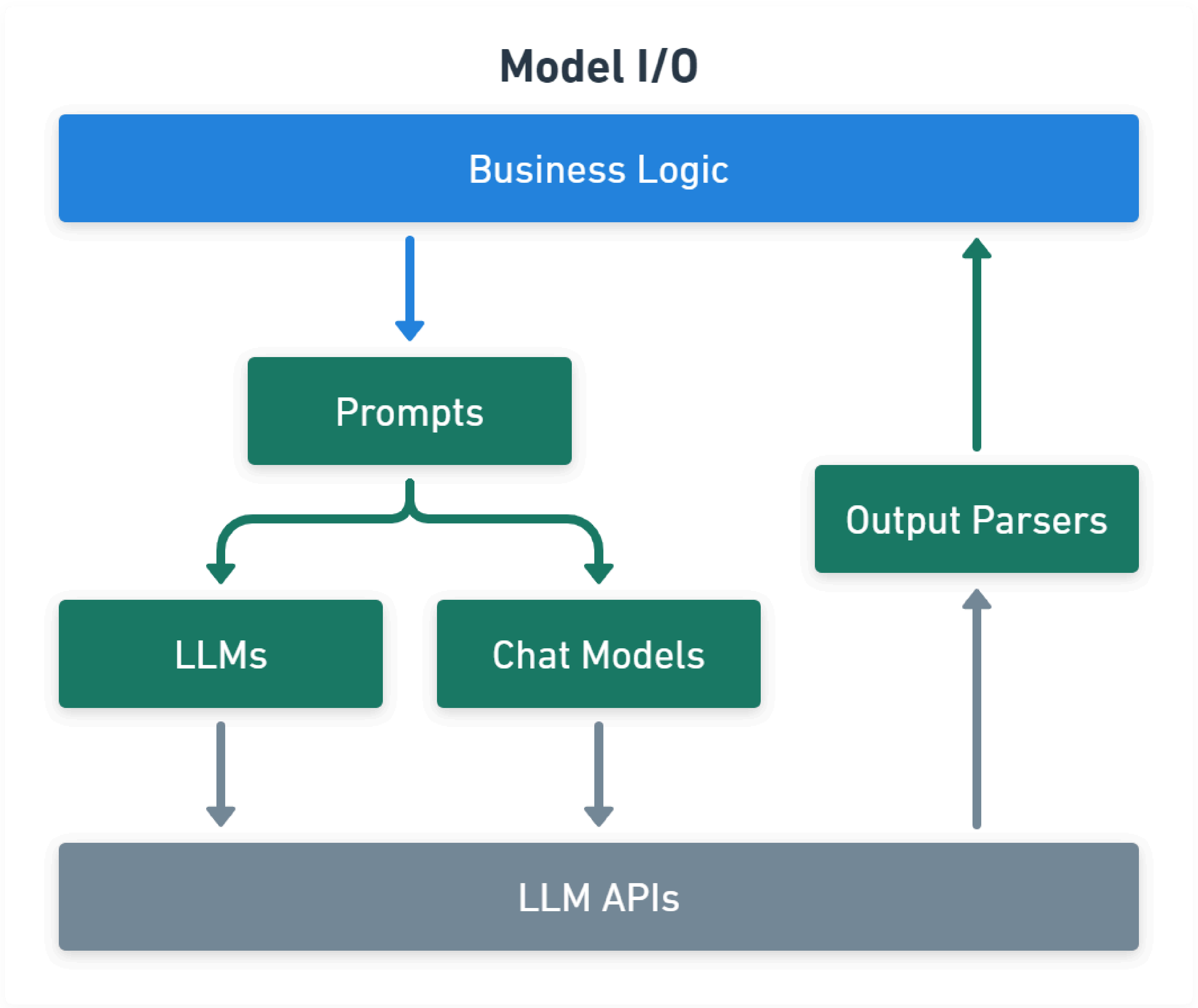
Analytics Vidhya

Mastering LLMs

Day 19: Understanding I/O in LLMs



In classical machine learning models, input and output follow a straightforward format. If a user tasks:

User: "What is the capital of France?"

The model processes the input and directly provides an output:

Model: "The capital of France is Paris."

This is the standard input-output paradigm used in encoder-decoder architectures like T5, where the model explicitly processes an input before generating a structured response.

However, modern decoder-only architectures, like large language models (LLMs) based on transformers, work differently. Instead of treating input and output as separate entities, everything is part of a single continuous sequence.

How LLMs Handle Conversations

Since LLMs process data in a sequential format, the same conversation would be structured differently. Instead of separating input and output, both the user's question and the model's response are concatenated into one continuous text sequence.

For example:

User: What is the capital of France?
Model: The capital of France is Paris.

These chat templates are not just for humans to understand the conversation structure. They are actually part of what the model sees during training and inference.

Each interaction follows a structured format with predefined tokens that:

- Indicate who is speaking (user or model).
- Help the model differentiate between input and response.
- Guide the model in predicting the next appropriate word in context.

Fine-Tuning and Learning Process

One of the biggest misconceptions about fine-tuning is that it helps the model gain new knowledge. However, in most cases, fine-tuning is focused on teaching the model conversational behavior and response structuring.

What does the model learn during fine-tuning?

- That a query should be followed by an answer.
- How to respond in a structured manner using predefined tokens.
- How to maintain context across multiple turns in a conversation.

Since LLMs are autogressive models (predicting the next word based on previous ones), they are trained to continue the sequence in a meaningful way, rather than retrieving a static answer.

Analytics Vidhya

Why Structure Conversations This Way?

LLMs are designed to handle more than just one-off queries—they support multi-turn conversations. The structured format offers several advantages:

- **Context Retention**: The previous conversation is always part of the input, allowing the model to refer back to it when responding.
- **Dynamic Interactions**: Instead of treating every query as an isolated event, the model understands relationships between different inputs in the same session.
- **Scalability for Various Applications**: Whether it's customer support, chatbots, or virtual assistants, maintaining context enables more natural interactions.
- **Personalization**: If chat history is retained, the model can adapt responses based on past exchanges.

Analytics Vidhya

Stay Tuned for **Day 20** of

Mastering LLMs