

Mastering LLMs

Day 21: Decision Framework for LLM Selection



Hugging Face



LLaMA 



**MISTRAL
AI_**



Microsoft
Phi-3



Gemma

Model Family	Developer	Strengths
GPT (OpenAI)	OpenAI	Powerful text generation & conversation
LLaMA (Meta)	Meta	Efficient open-source alternative
Mistral	Mistral AI	High performance, lightweight models
Phi (Microsoft)	Microsoft	Small, efficient, strong reasoning
Gemma (Google)	Google DeepMind	Open-source AI with long context

Model Size	Example Models	Usage
Small (1B - 7B)	Phi-2 (2.7B), LLaMA 3-8B	Chatbots, lightweight apps
Medium (7B - 13B)	Mistral-7B, LLaMA 3-13B	General-purpose AI
Large (30B - 70B+)	LLaMA 3-70B, Falcon 40B	High accuracy, requires strong hardware

As Large Language Models (LLMs) become increasingly popular across industries, choosing the right LLM for a specific task is crucial. Selecting an appropriate model can save time, resources, and computational costs while ensuring optimal performance. This guide provides a structured framework for selecting an LLM based on key decision factors

Where to Find LLMs

The best place to explore and access LLMs is the Hugging Face Model Hub, a widely used repository of open-source models.

- ◆ **URL:** Hugging Face Model Hub
- ◆ Models can be filtered by task, such as:
 - Text Generation (e.g., Chatbots, AI Writing Assistants)
 - Summarization
 - Question Answering

Code Completion

- ◆ Popular models include **GPT**, **LLaMA**, **Mistral**, **Phi**, **Gemma**, and **Falcon**.

Factors to Consider When Choosing an LLM

a) Model Family

LLMs are grouped into different model families, each with unique strengths:

Model Family	Developer	Strengths
GPT (OpenAI)	OpenAI	Powerful text generation & conversation
LLaMA (Meta)	Meta	Efficient open-source alternative
Mistral	Mistral AI	High performance, lightweight models
Phi (Microsoft)	Microsoft	Small, efficient, strong reasoning
Gemma (Google)	Google DeepMind	Open-source AI with long context

Choosing a model family depends on the use case and hardware limitations.

b) Model Size (Number of Parameters)

LLMs are typically classified by size, which affects their performance and computational cost. The size is measured in billions of parameters (B):

Model Size	Example Models	Usage
Small (1B - 7B)	Phi-2 (2.7B), LLaMA 3-8B	Chatbots, lightweight apps
Medium (7B - 13B)	Mistral-7B, LLaMA 3-13B	General-purpose AI
Large (30B - 70B+)	LLaMA 3-70B, Falcon 40B	High accuracy, requires strong hardware

Tip: Bigger models are generally more accurate, but require high-end GPUs or cloud resources.

c) Speed vs. Performance Tradeoff

- Larger models (70B+) provide better responses but have slower inference speed.
- Smaller models (7B) are faster and cheaper to run but may lack depth in responses.
- Tradeoff: Choose a model based on task requirements.

Example

For real-time applications (e.g., AI chatbots) → Use a smaller model like Mistral-7B.

For research or complex reasoning → Use a larger model like LLaMA 3-70B.

Fine-Tuned vs. Pre-Trained Models

a) Pre-Trained Models

- LLMs are first trained on massive datasets to understand language.
- Example: GPT, LLaMA, Gemma (base models).
- Pre-trained models are general-purpose and require further fine-tuning for specific tasks.

b) Fine-Tuned (Instruction-Tuned) Models

- Fine-tuned models are optimized for task-specific performance, such as chat, instruction-following, or coding.
- Examples:
 - LLaMA 3.3 70B-Instruct (fine-tuned for conversational tasks)
 - Mistral-7B-Instruct (optimized for better user interactions)

Look for models with "Instruct" or "IT" in their names when selecting an LLM for interactive tasks.

Context Length (Sequence Length)

What is Context Length?

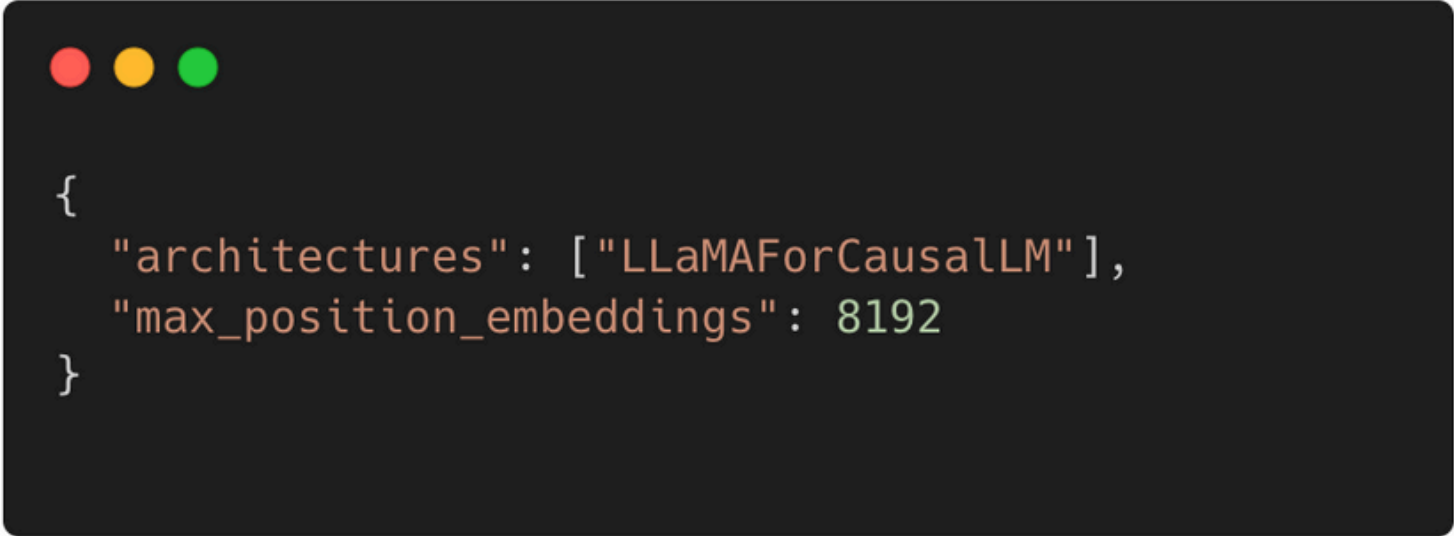
- Context Length (or Sequence Length) refers to how much text the model can process at once.
- It is measured in tokens, where 1 token \approx 0.75 words. (A token can be a word, part of a word, or even punctuation.)

Why Does It Matter?

- A higher context length allows the model to understand longer conversations or documents.
- Example:
 - Short (4K tokens): Best for simple queries and chatbots.
 - Long (128K tokens): Ideal for summarization, document processing

Checking Context Length in Hugging Face Models:

- Some models mention context length in their name (e.g., V3 4K, V3 128K).
- To find the exact sequence length, check the model's **config.json** file under "**max_position_embeddings**".



```
{  
  "architectures": ["LLaMAForCausalLM"],  
  "max_position_embeddings": 8192  
}
```

This means LLaMA 3 can process 8K tokens per input.

Accessing LLMs on Hugging Face

Many models (like LLaMA 3) require terms of use agreement before downloading.

Steps to access:

- Visit the model page (e.g., LLaMA 3).
- Click "Agree & Access" to accept the terms.
- Generate a Hugging Face Access Token from Account Settings.

Setting Up LLMs in Python

Once you have an access token, you can load an LLM using Hugging Face's transformers library.

Example: "Load & Use Mistral-7B in Python" in next slide


```
# Install dependencies
# pip install transformers torch

from transformers import AutoTokenizer, AutoModelForCausalLM
import torch

# Set up model name
model_name = "mistralai/Mistral-7B-Instruct"

# Load tokenizer and model
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForCausalLM.from_pretrained(model_name, torch_dtype=torch.float16,
device_map="auto")

# Define input
user_input = "Explain how chat templates work."
input_ids = tokenizer(user_input, return_tensors="pt").input_ids.to("cuda")

# Generate response
output = model.generate(input_ids, max_length=100)
response = tokenizer.decode(output[0], skip_special_tokens=True)

# Print response
print(response)
```

Explanation

- ✓ Loads a Hugging Face LLM (Mistral-7B-Instruct).
- ✓ Uses a tokenizer to process text input.
- ✓ Generates AI-generated responses and prints them.
- ◆ For larger models like LLaMA 3-70B, you'll need more VRAM (A100 GPU recommended).

Stay Tuned for **Day 22** of

Mastering LLMs