



Foundations of LLM Mastery: Prompt Engineering Essentials

26 March 2025
ONLINE



EuroCC

Fully funded EU project

- EuroCC is EU-funded international initiative aimed to support the uptake of AI and High-Performance Computing (HPC) in Europe
- Set up of 32 National Competence Centres (NCCs) across Europe
- EuroCC Austria is one of them
- Service Provider for AI, HPC and HPDA

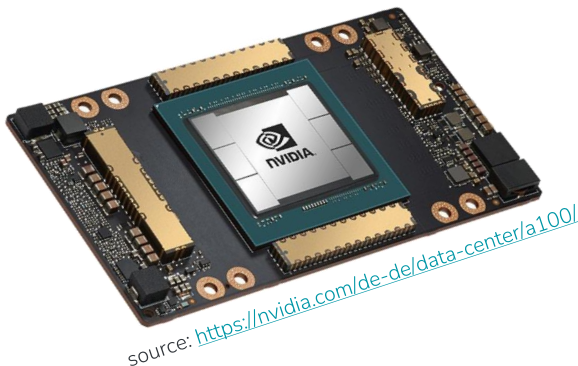


The Vienna Scientific Cluster

VSC-4 (2019)

790 CPU nodes

- 2x Intel Skylake Platinum CPUs
- 2x 24 cores per CPU
- 96 GB of memory per node



source: <https://nvidia.com/de-de/data-center/a100/>

VSC-5 (2022)

770 CPU nodes

- 2x AMD EPYC Milan
- 2x 64 cores per CPU
- 512 GB of memory per node

60 GPU nodes 2x NVIDIA A100,

- 40 GB memory per GPU

40 GPU nodes 2x NVIDIA A40

- 40 GB memory per GPU

Need More Compute-Power?

LUMI

- Third most powerful supercomputer in Europe and the 8th globally (Nov 2024)
- Sustained computing power (HPL) is 380 petaflops
- Over 262 000 AMD EPYC CPU cores
- Equipped with AMD Radeon Instinct MI250X GPUs

<https://www.lumi-supercomputer.eu/>

Leonardo

- 4th most powerful supercomputer in Europe and the 9th globally (Nov 24)
- Sustained computing power (HPL) is 239 petaflops
- Intel new gen Sapphire Rapids 56 cores
- Equipped with custom NVIDIA A100 SXM6 64GB GPUs

<https://leonardo-supercomputer.cineca.eu/>

European HPC Landscape

EuroHPC JU systems

Different access modes:
Calls for Proposals

EuroHPC development access:
Opportunity to test the system

Applicants can request a small number of node hours to get acquainted with the supercomputers to further develop their software.



AI Factory

Coming to Austria very soon!

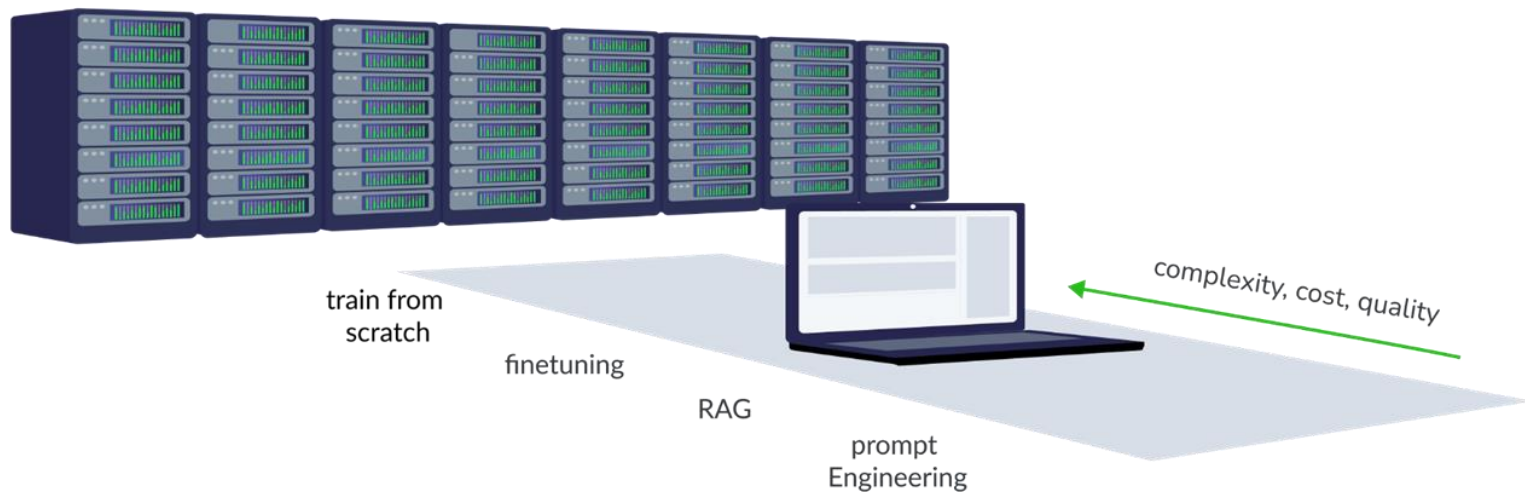
[Join the webinar on 31 March 2025](#)

Large Language Models on Supercomputers

A brief recap

Speaker: Simeon Harrison
Trainer at EuroCC Austria

How can you influence LLMs?



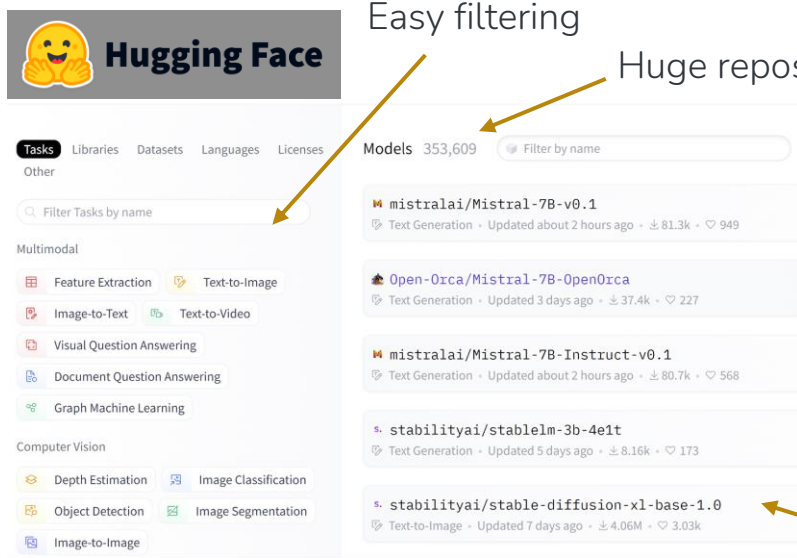
The Hugging Face Ecosystem

From 🤗 Transformers to the 🤗 Hub

Speaker: Simeon Harrison
Trainer at EuroCC Austria

Transformer Models

Spoilt for Choice at <https://huggingface.co/>



The screenshot shows the Hugging Face homepage. On the left, there's a sidebar with the 'Hugging Face' logo and navigation links: Tasks, Libraries, Datasets, Languages, Licenses, and Other. Below these are filters for 'Multimodal' (Feature Extraction, Text-to-Image, Image-to-Text, Text-to-Video, Visual Question Answering, Document Question Answering, Graph Machine Learning) and 'Computer Vision' (Depth Estimation, Image Classification, Object Detection, Image Segmentation, Image-to-Image). The main content area is titled 'Models 353,609' and includes a 'Filter by name' search bar. A list of models is displayed, including 'mistralai/Mistral-7B-v0.1', 'Open-Orca/Mistral-7B-OpenOrca', 'mistralai/Mistral-7B-Instruct-v0.1', 'stabilityai/stablelm-3b-4e1t', and 'stabilityai/stable-diffusion-xl-base-1.0'. Annotations with arrows point to the 'Filter Tasks by name' search bar (labeled 'Easy filtering'), the 'Models 353,609' header (labeled 'Huge repository'), and the 'stabilityai/stable-diffusion-xl-base-1.0' model entry (labeled 'All the relevant info').

Source: <https://huggingface.co/>



Pick the Right Model

mistralai/**Mistral-7B-Instruct-v0.2**   like 1.04k



Text Generation



Transformers



PyTorch



Safetensors

mistral

finetuned

conversational



arxiv:2310.06825



License: apache-2.0



Model card



Files and versions



Community 61



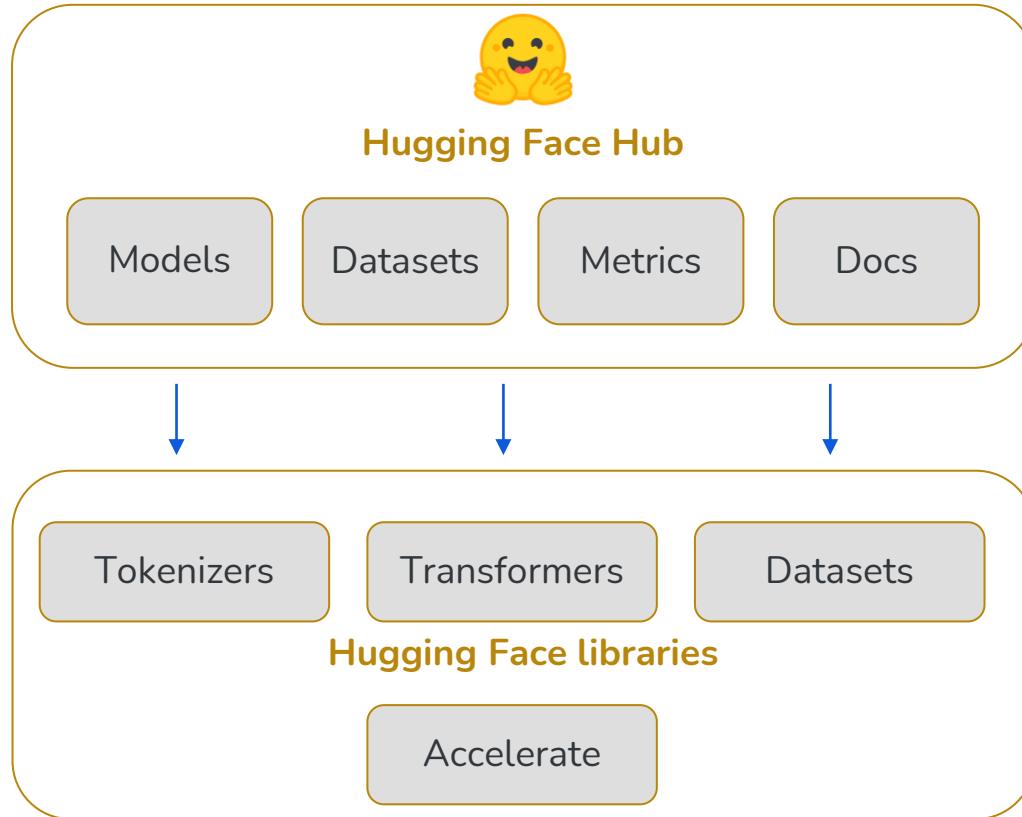
Edit model card

Model Card for Mistral-7B-Instruct-v0.2

The Mistral-7B-Instruct-v0.2 Large Language Model (LLM) is an improved instruct fine-tuned version of [Mistral-7B-Instruct-v0.1](#).

For full details of this model please read our [paper](#) and [release blog.post](#).

Hub and libraries



Transformer Anatomy

Attention is really all you need?

Speaker: Simeon Harrison
Trainer at EuroCC Austria

Transformer Anatomy

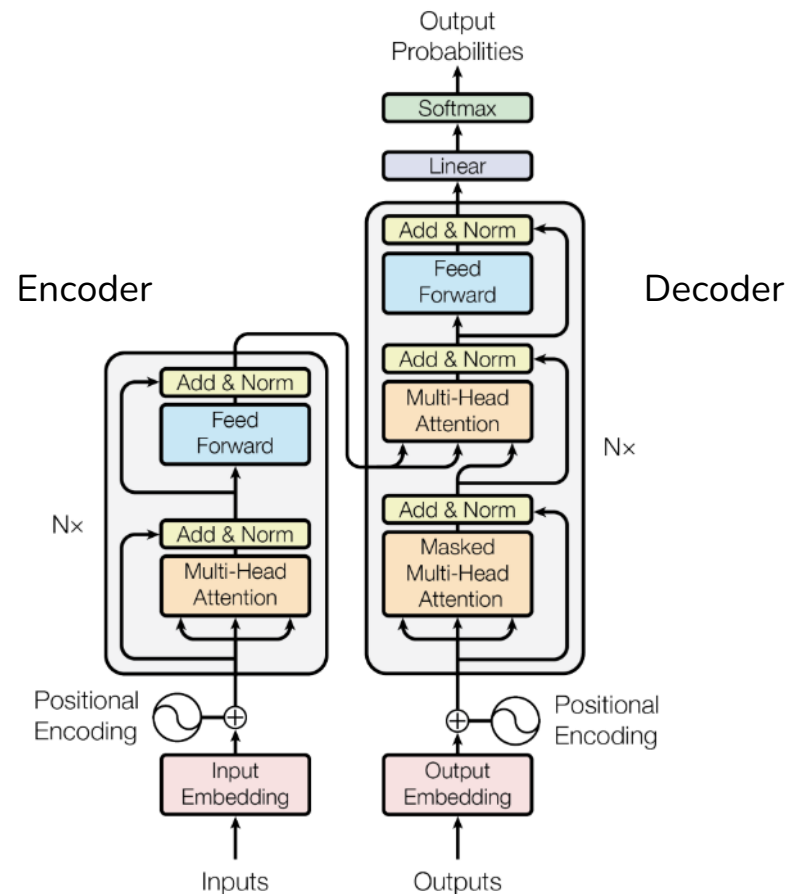
The original architecture

A transformer consists of an encoder and/or decoder block.

Words (tokens) are input as numerical representations (embeddings).

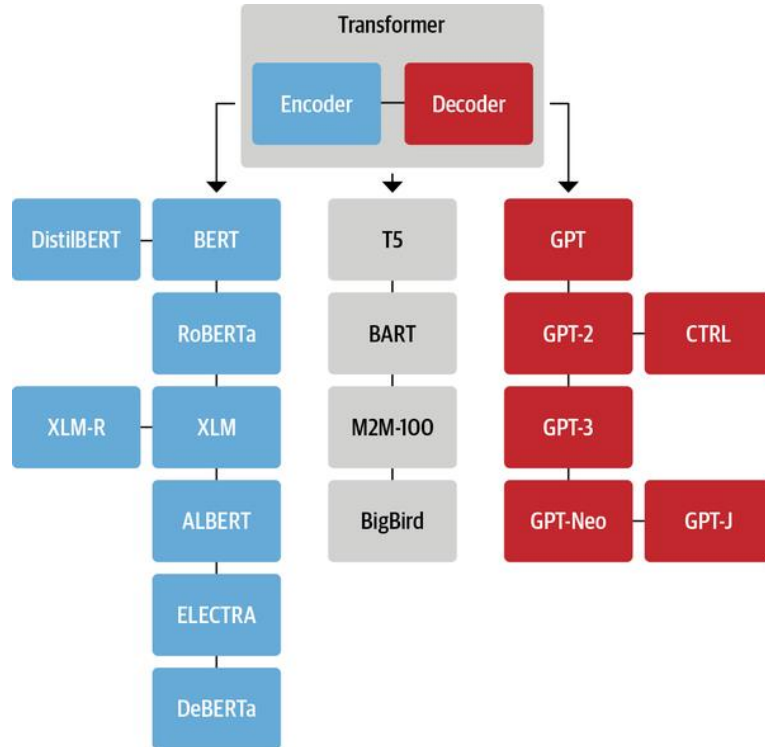
About 1/3 of all parameters are in the multi-head attention blocks

About 2/3 of all parameters are in the feed forward networks (also known as multi layer perceptron)



Source: "Attention Is All You Need", Vaswani et al.

Transformer Family



Encoder only:

These models excel at text classification, named entity recognition, and question answering

Decoder only:

Very good at predicting the next word in a sequence, therefore mostly used for text generation

Encoder-Decoder:

These models are often used for machine translation or summarization tasks.

From Text to Tokens

Word Tokenization

- Model does not have to learn words from characters
- Each word has specific ID
- Size of vocabulary explodes
- Model needs to learn different tokens for e.g. singular and plural

Large language models
on supercomputers

From Tokens to Vectors

Embeddings

Tokens are mapped to unique integers according to the vocabulary size of the tokenizer.

Now, the tokens need to be embedded, which means turned into a vector representation.

This is done by an embedding layer of a model. The model takes each token ID and looks it up in an embedding matrix. The embedding matrix is a learned set of weights that maps each token ID to a corresponding high-dimensional vector (embedding).

Woman →

$$\begin{bmatrix} 1.2 \\ 0.4 \\ 3.6 \\ 5.0 \\ 3.9 \\ 8.5 \\ 2.1 \\ 0.6 \\ 7.0 \\ 8.2 \\ 4.2 \\ 9.8 \\ 3.0 \\ \vdots \\ 1.1 \\ 4.3 \\ 2.7 \\ 3.3 \\ 0.0 \end{bmatrix}$$

Context Is All You Need

Embeddings

Here, we will refer to “word” instead of “token”, as it makes the content easier to explain.

A word embedding comes as a multi dimensional vector (e.g. 12.000 dim).

The initial word embedding in all of the examples of the word „mole“ is the same.



The European mole is a mammal



Take a biopsy of the mole

$$6.02 \times 10^{23}$$

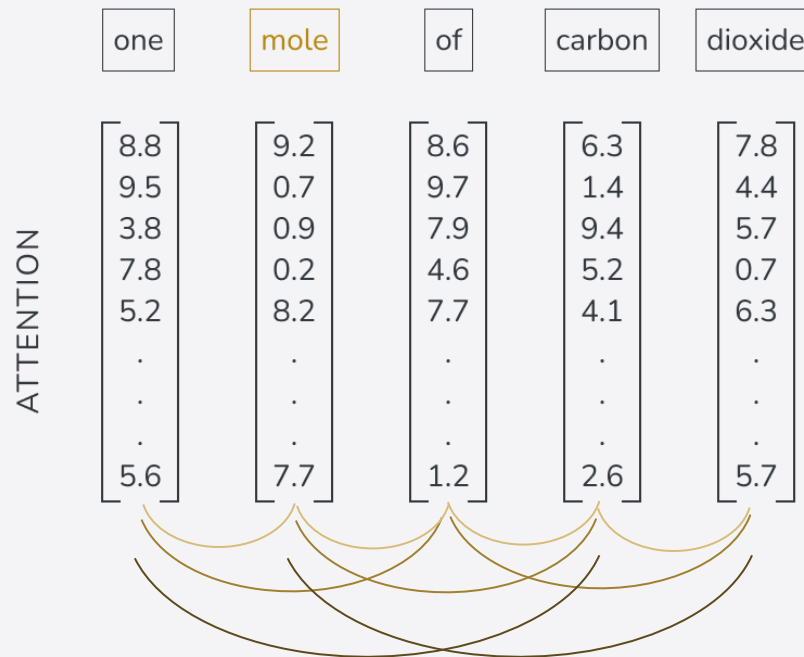
One mole of carbon dioxide

Context Is All You Need

Attention

The word „mole“ should be represented by a **unique vector** in the embedding space, depending on its **context**.

An **attention** block should **compute the vectors that you need to add** to the original, generic vector to get it to the correct, meaningful, rich representation, depending on the context in which the word is used.



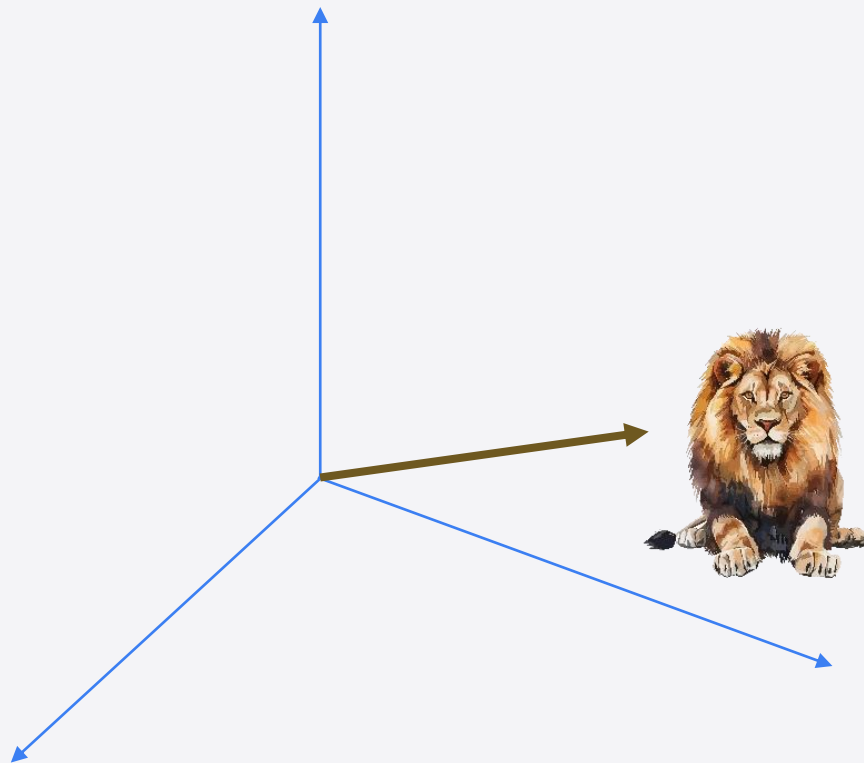
Context Is All You Need

Lion

We associate the word „lion“ with a big cat, living wild on the African continent.

We probably imagine a majestic predator with a big mane.

The embedding of the word „lion“ is a vector with a certain length and direction within the embedding space.

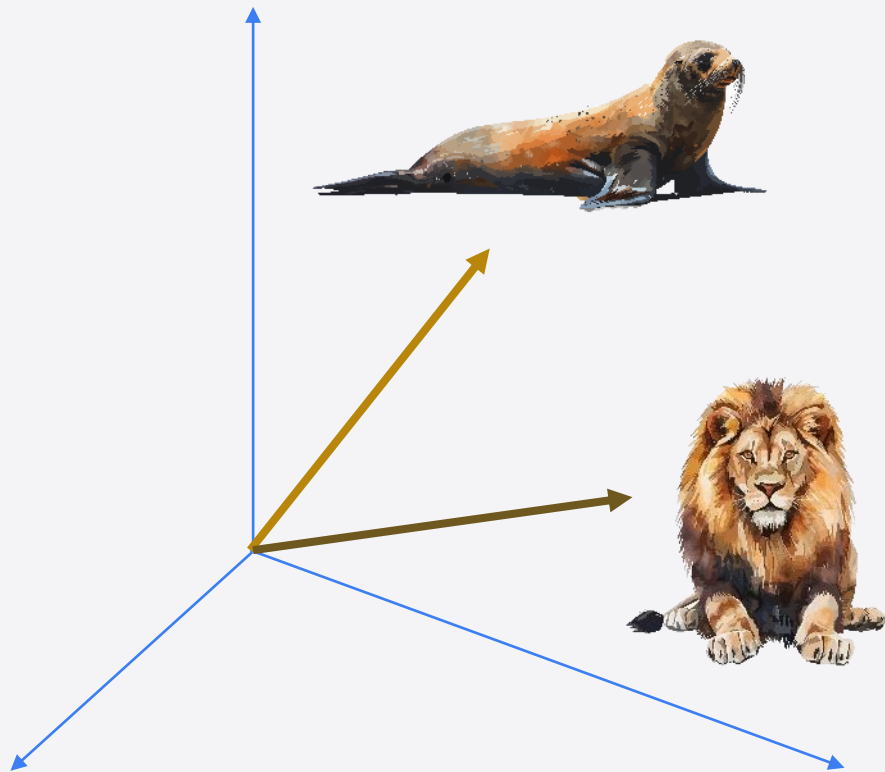


Context Is All You Need

Sea Lion

However, as soon we add the word „sea“ in front of „lion“ we imagine a totally different animal.

The same goes for the embedding. The attention mechanism needs to update the direction and length of the vector so that it represents the animal in question correctly.



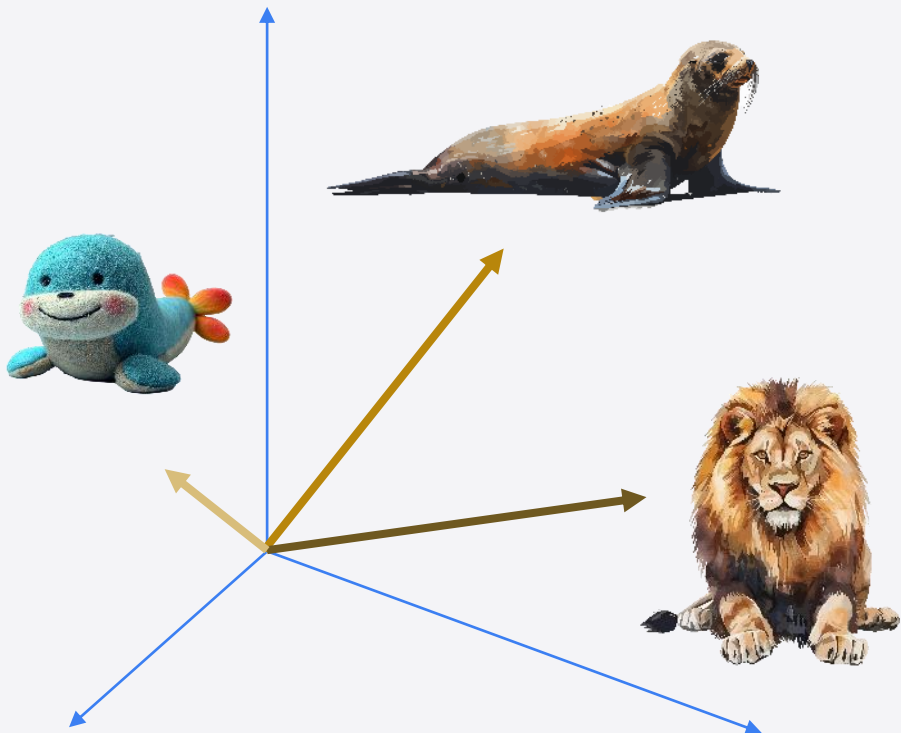
Context Is All You Need

Sea Lion Cuddly Toy

The context depends on more than just the immediate words to the left and right.

The embedding of „sea lion cuddly toy“ will certainly be very different of just „lion“.

In order to achieve that the vector for „lion“ needs to attend to all the other words in the input (context size).



Problems Arise

Data and Model too large

You might quickly encounter a situation in which your data and model no longer fit in your GPU's memory.

Memory footprint estimation for Mistral 7B (half precision):

$7 \times 2 = 14$ GB for the weights

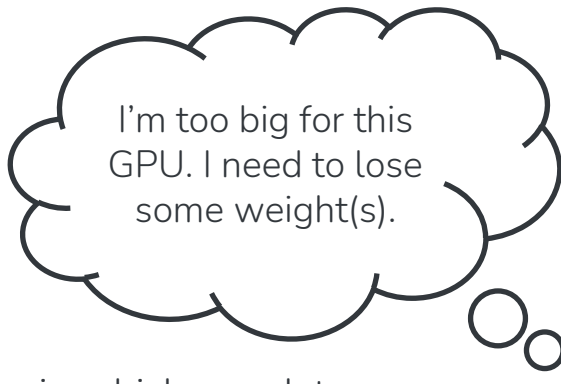
$7 \times 2 = 14$ GB for the gradients

$7 \times 2 \times 2 = 28$ GB for the optimizer state(s)




Total of 56GB for the model only!

7 comes from 7B parameters

2 stands for 2 Bytes per parameter



Limited GPU memory

Bits per parameter	Data type		Largest number possible
32 bits	FP32		3.389×10^{38}
16 bits	FP16		65504
16 bits	BFLOAT16		3.389×10^{38}

Fewer bits



Quantization



sign



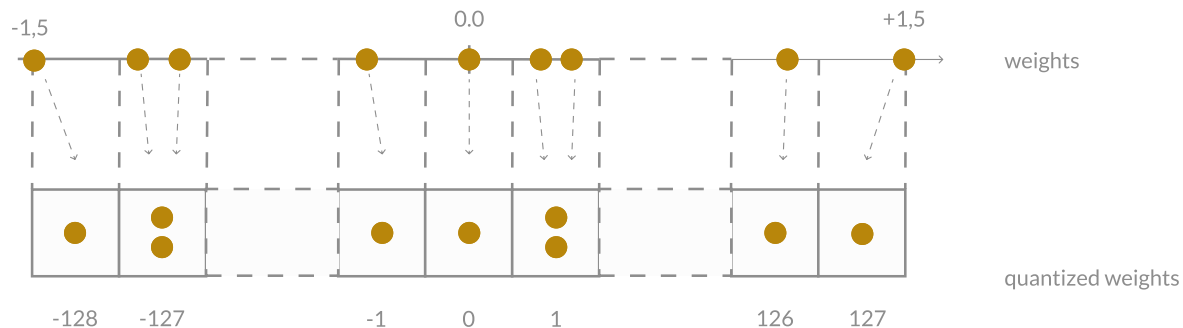
exponent



mantissa

$$Value = (-1)^S * 2^{(E-15)} * (1 - M)$$

Quantization



Unslot

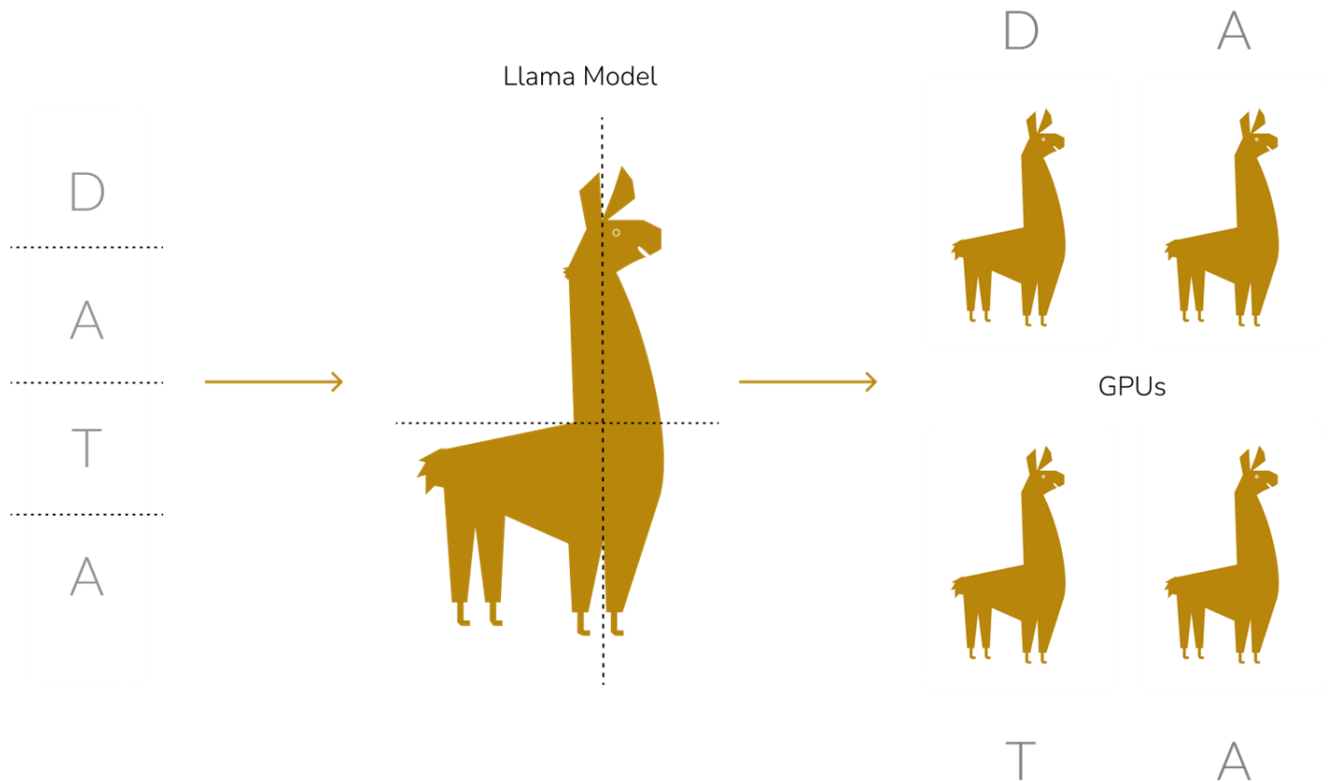


unslot

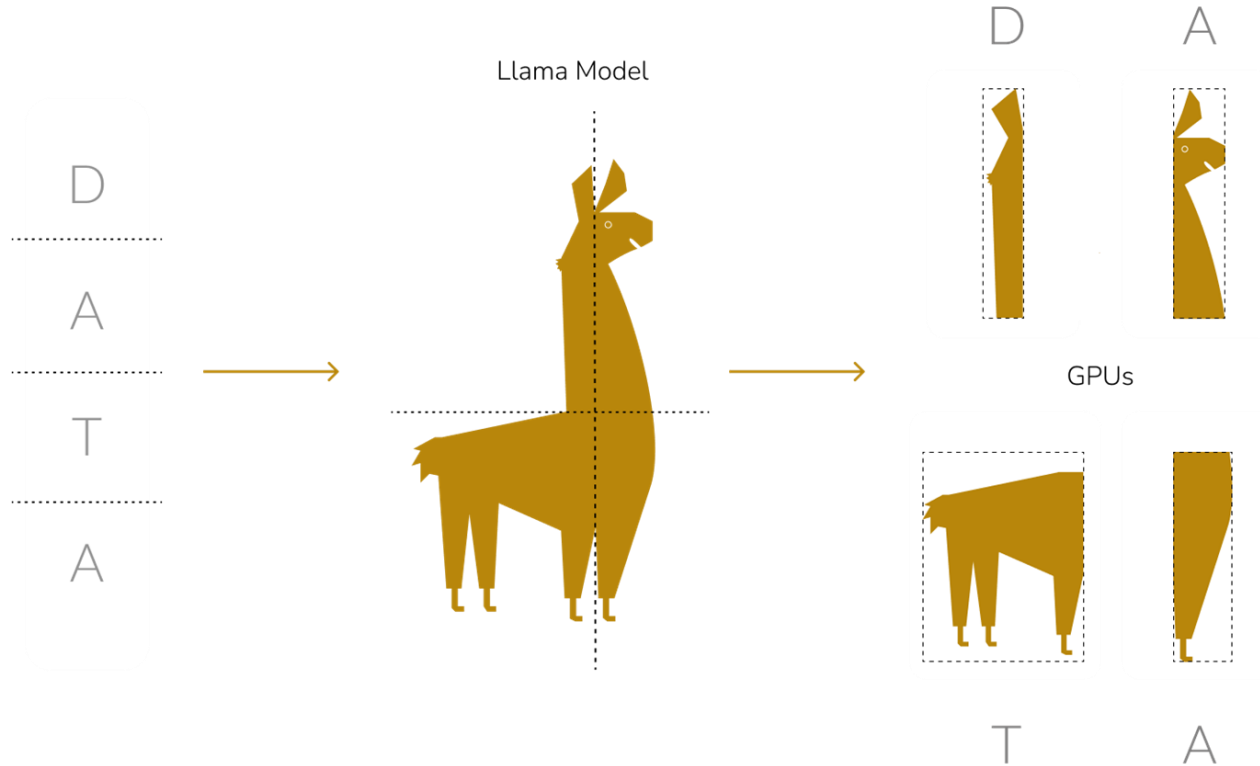
Optimized GPU kernels

created by manually deriving all
compute heavy maths steps

Data Parallelism



Model Parallelism



Prompt Engineering

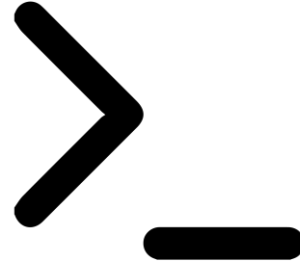
Definition:

The practice of designing inputs ("prompts") to guide the behavior and output of large language models (LLMs).

Goal:

Improve

- relevance
- accuracy
- reliability (or reproducibility)
- controllability



Role Prompting

Definition

Instructing the model to take on a specific role or persona.

Examples

- "You are a helpful legal assistant."
- "Act as a professional proofreader."

Effect

- Sets the tone and style
- Influences vocabulary and formality

Meta Prompting

Definition

Prompts that instruct the model on how to behave or how to generate prompts themselves.

Examples

- “Generate a prompt that would help someone learn about contract law.”
- “List questions that would clarify this legal clause.”

Effect

Self-reflection and model steering

Prompt Chaining

Definition

Connecting multiple prompts so that the output of one becomes the input of the next.

Example

1. P1: Summarize this legal clause
2. P2: Based on this summary, list 3 follow-up questions

Effect

- Enables complex workflows
- Decomposes task

LangChain



LangChain is an open-source framework for building applications with large language models. It provides abstractions for prompt templates, chains, memory, agents and tool integration.

[Documentation](#)

THANK YOU



This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101101903. The JU receives support from the Digital Europe Programme and Germany, Bulgaria, Austria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Greece, Hungary, Ireland, Italy, Lithuania, Latvia, Poland, Portugal, Romania, Slovenia, Spain, Sweden, France, Netherlands, Belgium, Luxembourg, Slovakia, Norway, Türkiye, Republic of North Macedonia, Iceland, Montenegro, Serbia

STAY IN TOUCH



EuroCC Austria



@eurocc_austria



eurocc-austria.at