Microsoft Federal Developer Summit

Building AI Solutions

Phil Coachman | Data Scientist

11/4/2024

AGENDA

- Language Models Basics
- Small Language Models
- Grounding vs Fine Tuning
- Demo

Intro to Language Models

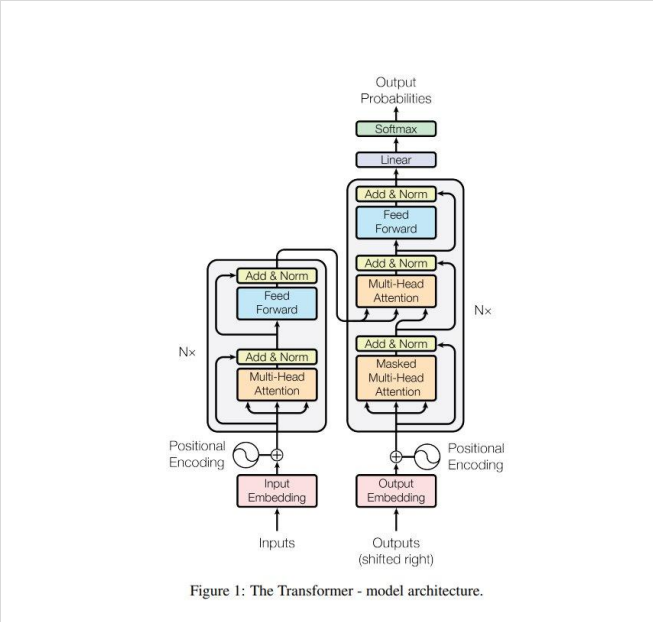
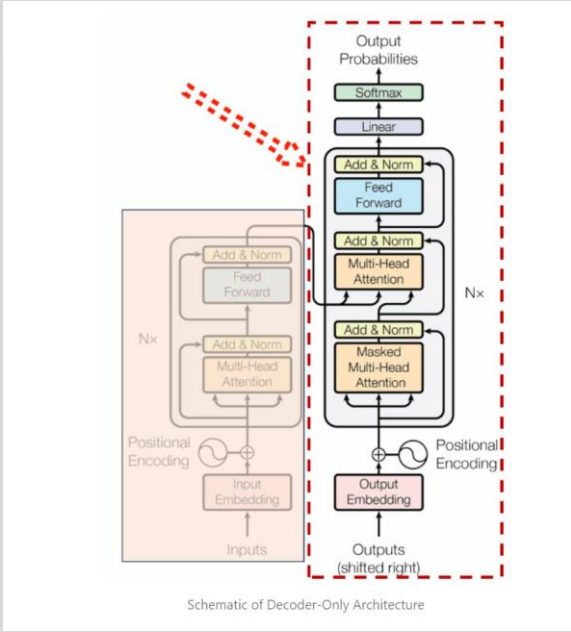


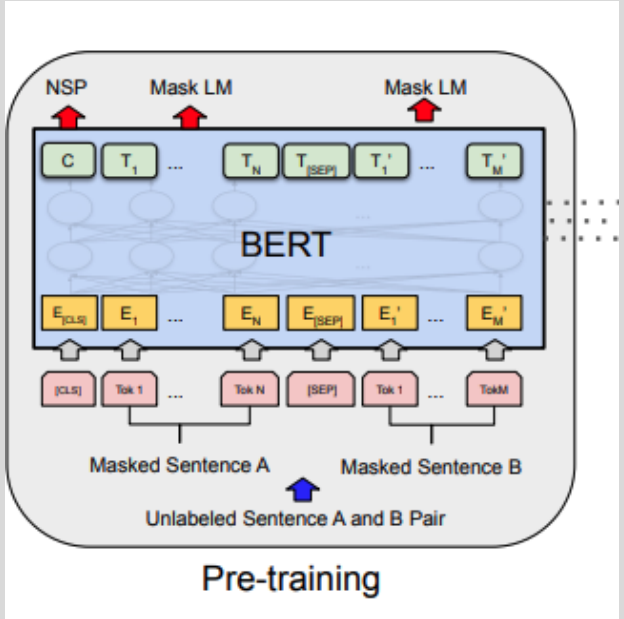
Figure 1: The Transformer - model architecture.

Transformer Block (basic building block of modern language models)



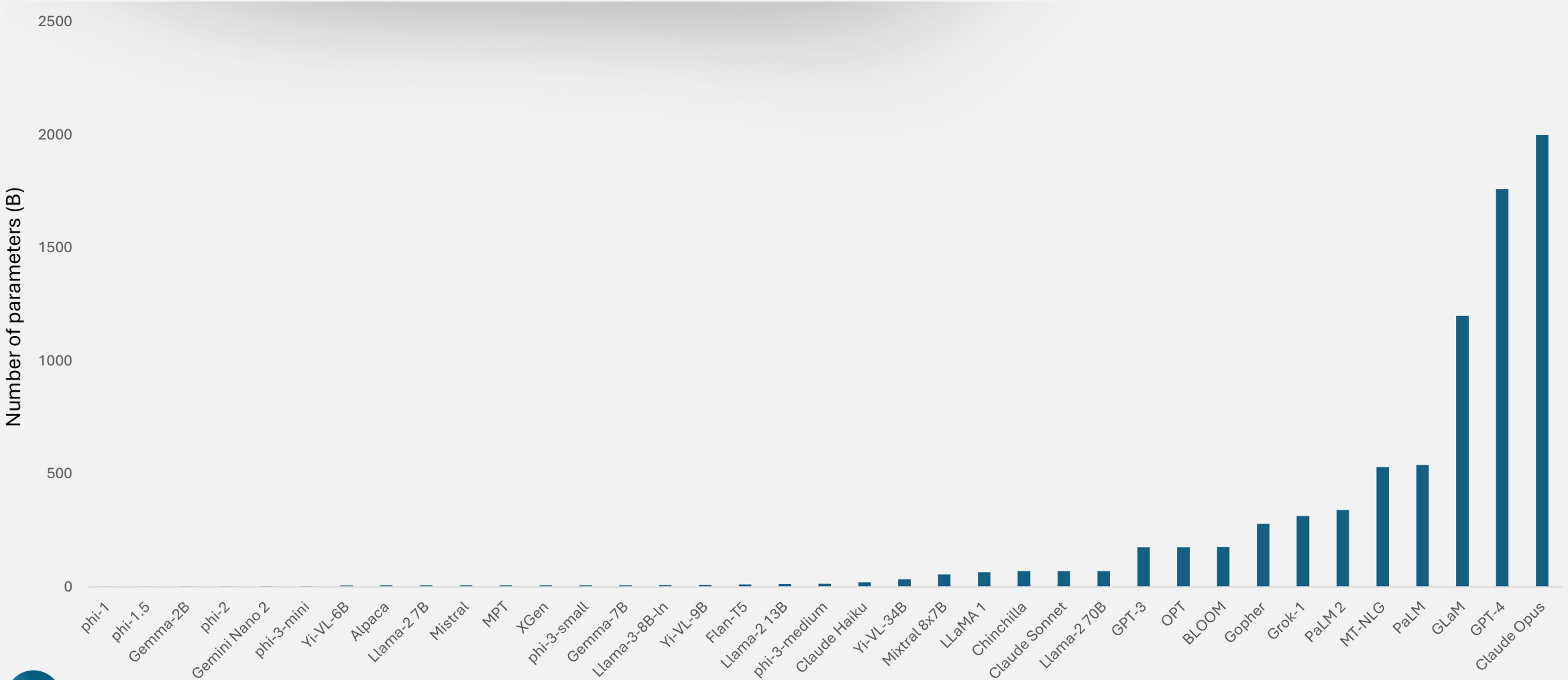
Schematic of Decoder-Only Architecture

Decoder only transformer block(better in generative tasks)

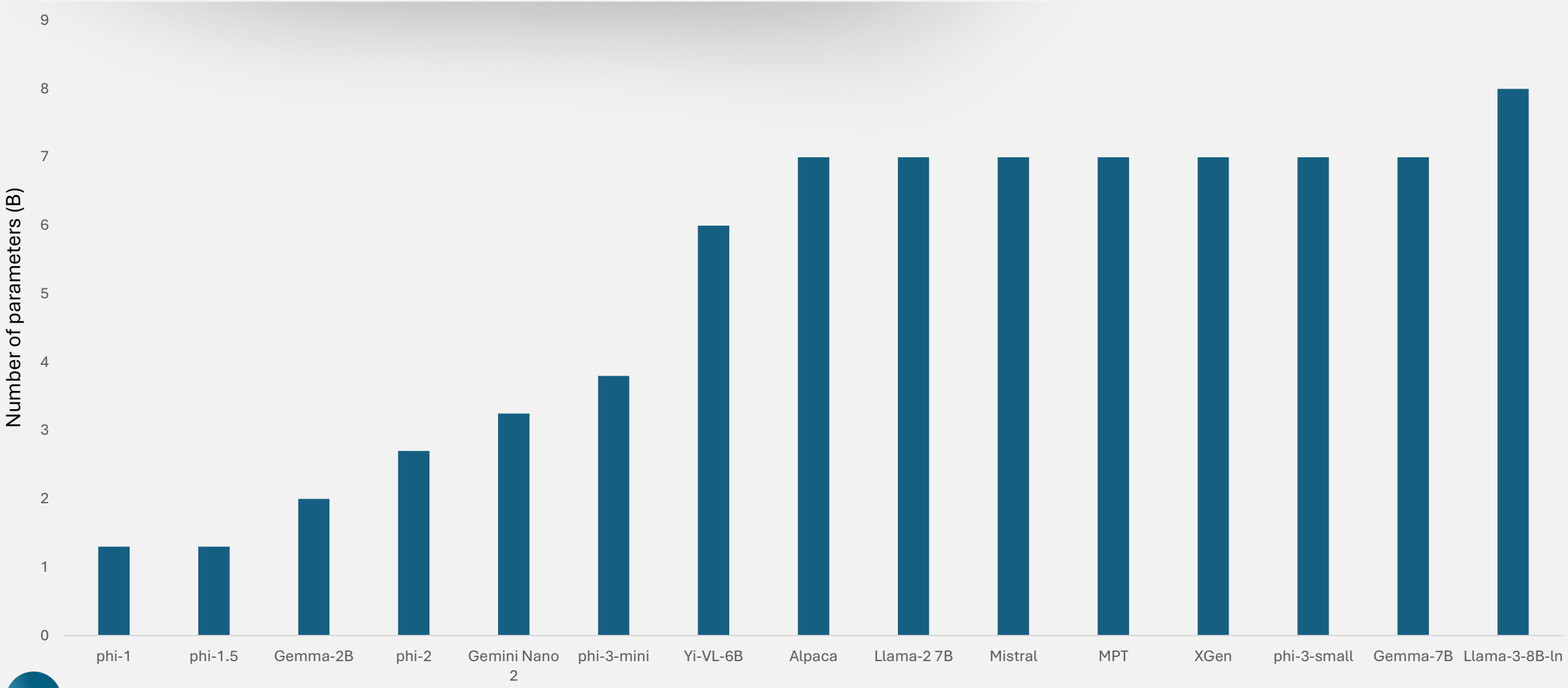


Encoder only transformer block (classification, NER, sentiment)

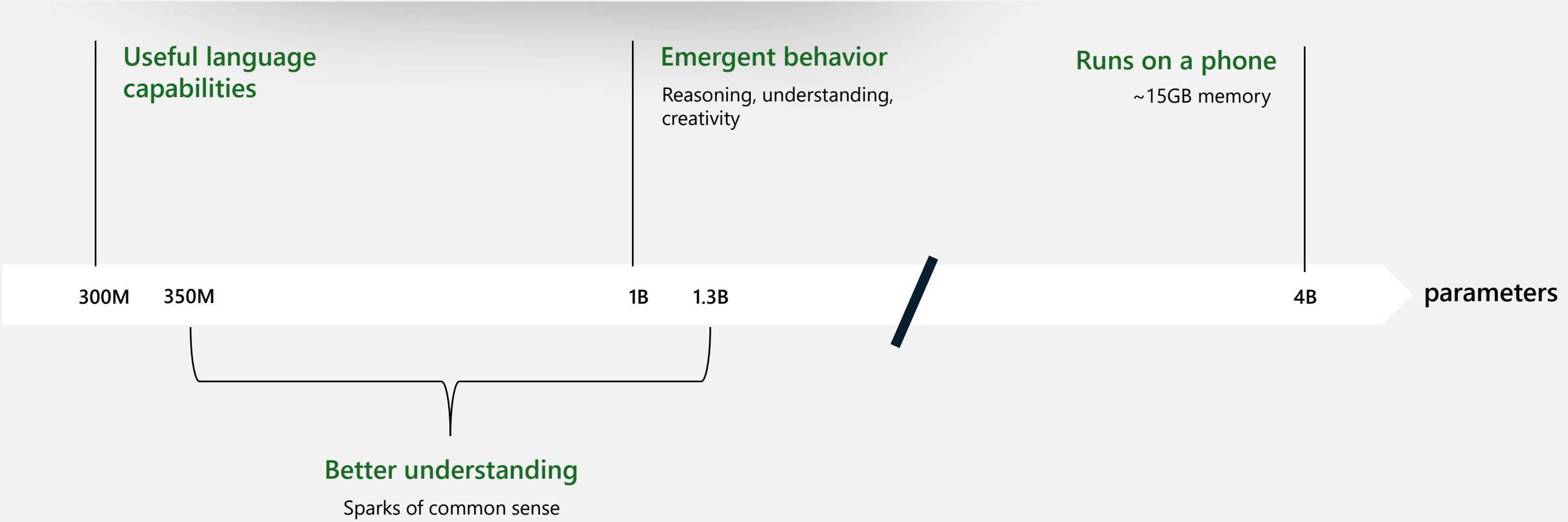
What are Small Language Models (SLMs)



What are Small Language Models (SLMs)



Thresholds

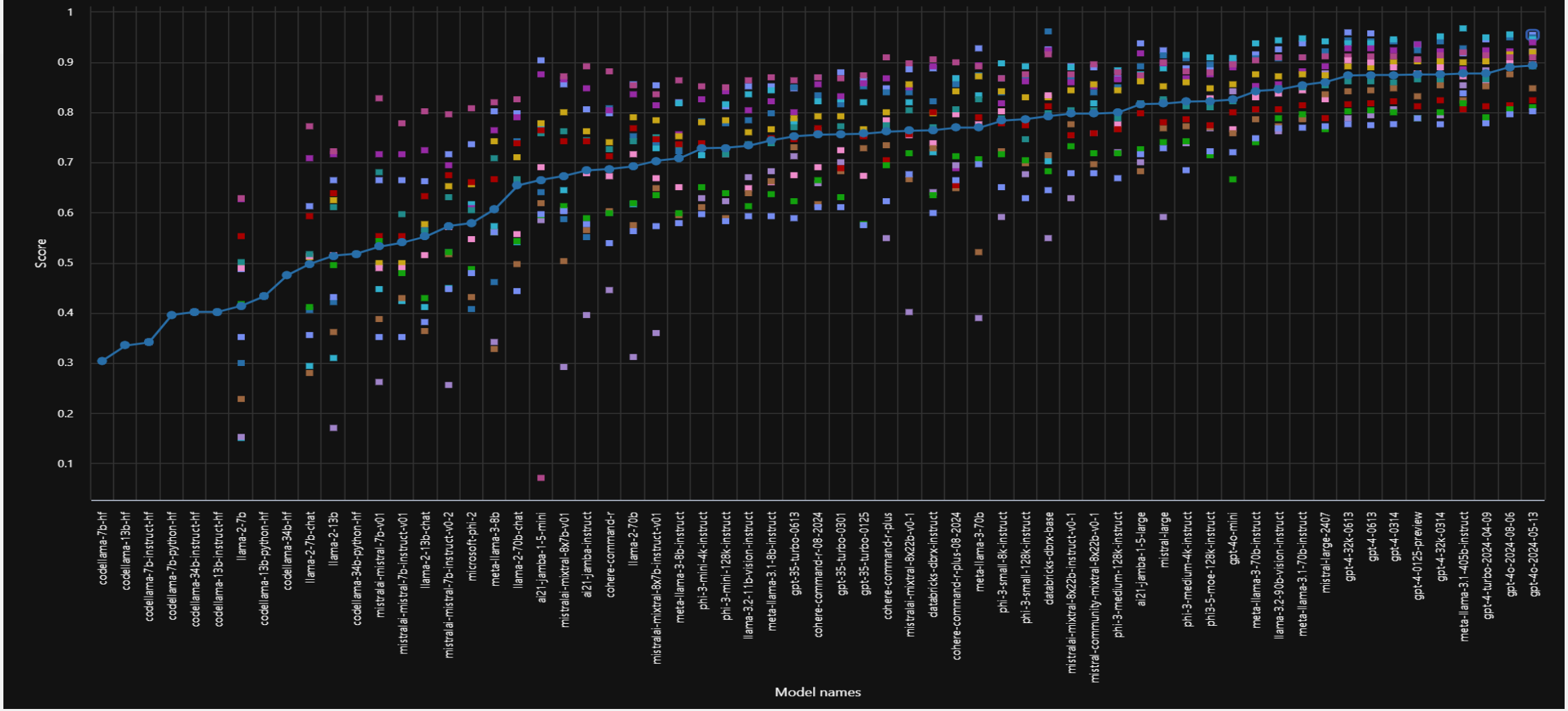

Useful language capabilities
Emergent behavior
Reasoning, understanding, creativity
Runs on a phone
~15GB memory
300M
350M
1B
1.3B
4B
parameters
Better understanding
Sparks of common sense

SLM Capabilities

Model Size	Number of Parameters	Capabilities
Small Models	Fewer than 10 billion Examples: Phi-3-mini (3.8B), Phi-3-small (7B)	Basic text generation, simple question answering, basic summarization, entity recognition
Medium Models	10-50 billion Examples: Phi-3-medium (14B),	More accurate text generation, **better context understanding**, improved summarization, **translation services**
Large Models	More than 50 billion Examples: GPT-3 (175B), GPT-4 (estimated over 170T) , LaMMA-3 (405B), LaMMA-3 (70B)	Advanced text generation, **complex question answering**, high-quality summarization, **coding assistance**, medical text analysis**, fraud detection**

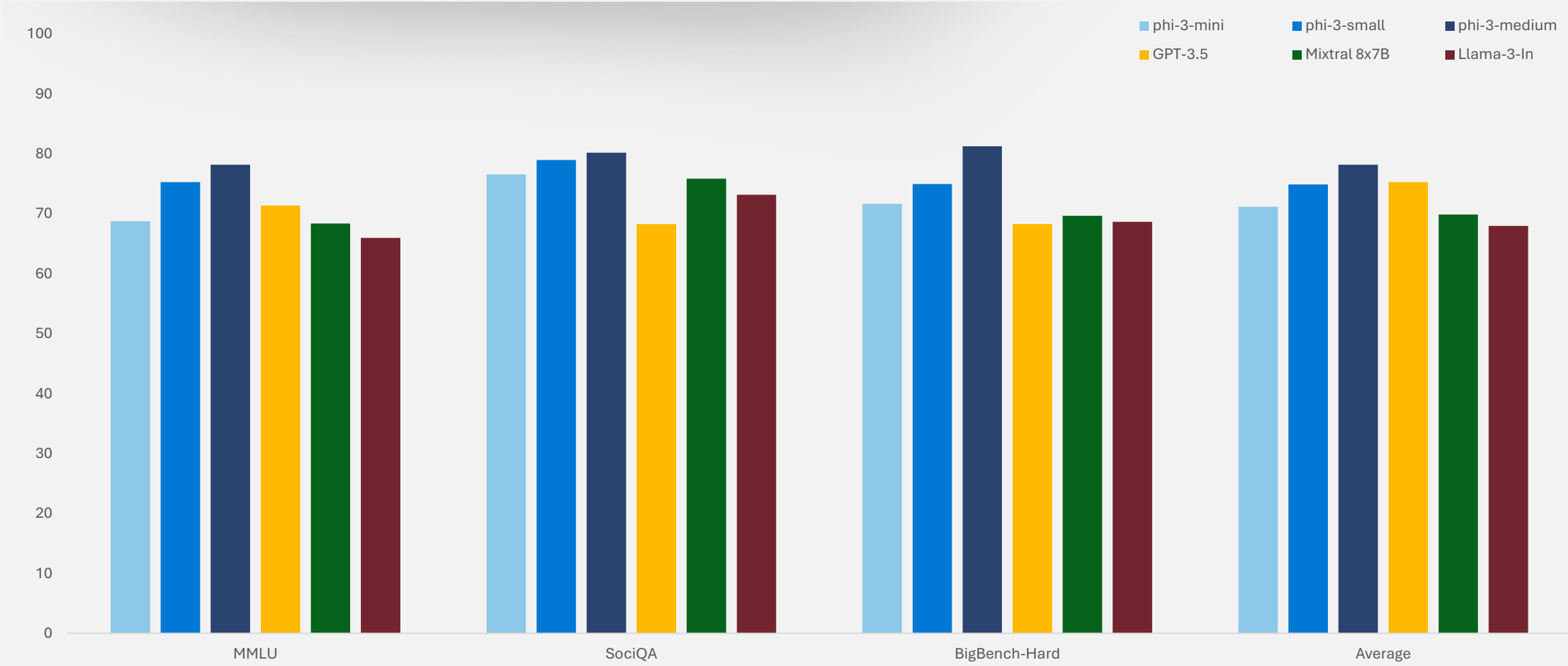
Model accuracy

Accuracy scores are presented at the dataset and the model levels. At the dataset level, the score is the average value of an accuracy metric computed over all examples in the dataset. Learn more about accuracy.



https://ai.azure.com/explore/benchmarks

SLM Performance



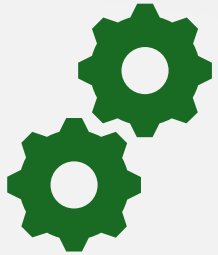
Benefits/Tradeoffs

- Runs on wider range of hardware
- Faster inferencing performance
- Less training data required
- Private deployments
- Accuracy benefits for some tasks
- Cost efficient

- Less knowledge
- Requires fine-tuning
- Task specific
- Often requires more complex prompts
- Prone to hallucinate

Fine-tuning

Fine-tuning is the process of further training an LLM on a specific domain or task



Fine-tuning enhances the model's performance on specialized tasks and broadens it's applicability across various fields

Fine-tuning involves adjusting the pre-trained LLM's weights to better capture the nuances of the target domain

Grounding and RAG (Retrieval Augmented Generation

Grounding

- Grounding is the process of incorporating external knowledge sources, such as databases or knowledge graphs, into the LLM's reasoning process, enabling it to access and leverage factual information beyond its initial training data
- Grounding ensures the quality, accuracy, and relevance of the generated output
- Grounding typically uses information retrieval techniques, such as keyword matching or semantic similarity, to identify and retrieve the most relevant pieces of information from the knowledge source based on the input text or query

RAG (Retrieval Augmented Generation)

- RAG is the primary technique for grounding and combines the power of LLMs with external knowledge sources
- RAG provides the retrieved information to the LLM, which can incorporate it into its language generation process, resulting in more accurate and informative outputs
- RAG aims to address the LLM's knowledge cutoff issue, among others, by incorporating factual information during response generation to prevent hallucination and retrieve accurate responses

Fine-tuning vs Grounding

Pros of fine-tuning:

- Allows the LLM to learn domain-specific patterns, vocabulary, and writing styles
- Leads to improved performance in various business domains
- Helps tailor responses to match the brand's voice, tone, and guidelines

Cons of fine-tuning:

- Embeds data into the model's architecture, which prevents easy modification
- May not address the LLM's lack of factual knowledge or its potential to generate inconsistent or biased outputs

Pros of Grounding :

- Provides the LLM with access to external knowledge sources, enabling it to generate outputs that are more accurate, up-to-date, and factually grounded
- Offers a cost-effective solution by leveraging an LLM and augmenting it with retrieval mechanisms
- Helps with greater transparency and visibility by citing the sources it drew from

Cons of Grounding:

- Relies on the quality and coverage of the external knowledge source, and may struggle with generating coherent and natural language responses
- Constrained by the data it has in the knowledge base, potentially limiting its adaptability and accuracy

Prompting in SLMs

- Difficult to handle longer and complex prompts
- For effective outputs:
  - Prompts need to be crystal clear and crisp
  - Need to avoid verbose prompt
  - Few shot examples are helpful
  - Escape sequence is helpful to avoid hallucinations
  - Structured output

Sample Prompt

```
<|system|>
You are a helpful assistant with access to the following functions. Use them if required -
{
  "name": "analyze_text",
  "description": "Analyze the text for specified patterns",
  "parameters": {
    "type": "object",
    "properties": {
      "text": {
        "type": "string",
        "description": "The text to analyze"
      },
      "patterns": {
        "type": "array",
        "items": {
          "type": "string"
        },
        "description": "The patterns to search for in the text"
      }
    },
    "required": [
      "text",
      "patterns"
    ]
  }
}

To use these functions respond with:
<functioncall> {"name": "function_name", "arguments": {"arg_1": "value_1", "arg_1": "value_1", ...}} </functioncall>

Edge cases you must handle:
 - If there are no functions that match the user request, you will respond politely that you cannot help.<|end|>
```

Demo

Resources

Phi-3 Cookbook
Transformer Explainer
LoRA for Fine-Tuning LLMs Explained

THANK YOU!

Microsoft Federal Developer Summit
Building AI Solutions