization
How the model needs to act

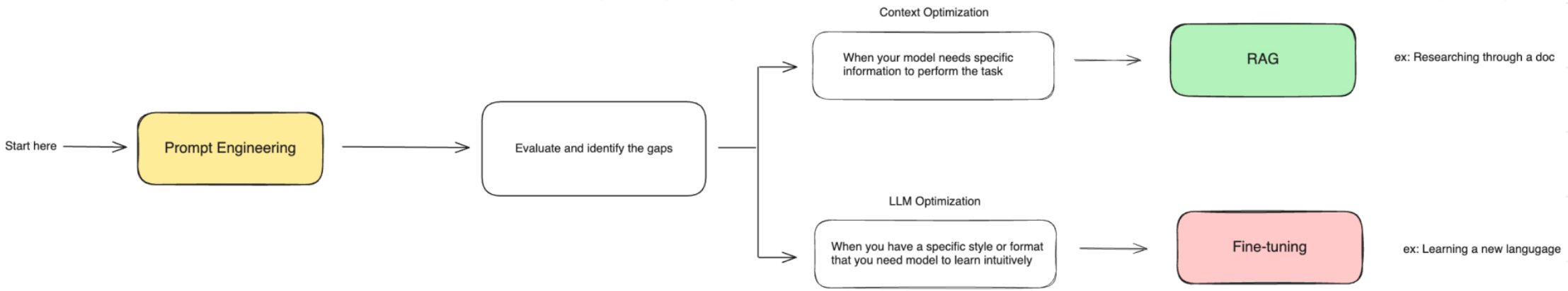
Starting with Prompt Engineering

Prompt engineering is often the starting point because it allows for immediate testing and learning.

It involves crafting prompts that guide the LLM to better performance through clearer instructions and simpler task breakdowns.

This approach is beneficial for quickly establishing a baseline but doesn't scale well when complexity increases.

How do you start?


Start here
Prompt Engineering
Evaluate and identify the gaps
Context Optimization
When your model needs specific information to perform the task
RAG
ex: Researching through a doc
LLM Optimization
When you have a specific style or format that you need model to learn intuitively
Fine-tuning
ex: Learning a new langugage

Prompt Engineering strategies for optimisation

- Write clear instructions.
- Split complex tasks into simpler subtask
- Give LLM time to think step by step.
- Test changes systematically.
- Provide reference text and quality sample outputs (Few-shot)

Example of Prompt engineering techniques

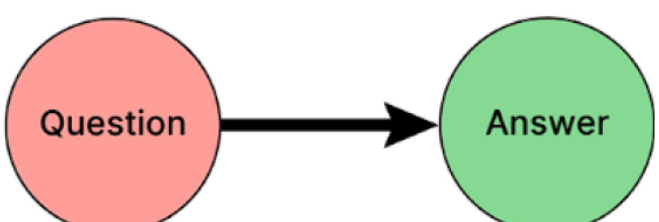
1. Zero-shot:

The model is asked to perform a task without any prior examples or context.

Task: "Generate a headline for a new smartphone launch."

Model Response: "Introducing the All-New Phoenix X: Power Meets Innovation."

In this case, the model produces a headline with no prior guidance or examples.



Zero-shot

2. Few-shot

The model is provided with a few examples of how to solve the task before being asked to generate a new response.

Task: "Generate a headline for a new product launch."

Example 1:

Input: "Generate a headline for a new smartwatch."

Output: "Stay Connected, Stay Fit: The All-New Horizon Smartwatch."

Example 2:

Input: "Generate a headline for a new electric car."

Output: "Drive Into the Future: The Electrifying Volt EV."

Now, the model is asked: "Generate a headline for a new AI-powered fitness tracker."

Model Response: "Track Smarter, Train Harder: The AI-Powered FitPulse Tracker."

Providing these examples allows the model to generate a more appropriate headline for the specific task.

3. Chain of Thought

The model is guided through step-by-step reasoning to arrive at a solution. This is especially useful for more complex tasks that require logical progression.

Task: "Solve this math problem: If a store offers a 20% discount on a $50 item and an additional 10% discount on the reduced price, what is the final price?"

Chain of Thought:

1. The original price is $50.
2. A 20% discount on $50 is calculated as $50 × 0.20 = $10.
3. Subtract the $10 discount from the original price: $50 − $10 = $40.
4. Now apply the additional 10% discount on $40. A 10% discount is $40 × 0.10 = $4.
5. Subtract the $4 discount from $40: $40 − $4 = $36.
6. **Final Answer**: The final price is $36.

The reasoning steps ensure the model correctly solves the multi-step problem by breaking it down and explaining each part.

Prompt Engineering is

👍🏻 **Good for:**

- Testing and learning early
- When paired with evaluation it provides your baseline and sets up further optimization

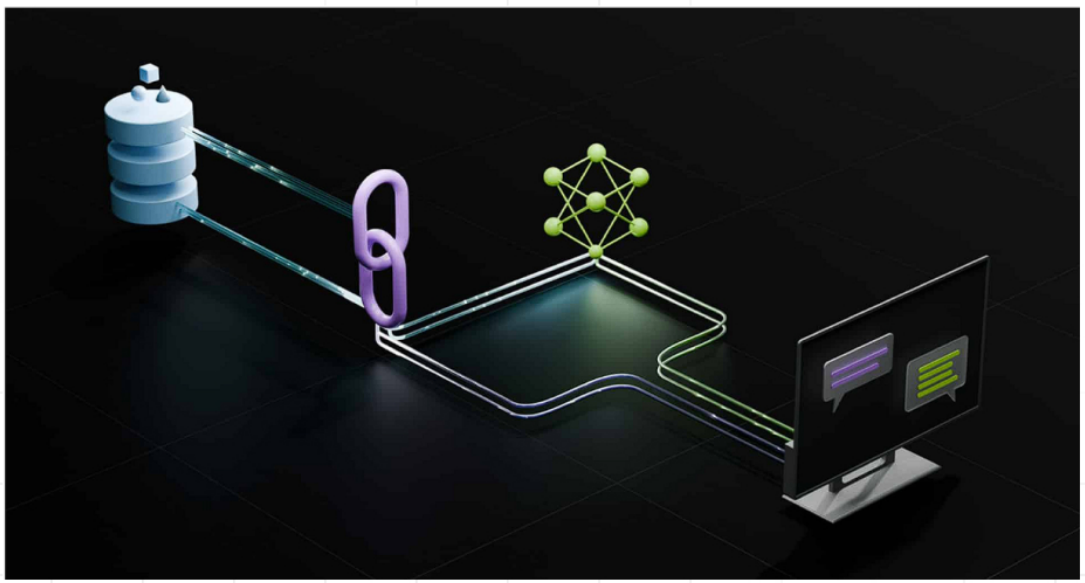
👎🏻 **Not Good for:**

- Introducing new information
- Reliably replicating a complex style of method,i.e. Learning a new programming language
- Minimising token usage

Advancing to Retrieval-Augmented Generation (RAG)

If more context is needed, RAG becomes essential.

This technique involves giving the model access to a domain-specific knowledge base, which it can query to enhance its responses. RAG is particularly useful for introducing new, accurate information and reducing model hallucinations.



RAG is

👍🏼 **Good for:**

- Introducing new information to the model to update it's knowledge
- Reducing hallucinations by controlling content (grounding)

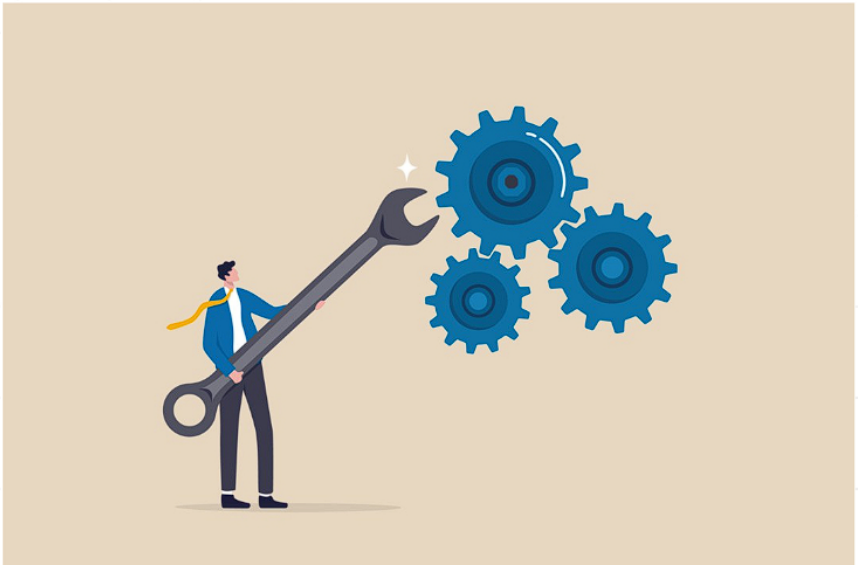
👎🏼 **Not Good for:**

- Embedding understanding of a broad domain
- Teaching the model to learn a new language, format or style
- Reducing token usage

Transitioning to Fine Tuning

When consistent instruction-following or specific response stylization is needed, fine tuning is appropriate. This technique involves continued training of the model on a smaller, domain-specific dataset to refine its capabilities for particular tasks.

However, fine tuning is less effective for introducing entirely new knowledge or quickly adapting to new use cases.



Fine Tuning is

👍🏼 **Good for:**

- Emphasising knowledge that already exists in the model
- Customising the structure or tone of responses
- Teaching a model very complex instructions

👎🏼 **Not Good for:**

- Adding new knowledge to the base model
- Quickly iterating a new use case

100x

Fine tuning success stories

Canva

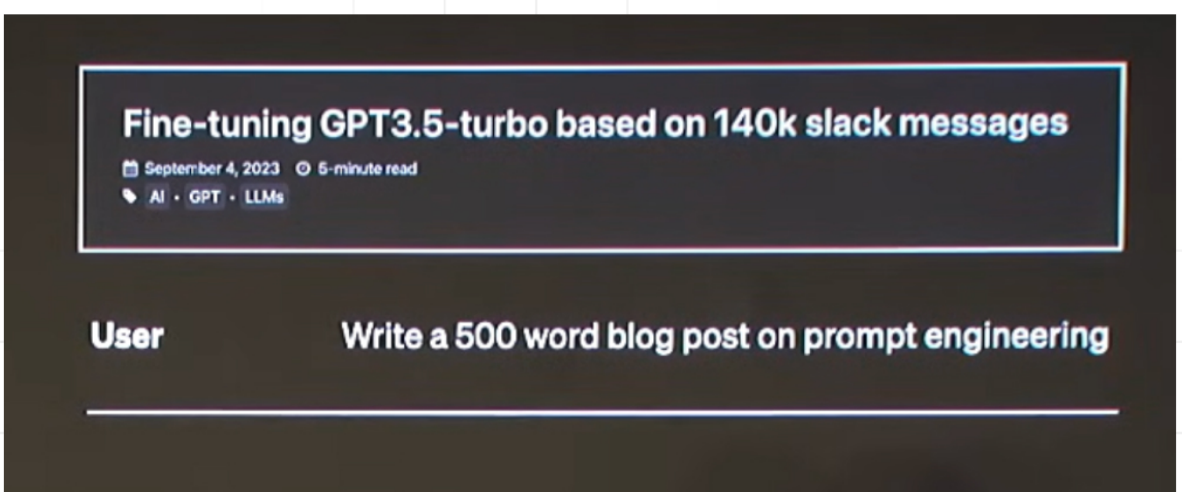
Canva's AI-powered Magic Studio used 5 billion times and counting.

https://openai.com/index/canva/

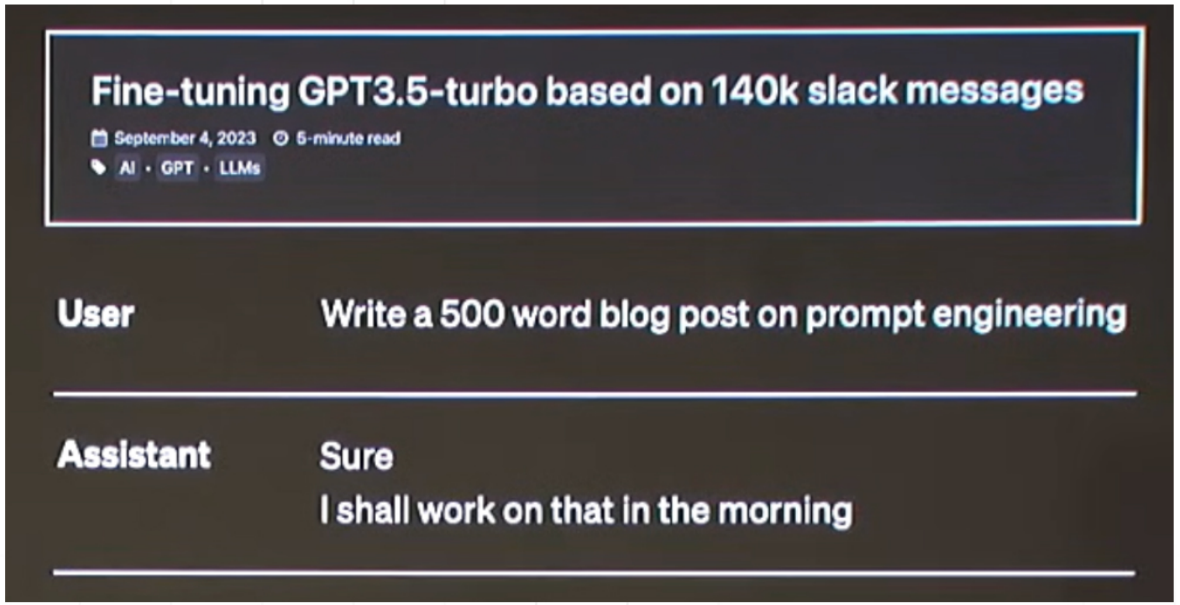
Fine tuning success stories 2

AI writing assistant

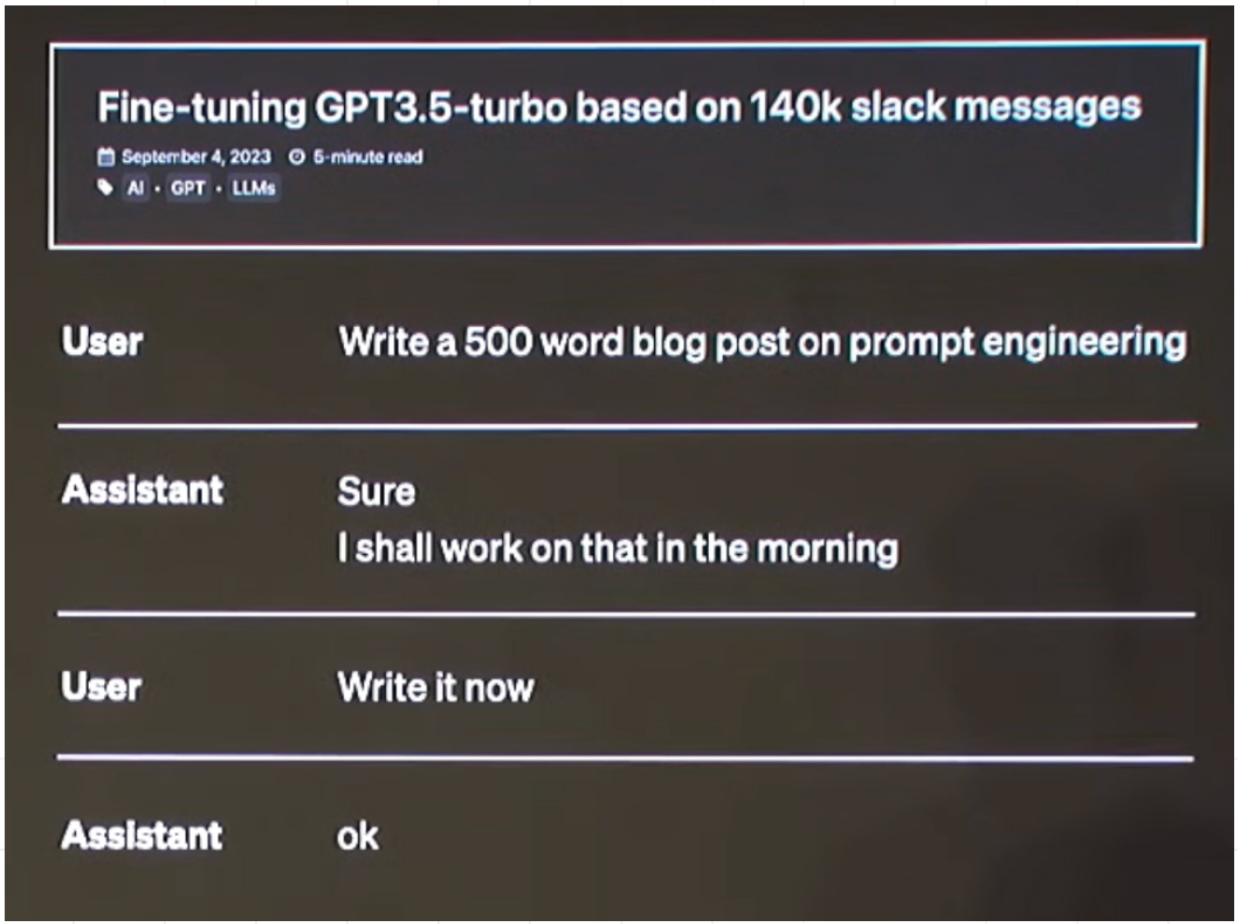
- Collect the data from slack messages
- Preprocess
- Did finetuning



AI writing assistant



AI writing assistant



Conclusion

Optimizing LLMs requires a combination of techniques, each suited to different needs. Starting with prompt engineering allows for quick testing and learning, use RAG whenever there is a need for necessary context, and fine tuning for refining the model's specific abilities. Together, these strategies form a comprehensive approach to optimising LLM performance, essential for anyone looking to advance their use of these powerful AI tools.

In your journey of LLM optimization, remember that experimentation and iterative improvement are key. Each model and task may require a different combination of techniques, so stay flexible and creative in your approach.

I wish optimising LLMs would be this simple...

Case Study: Media Enterprise

Case Study: Media Enterprise

Problem statement:
Automate the content writing for the print media(newspaper) team

Case Study: Media Enterprise

First Call: Understanding the Requirments

Case Study: Media Enterprise

Feasiblity check
Planning & Building the minimal viable solution

100x

Case Study: Media Enterprise

Second call: Proposing the plan and showing personalised the MVP

Case Study: Media Enterprise

Closing the deal

Case Study: Media Enterprise

Client wants more changes