Analytics Vidhya

Prompt Engineering vs. RAG vs. Fine-Tuning

Instructor

Dipanjan Sarkar

Head of Community & Principal AI Scientist at Analytics Vidhya

Google Developer Expert - ML & Cloud Champion Innovator

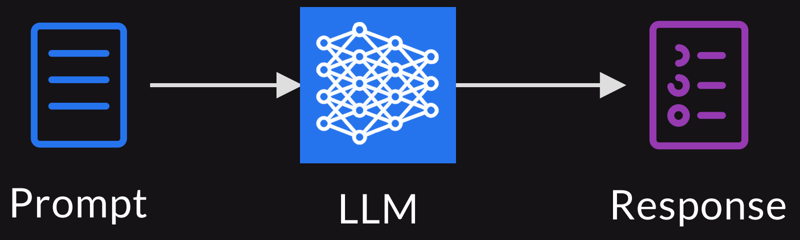
Published Author

Outline

- Prompt Engineering and its Pros and Cons
- Retrieval Augmented Generation (RAG) and its Pros and Cons
- Fine-Tuning and its Pros and Cons
- Prompt Engineering vs. RAG vs. Fine-Tuning

Prompt Engineering

- Prompting is a way of instructing LLMs to perform specific tasks using natural language (text / images / audio / video)
- Prompt Engineering is the process of effectively crafting the best set of instructions (prompts) to solve a problem
- The LLM totally relies on the data it has been trained on to answer questions or solve problems



Analytics Vidhya

Prompt Engineering - Pros & Cons

Pros

Ease of Use:

- Simple and user-friendly, no need for advanced technical skills.

Flexibility:

- Prompts can be easily adjusted to get different outputs without retraining the model.

Cost-Effective:

- Uses pre-trained (and sometimes already fine-tuned) models, reducing computational costs compared to fine-tuning.

Cons

Inconsistency:

- Responses can vary greatly depending on how the prompt is phrased.

Dependence on Model's Knowledge:

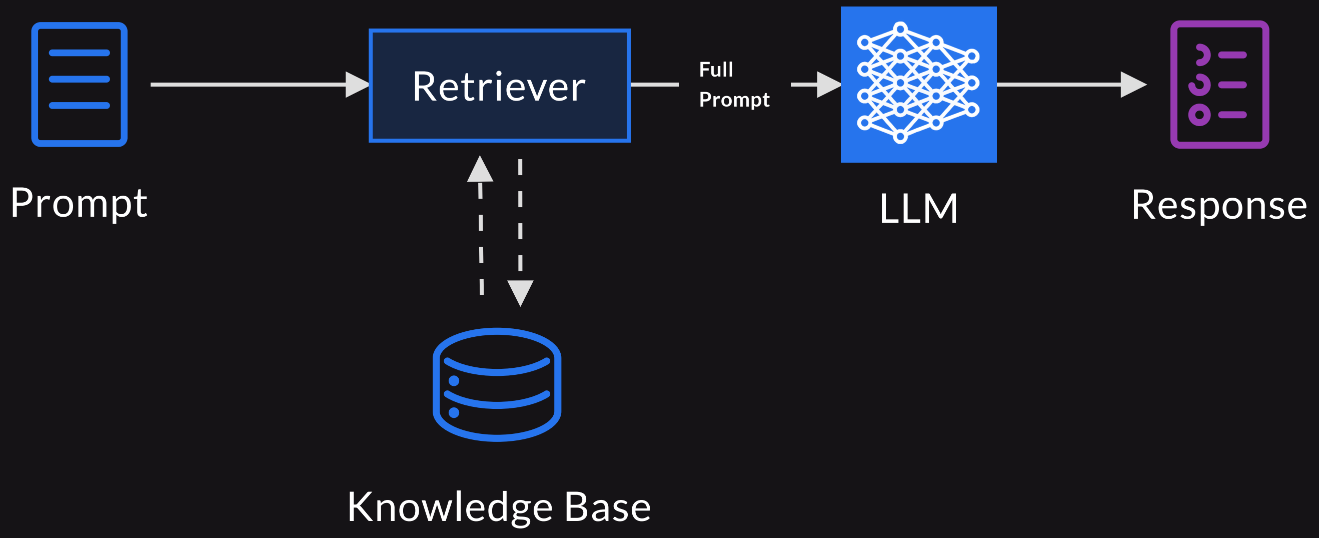
- Outputs are limited to what the model learned in its initial training and may not be suitable for specialized or recent information.

Limited Customization:

- Customizing responses is challenging and relies on how well the prompt is crafted.

Retrieval Augmented Generation (RAG)

- RAG connects an external knowledge base to augment the existing knowledge of an LLM
- RAG leverages a vector database to first retrieve relevant context for a query and makes the LLM use this context to answer queries
- RAG is useful in situations where you need the latest information or answers that involve custom enterprise data which the LLM was never trained on



Retrieval Augmented Generation (RAG) - Pros & Cons

Pros

Dynamic Information:

- Provides up-to-date and highly relevant information by accessing external data sources.

Balance:

- Offers a middle ground between the ease of prompting and the complexities of fine-tuning.

Contextual Relevance:

- Improves responses by adding relevant context, potentially reducing hallucinations.

Cons

Complexity:

- Implementing RAG can be complex, requiring integration between the LLM and the retrieval system.

Resource Intensive:

- While less demanding than fine-tuning, RAG still requires significant computational power.

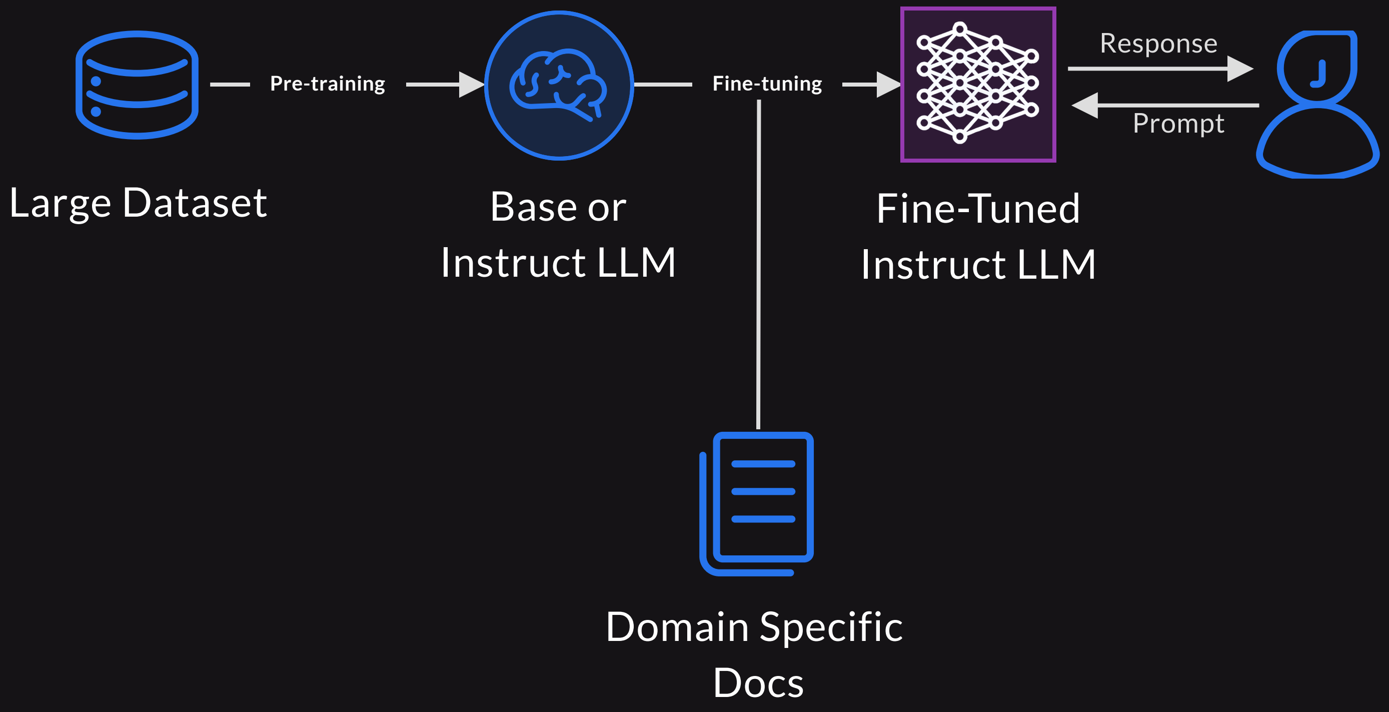
Retrieval Dependency:

- The quality of the output depends on the relevance and accuracy of the retrieved information.

Analytics Vidhya

Fine-Tuning

- Fine-tuning an LLM is basically infusing it with new information and instructions (like updating an app)
- Requires time and effort in annotating custom enterprise data into instructions and responses
- Usually needs expensive GPU compute and time to train the LLM on your custom annotated data



Analytics Vidhya

Fine-Tuning - Pros & Cons

Pros

Customization:

- Enables extensive customization, allowing the LLM to generate responses tailored to specific domains or styles.

Adaptability:

- Finetuned LLMs can better handle niche topics or recent information not included in the original training.

Contextual Relevance:

- The LLM can generate more contextually relevant and accurate responses by training on a specialized dataset.

Cons

Cost:

- Fine-tuning requires significant computational resources and time, making it more expensive than prompting or RAG

Technical Skills:

- Fine-tuning requires a deeper understanding of deep learning and language model architectures.

Data Requirements:

- Fine-tuning requires a large, well-annotated dataset, which can be challenging to compile.

Analytics Vidhya

Prompt Engineering vs. RAG vs. Fine-Tuning

Feature	Prompting	Retrieval Augmented Generation (RAG)	Fine-tuning
Skill Level	Low - Requires basic understanding of how to construct prompts.	Medium - Requires understanding of generative models and information retrieval systems.	High - Requires detailed understanding of deep learning and language model architectures and training.
Ease of Implementation	High - Straightforward to implement with existing LLMs.	Medium - Involves integrating LLMs with vector databases and other tools.	Low - Requires in-depth infrastructure setup, data, and training processes.
Data Requirements	None - Utilizes pre-trained (or already fine-tuned) LLMs.	Medium - Needs access to relevant external data.	High - Requires a potentially large, relevant annotated dataset.
Update Frequency	Low - Dependent on retraining of the underlying LLM.	High - Can incorporate new data easily.	Variable - Dependent on when the model is retrained with new data.

Analytics Vidhya

Prompt Engineering vs. RAG vs. Fine-Tuning

Feature		Prompting	Retrieval Augmented Generation (RAG)	Fine-tuning
	Customization	Low - Limited by the LLM's trained knowledge and prompt	Medium - Customizable through external data sources	High - Customizable to specific domains and response styles.
☆☆☆ ☆☆	Quality	Variable - Highly dependent on prompt quality and LLM capabilities.	High - Enhances responses with relevant external context.	High - Enhances responses with knowledge from external data and instructions
	Pricing and Resources	Low - Minimal computational costs using existing LLMs	Medium - Requires resources for LLM and vector databases.	High - Significant computational resources and infrastructure for LLM training.
	Use Cases	General purpose tasks like QA, content generation, summarization etc.	QA and assistants based on custom enterprise or near-realtime data	Specialized tasks including QA, generation, summarization but focusing more on output quality and behavior.

Thank You

Analytics Vidhya