

Small and large language models

Sonja Aits

Lund University
2026-09-15



Encoding

Data must be converted to numerical representations → Many ways to do this!

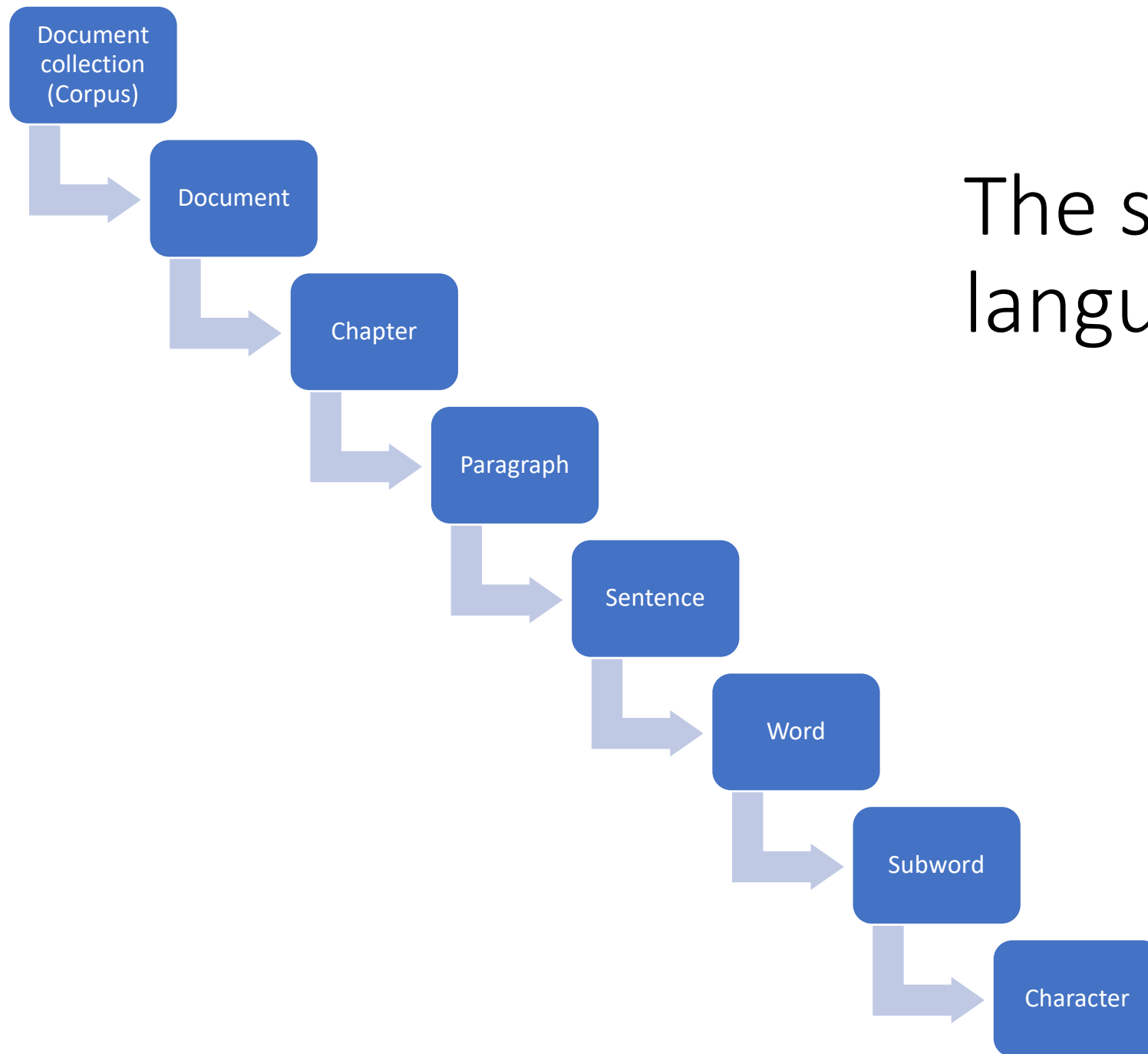
Binary encoding

Cancer	Healthy
1	0

One-hot encoding

Sample	Diseases	Sample	D	C	S
1	diabetes, cancer	1	1	1	0
2	diabetes, stroke	2	1	0	1
3	stroke	3	0	0	1
4	cancer, stroke	4	0	1	1

The structure of language data



How would you encode
sequences of text?

Tokenization splits text into meaningful pieces

Bumble bees can transport microorganisms.

Tokenization splits text into meaningful pieces

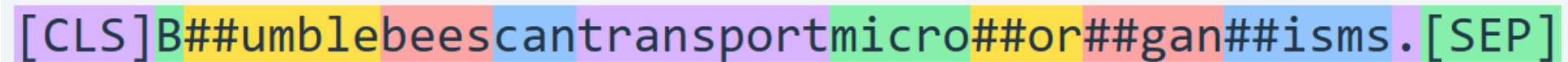
Bumble bees can transport microorganisms.

GPT-4

The image shows the sentence "Bumble bees can transport microorganisms." with each word highlighted in a different color: "Bumble" (purple), "bees" (yellow), "can" (red), "transport" (blue), "microorganisms" (green), and "." (yellow).

Bumble bees can transport microorganisms.

BERT-base-cased

The image shows the sentence "Bumble bees can transport microorganisms." with each word highlighted in a different color: "Bumble" (purple), "bees" (yellow), "can" (red), "transport" (blue), "micro" (green), "or" (yellow), "gan" (red), "isms" (blue), and "." (green). The tokens are separated by double hash symbols (#).

[CLS] B##umblebeescantransportmicro##or##gan##isms.[SEP]

[CLS] (Classification Token): At the beginning of a sentence

[SEP] (Separator Token): At the end of a sentence

Tokens are encoded as vectors ("embeddings")

- Today, embeddings are typically learnt during training
- Length: 100s-1000s

$$\text{bee} = \begin{bmatrix} -1.745 \\ 0.312 \\ -0.988 \\ 1.234 \\ -0.567 \end{bmatrix}$$

State-of-the-art NLP models encode words dynamically based on their context

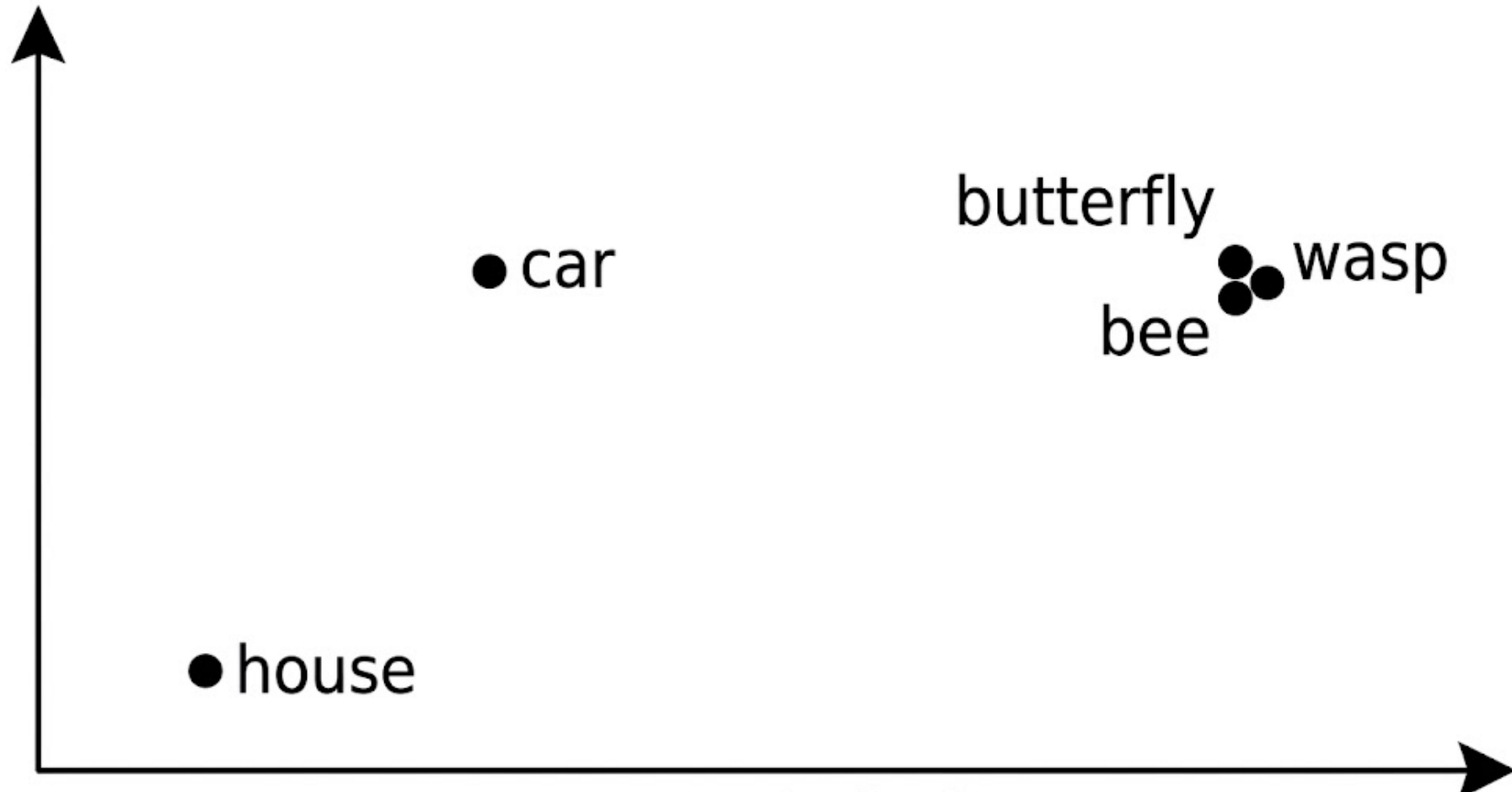
Occupation groups who spend many hours with computer work often suffer from wrist pain related to **mouse** use.

mouse → vector a

This study used a transgenic **mouse** model to evaluate the impact of TP53 mutations on lung cancer survival.

mouse → vector b

Related words have similar embeddings



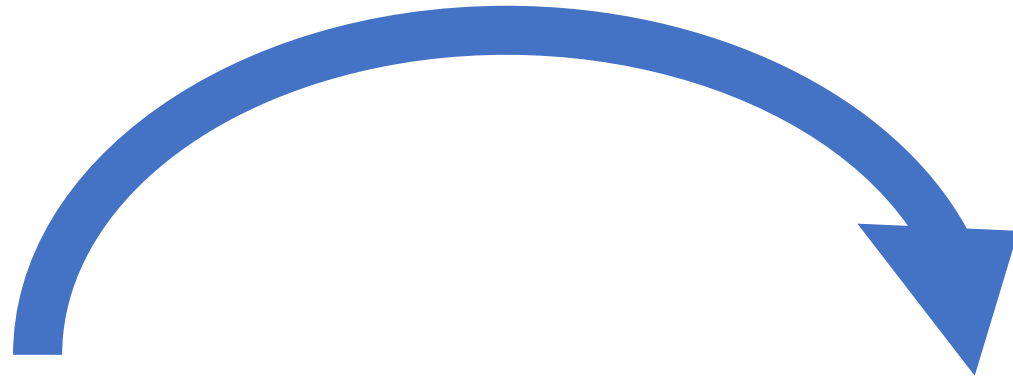
Models need to learn relationships across the sequence

The bee landed on the flower. Then it collected pollen.

≠

The	.	Then
bee	flower	it
landed		pollen
on	collected	the .

Models need to learn relationships across the sequence



The bee landed on the flower. Then it collected pollen.

Language models (LMs)

= models that learn statistical patterns in sequences of tokens and can thereby make predictions about those sequences.

LMs are trained on self-supervised proxy tasks

- **Masked-token prediction:** predict a missing token from the surrounding tokens

The bee landed on the ????. Then it collected pollen.

- **Next-token prediction:** predict the next token from the tokens that came before

The bee landed on the ???

→ **Enables learning from unlabeled texts**



are bumble bees|



are bumble bees — Search with Google



are bumble bees **dangerous**



are bumble bees **intelligent**



are bumble bees **dangerous to cats**



are bumble bees **friendly**



are bumble bees **independent** — Search



are bumble bees bees



are bumble bees **endangered**

Attention makes it possible to assign different importance to the input tokens

- Not all tokens are equally informative for the prediction

The bee landed on the ????. Then it collected pollen.

Attention makes it possible to assign differences in importance

- Not all tokens are equally informative for the prediction

*The **bee** landed on the ????. Then it collected **pollen**.*

Transformers are deep neural networks built around attention

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Lukasz Kaiser*
Google Brain
lukaszkaizer@google.com

Illia Polosukhin* †
illia.polosukhin@gmail.com

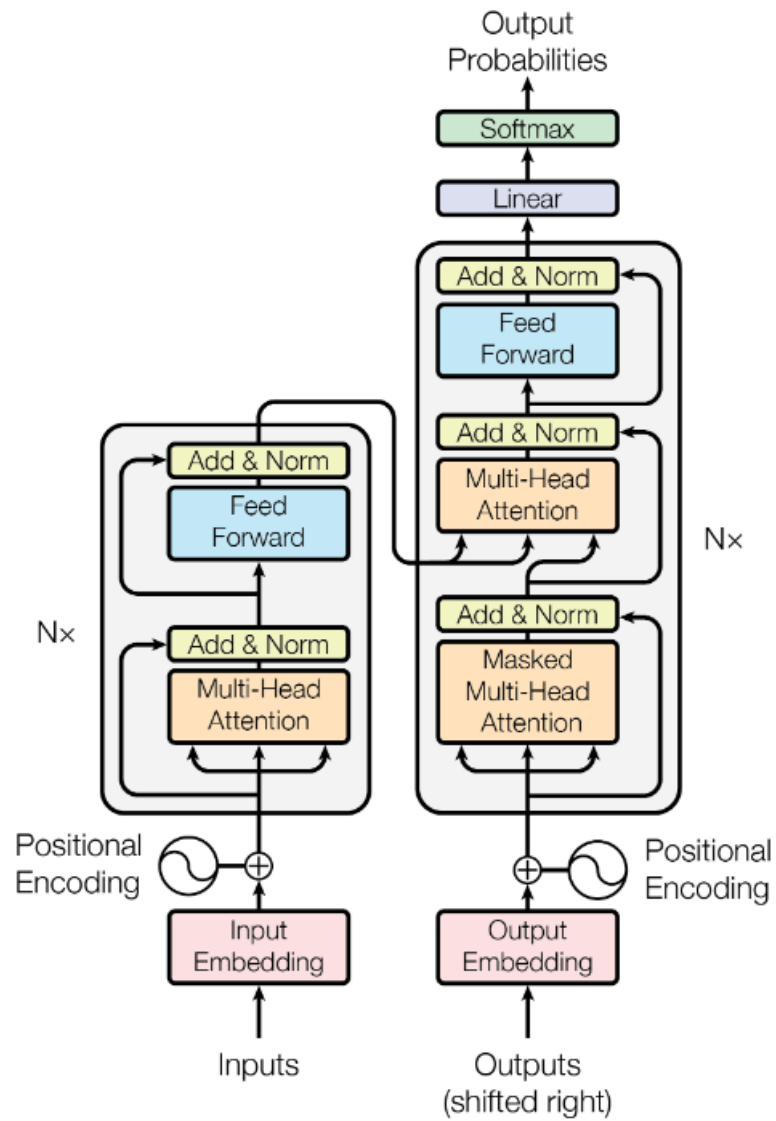
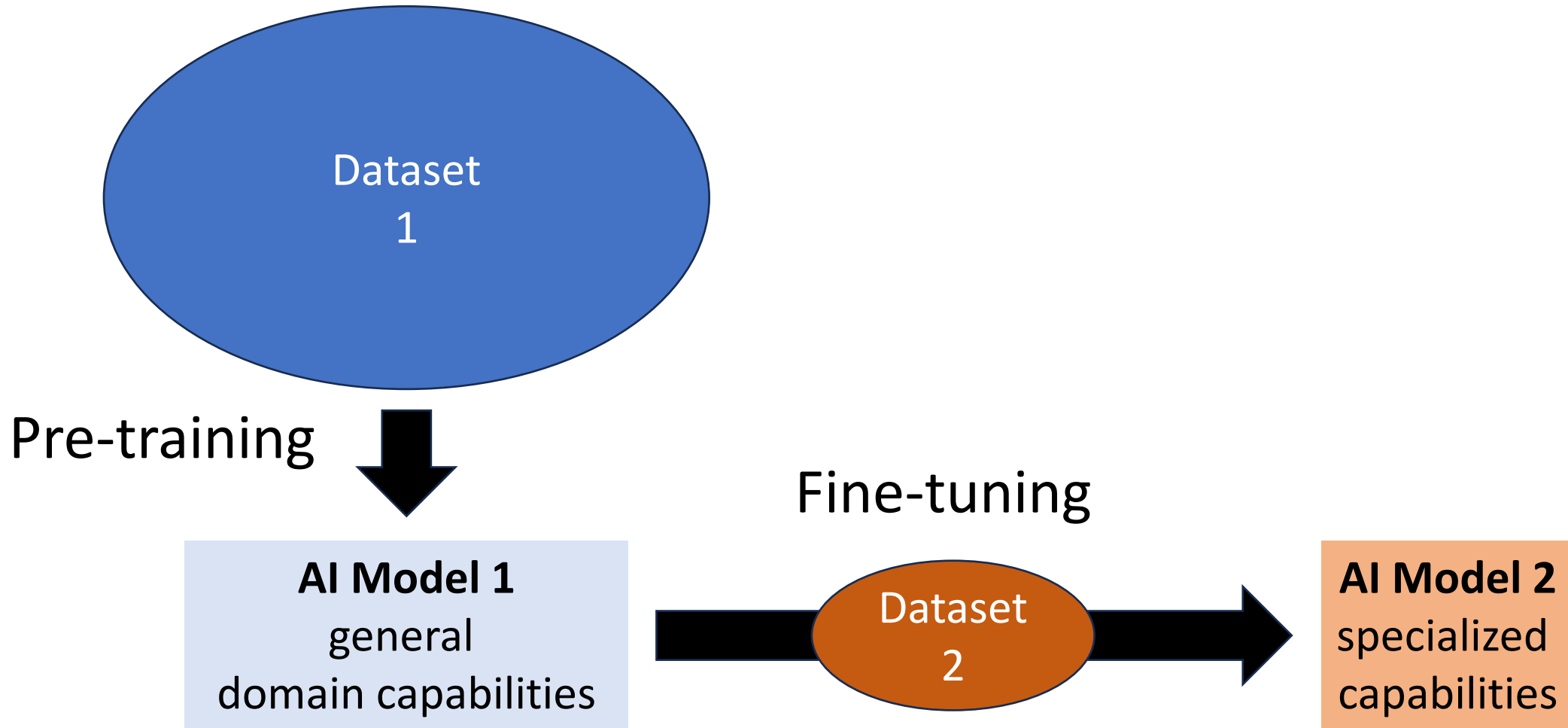


Figure 1: The Transformer - model architecture.

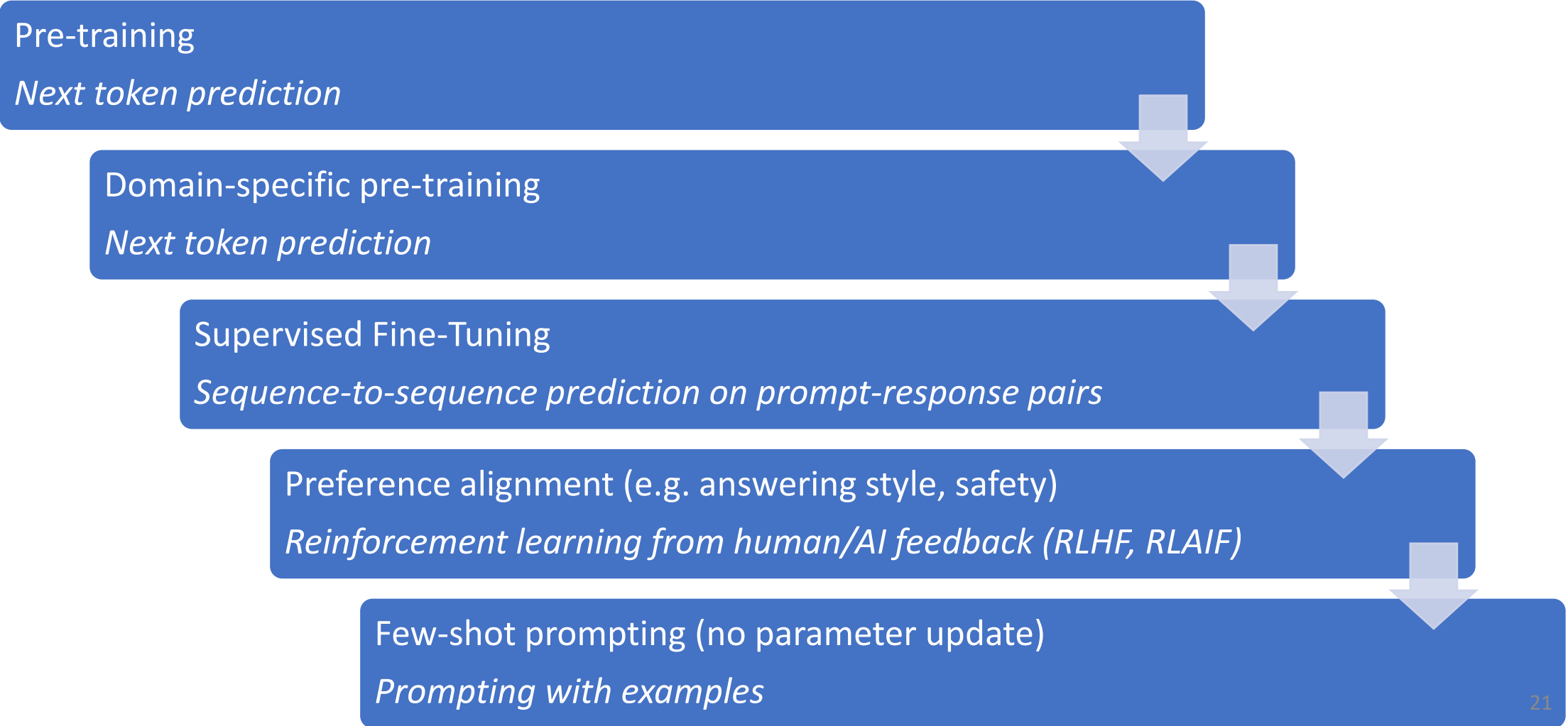
There are many different transformer architectures

Architecture	Main purpose	Attention pattern	Typical examples
Encoder-only	Understand/analyze input	Bidirectional: can see the whole input	BERT, RoBERTa
Decoder-only	Generate text	Causal: can only see previous tokens	GPT-style models, Llama
Encoder–decoder	Transform one sequence into another	Encoder sees source bidirectionally; decoder generates causally and attends to encoder	T5, original Transformer

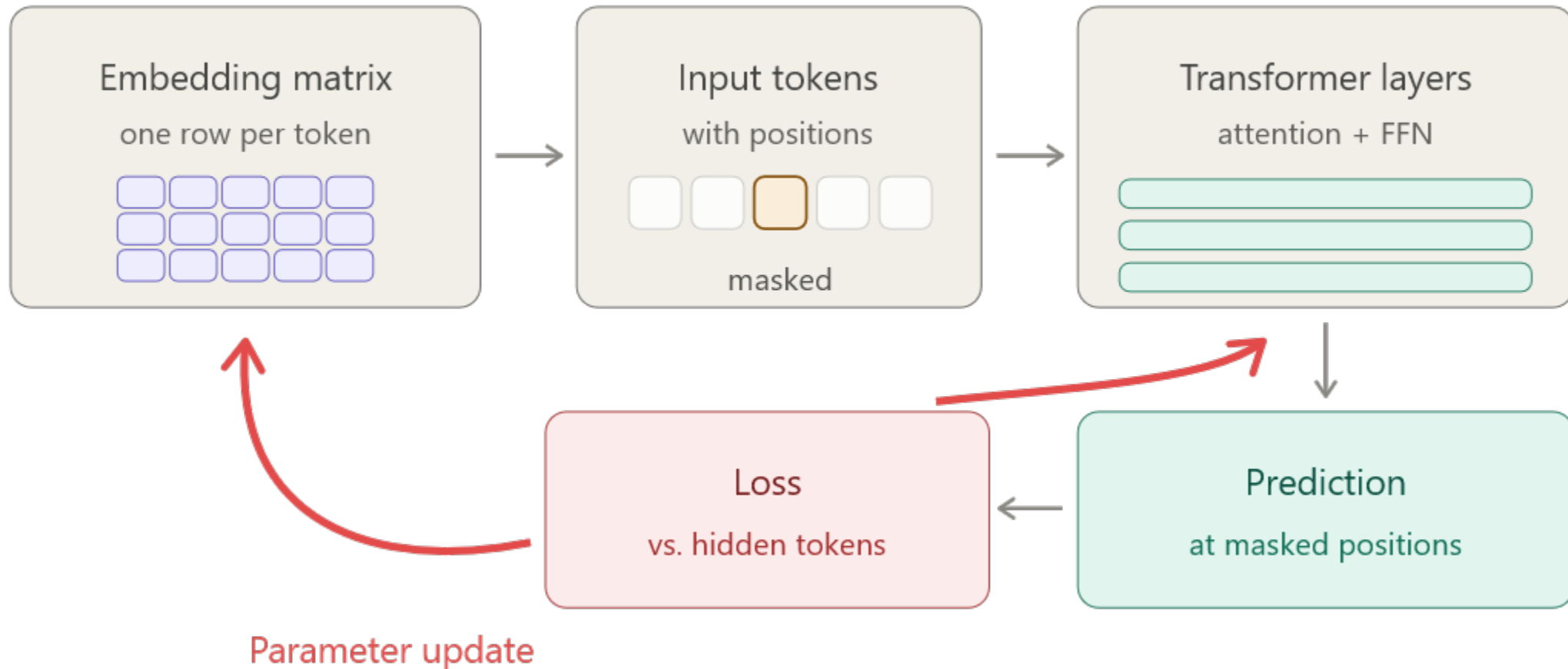
Smaller encoder-only language models are often trained with transfer learning

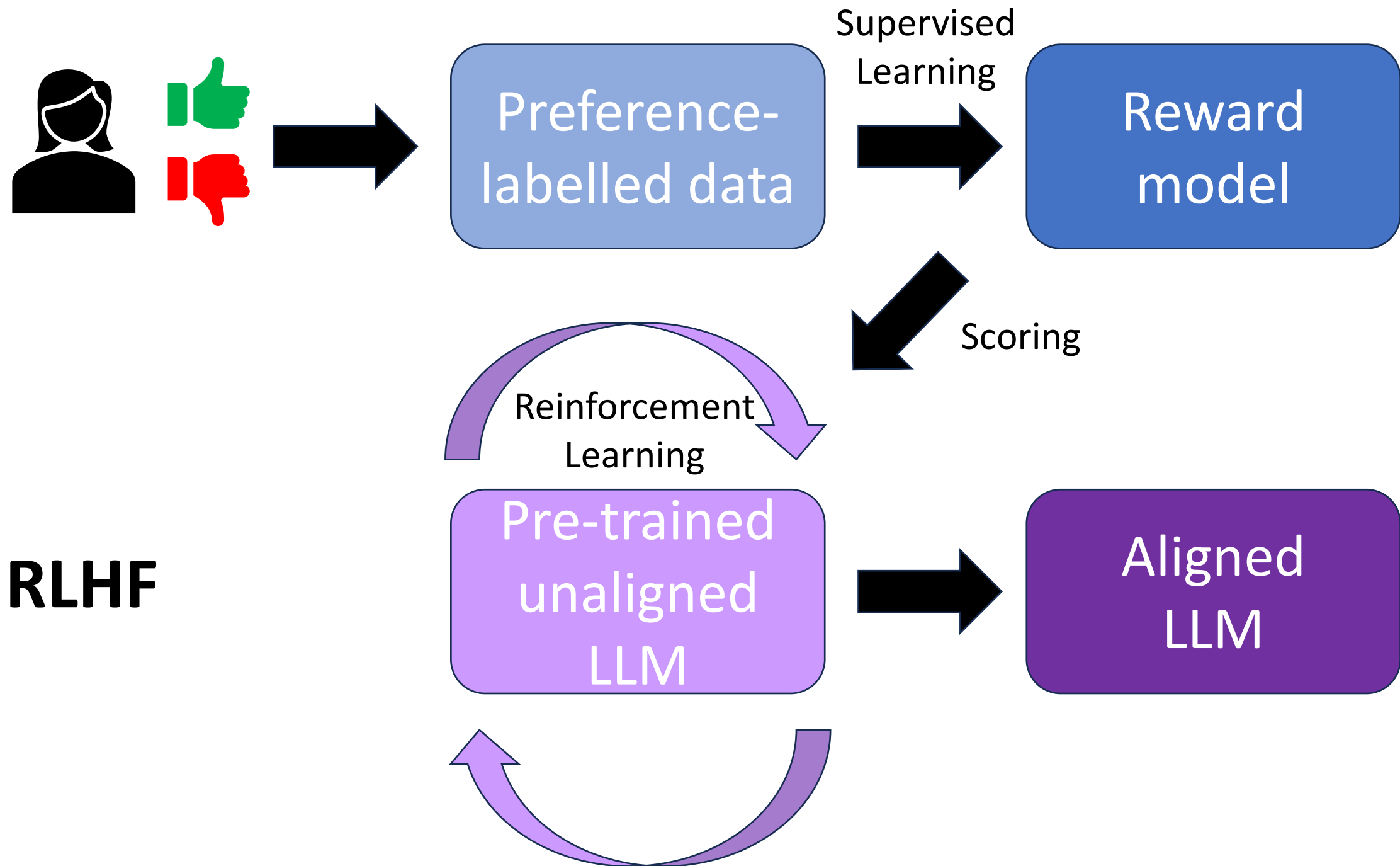


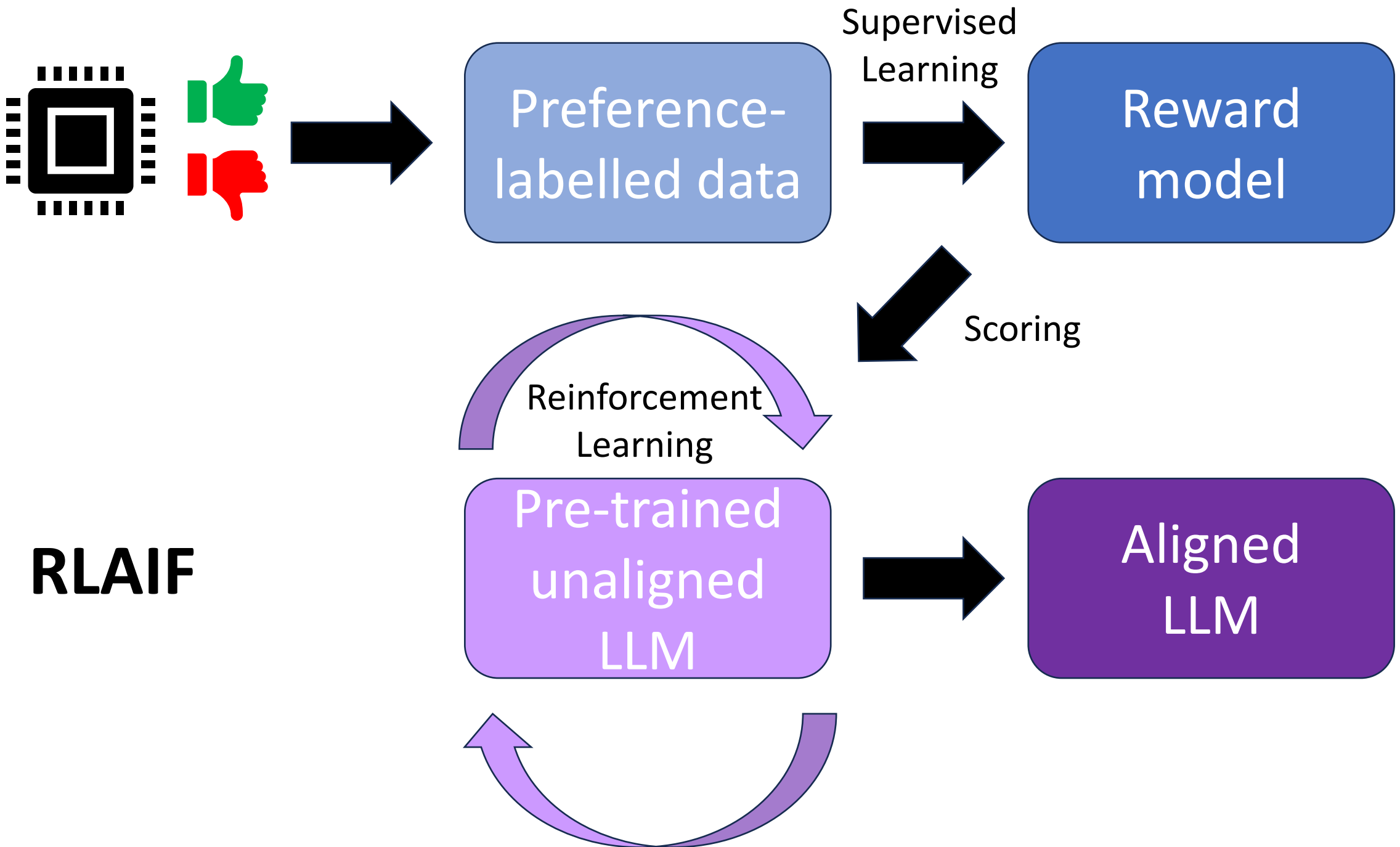
General-purpose large language models are usually trained in a stepwise manner



Pre-Training procedure (masked token prediction)



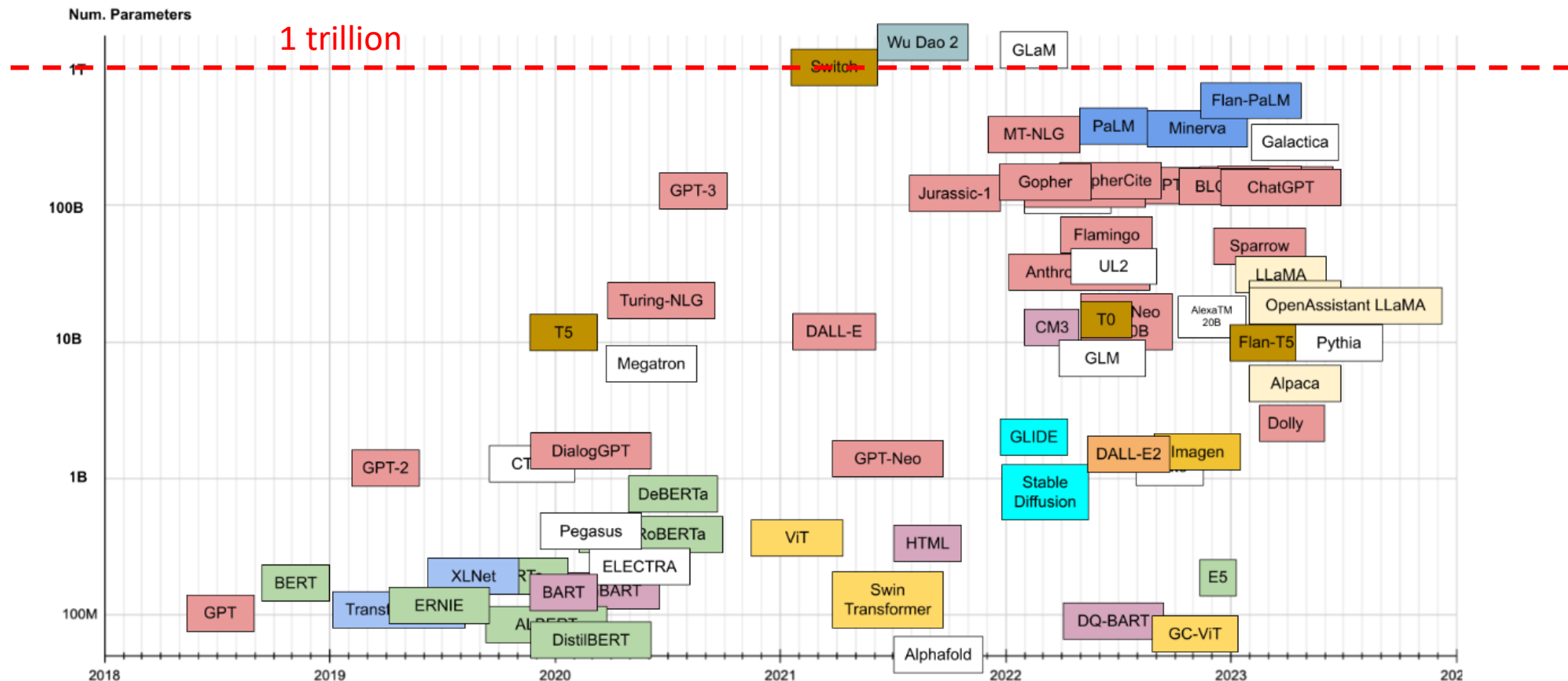




Computational resources

- Models usually trained on GPUs
- Small language models can be trained on CPUs (even on laptop)
- LU students/staff have access to free distributed supercomputing infrastructure: NAISS
- SOTA models often require enormous computational resources

Frontier LMs are getting bigger and bigger



Frontier LMs are getting bigger and bigger

Kimi K3

Total parameters: 2.8 trillion

Vocabulary: 160 000

Attention Hidden Dimension: 7168

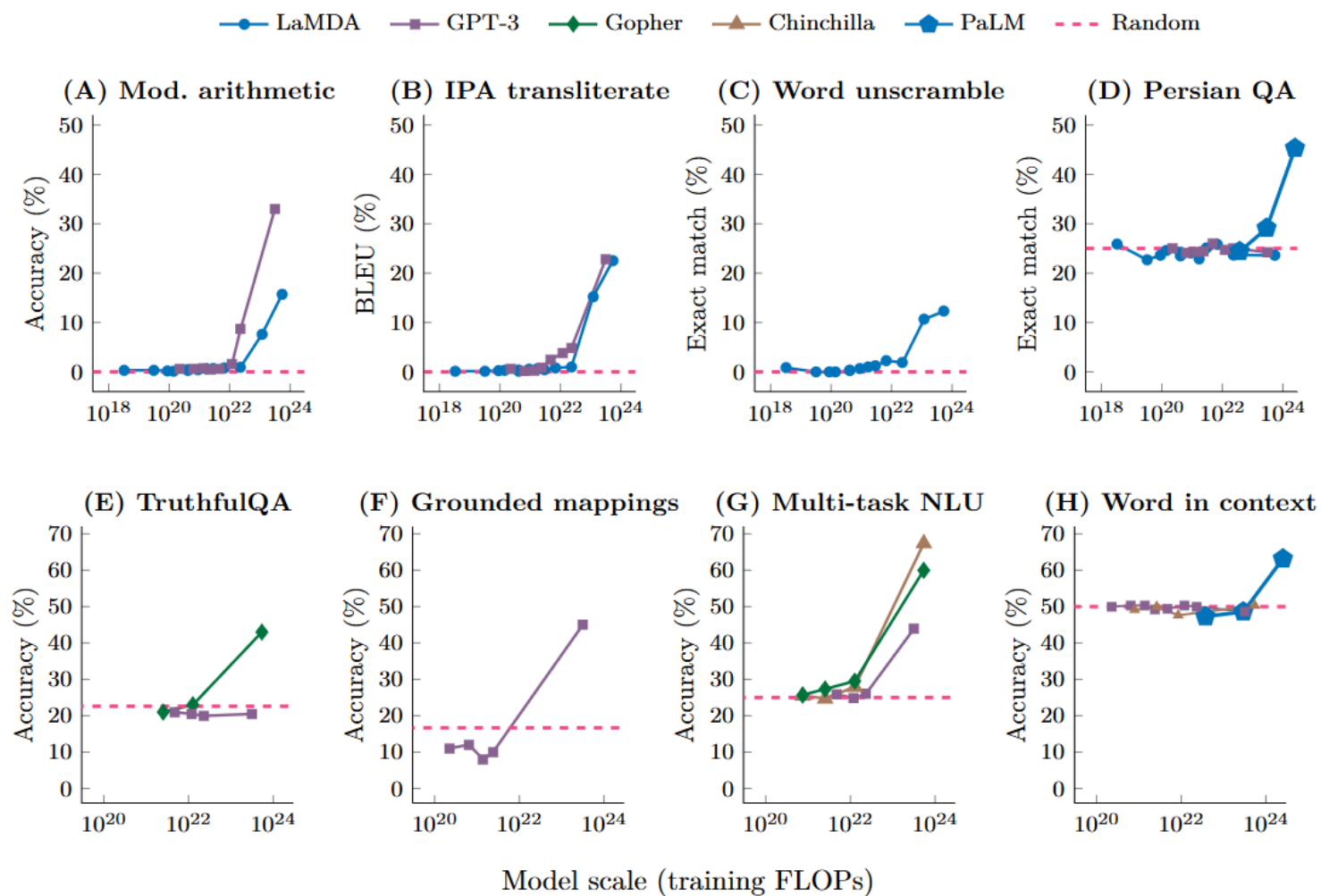
Context window: 1048576

Layers: 93

Attention heads: 96

<https://huggingface.co/moonshotai/Kimi-K3>

New capabilities emerge at larger scales



Wei et al. 2022

<https://doi.org/10.48550/arXiv.2206.07682>

LLMs can operate across a variety of tasks and domains



Computer Science > Computation and Language

[Submitted on 22 Mar 2023 (v1), last revised 13 Apr 2023 (this version, v5)]

Sparks of Artificial General Intelligence: Early experiments with GPT-4

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, Yi Zhang

One-shot vs Few-shot vs Fine-tuning

Same goal — adapt a pre-trained model to a new task — different ways to provide the task information.

One-shot

Provide a single example in the prompt.

The model infers the task from just one example
(at inference time, no weight updates).



- ✓ No training or weight updates
- ✓ Uses just one example
- ✓ Simpler prompt, but less reliable than few-shot

Few-shot

Provide a few examples in the prompt.

The model learns the task from these examples
(at inference time, no weight updates).

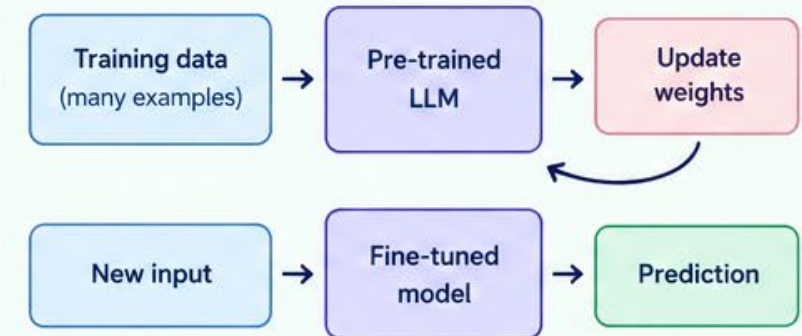


- ✓ No training or weight updates
- ✓ Uses just a few examples
- ✓ More reliable than one-shot

Fine-tuning

Update the model's weights using labeled examples from the target task.

After fine-tuning, the model is better adapted to the task and can generalize to new inputs.



- ✓ Requires training with labeled data
- ✓ Updates model weights
- ✓ Sometimes give the best performance (especially for complex or specialized tasks)

LLMs can reason in a stepwise manner (Chain of Thought reasoning)

Question: A mutant bacterium synthesizes RNA at 30°C, but halts all RNA synthesis instantly at 42°C. Is the mutation in RNA Polymerase or a specific operon regulator?

Chain of Thought:

Mutation type: Instant shutoff at 42°C indicates a heat-sensitive protein unfolding.

Impact scope: The shutdown is global (all RNA), not local (one gene set).

Regulator vs. Core enzyme: A regulator affects only its own operon; RNA Polymerase drives all transcription.

Conclusion: RNA Polymerase. A core enzyme failure halts global transcription; a regulator failure would only affect specific genes.

But reasoning not always correct

LLM failures

- Hallucinations
- Sycophancy
- Prompt sensitivity to wording
- Context limitations
- Bias
- Knowledge limitations
- Distribution shift
- Unproductive loops

Capability ≠ Reliability ≠ Truth

Pick the appropriate tool for the task

	Small models, e.g. BERT	Frontier models, e.g. GPT-5
Best for	Narrow task	Complex tasks/Diverse tasks
Speed	Fast	Slower
Compute requirements	Low	Higher
Costs	Low (but development cost)	Higher
Reasoning	None	Strong
Flexibility	Low	High
Examples	Classification, information extraction	Coding, data analysis, text generation, reasoning
Use when...	Scale & efficiency matter	Diverse capabilities matter

Deep neural networks are not always the optimal solution

- Keyword matching (Dictionary-based NLP)
- Rule-based systems

Advantages	Disadvantages
High explainability	Can be slow to scale
Fast inference and prototyping	Rules not always possible
Low infrastructure requirements	Context not considered
No encoding required	Often poor performance

Transformers can also be used for other sequential data

- Long sequences are split into chunks (like tokenization)
 - Speech (continuous wave) → Audio frames → Audio encoder
 - DNA sequence → Codons (triplets), motifs
 - Protein sequence → Amino acids, motifs
- Audio models: Whisper, Wav2Vec2