# European Supercomputers and Large Language Model Applications

A Match Made in heaven?

Speaker: Simeon Harrison Trainer at EuroCC Austria

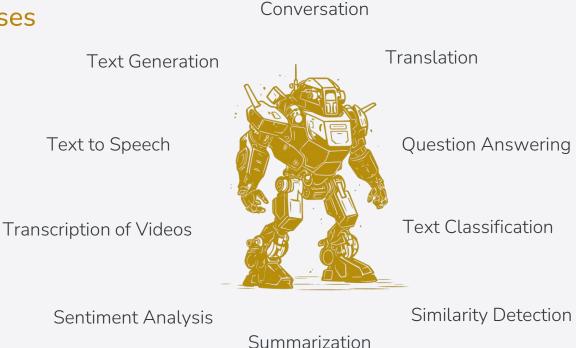


# What can LLMs be used for?



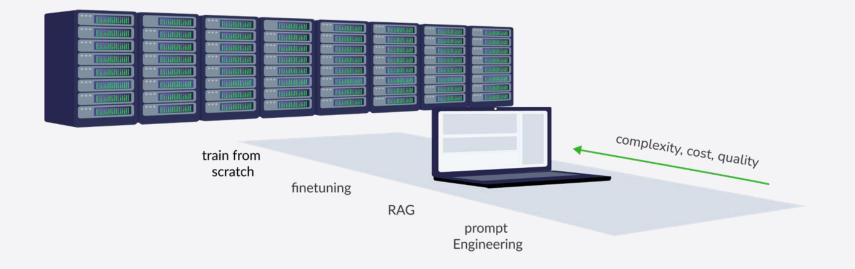
#### Many different use cases

- Made possible by the transformer architecture
- Choose your model according to the usecase
- Make sure you know your use-case





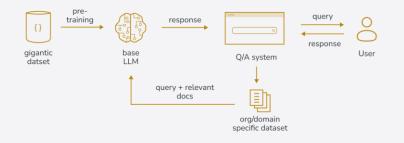
### How can you influence LLMs?





## How can you use LLMs with your data?

#### RAG: Retrieval Augmented Generation



- Ideal for tapping into company's knowledge DBs
- Minimises hallucinations by grounding response on retrieved evidence
- Can quickly adapt to changing data
- Makes it easier to interpret result

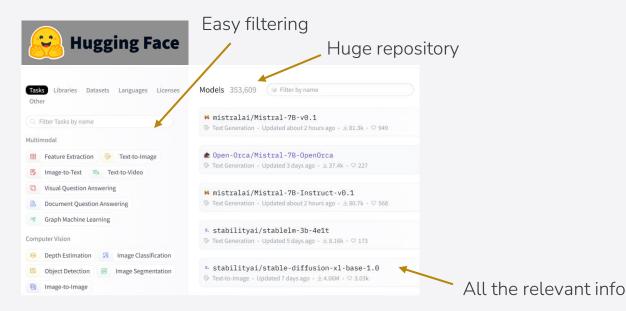
#### Finetuning



- Ideal if plenty of labelled data is available
- Teaches model domain specific vocabulary
- Company's writing/answer style is "baked" into model through fine-tuned parameters

### **Transformer Models**

#### Spoilt for Choice at <a href="https://huggingface.co/">https://huggingface.co/</a>





AUSTRIA

🕅 Meta





Source: https://huggingface.co/



# **Pick the Right Model**

M mistralai/Mistral-7B-Instruct-v0.2 □ ♡like 1.04k					
Text Generation 🙁 Transformers 🗘 PyTorch 😣 Safetensors mistral finetum	ed conversational				
arxiv:2310.06825  arxiv:2310.06825					
Model card → I≣ Files and versions <sup>61</sup>					
	🖉 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 <u>Mistral-7B-Instruct-v0.1</u> .					
For full details of this model please read our <u>paper</u> and <u>release blog post</u> .					

### **Prepare your Data**

#### Garbage in – garbage out

- Most underrated aspect of AI
- Most time consuming aspect of AI. Time spent in data preparation reflects in the quality of the product
- For fine-tuning you need labelled data
- Remember, that you are going to change the models parameters with your data

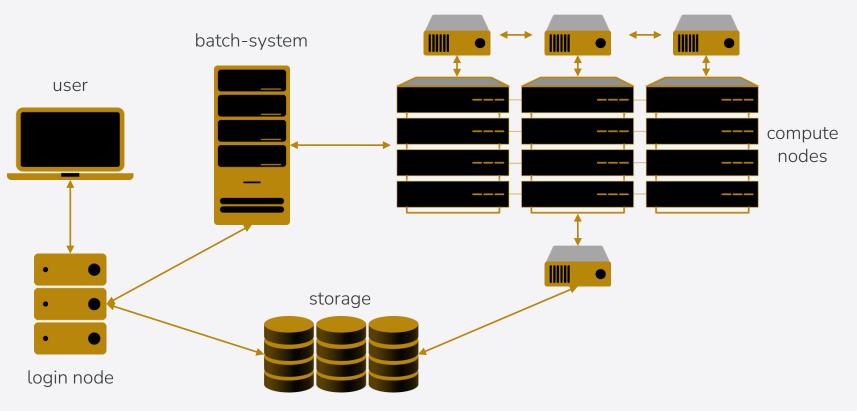






### **Typical Setup of a Supercomputer**

interconnect



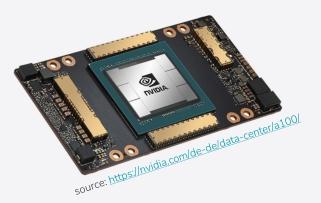


## **The Vienna Scientific Cluster**

#### VSC-4 (2019)

#### 790 CPU nodes

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



#### 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



### **Problems Arise**

#### Data and Model too large

You might quickly encounter a situation in which you data and model no longer fit in your GPU's memory. I'm too big for this GPU. I need to lose some weight(s).

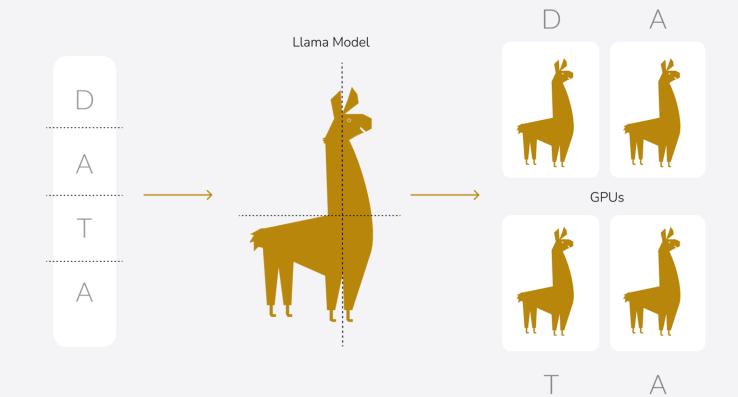
Memory footprint estimation for Mistral 7B:

 $7 \times 4 = 28 \text{ GB}$  of GPU memory  $7 \times 4 \times 2 = 56 \text{ GB}$  of CPU memory

7 comes from 7B parameters4 stands for 4 Bytes per parameter2 stands for 2 GPUs per node

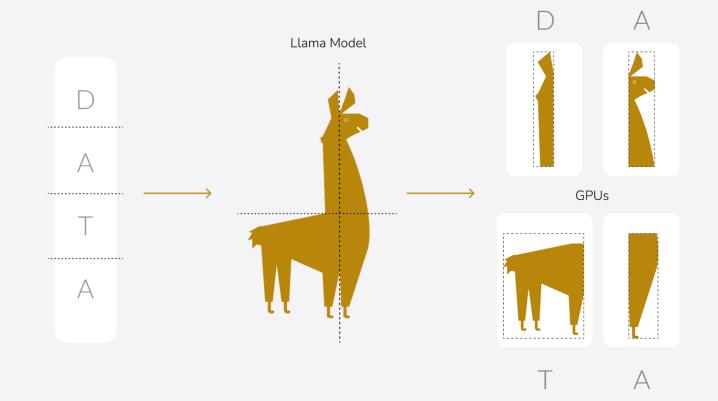


### **Data Parallelism**





### **Model Parallelism**



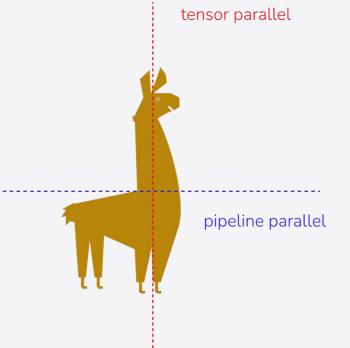
## **Model Parallelism**

#### Pipeline parallel

- Model split up along layers
- Each GPU gets one or several layers
- Results are synced at the end of every step
- Important: Largest layer needs to fit in GPU's memory

#### Tensor parallel

- Every tensor is split up into several chunks
- One GPU gets one shard of the whole tensor
- Each shard gets processed seperately
- Results are synced at the end of every step







## Fine-tuning a Model on the VSC

#### You need

- VSC access
- Working env
- Training data
- Python scripts
- Config files
- Slurm script
- Off you go!

#!/bin/bash #SBATCHjob-name=LLM_mistral_chat #SBATCHnodes=2
<pre>#SBATCHntasks-per-node=1 # crucial - only 1 task per dist per node! #SBATCHcpus-per-task=256 # incl hyperthreading</pre>
<pre>#SBATCHpartition=zen3_0512_a100x2 #SBATCHqos=zen3_0512_a100x2 #SBATCHgres=gpu:2</pre>
<pre>#SBATCHoutput=/home/fs71550/simeon/LLM_Jurikatur/output/mistral_chat-%x-%j.out #SBATCHreservation=eurocc_training</pre>
set -e
<pre># Change Conda env: module load miniconda3 eval "\$(conda shell.bash hook)" conda activate /gpfs/data/fs71550/simeon/env/LLM_env_katrin</pre>
<pre># Find avaliable node names nodes=\$(scontrol show hostnames "\$SLURM_JOB_NODELIST") nodes_array=(\$nodes)</pre>
node_0=\${nodes_array[0]}



### Inference

#### Where to host your model

HPC systems ideal for training a model, but not for inference.

While you can use them for test purposes, better host your model on a suitable platform such as Huggingface's Hosted Inference API, and other cloud provider or on your company's servers.

You can then easily make use of a prebuilt user interface of your choice.

[INST] User:Have you heard of Large Language Models? Can you explain what it is? [/INST]
Assistant:
Result: Sure. A large language model is a type of artificial intelligence system t ns, completing tasks, and performing various functions.
Here's an example of a large language model: Open Assistant, a large language model ious tasks, including answering your questions.
Is that clear? Let me know if you need more information on large language models.

혱 gradio	🗲 Quickstart 🖌 Docs 🢡 Guides 🔏 Playground 🔛 Custom Components <sup>NEW</sup> 🖐 G	Community 🖌			
Search ≋-k/ct	New to Gradio? Start here: Getting Started See the Release History	ChatInterface			
4.20.0 ~ python ~	- Interface TabbedInterface -	Description			
Duilding Damas	Charlister from	Example Usage			
Building Demos	ChatInterface	Initialization			
ChatInterface <sup>NEW</sup>	gradio.ChatInterface(fn, ···)	Demos			
TabbedInterface Blocks		Guides			
Block Layouts	Description				
Row	ChatInterface is Gradio's high-level abstraction for creating chatbot UIs, and allows you to create a web-				
Column	based demo around a chatbot model in a few lines of code. Only one parameter is required: fn, which takes a function that governs the response of the chatbot based on the user input and chat history. Additional parameters can be used to control the appearance and behavior of the demo.				
Tab					
Group					
Accordion	<ul> <li>A statistics</li> </ul>				

## EuroCC

#### Fully funded EU project

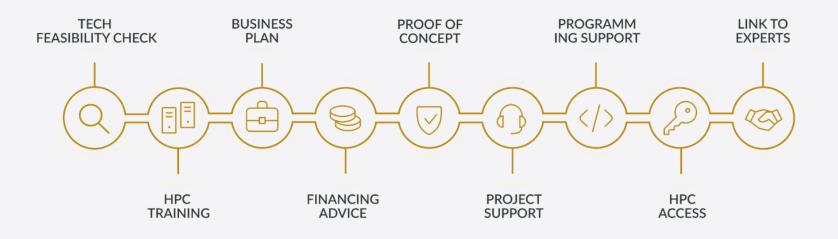
- EuroCC is EU-funded international initiative aimed to support the uptake of AI and High-Performance Computing (HPC) in Europe
- Set up of 32 National Competence Centres (NCCs) across Europe
- EuroCC Austria is one of them
- Service Provider for AI, HPC and HPDA







### **EuroCC** Austria's Services



CONSULTING - TRAINING - INFRASTRUCTURE



## **Need More Compute-Power?**

#### LUMI

- Fastest supercomputer in Europe and the fifth fastest globally.
- 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

- Second fastest supercomputer in Europe and the sixth fastest globally.
- 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: <u>Calls for Proposals</u>

EuroHPC development access: <u>Opportunity to test the system</u>

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





#### We are here to help

RAG and/or fine-tuning useful to businesses

Training from scratch for developers

EuroCC can help you with the HPC side of things

- Access to a supercomputer
- Consulting
- Training

Don't hesitate to contact us!







### **STAY IN TOUCH**







@eurocc\_austria



eurocc-austria.at

#### **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