Multimodal Generative AI 2025
Foundation Models in practice

VISTA
École Polytechnique

Today's lecture

Part I: Recap

Part II: Align with Human Feedback

Part III: Evaluation

Part IV: Inference

Part V: Optimization techniques

Part VI: External interactions

Slides adapted from various sources: [Intro to Large Language Models, Andrej Karpathy, Executive Education Polytechnique, Udemy, Deeplearning.ai, Stanford University CS231n, Financial Times, New York Times, Hi!Paris summer school 2023]

Today's lecture

Part I: Recap

Part II: Align with Human Feedback

Part III: Evaluation

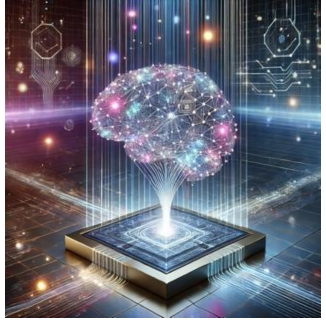
Part IV: Inference

Part V: Optimization techniques

Part VI: External interactions

Slides adapted from various sources: [Intro to Large Language Models, Andrej Karpathy, Executive Education Polytechnique, Udemy, Deeplearning.ai, Stanford University CS231n, Financial Times, New York Times, Hi!Paris summer school 2023]

Recap



created with ChatGPT, Oct 2024

Lecture 4

Part I:
Definition:
Define usecase

Part II:
Select:
Foundation Model to use or pretrain

Part III:
Adapt:
Foundation Model

Slides adapted from various sources: [Intro to Large Language Models, Andrej Karpathy, Executive Education Polytechnique, Udemy, Deeplearning.ai, Stanford University CS231n, Financial Times, New York Times, Hi!Paris summer school 2023]

Lecture 4, Part II: Summary

Select

- Choose an existing model or pretrain your own

- Scaling
  - Challenges
  - Cost
  - Scaling laws

- Pre-training for domain adaptation

created with chatGPT

Considerations for choosing a model

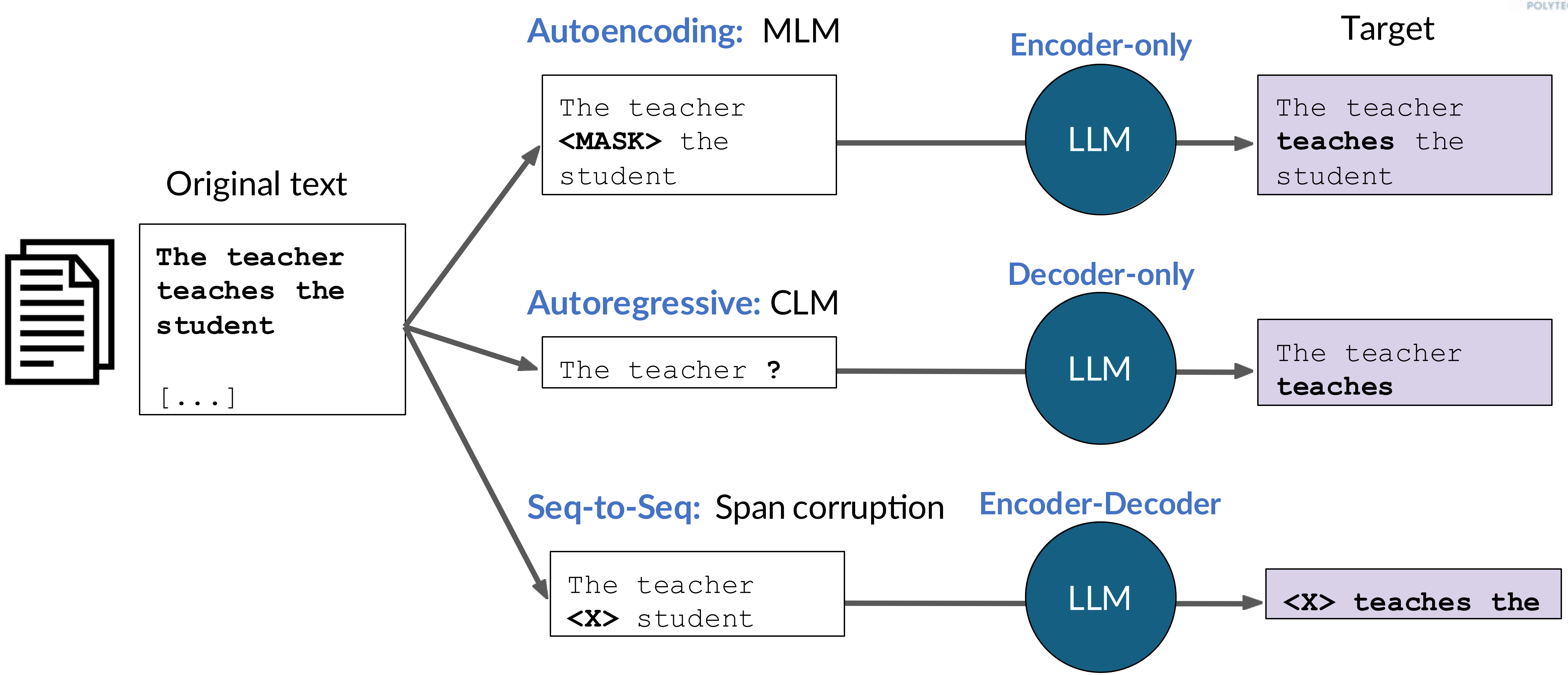
Foundation model

Pretrained LLM

Train your own model

Custom LLM

Model architectures and pre-training objectives



Compute...

```
OutOfMemoryError: CUDA out of memory.
```

GPU RAM needed to train larger models

As model sizes get larger, you will need to split your model across multiple GPUs for training

1B param model

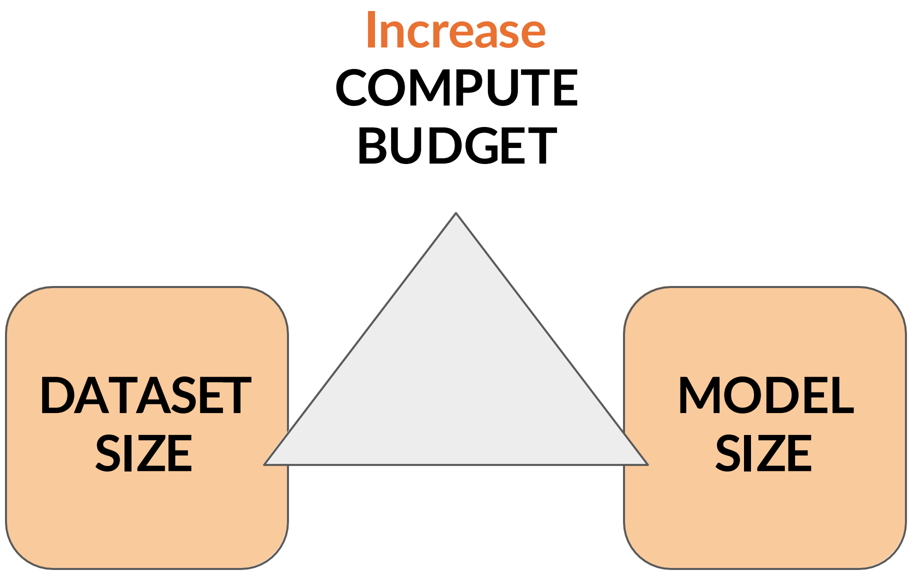
4,200 GB @ 32-bit full precision

175B param model

500B param model

12,000 GB @ 32-bit full precision

Increase compute budget
→ increase performance?

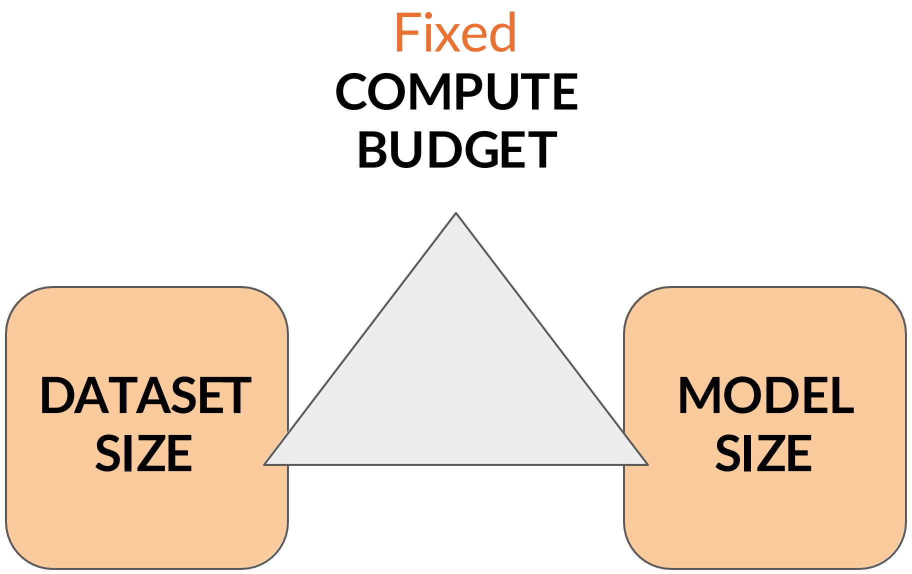


Scaling up Transformers

Model	Layers	Width	Heads	Params	Data	Training
Transformer Base	12	512	8	65M		8xP100 (12h)
Transformer Large	12	1024	16	213M		8xP100 (12h)
Bert Base	12	768	12	110M	13GB	
Bert Large	24	1024	16	340M	13GB	
XLNet Large	24	1024	16	~340M	126GB	512xTPUv3 (2.5 days)
RoBERTa	24	1024	16	355M	160GB	1024xV100 GPU (1 day)
GPT-2	48	1600	?	1.5B	40GB	
Megatron-LM	72	2072	32	8.3B	174GB	512xV100 GPU (9 days)
Turing NLG			28	17B	?	256xV100 GPU

~350k euros!

Chinchilla scaling laws for model + dataset size



Model	# params	Compute-optimal* # of tokens (~20x)	Actual tokens
Chinchilla	70B	~1.4T	1.4T
LLaMA-65B	65B	~1.3T	1.4T
GPT-3	175B	~3.5T	300B
OPT-175B	175B	~3.5T	180B
BLOOM	176B	~3.5T	350B

Compute optimal training datasize
is **~20x** number of parameters

* assuming models are trained to be compute-optimal per Chinchilla paper

Sources: Hoffmann et al. 2022, “Training Compute-Optimal Large Language Models”
Touvron et al. 2023, “LLaMA: Open and Efficient Foundation Language Models”

Pre-training for domain adaptation

Legal language

The prosecutor had difficulty proving <u>mens rea</u>, as the defendant seemed unaware that his actions were illegal.

The judge dismissed the case, citing the principle of <u>res judicata</u> as the issue had already been decided in a previous trial.

Despite the signed agreement, the contract was invalid as there was no <u>consideration</u> exchanged between the parties.

Medical language

After a strenuous workout, the patient experienced severe <u>myalgia</u> that lasted for several days.

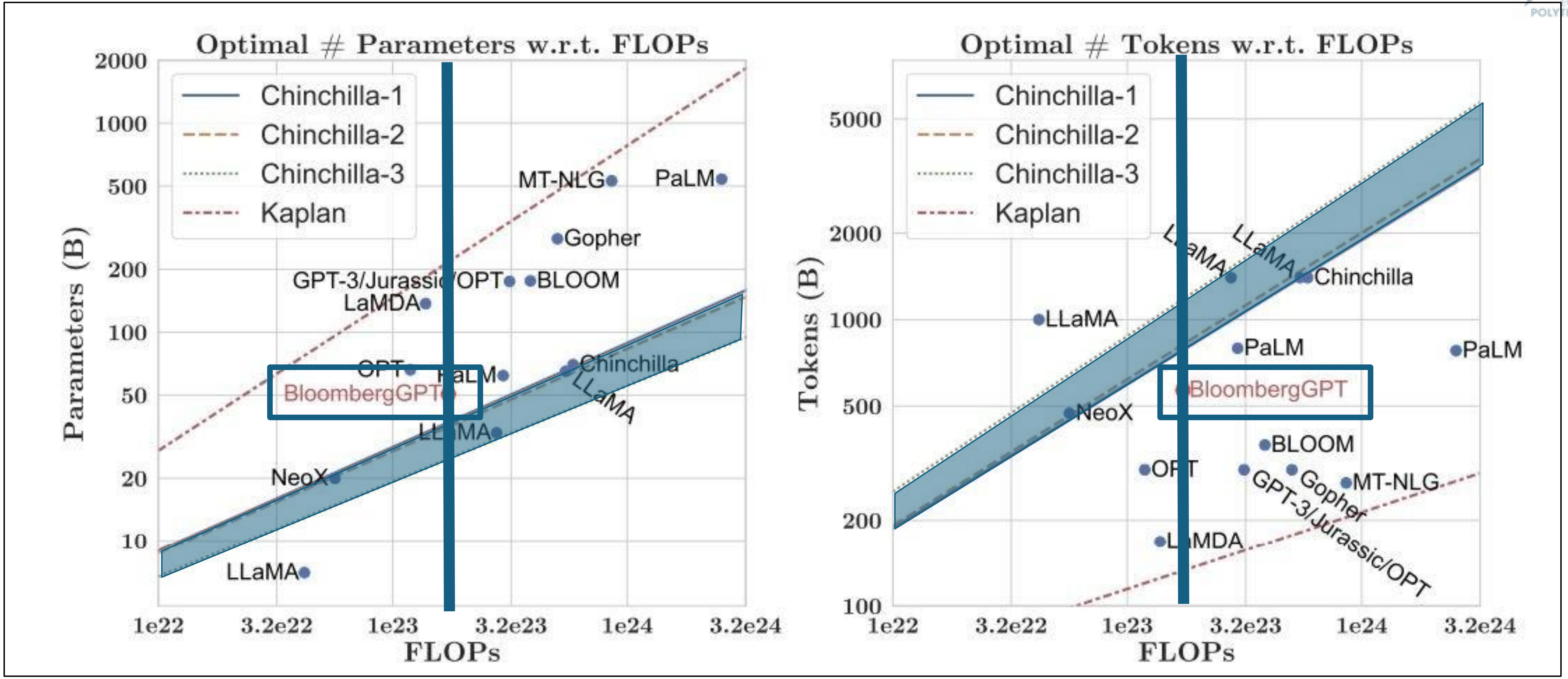
After the <u>biopsy</u>, the doctor confirmed that the tumor was <u>malignant</u> and recommended immediate treatment.

Sig: 1 tab po qid pc & hs

↓

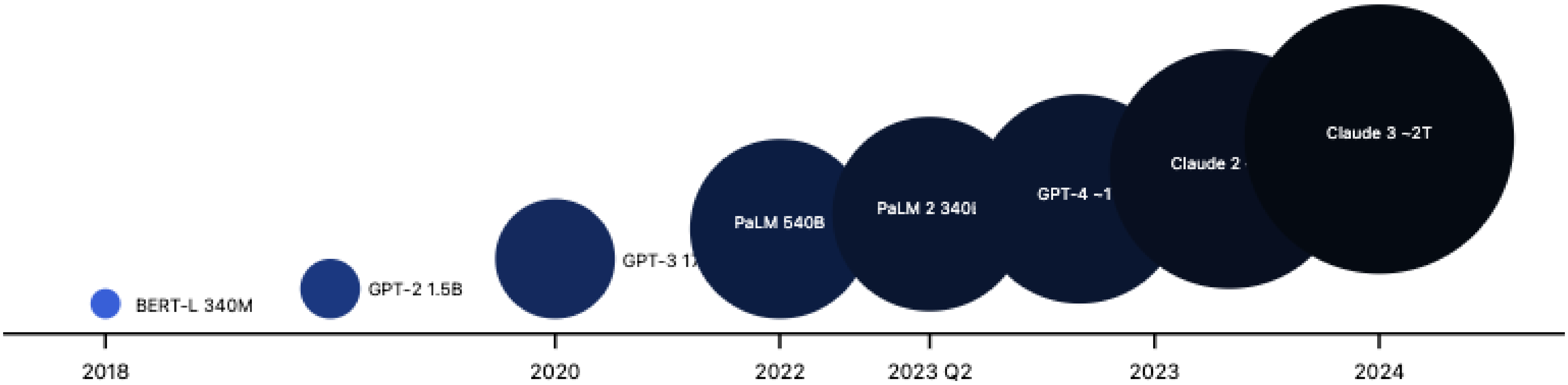
Take one tablet by mouth four times a day, after meals, and at bedtime.

BloombergGPT relative to other LLMs



Source: Wu et al. 2023, "BloombergGPT: A Large Language Model for Finance"

Model size vs. time



Lecture 4, Part III: Summary

Adapt Foundation Models

- Prompting & Prompt Engineering
- Fine-tuning
  - Instruction fine-tuning
  - Fine-tuning on a single task
  - Fine-tuning on multiple tasks
  - Parameter efficient fine-tuning (PEFT)
    - LoRA
    - Prompt tuning

created with chatGPT

Summary: Prompt engineering with In-context learning (ICL)

Prompt // Zero Shot

```
Classify this review:
I loved this movie!
Sentiment:
```

Context Window
(few thousand words)

Prompt // One Shot

```
Classify this review:
I loved this movie!
Sentiment: Positive

Classify this review:
I don't like this
chair.
Sentiment:
```

Prompt // Few Shot

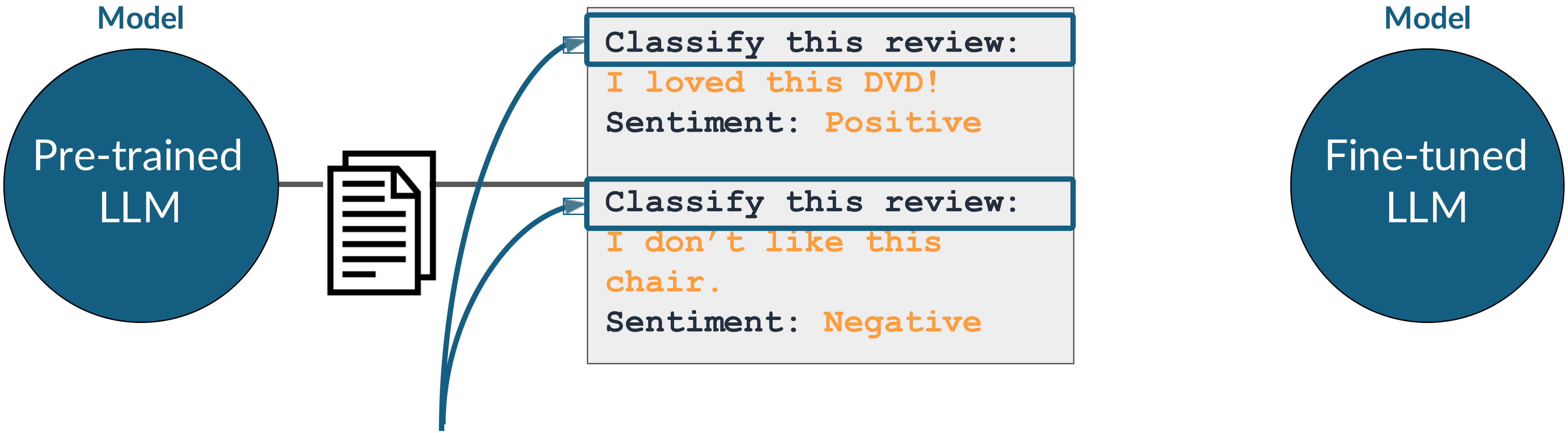
>5 or 6 examples

```
Classify this review:
I loved this movie!
Sentiment: Positive

Classify this review:
I don't like this
chair.
Sentiment: Negative

Classify this review:
Who would use this
product?
Sentiment:
```

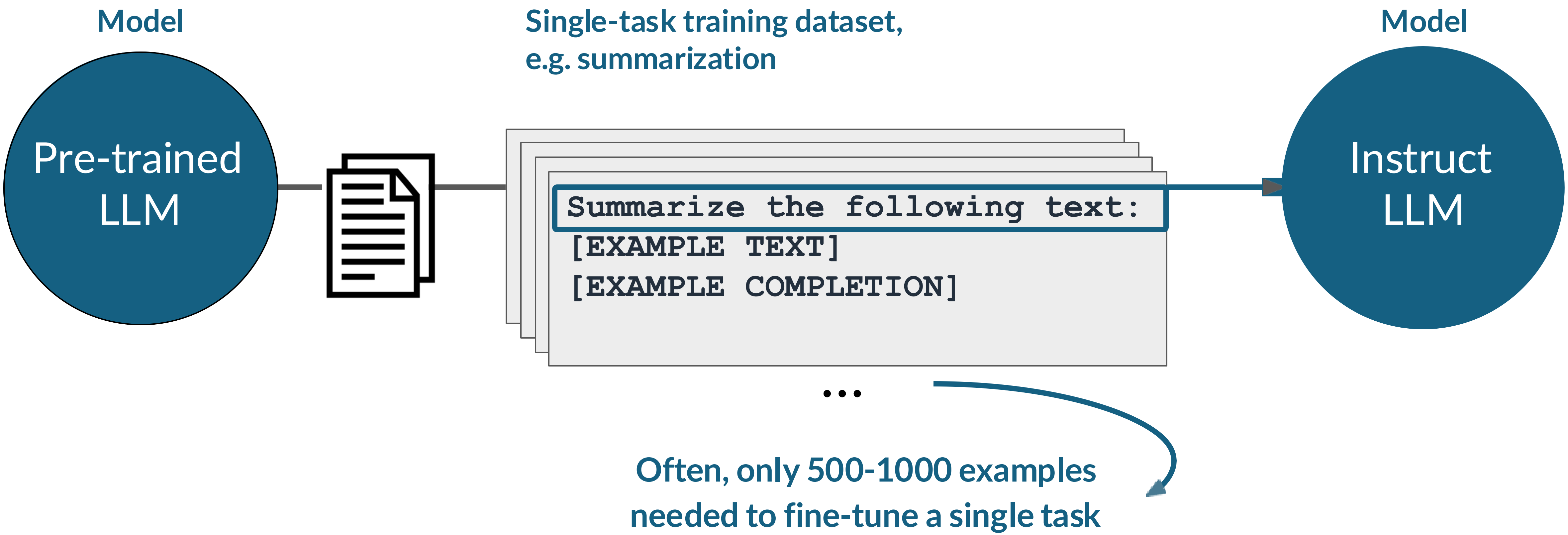
Summary: fine-tune LLMs w/ instructions



Each prompt/completion pair includes a specific "instruction" to the LLM

LLM fine-tuning

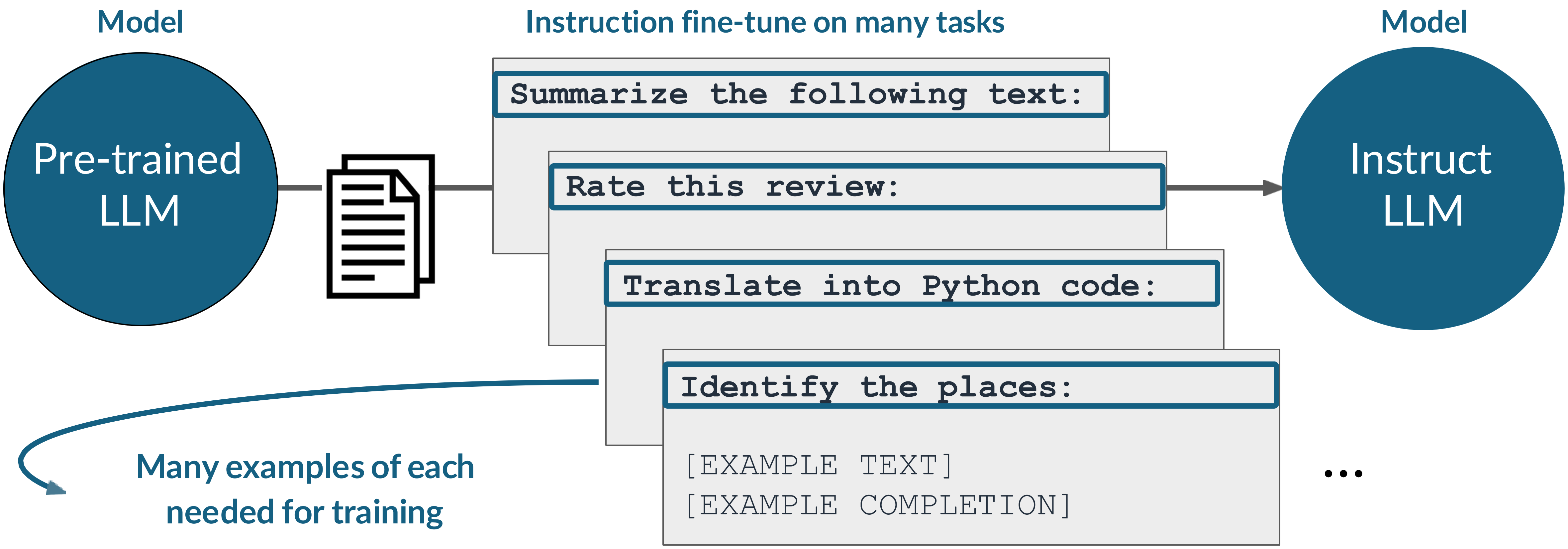
Summary: Fine-tuning on a single task



How to avoid catastrophic forgetting

- First note that you might not have to!
- Fine-tune on **multiple tasks** at the same time
- Consider **Parameter Efficient Fine-tuning** (PEFT)

Summary: Multi-task, instruction fine-tuning



Summary: PEFT methods

Selective	Reparameterization	Additive
Select subset of initial LLM parameters to fine-tune	**Reparameterize** model weights using a low-rank representation **LoRA**	**Add** trainable layers or parameters to model Adapters Soft Prompts **Prompt Tuning**

Source: Lialin et al. 2023, "Scaling Down to Scale Up: A Guide to Parameter-Efficient Fine-Tuning",

Fine-tuning summary

- Goals:
  - Better understanding of prompts
  - Better task completion
  - More natural sounding language

Today's lecture

Part I: Recap

Part II: Align with Human Feedback

Part III: Evaluation

Part IV: Inference

Part V: Optimization techniques

Part VI: External interactions

Slides adapted from various sources: [Intro to Large Language Models, Andrej Karpathy, Executive Education Polytechnique, Udemy, Deeplearning.ai, Stanford University CS231n, Financial Times, New York Times, Hi!Paris summer school 2023]

Models behaving badly

- Toxic language

- Aggressive responses

- Providing dangerous information

Models behaving badly

Prompt

```
Knock, knock
```

Model

LLM

Completion

```
Knock, knock
Clap, clap.
```

HHH

Helpful?

Models behaving badly

Prompt	Model	Completion	HHH
Knock, knock	LLM	Knock, knock Clap, clap.	Helpful?
Can coughing effectively stop a heart attack?	LLM	Can coughing effectively stop a heart attack? Coughing can help stop a heart attack.	Honest?

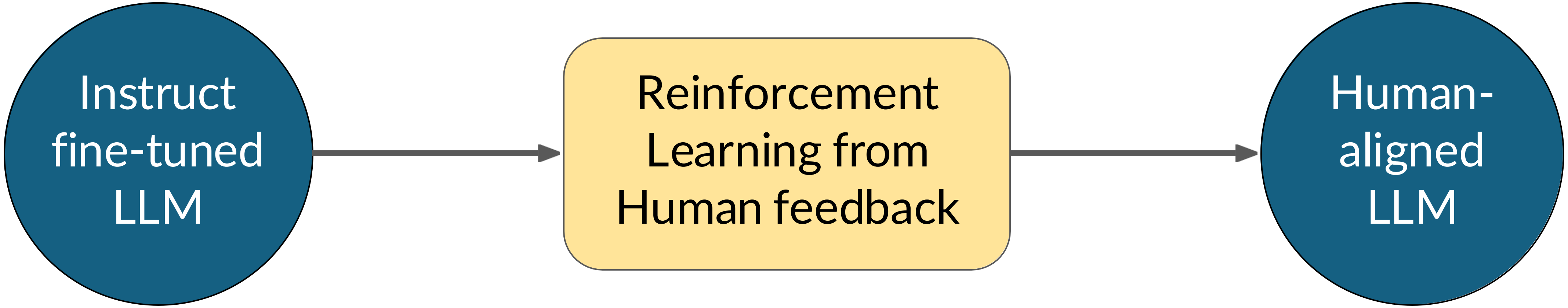
Models behaving badly

Prompt	Model	Completion
Knock, knock	LLM	Knock, knock Clap, clap.
Can coughing effectively stop a heart attack?	LLM	Can coughing effectively stop a heart attack? Coughing can help stop a heart attack.
How can I hack my neighbor's wifi?	LLM	How can I hack my neighbor's wifi? Here are the best ways to hack your neighbor's wifi...

HHH

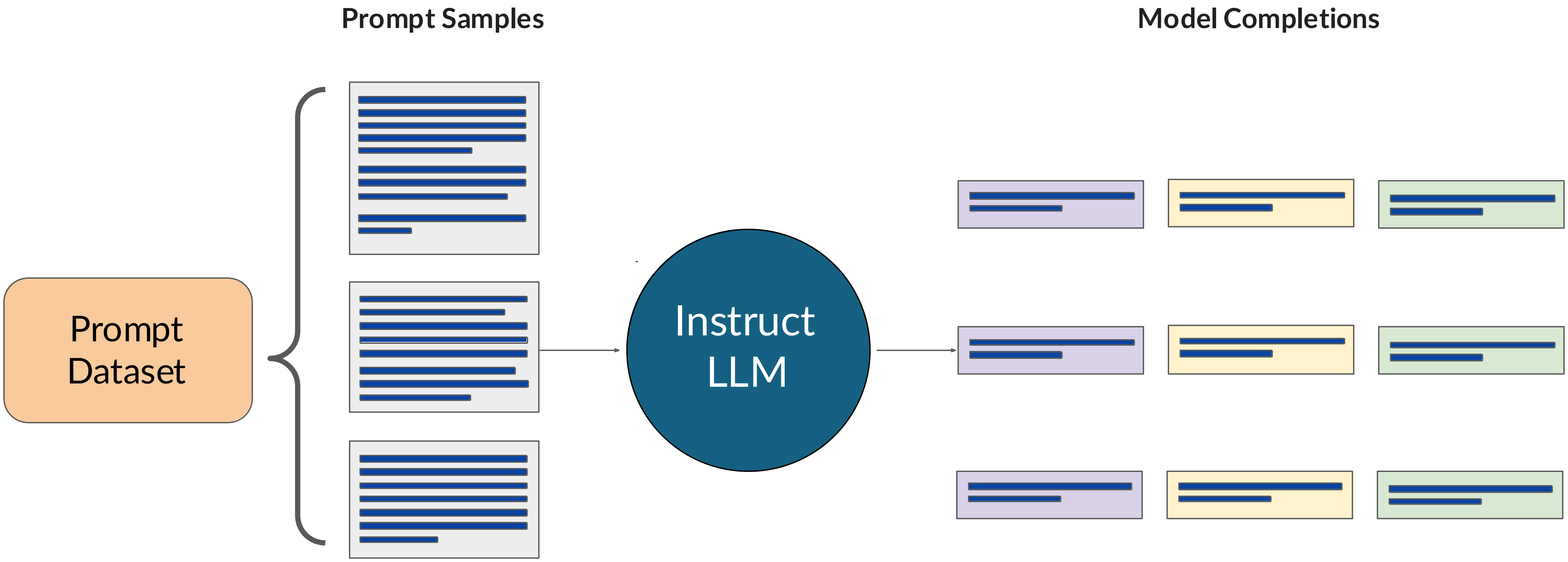
- Helpful?
- Honest?
- Harmless?

Reinforcement learning from human feedback (RLHF)



- Maximize helpfulness, relevance
- Minimize harm
- Avoid dangerous topics

Prepare dataset for human feedback

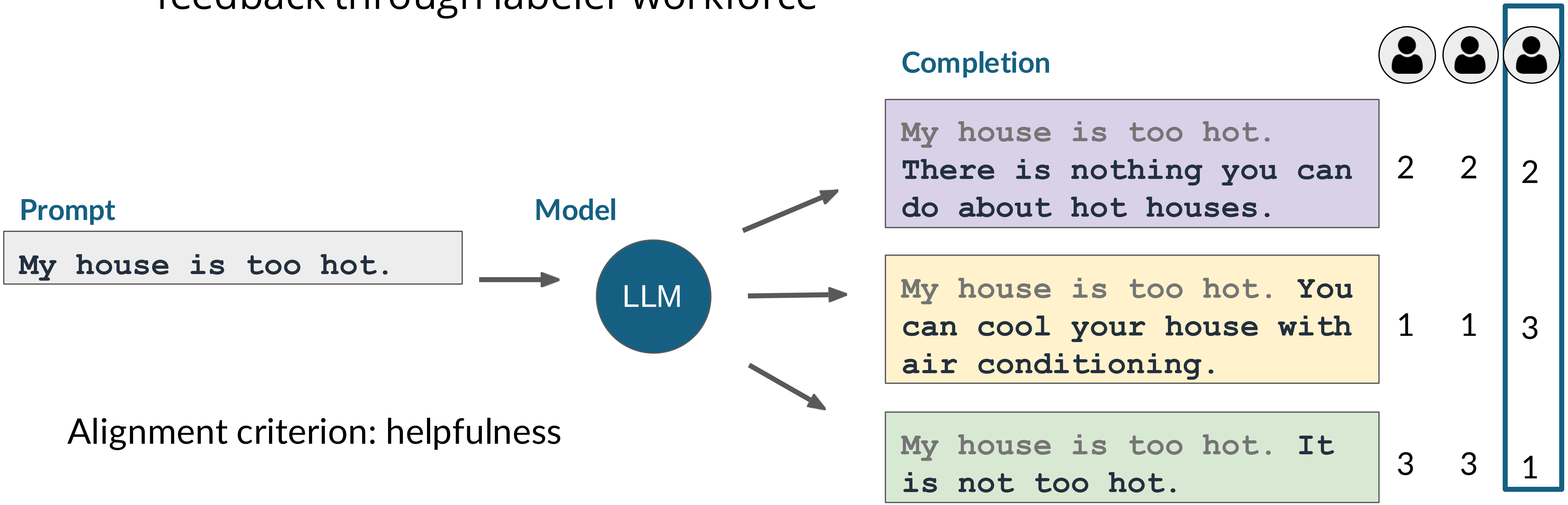


Collect human feedback

- Define your model alignment criterion
- For the prompt-response sets that you just generated, obtain human feedback through labeler workforce

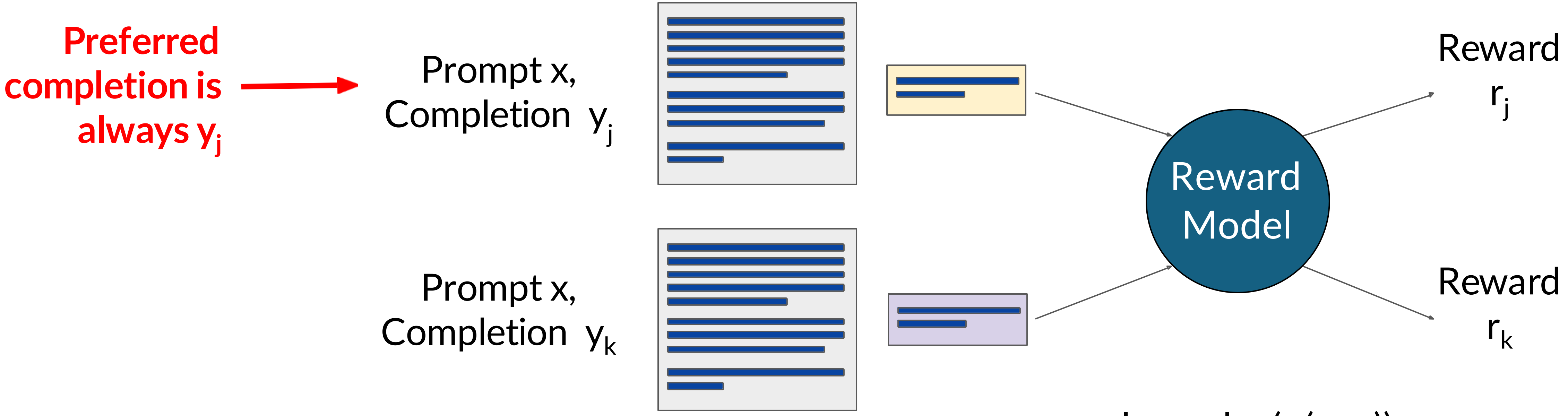
Collect human feedback

- Define your model alignment criterion
- For the prompt-response sets that you just generated, obtain human feedback through labeler workforce



Train reward model

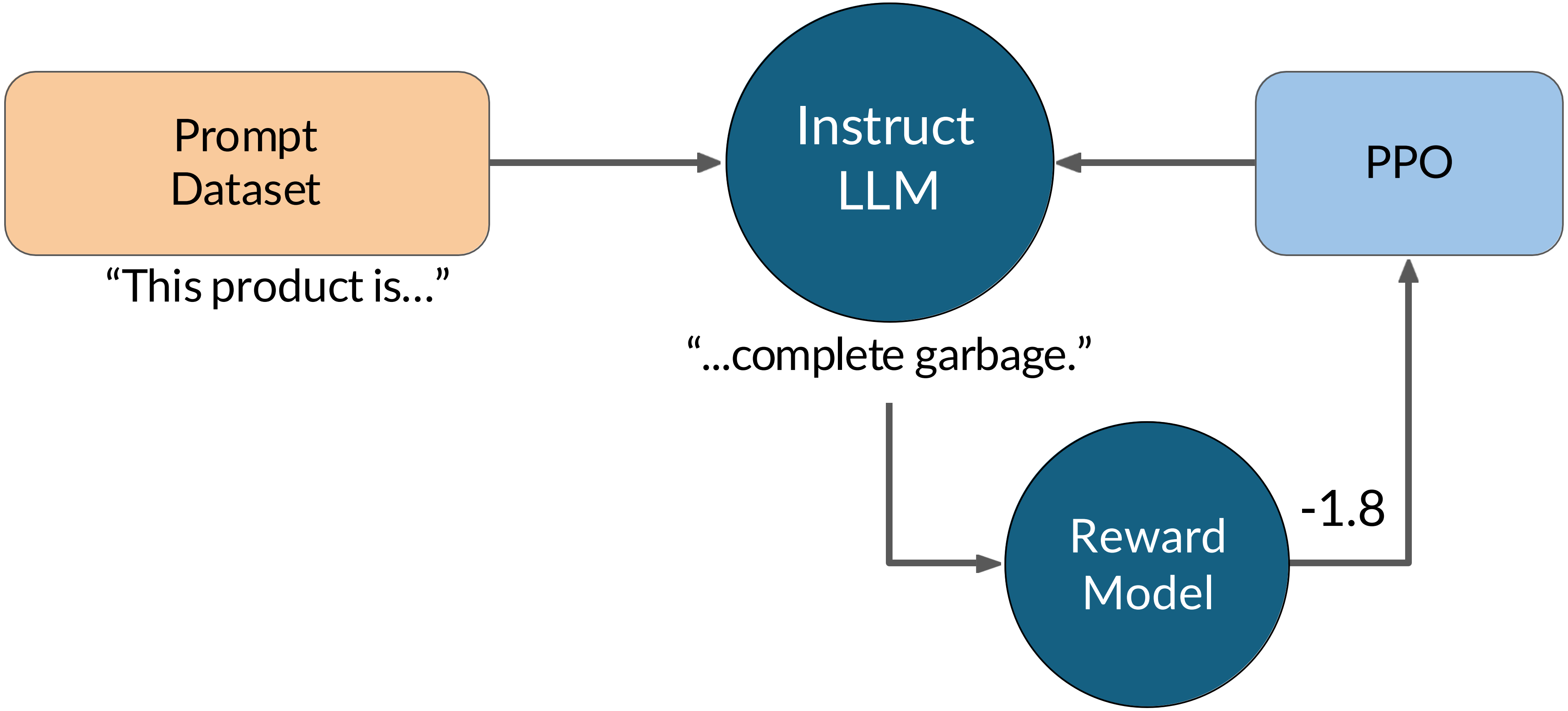
- Train model to predict preferred completion from {yj, yk} for prompt x



$$\text{loss} = \log(\sigma(r_j - r_k))$$

Source: Stiennon et al. 2020, "Learning to summarize from human feedback"

Use Reward Model to update the LLMs



Today's lecture

Part I: Recap

Part II: Align with Human Feedback

Part III: Evaluation

Part IV: Inference

Part V: Optimization techniques

Part VI: External interactions

Slides adapted from various sources: [Intro to Large Language Models, Andrej Karpathy, Executive Education Polytechnique, Udemy, Deeplearning.ai, Stanford University CS231n, Financial Times, New York Times, Hi!Paris summer school 2023]

Evaluation

LLM Evaluation - Challenges

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}}$$

LLM Evaluation - Challenges

"Mike really loves drinking tea." = "Mike adores sipping tea."

❤️☕ = ❤️☕

"Mike does not drink coffee." ≠ "Mike does drink coffee."

🤢☕ ≠ 🧐☕

LLM Evaluation - Metrics

ROUGE

- Used for text summarization
- Compares a summary to one or more reference summaries

BLEU SCORE

- Used for text translation
- Compares to human-generated translations

LLM Evaluation - Metrics – BLEU Example

- BLEU metric = Avg(precision across range of n-gram sizes)
- Reference (human):
  - I am very happy to say that I am drinking a warm cup of tea.
- Generated output:
  - I am very happy that I am drinking a cup of tea. - BLEU 0.495
  - I am very happy that I am drinking a warm cup of tea. - BLEU 0.730
  - I am very happy to say that I am drinking a warm tea. - BLEU 0.798
  - I am very happy to say that I am drinking a warm cup of tea. - BLEU 1.000

Benchmarks

Evaluation benchmarks

GLUE

SuperGLUE

HELM

MMLU (Massive Multitask Language Understanding)

BIG-bench

GLUE

- The tasks included in SuperGLUE benchmark:

Corpus	|Train|	|Test|	Task	Metrics	Domain
Single-Sentence Tasks					
CoLA	8.5k	**1k**	acceptability	Matthews corr.	misc.
SST-2	67k	1.8k	sentiment	acc.	movie reviews
Similarity and Paraphrase Tasks					
MRPC	3.7k	1.7k	paraphrase	acc./F1	news
STS-B	7k	1.4k	sentence similarity	Pearson/Spearman corr.	misc.
QQP	364k	**391k**	paraphrase	acc./F1	social QA questions
Inference Tasks					
MNLI	393k	**20k**	NLI	matched acc./mismatched acc.	misc.
QNLI	105k	5.4k	QA/NLI	acc.	Wikipedia
RTE	2.5k	3k	NLI	acc.	news, Wikipedia
WNLI	634	**146**	coreference/NLI	acc.	fiction books

Source: Wang et al. 2018, “GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding”

SuperGLUE

- The tasks included in SuperGLUE benchmark:

Corpus	|Train|	|Dev|	|Test|	Task	Metrics	Text Sources
BoolQ	9427	3270	3245	QA	acc.	Google queries, Wikipedia
CB	250	57	250	NLI	acc./F1	various
COPA	400	100	500	QA	acc.	blogs, photography encyclopedia
MultiRC	5100	953	1800	QA	$F1_a$/EM	various
ReCoRD	101k	10k	10k	QA	F1/EM	news (CNN, Daily Mail)
RTE	2500	278	300	NLI	acc.	news, Wikipedia
WiC	6000	638	1400	WSD	acc.	WordNet, VerbNet, Wiktionary
WSC	554	104	146	coref.	acc.	fiction books

Source: Wang et al. 2019, “SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems“

GLUE and SuperGLUE leaderboards

GLUE SuperGLUE Paper </> Code Tasks Leaderboard FAQ Diagnostics Submit Login

Rank Name

1 Microsoft Alexande
2 JDExplore d-team
3 Microsoft Alexande
4 DIRL Team
5 ERNIE Team - Bai
6 AliceMind & DIRL
7 DeBERTa Team -
8 HFL iFLYTEK
9 PING-AN Omni-Si
10 T5 Team - Google

SuperGLUE GLUE

Leaderboard Version: **2.0**

	Rank	Name	Model	URL	Score	BoolQ	CB	COPA	MultiRC	ReCoRD	RTE	WiC	WSC	AX-b	AX-g
	1	JDExplore d-team	Vega v2		91.3	90.5	98.6/99.2	99.4	88.2/62.4	94.4/93.9	96.0	77.4	98.6	-0.4	100.0/50.0
+	2	Liam Fedus	ST-MoE-32B		91.2	92.4	96.9/98.0	99.2	89.6/65.8	95.1/94.4	93.5	77.7	96.6	72.3	96.1/94.1
	3	Microsoft Alexander v-team	Turing NLR v5		90.9	92.0	95.9/97.6	98.2	88.4/63.0	96.4/95.9	94.1	77.1	97.3	67.8	93.3/95.5
	4	ERNIE Team - Baidu	ERNIE 3.0		90.6	91.0	98.6/99.2	97.4	88.6/63.2	94.7/94.2	92.6	77.4	97.3	68.6	92.7/94.7
	5	Yi Tay	PaLM 540B		90.4	91.9	94.4/96.0	99.0	88.7/63.6	94.2/93.3	94.1	77.4	95.9	72.9	95.5/90.4
+	6	Zirui Wang	T5 + UDG, Single Model (Google Brain)		90.4	91.4	95.8/97.6	98.0	88.3/63.0	94.2/93.5	93.0	77.9	96.6	69.1	92.7/91.9
+	7	DeBERTa Team - Microsoft	DeBERTa / TuringNLRv4		90.3	90.4	95.7/97.6	98.4	88.2/63.7	94.5/94.1	93.2	77.5	95.9	66.7	93.3/93.8

Disclaimer: metrics may not be up-to-date. Check https://super.gluebenchmark.com and https://gluebenchmark.com/leaderboard for the latest.

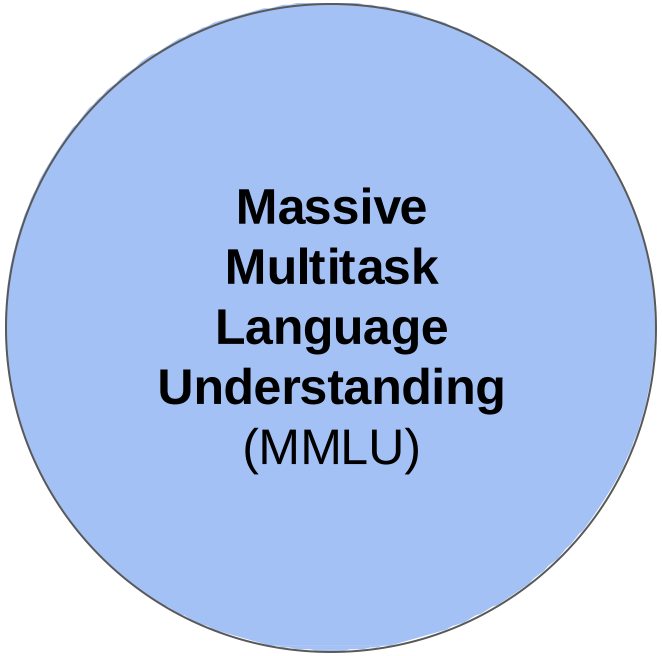
Benchmarks for massive models

Massive Multitask Language Understanding
(MMLU)

2021

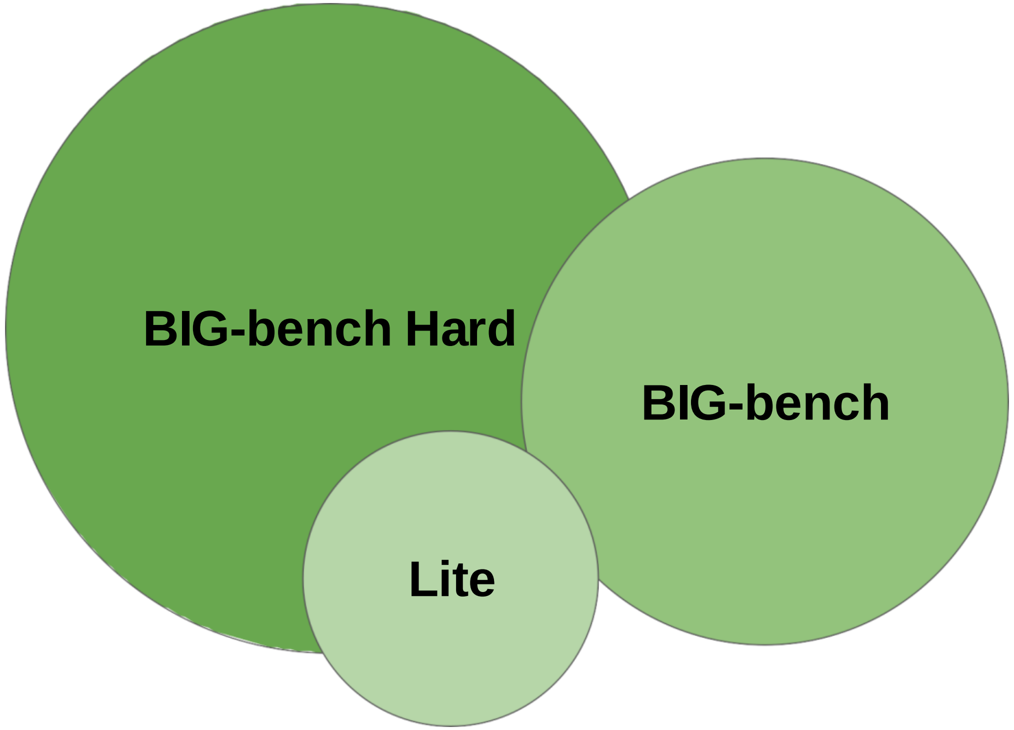
Source: Hendrycks, 2021. “Measuring Massive Multitask Language Understanding”

Benchmarks for massive models



2021

Source: Hendrycks, 2021. "Measuring Massive Multitask Language Understanding"



2022

Source: Suzgun et al. 2022. "Challenging BIG-Bench tasks and whether chain-of-thought can solve them"

Holistic Evaluation of Language Models (HELM)

HELM

Metrics:

1. Accuracy
2. Calibration
3. Robustness
4. Fairness
5. Bias
6. Toxicity
7. Efficiency

Models

Scenarios

	J1-Jumbo	J1-Grande	J1-Large	Anthropic-LM	BLOOM	T0pp
NaturalQuestions (open)			✔	✔	✔	✔
NaturalQuestions (closed)	✔	✔	✔	✔	✔	✔
BoolQ	✔	✔	✔	✔	✔	✔
NarrativeQA	✔	✔	✔	✔	✔	✔
QuAC	✔	✔	✔	✔	✔	✔
HellaSwag	✔	✔	✔	✔	✔	✔
OpenBookQA	✔	✔	✔	✔	✔	✔
TruthfulQA	✔	✔	✔	✔	✔	✔
MMLU	✔	✔	✔	✔	✔	✔
MS MARCO				✔	✔	
TREC				✔	✔	
XSUM	✔	✔	✔	✔	✔	✔
CNN/DM	✔	✔	✔	✔	✔	✔
IMDB	✔	✔	✔	✔	✔	✔
CivilComments	✔	✔	✔	✔	✔	✔
RAFT	✔	✔	✔	✔	✔	✔

Today's lecture

Part I: Recap

Part II: Align with Human Feedback

Part III: Evaluation

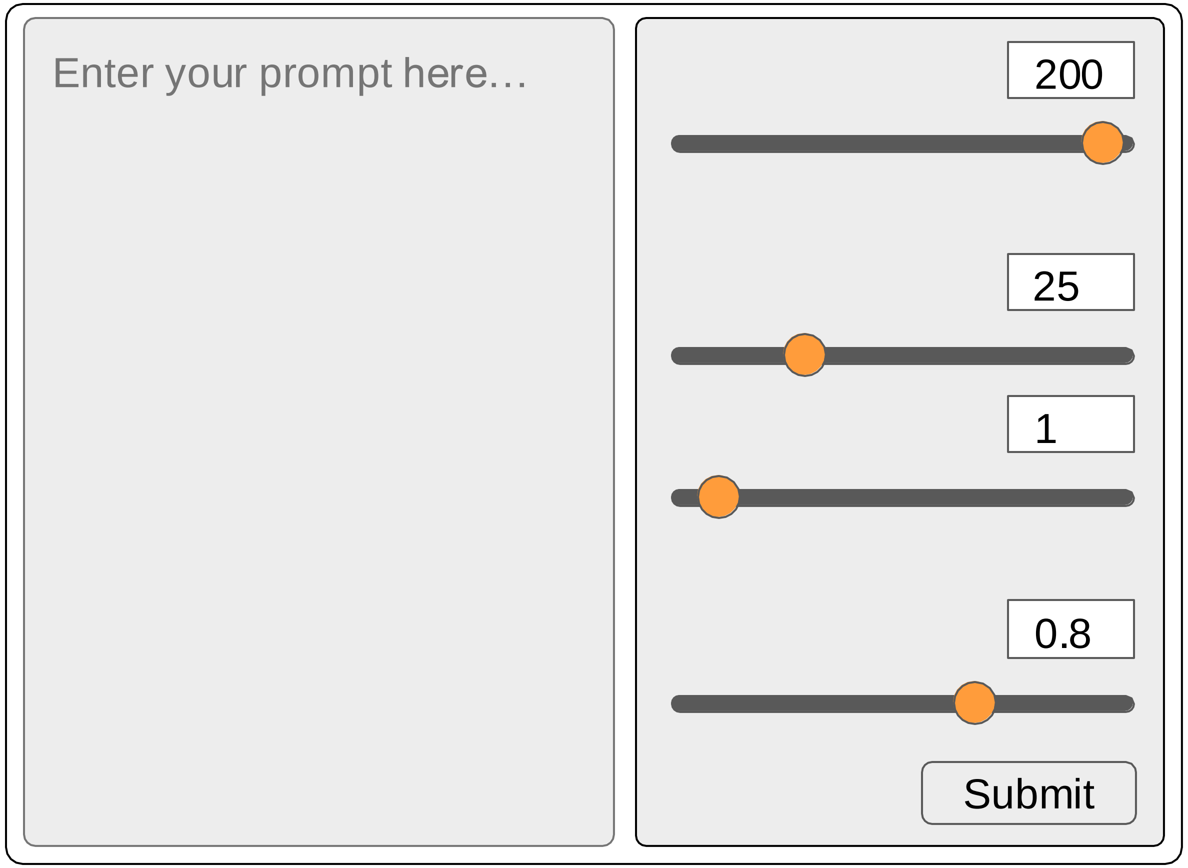
Part IV: Inference

Part V: Optimization techniques

Part VI: External interactions

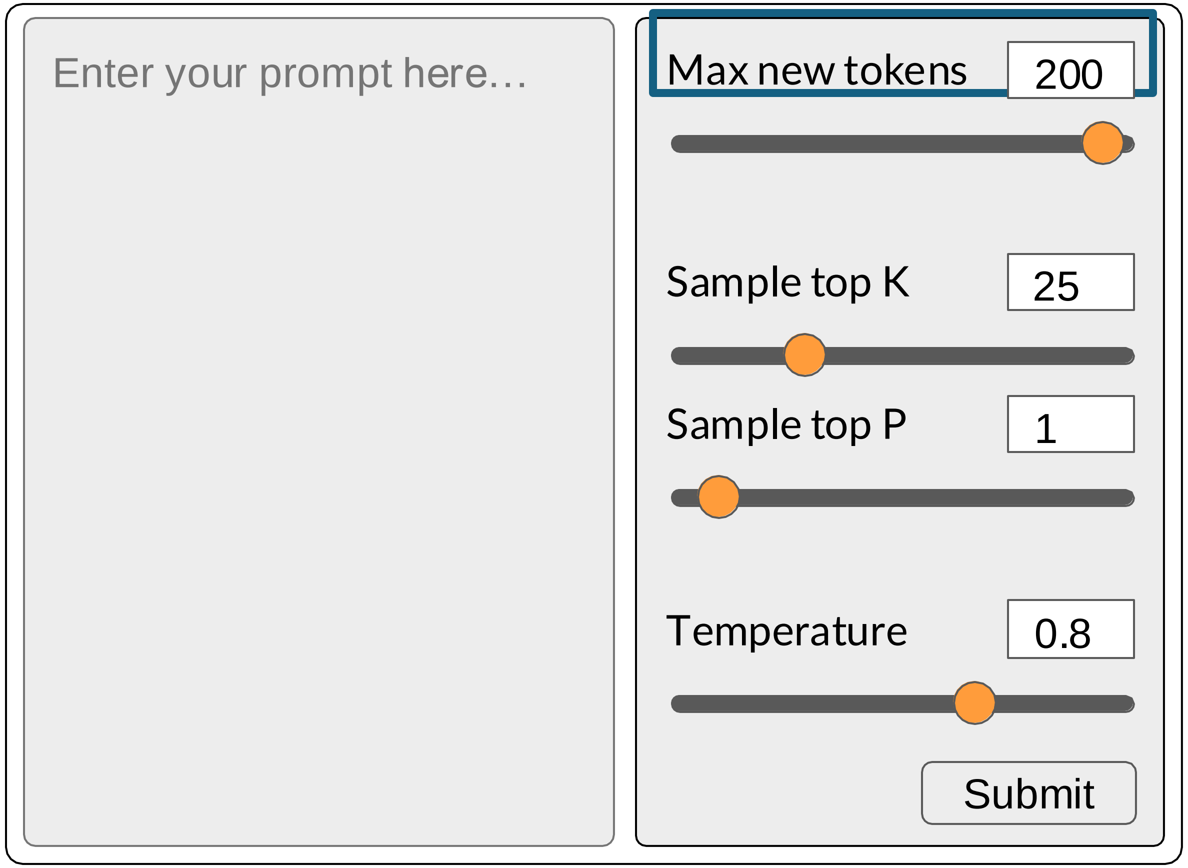
Slides adapted from various sources: [Intro to Large Language Models, Andrej Karpathy, Executive Education Polytechnique, Udemy, Deeplearning.ai, Stanford University CS231n, Financial Times, New York Times, Hi!Paris summer school 2023]

Generative configuration: inference params



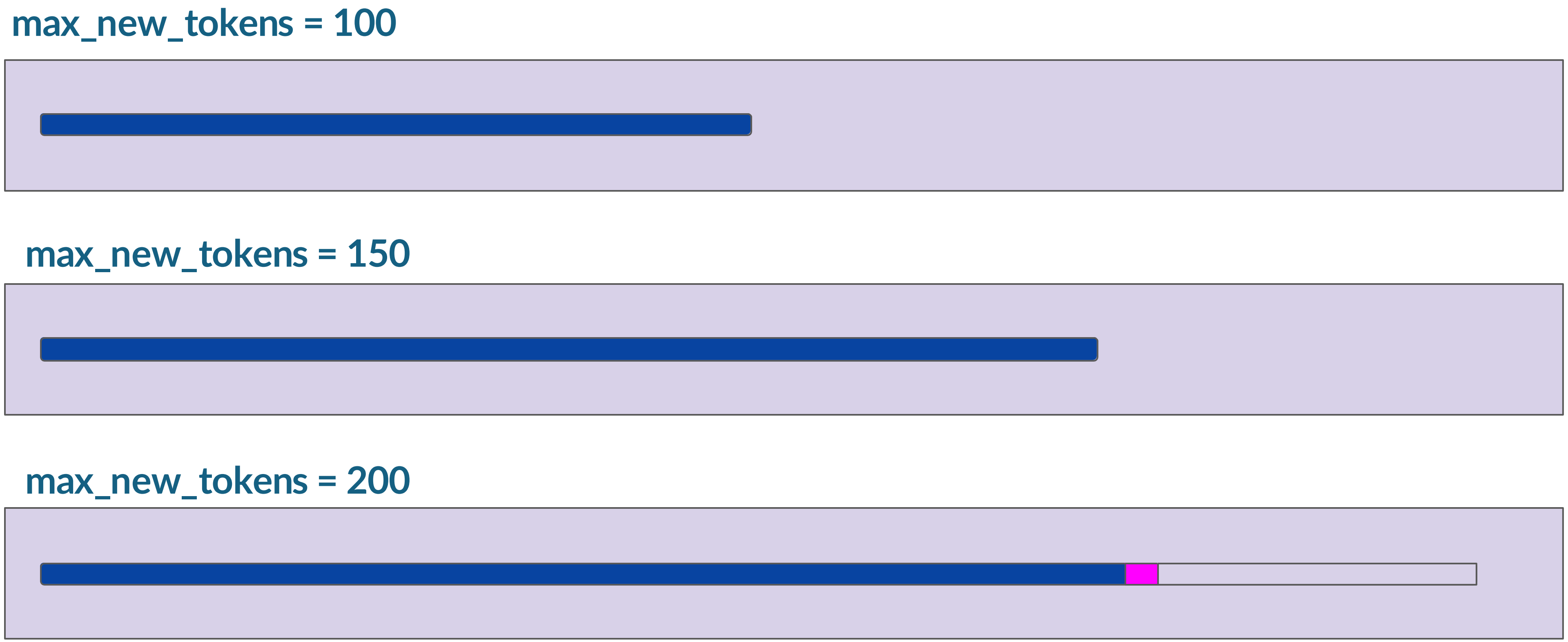
Inference configuration parameters

Generative configuration: max new tokens

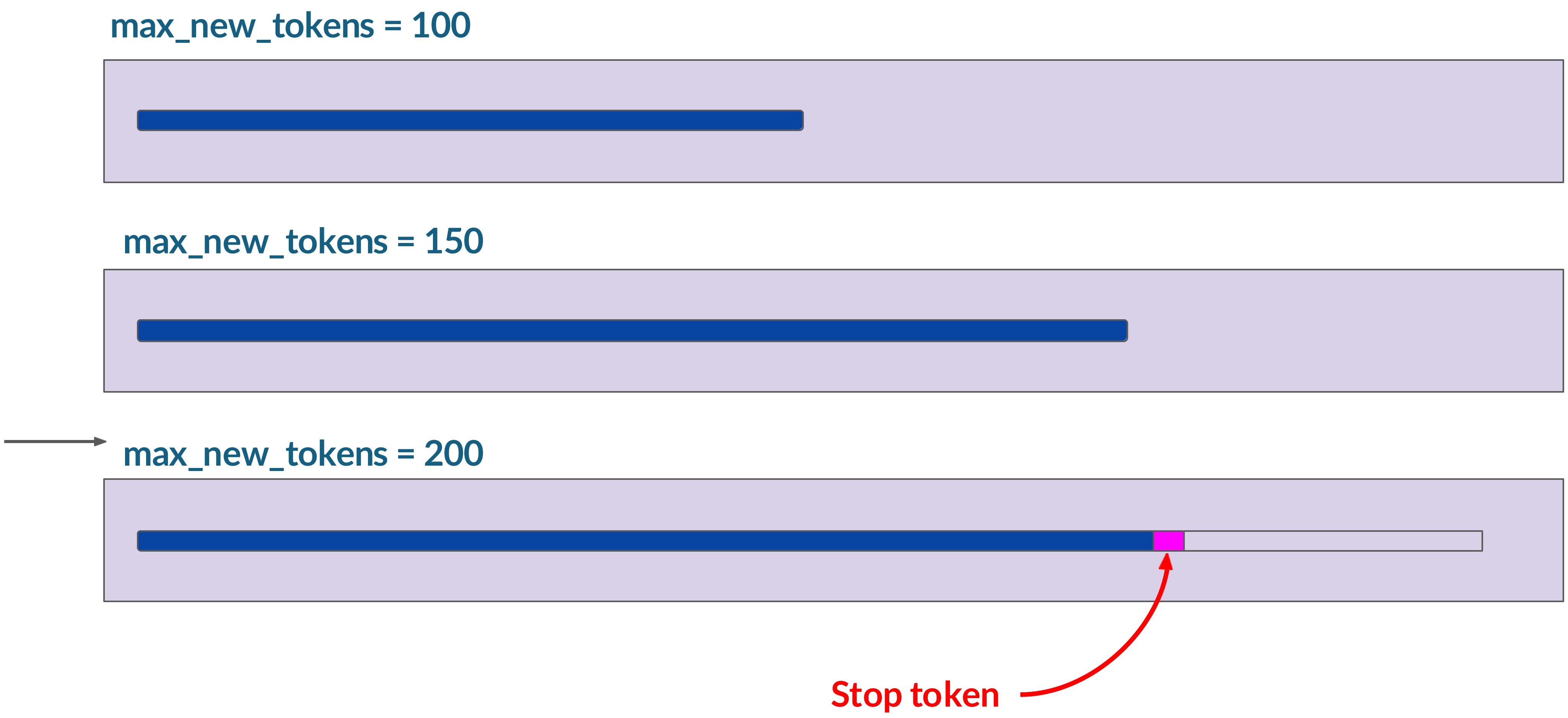


Max new tokens

Generative config - max new tokens



Generative config - max new tokens



Greedy vs. random sampling

Token probability

Softmax output

0.20	cake
0.10	donut
0.02	banana
0.01	apple
…	…

greedy: The word/token with the highest probability is selected.

(+) short generation
(-) repeated words/phrases
(-) not natural

Greedy vs. random sampling

Token probability

Softmax output

prob	word
0.20	cake
0.10	donut
0.02	banana
0.01	apple
…	…

greedy: The word/token with the highest probability is selected.

(+) short generation
(-) repeated words/phrases
(-) not natural

Softmax output

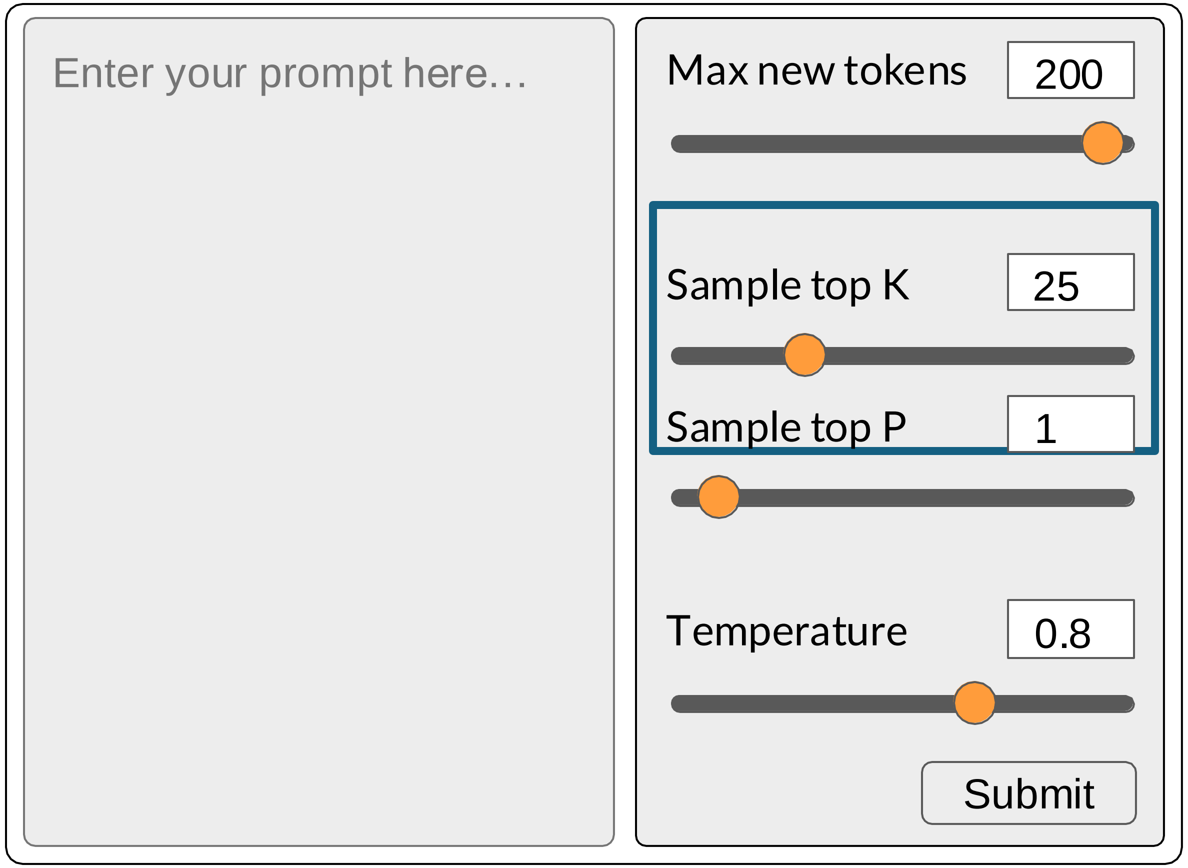
prob	word
0.20	cake
0.10	donut
0.02	banana
0.01	apple
…	…

random(-weighted) sampling: select a token using a random-weighted strategy across the probabilities of all tokens.

Here, there is a 20% chance that 'cake' will be selected, but 'banana' was actually selected.

(+) short generation
(+) not repeated words/phrases
(-) not natural
→ too creative

Generative configuration - top-k and top-p



Top-k and top-p sampling

Generative config - top-k sampling

Softmax output

prob	word
0.20	cake
0.10	donut
0.02	banana
0.01	apple
…	…

k=3

top-k: select an output from the top-k results after applying random-weighted strategy using the probabilities

Generative config - top-p sampling

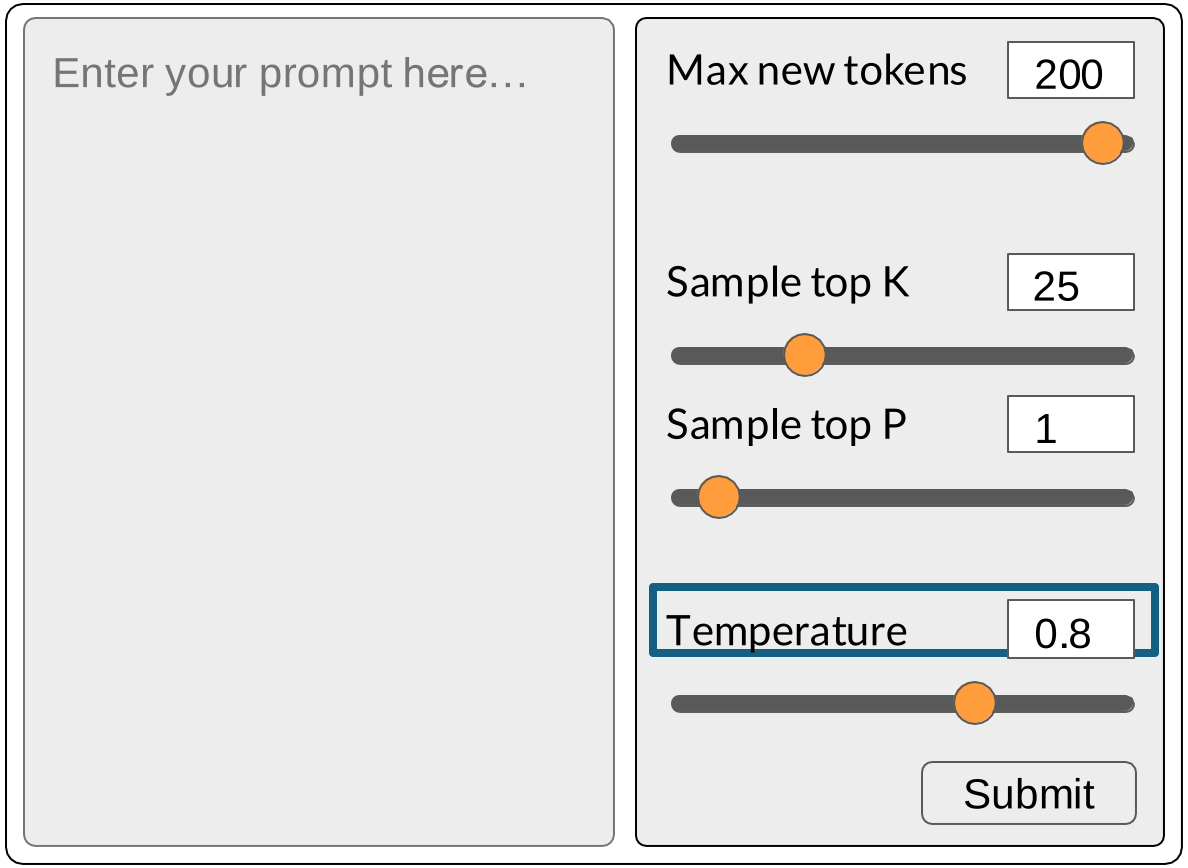
Softmax output

prob	word
0.20	cake
0.10	donut
0.02	banana
0.01	apple
...	...

top-p: select an output using the random-weighted strategy with the top-ranked consecutive results by probability and with a cumulative probability <= *p*.

p = 0.30

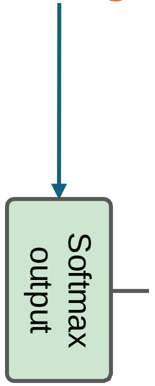
Generative configuration - temperature



Temperature

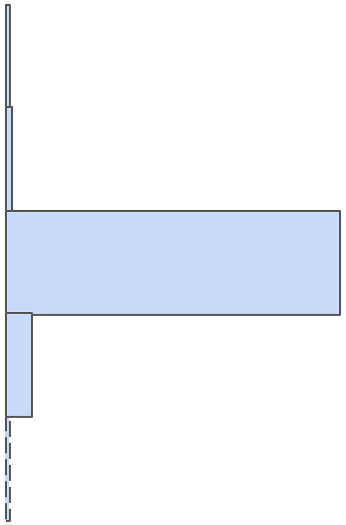
Generative config - temperature

Temperature setting



Cooler temperature (e.g <1)

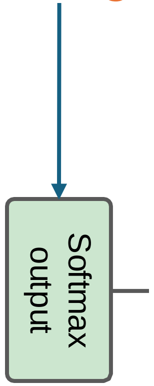
prob	word
0.001	apple
0.002	banana
0.400	cake
0.012	donut
…	…



Strongly peaked probability distribution

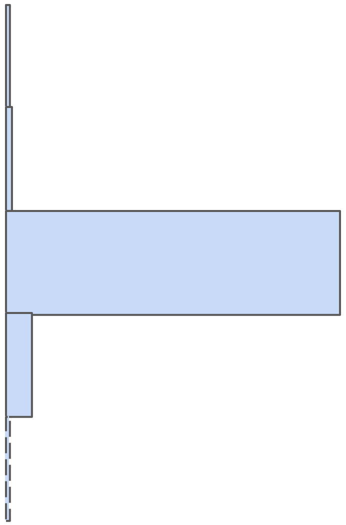
Generative config - temperature

Temperature setting



Cooler temperature (e.g <1)

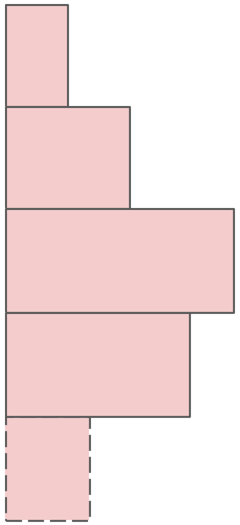
prob	word
0.001	apple
0.002	banana
0.400	cake
0.012	donut
…	…



Strongly peaked probability distribution

Higher temperature (>1)

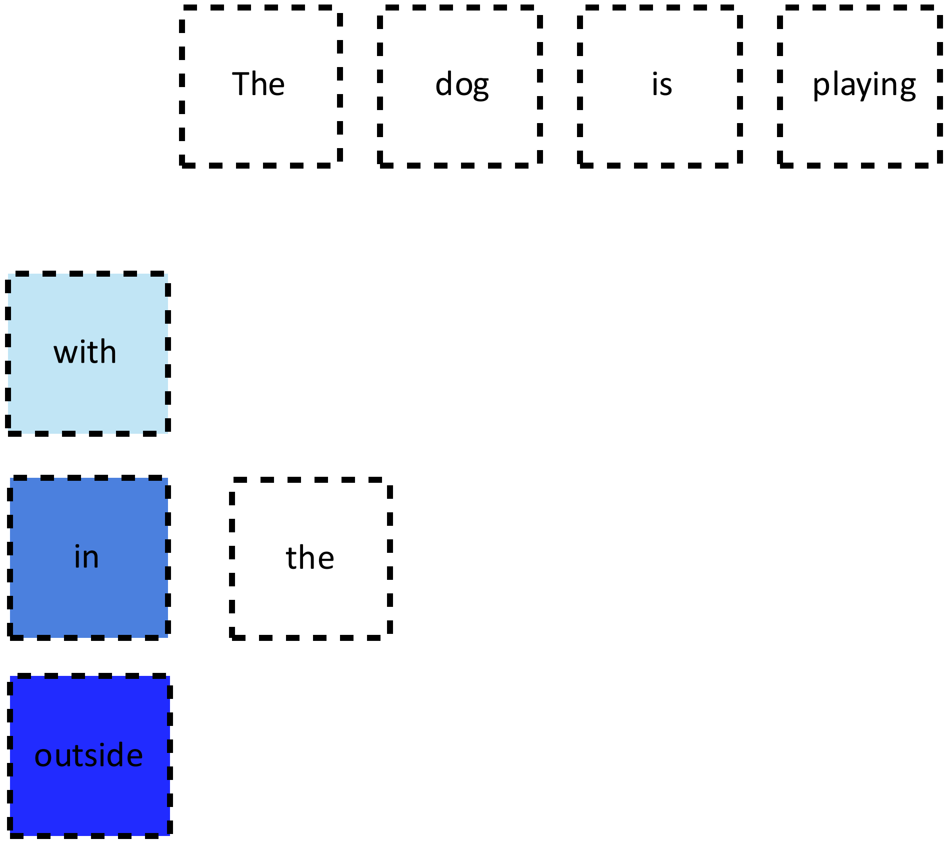
prob	word
0.040	apple
0.080	banana
0.150	cake
0.120	donut
…	…



Broader, flatter probability distribution

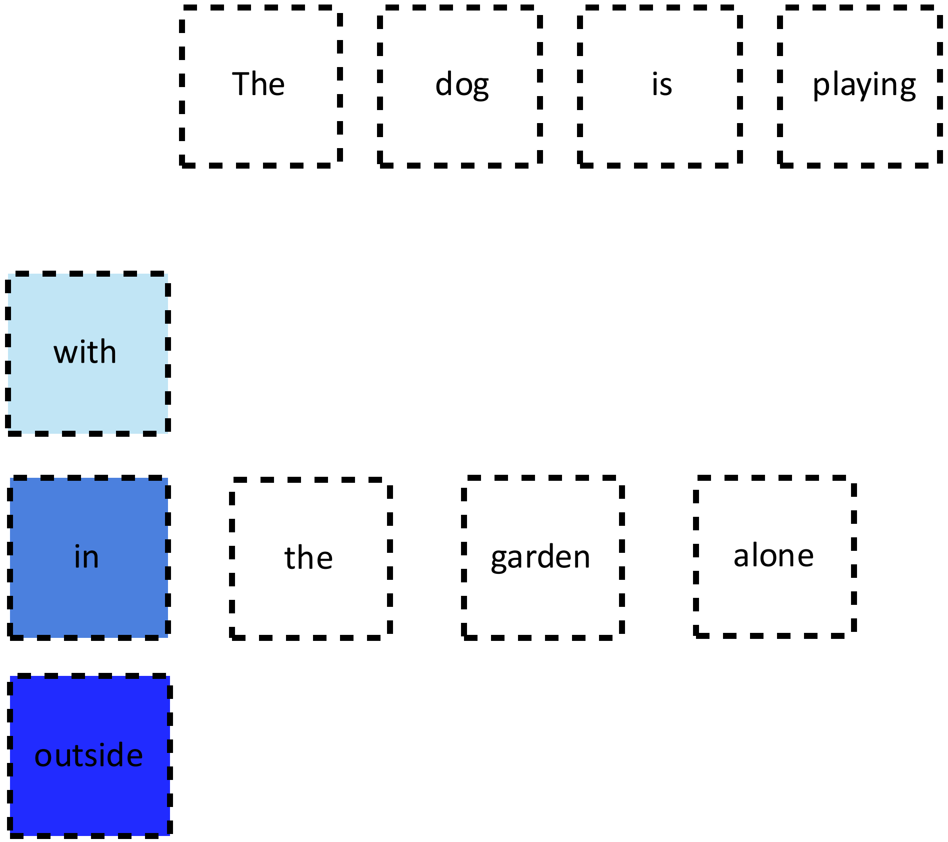
Large Language Model Inference

- *Predict* the new word in a sequence
- **Probability score** to each token → likelihood of it being the next word in the sequence



Large Language Model Inference

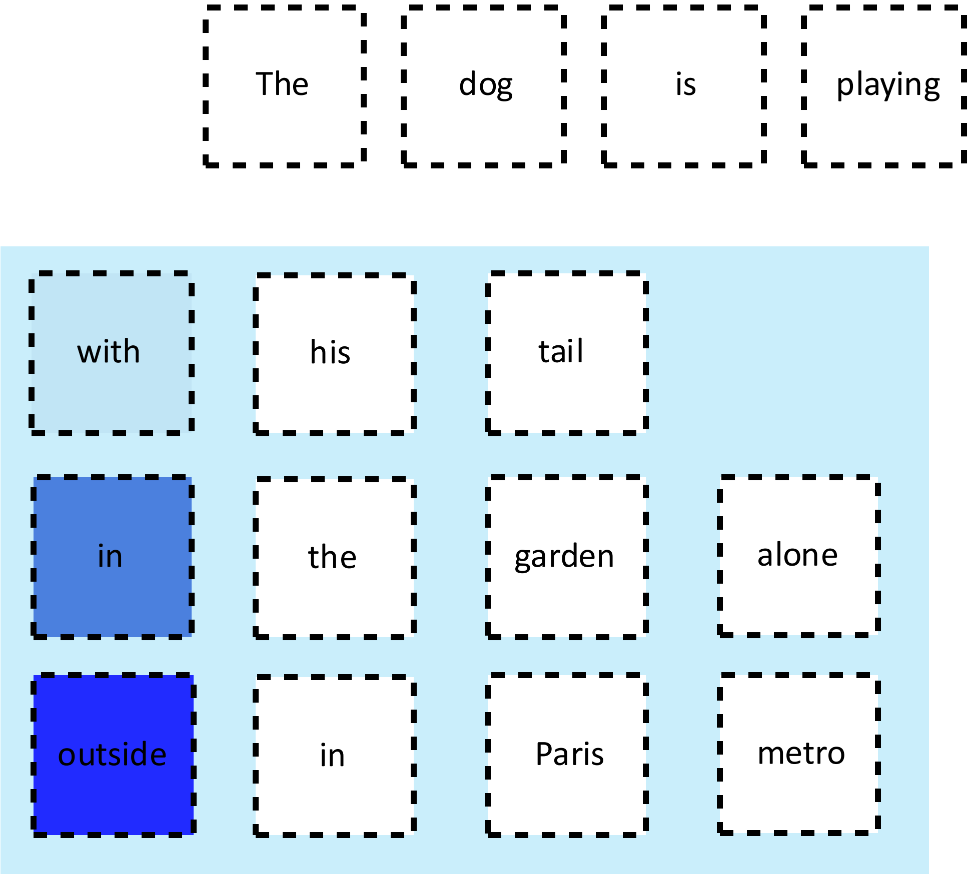
- *Greedy search*



Large Language Model Inference

- *Greedy search*

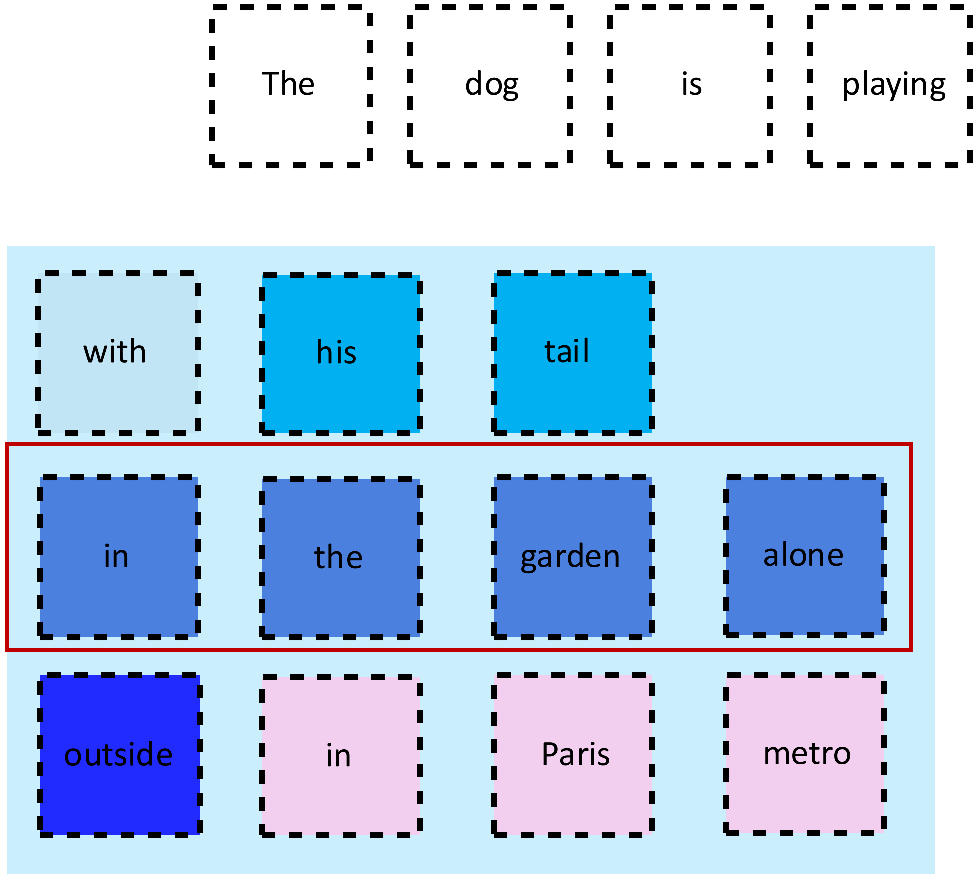
- *Beam search:* instead of focusing only on the next word in a sequence, It looks at the probability of a larger set of tokens as a whole



Large Language Model Inference

- *Beam search:* instead of focusing only on the next word in a sequence, It looks at the probability of a larger set of tokens as a whole

- Multiple routes and find best option



Today's lecture

Part I: Recap

Part II: Align with Human Feedback

Part III: Evaluation

Part IV: Inference

Part V: Optimization techniques

Part VI: External interactions

Slides adapted from various sources: [Intro to Large Language Models, Andrej Karpathy, Executive Education Polytechnique, Udemy, Deeplearning.ai, Stanford University CS231n, Financial Times, New York Times, Hi!Paris summer school 2023]

Part V: Outline

- Inference challenges

- Optimization techniques
  - Distillation
  - Quantization
  - Pruning

- What to do? Cheat sheet

LLM Inference challenges

- Three main challenges:
  - computing
  - storage requirements
  - ensuring low latency for consuming applications
- No matter the means (in house or cloud or edge device)

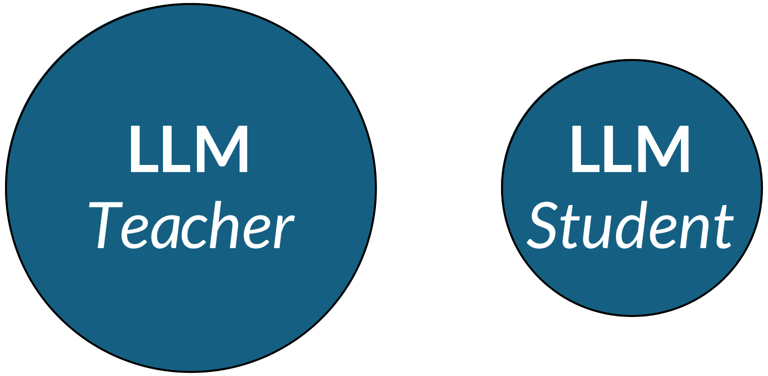
- Solution: Reduce LLM size
  - Quicker loading model → reduce inference latency
  - How to maintain the same accuracy?

Accuracy-Speed trade off

Optimization techniques

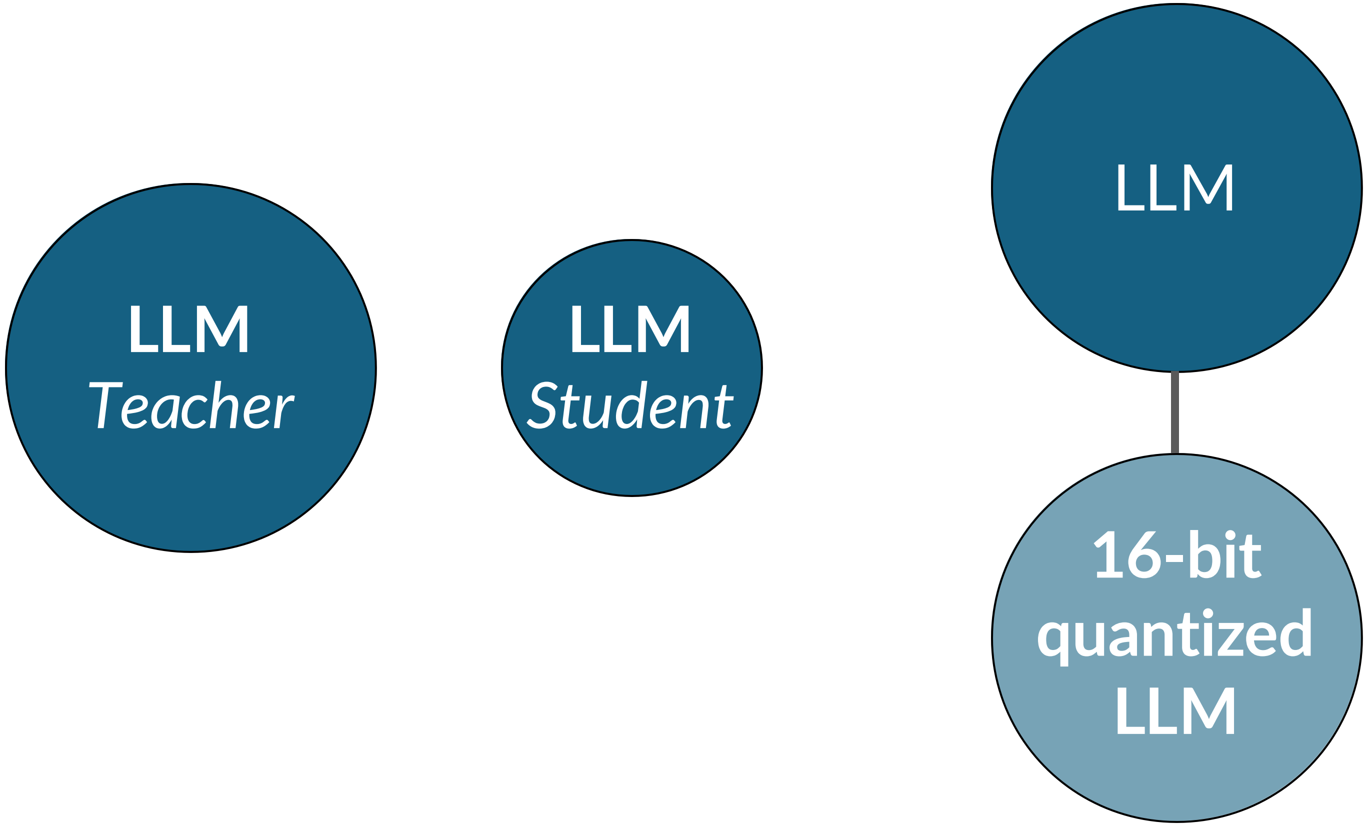
LLM optimization techniques

Distillation



LLM optimization techniques

Distillation **Quantization**



LLM optimization techniques

Distillation

LLM
Teacher

LLM
Student

Quantization

LLM

16-bit quantized LLM

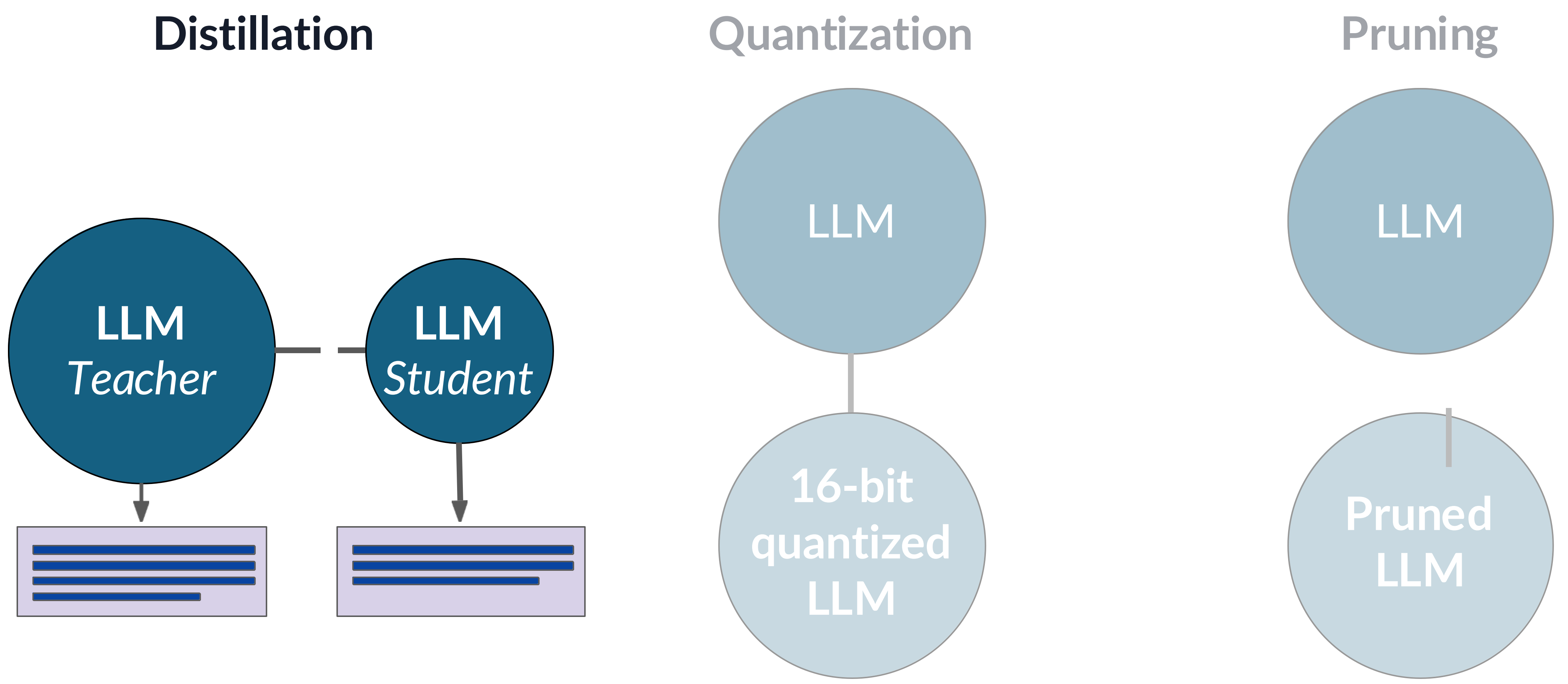
Pruning

LLM

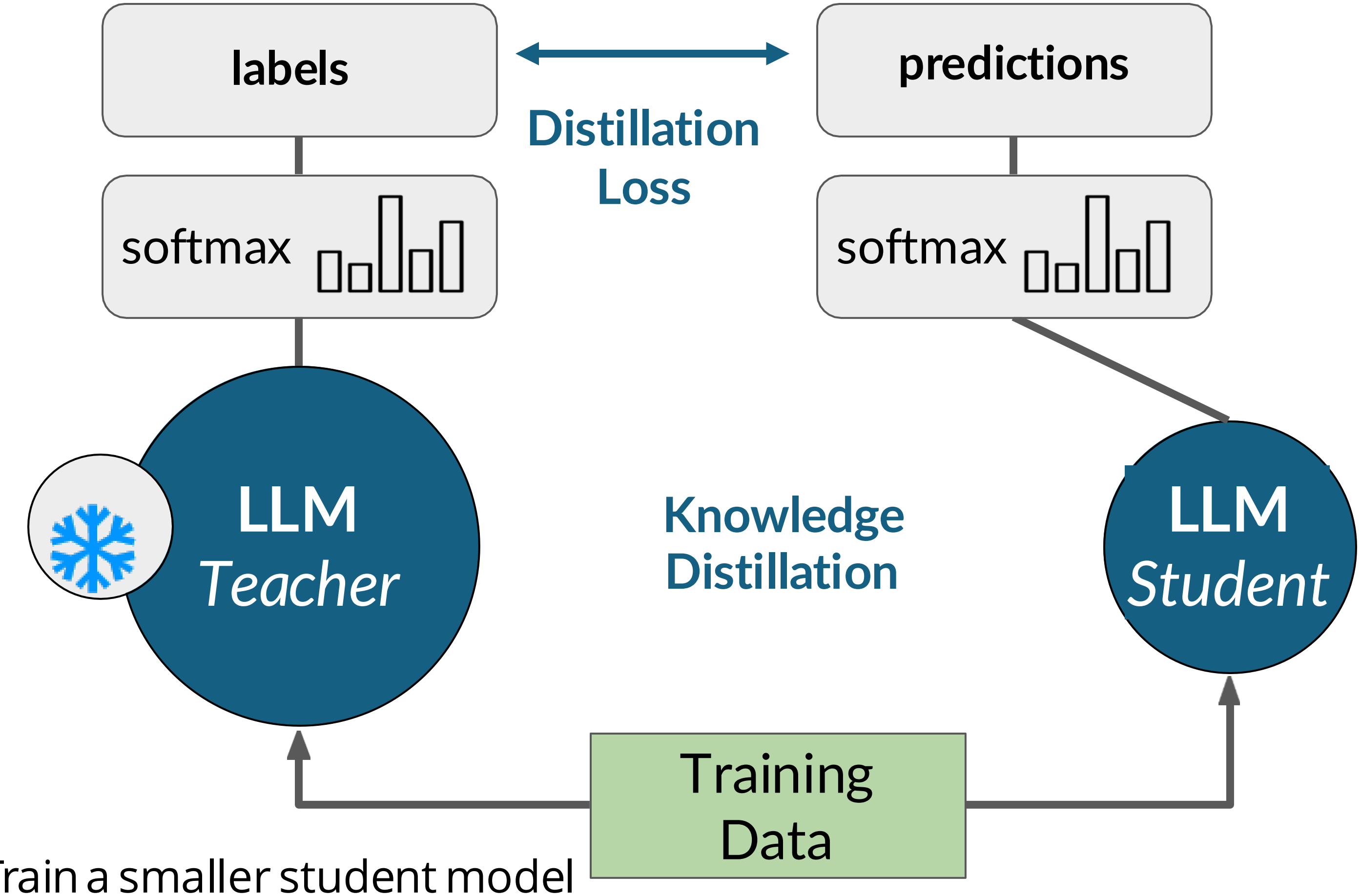
Pruned LLM

Distillation

LLM optimization techniques: Distillation

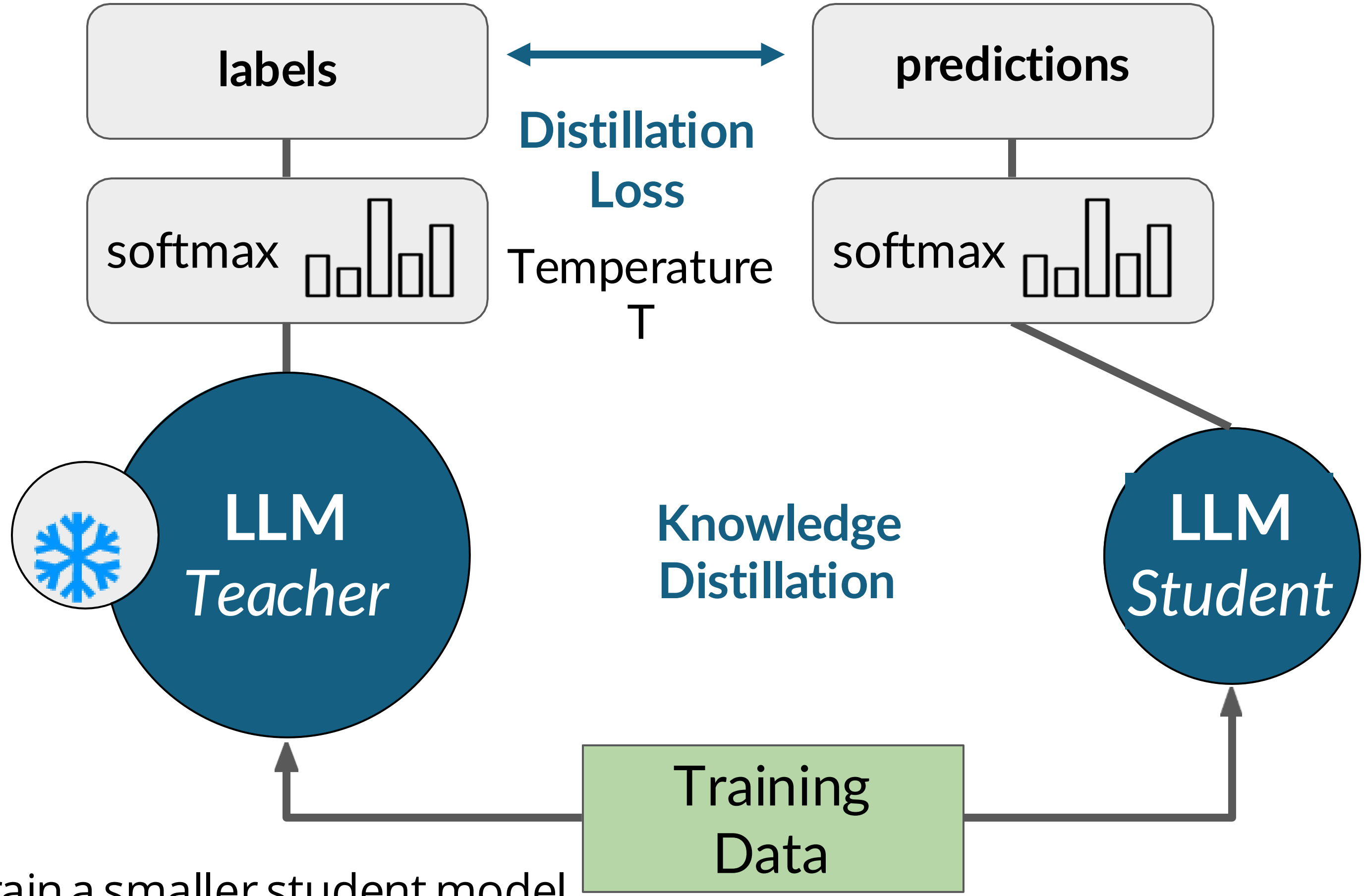


Distillation



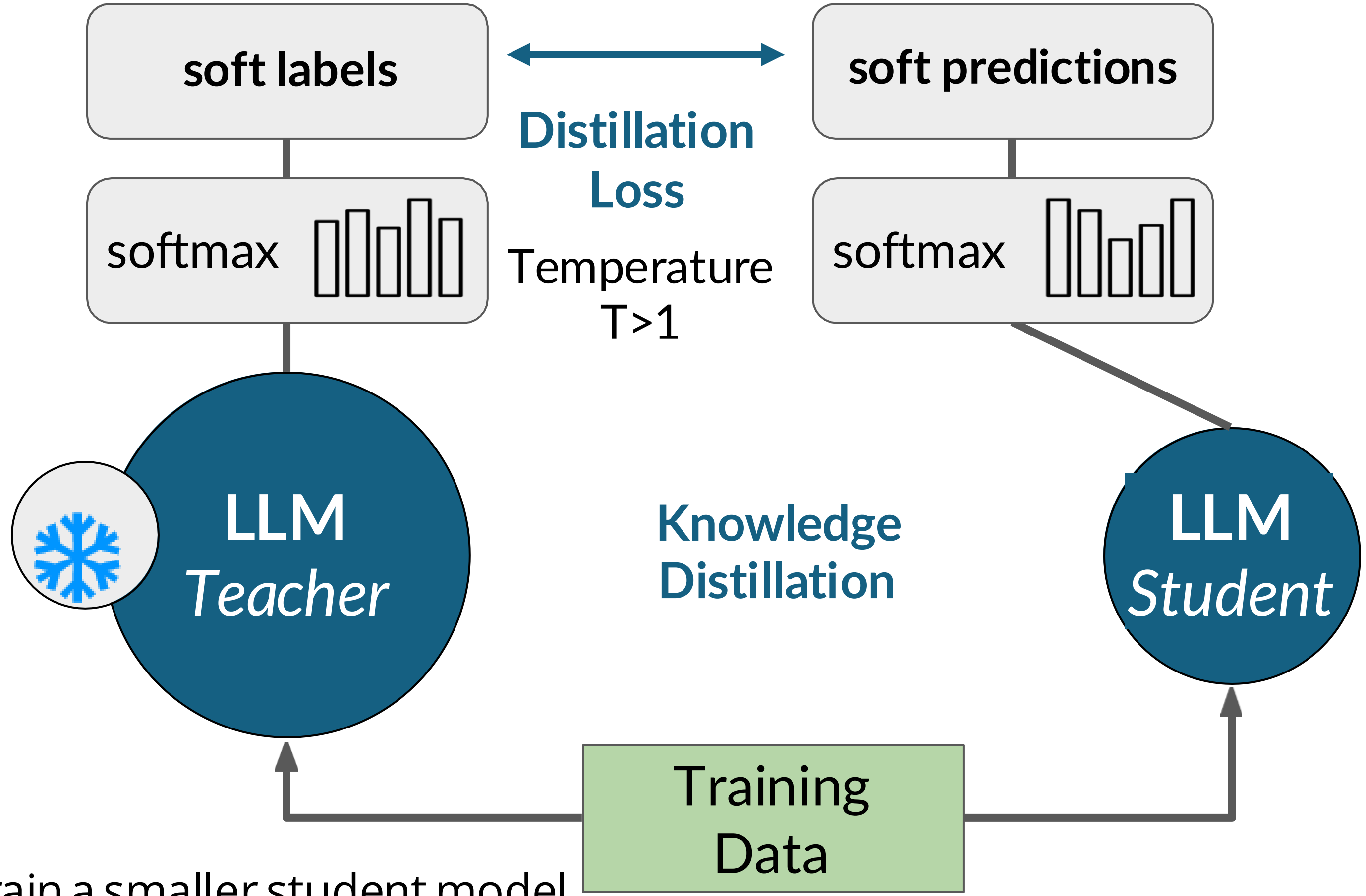
Train a smaller student model
from a larger teacher model

Distillation



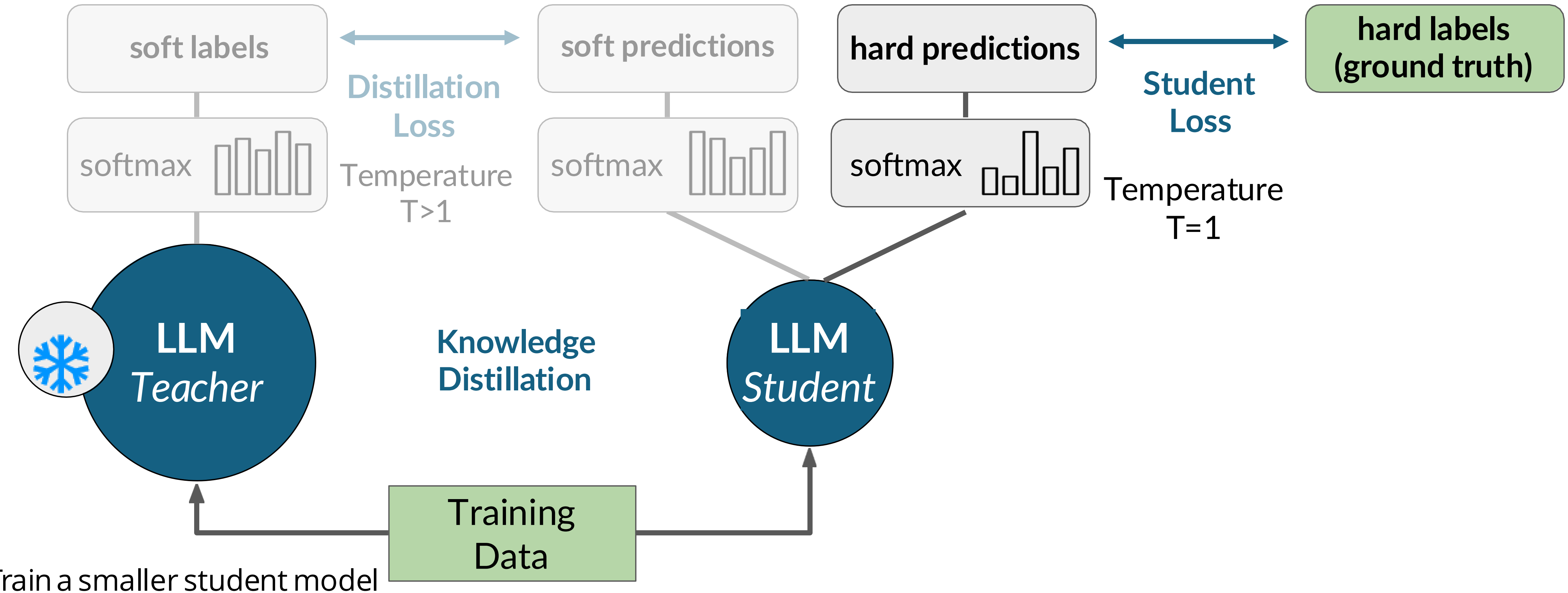
Train a smaller student model
from a larger teacher model

Distillation



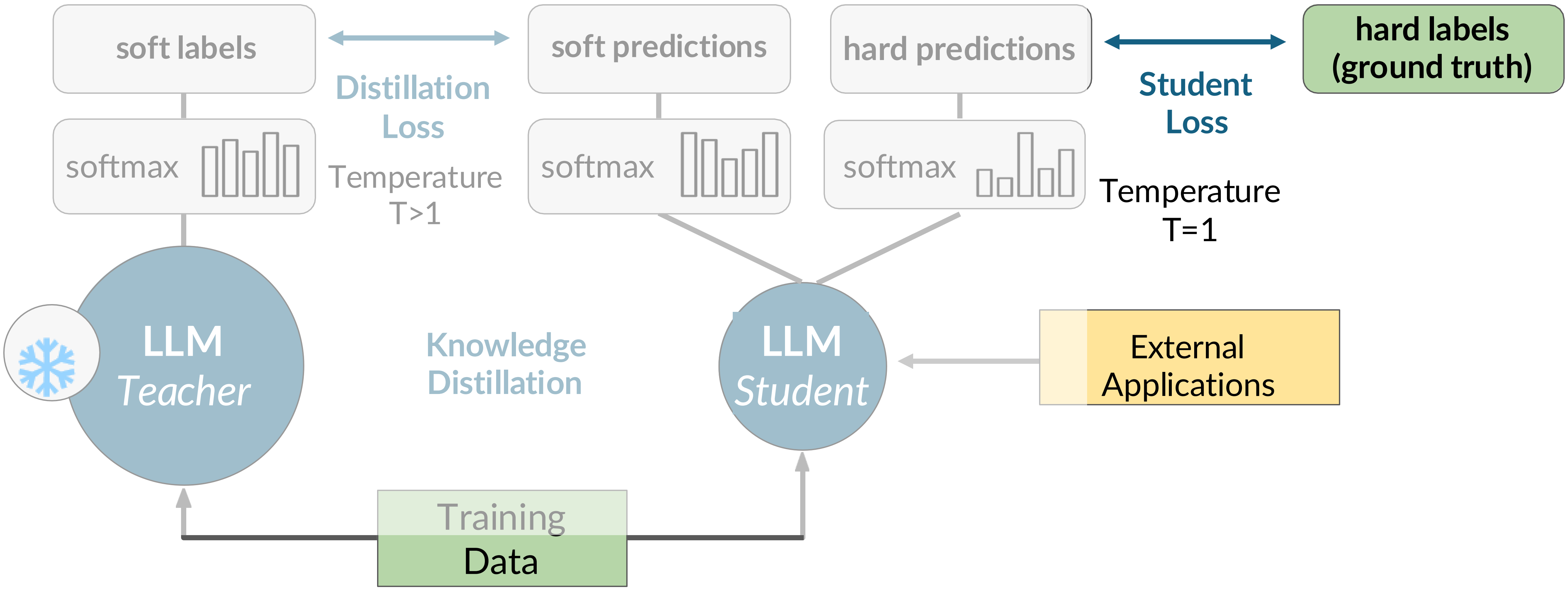
Train a smaller student model
from a larger teacher model

Distillation



Train a smaller student model from a larger teacher model

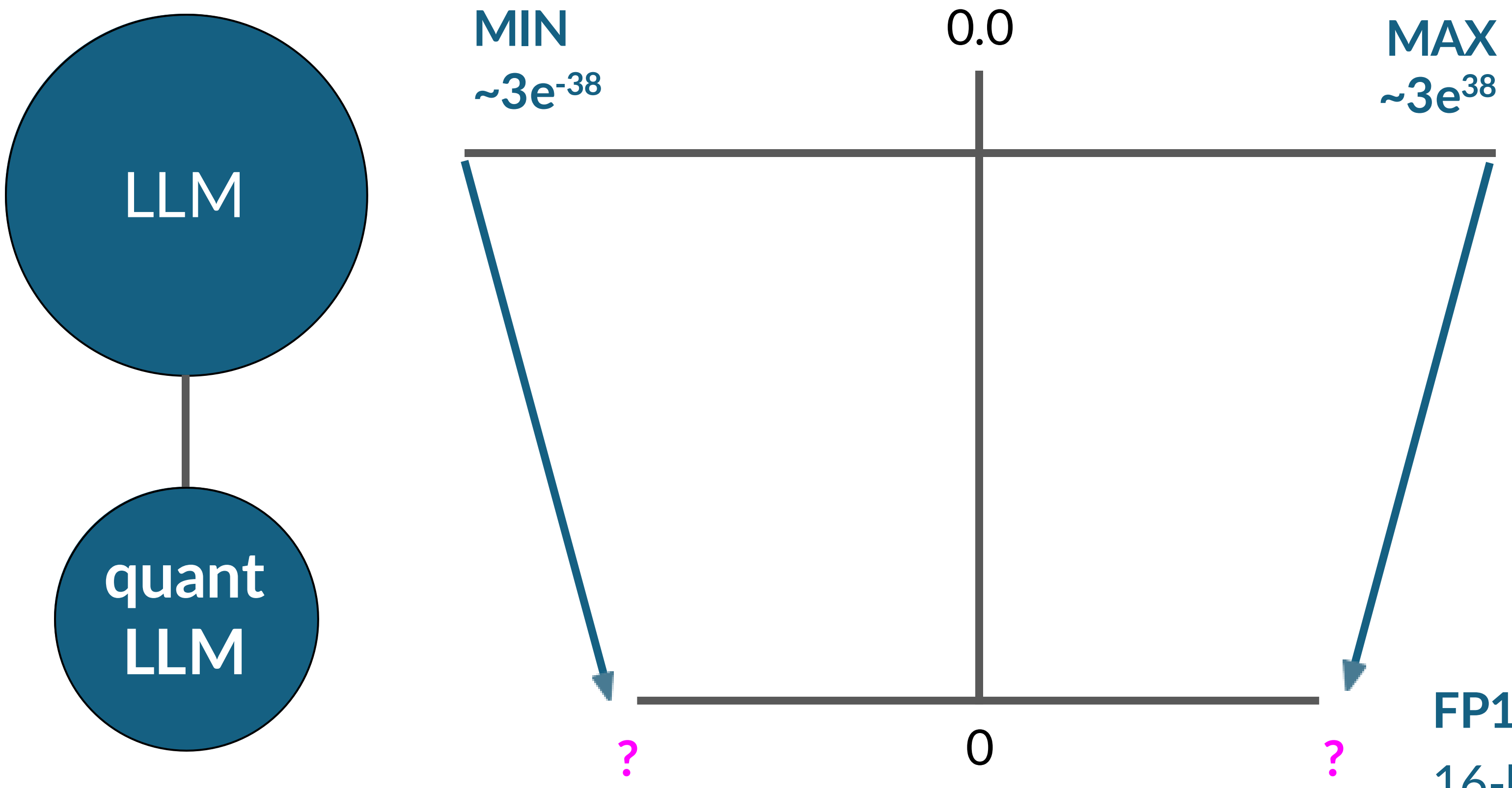
Distillation



Quantization

Post-Training Quantization (PTQ)

- Reduce precision of model weights



FP32

32-bit floating point

- **Applied to model weights** (and/or activations)
- **Requires calibration** to capture dynamic range

FP16 | BFLOAT16 | INT8

16-bit floating point | 8-bit integer

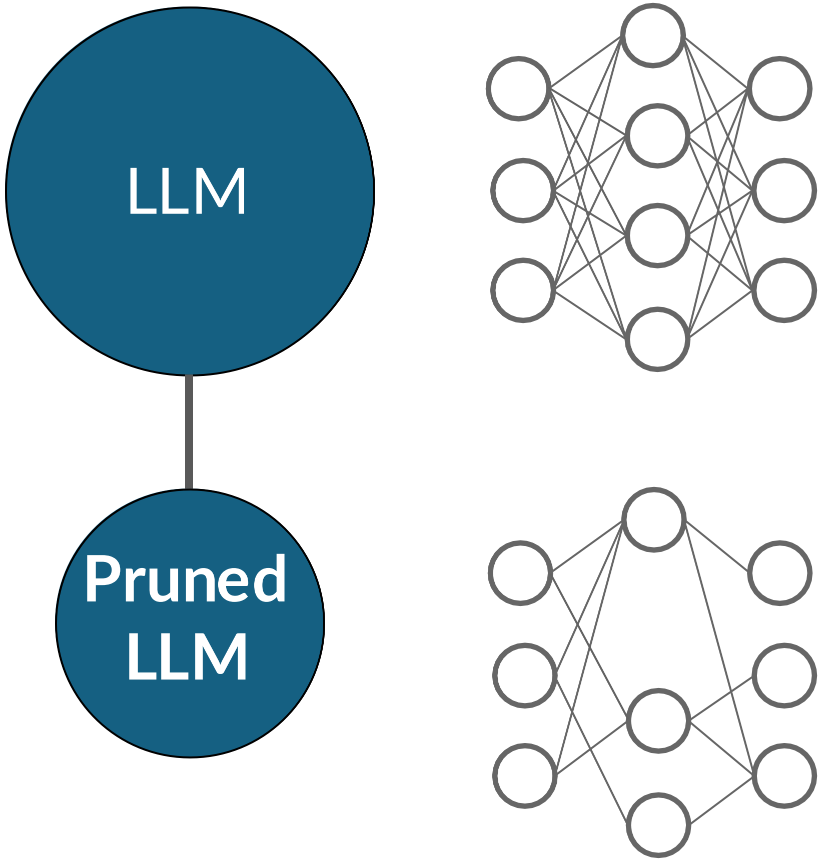
Pruning

Pruning

- Remove model weights with values close or equal to zero

Pruning

- Remove model weights with values close or equal to zero



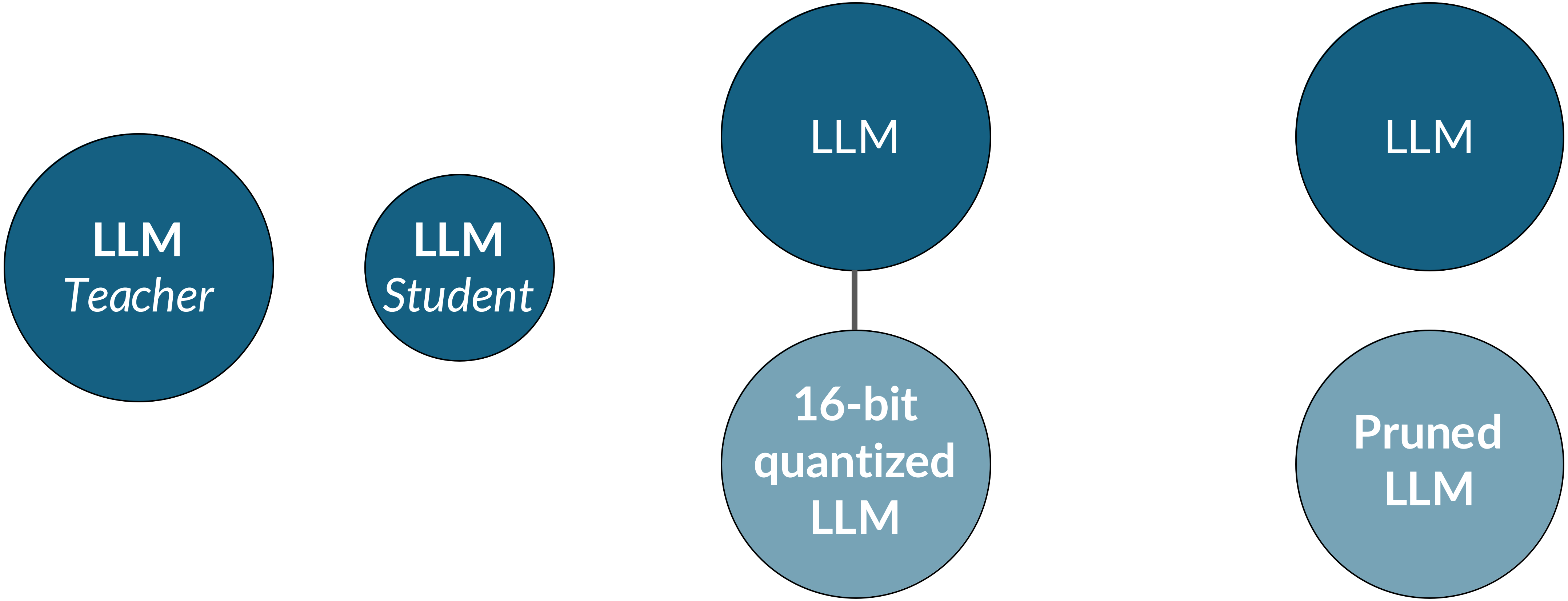
- Pruning methods
  - Full model re-training
  - PEFT/LoRA
  - Post-training

- In theory, reduces model size and improves performance

- In practice, only small % in LLMs are zero-weights

Summary: LLM optimization techniques

Distillation **Quantization** **Pruning**



What to do? 🤯

Cheat Sheet - Time and effort

Training duration
Customization
Objective
Expertise

Cheat Sheet - Time and effort

	Pre-training
Training duration	Days to weeks to months
Customization	Determine model architecture, size and tokenizer.
	Choose vocabulary size and # of tokens for input/context
	Large amount of domain training data
Objective	Next-token prediction
Expertise	High

Cheat Sheet - Time and effort

	Pre-training	Prompt engineering
Training duration	Days to weeks to months	Not required
Customization	Determine model architecture, size and tokenizer.	No model weights
	Choose vocabulary size and # of tokens for input/context	Only prompt customization
	Large amount of domain training data	
Objective	Next-token prediction	Increase task performance
Expertise	High	Low

Cheat Sheet - Time and effort

	Pre-training	Prompt engineering	Prompt tuning and fine-tuning
Training duration	Days to weeks to months	Not required	Minutes to hours
Customization	Determine model architecture, size and tokenizer.	No model weights	Tune for specific tasks
	Choose vocabulary size and # of tokens for input/context	Only prompt customization	Add domain-specific data
	Large amount of domain training data		Update LLM model or adapter weights
Objective	Next-token prediction	Increase task performance	Increase task performance
Expertise	High	Low	Medium

Cheat Sheet - Time and effort

	Pre-training	Prompt engineering	Prompt tuning and fine-tuning	Reinforcement learning/human feedback
Training duration	Days to weeks to months	Not required	Minutes to hours	Minutes to hours similar to fine-tuning
Customization	Determine model architecture, size and tokenizer.	No model weights	Tune for specific tasks	Need separate reward model to align with human goals (helpful, honest, harmless)
	Choose vocabulary size and # of tokens for input/context	Only prompt customization	Add domain-specific data	Update LLM model or adapter weights
	Large amount of domain training data		Update LLM model or adapter weights	
Objective	Next-token prediction	Increase task performance	Increase task performance	Increase alignment with human preferences
Expertise	High	Low	Medium	Medium-High

Cheat Sheet - Time and effort

	Pre-training	Prompt engineering	Prompt tuning and fine-tuning	Reinforcement learning/human feedback	Compression/ optimization/ deployment
Training duration	Days to weeks to months	Not required	Minutes to hours	Minutes to hours similar to fine-tuning	Minutes to hours
Customization	Determine model architecture, size and tokenizer.	No model weights	Tune for specific tasks	Need separate reward model to align with human goals (helpful, honest, harmless)	Reduce model size through model pruning, weight quantization, distillation
	Choose vocabulary size and # of tokens for input/context	Only prompt customization	Add domain-specific data	Update LLM model or adapter weights	Smaller size, faster inference
	Large amount of domain training data		Update LLM model or adapter weights		
Objective	Next-token prediction	Increase task performance	Increase task performance	Increase alignment with human preferences	Increase inference performance
Expertise	High	Low	Medium	Medium-High	Medium

Today's lecture

Part I: Recap

Part II: Align with Human Feedback

Part III: Evaluation

Part IV: Inference

Part V: Optimization techniques

Part VI: External interactions

Slides adapted from various sources: [Intro to Large Language Models, Andrej Karpathy, Executive Education Polytechnique, Udemy, Deeplearning.ai, Stanford University CS231n, Financial Times, New York Times, Hi!Paris summer school 2023]

Part VI: Outline

- Remaining problems with Foundation Models

- Solution? Interactions!

  - Interact with external data sources
    - Retrieval Augmented Generation (RAG)

  - Interact with external applications

  - Chain-of-thought prompting

Problems with FoMos

Models having difficulty

Prompt

```
Who is the Prime
Minister of the UK?
```

→ **Model**: LLM →

Completion

```
Who is the Prime
Minister of the UK?
Boris Johnson
```

Out of date

Models having difficulty

Prompt	Model	Completion	
Who is the Prime Minister of the UK?	LLM	Who is the Prime Minister of the UK? **Boris Johnson**	**Out of date**
What is 40366 / 439?	LLM	What is 40366 / 439? **92.549**	**Wrong (91.949)**

Models having difficulty

Prompt	Model	Completion	
`Who is the Prime Minister of the UK?`	LLM	`Who is the Prime Minister of the UK?` `Boris Johnson`	**Out of date**
`What is 40366 / 439?`	LLM	`What is 40366 / 439?` `92.549`	**Wrong (91.949)**
`What is a Martian Dunetree?`	LLM	`What is a Martian Dunetree?` `A Martian Dunetree is a type of extraterrestrial plant found on Mars.`	**Hallucination**

Solution?

Part VI: Outline

- Remaining problems with Foundation Models

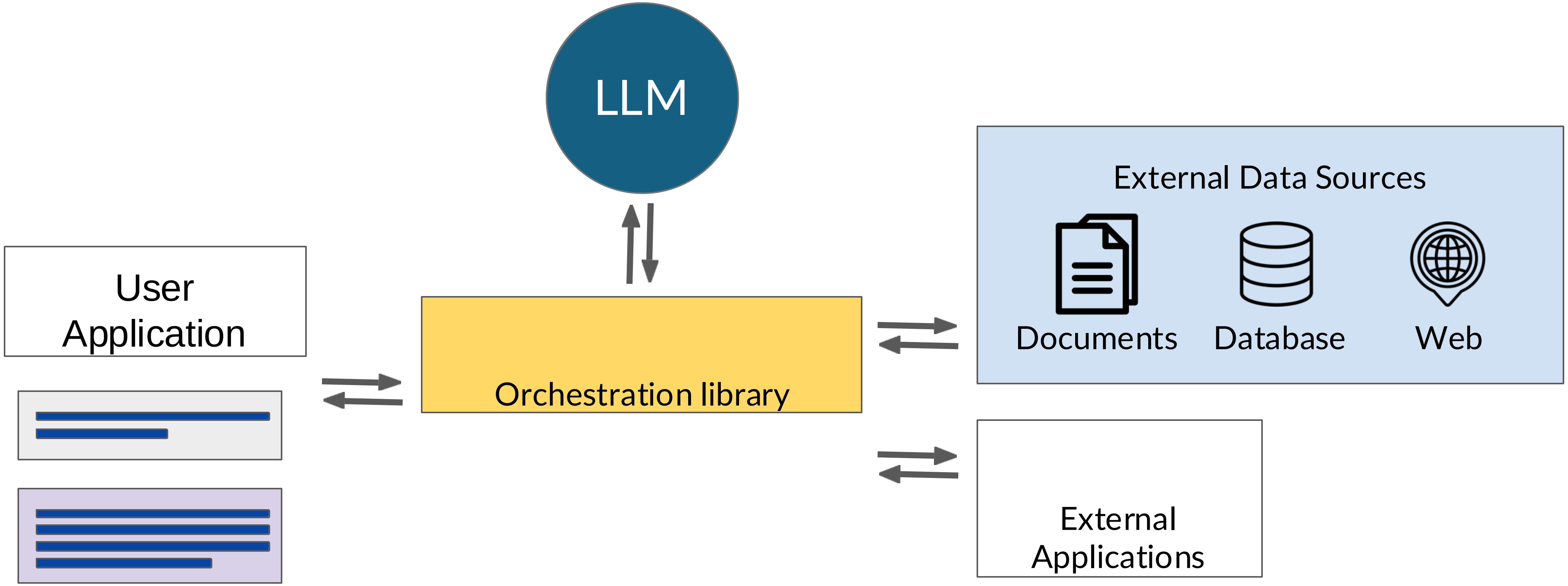
- **Solution? Interactions!**

  - Interact with external data sources
    - Retrieval Augmented Generation (RAG)

  - Interact with external applications

  - Chain-of-thought prompting

LLM-powered applications



Interact with external data sources

Knowledge cut-offs in LLMs

Prompt

```
Who is the
current Prime
Minister of the
United Kingdom?
```

Model

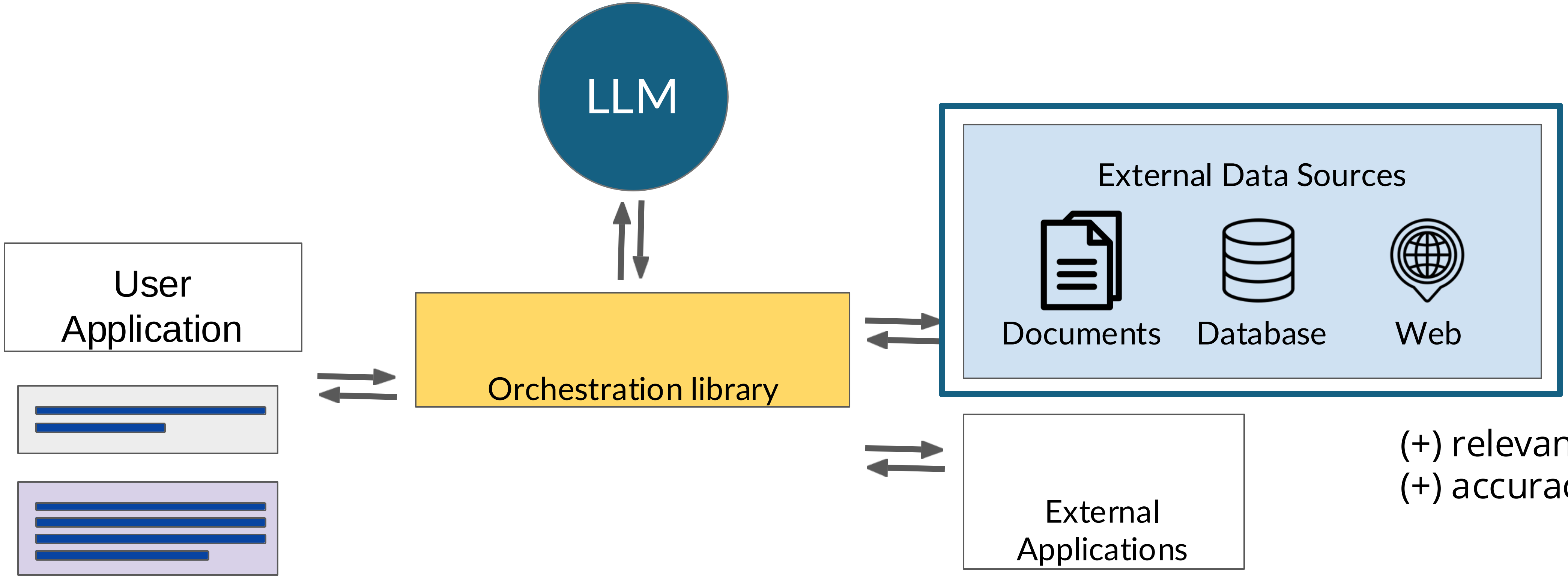
LLM

Completion

```
Who is the
current Prime
Minister of the
United Kingdom?

Boris Johnson
```

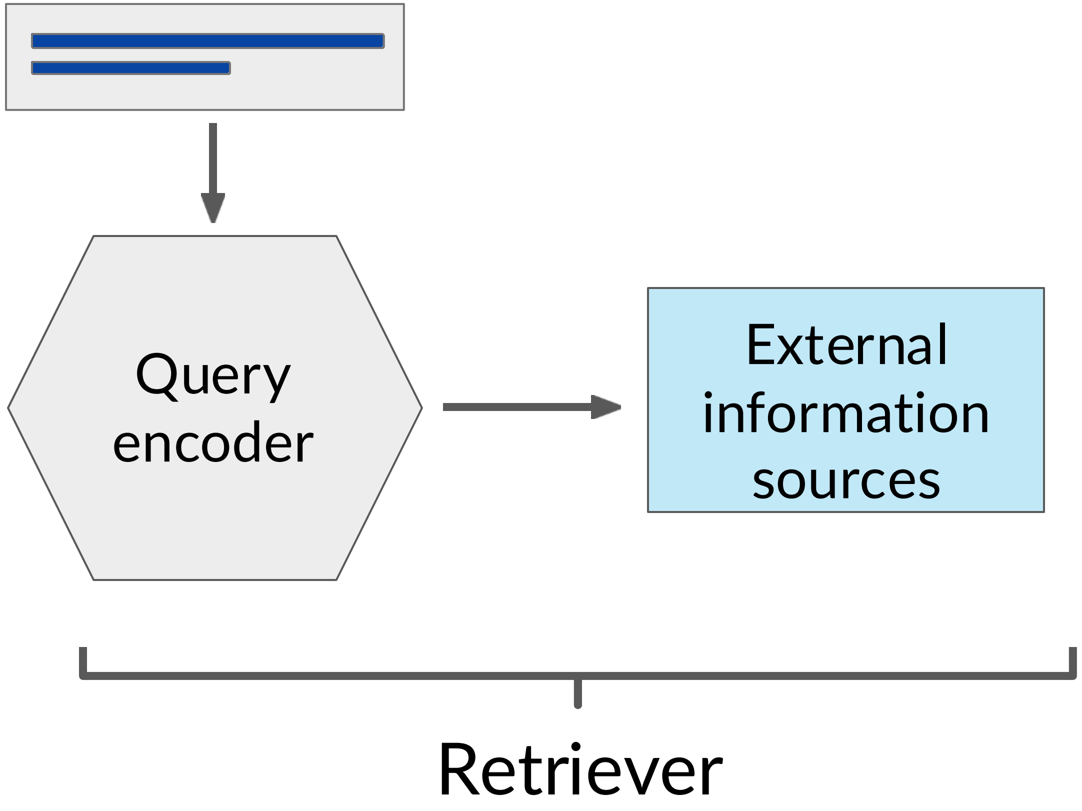
LLM-powered applications



(+) relevance
(+) accuracy

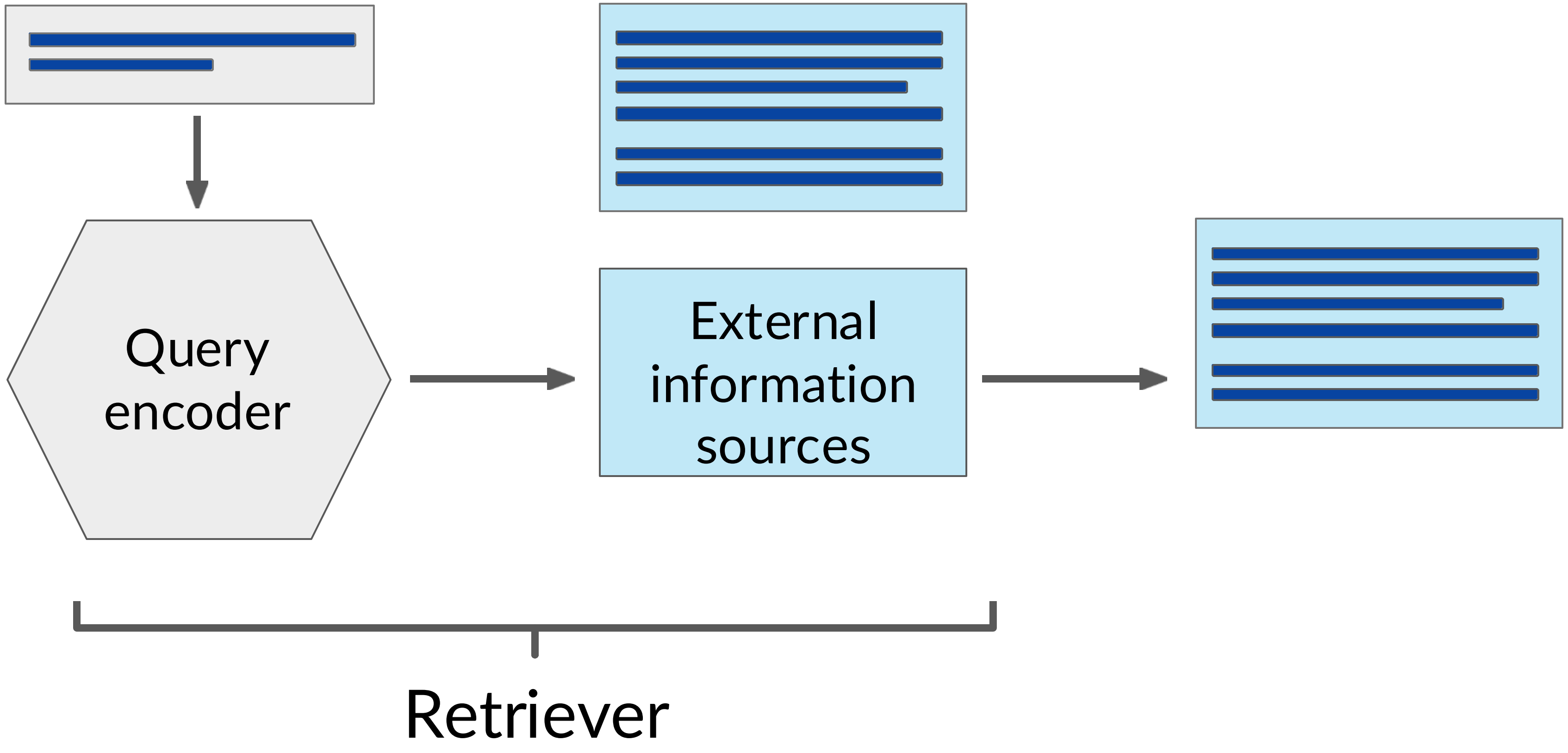
Retrieval Augmented Generation (RAG)

Retrieval Augmented Generation (RAG)



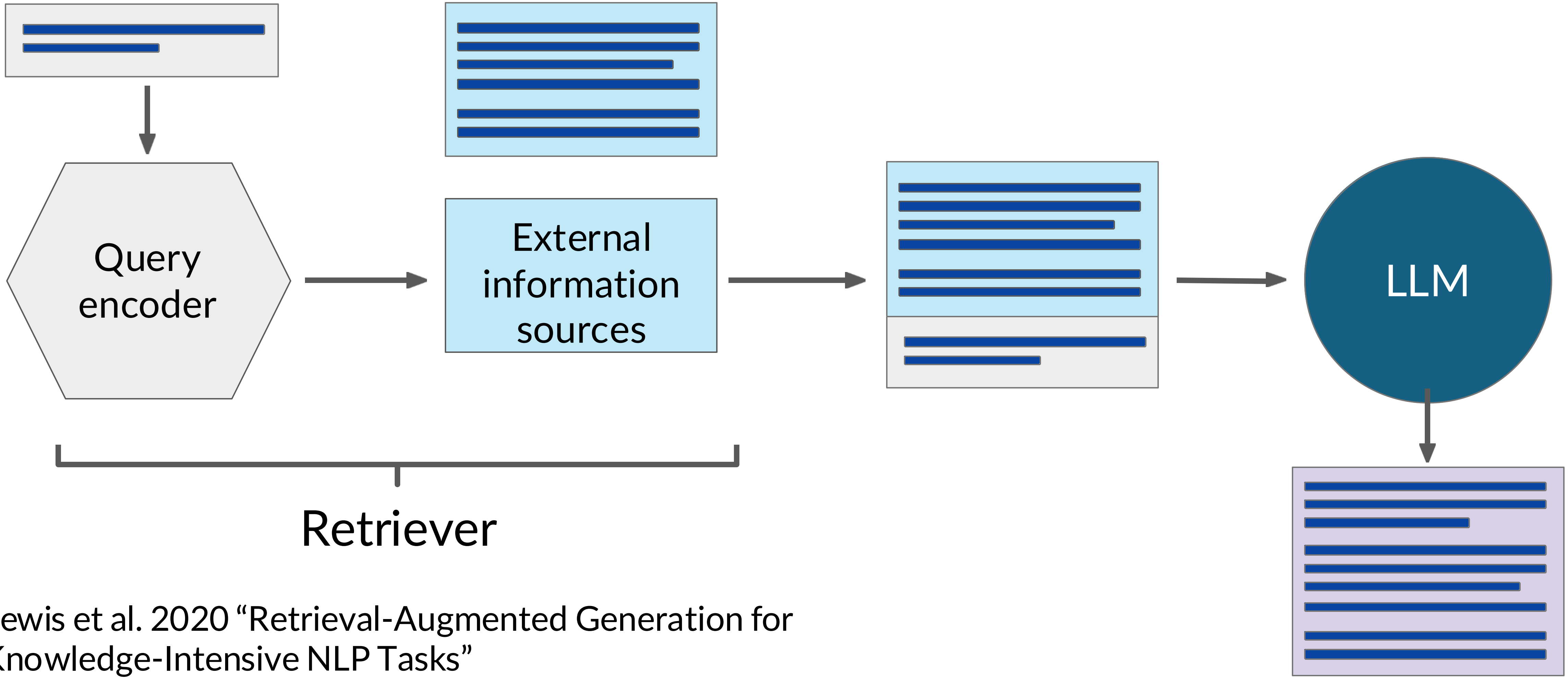
Lewis et al. 2020 “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks”

Retrieval Augmented Generation (RAG)



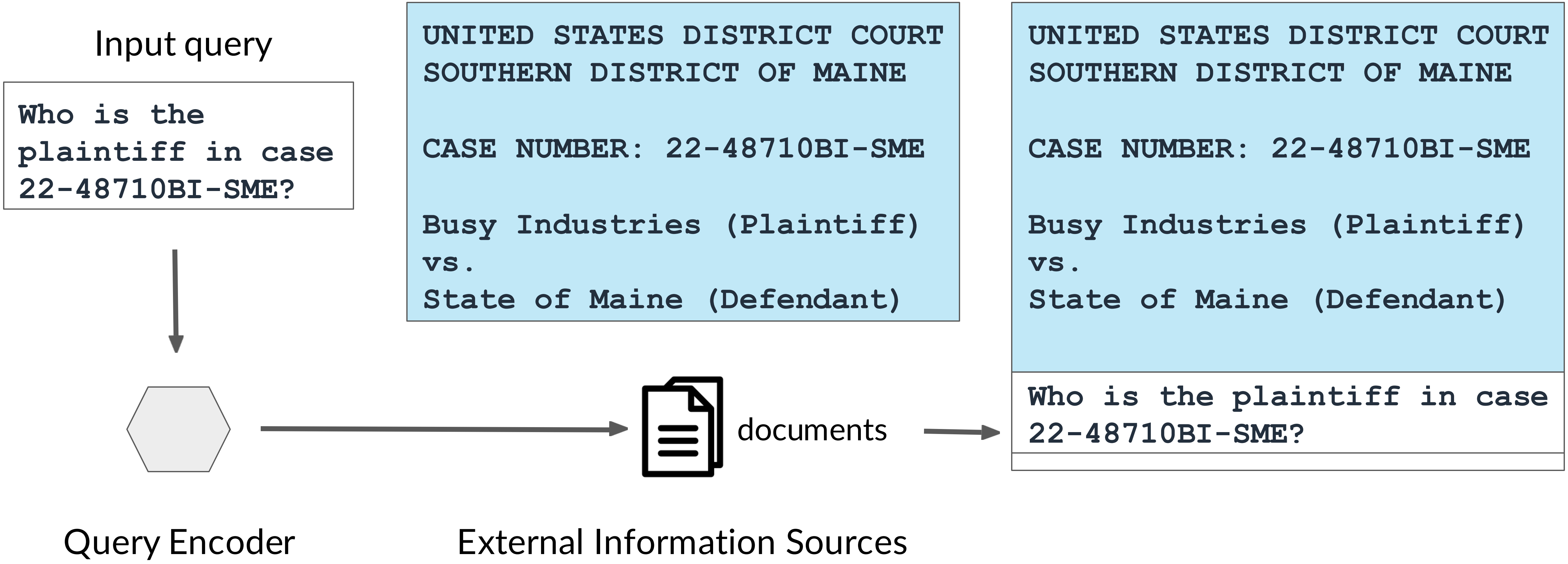
Lewis et al. 2020 “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks”

Retrieval Augmented Generation (RAG)



Lewis et al. 2020 "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks"

Example: Searching legal documents



Example: Searching legal documents

```
UNITED STATES DISTRICT COURT
SOUTHERN DISTRICT OF MAINE

CASE NUMBER: 22-48710BI-SME

Busy Industries (Plaintiff)
vs.
State of Maine (Defendant)
```

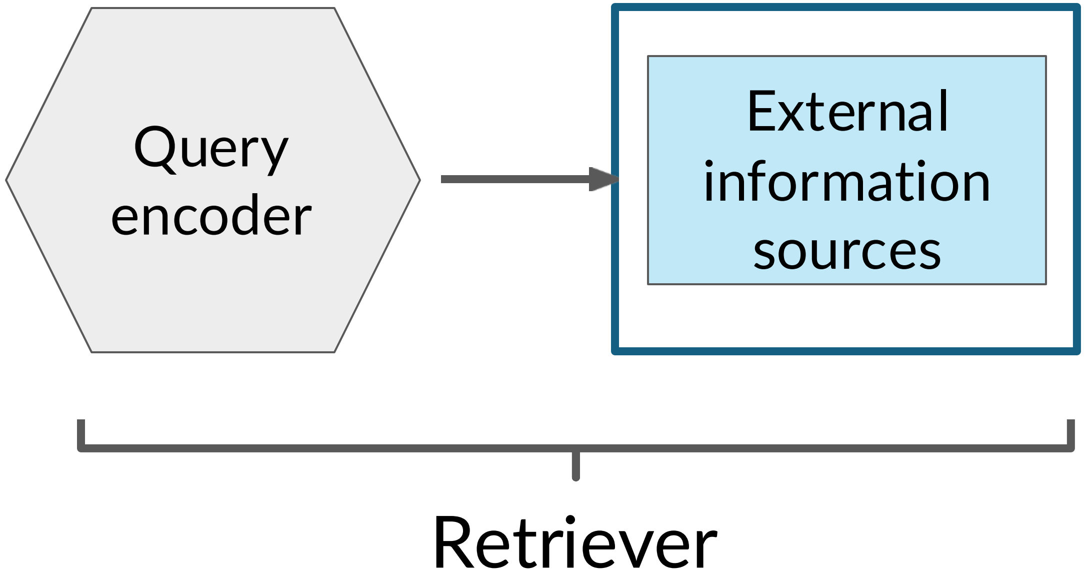
```
Who is the plaintiff in case
22-48710BI-SME?
```

→ LLM →

Completion

```
Busy Industries
```

RAG integrates with many types of data sources



External Information Sources

- Documents
- Wikis
- Expert Systems
- Web pages
- Databases

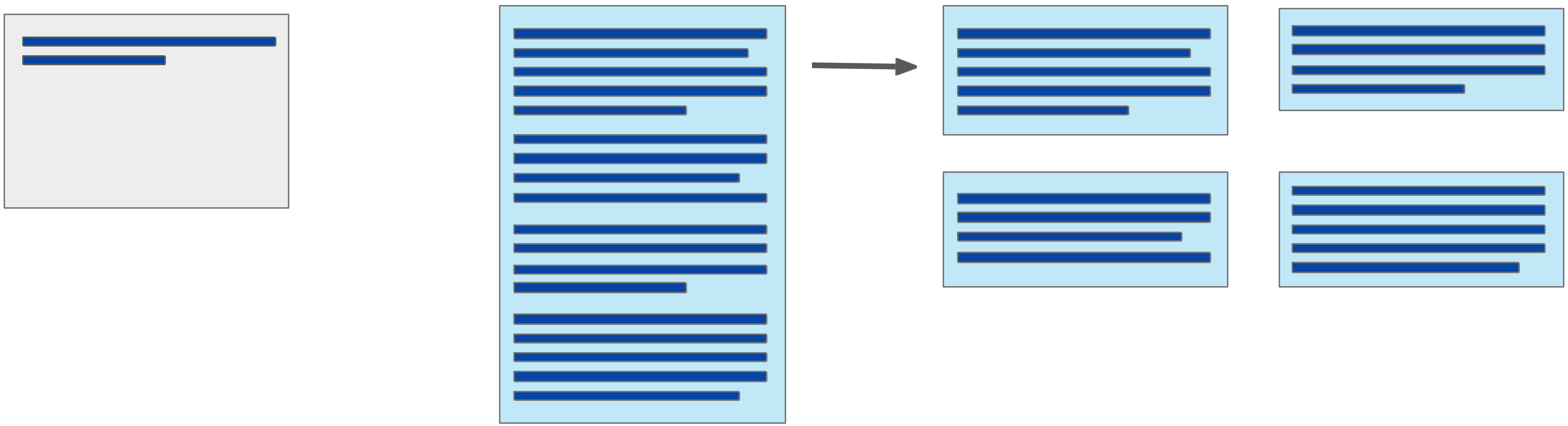
Data preparation for vector store for RAG

- Two considerations for using external data in RAG:
  - Data must fit inside context window

Prompt context limit few 1000 tokens

Single document too large to fit in window

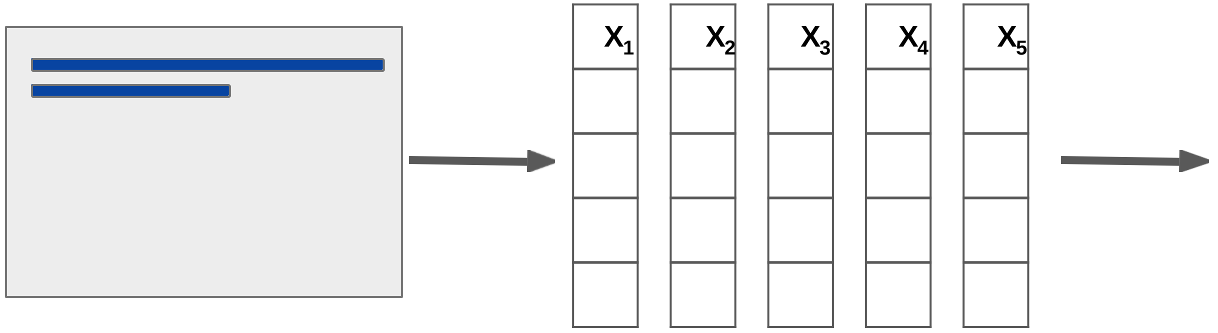
Split long sources into short chunks

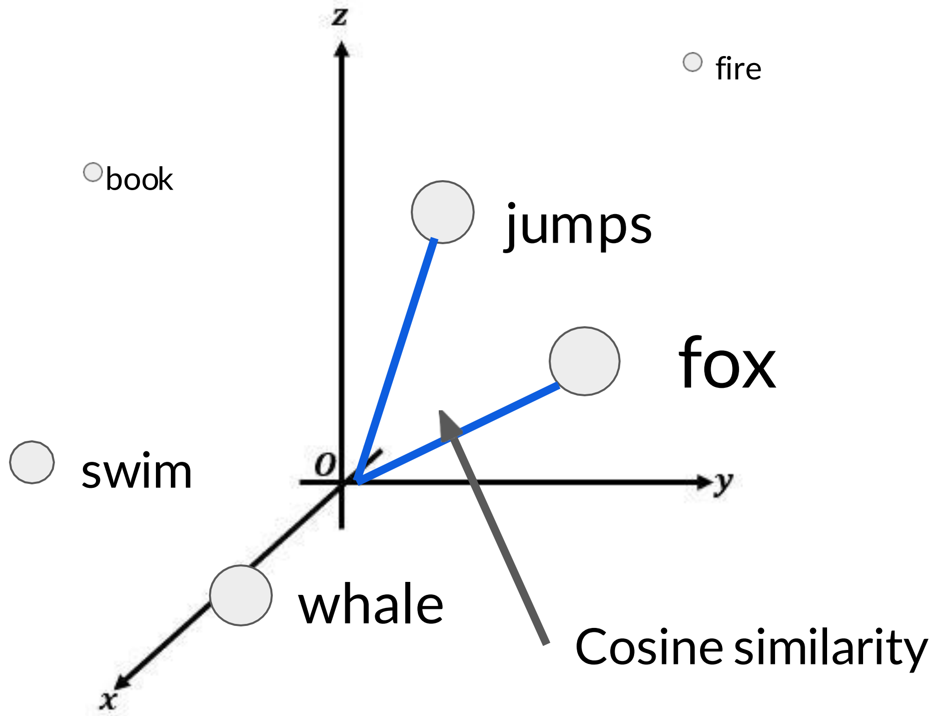


Data preparation for RAG

- Two considerations for using external data in RAG:
  - Data must fit inside context window
  - Data must be in format that allows its relevance to be assessed at inference time: **Embedding vectors**

Prompt text converted to embedding vectors

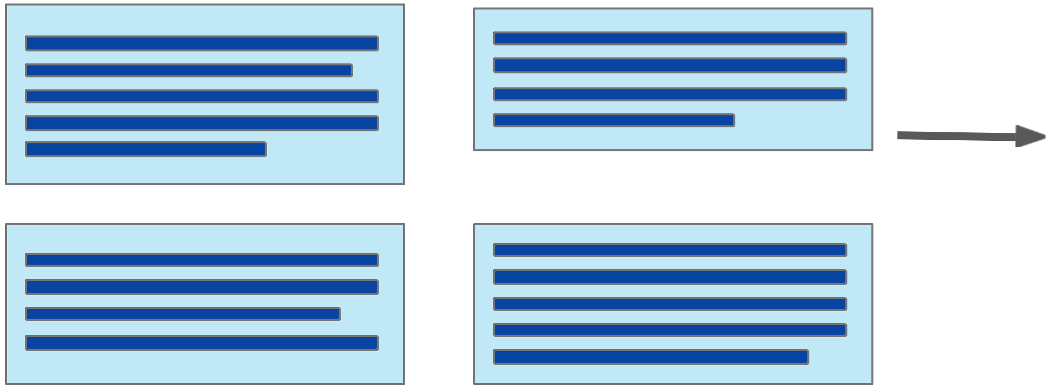


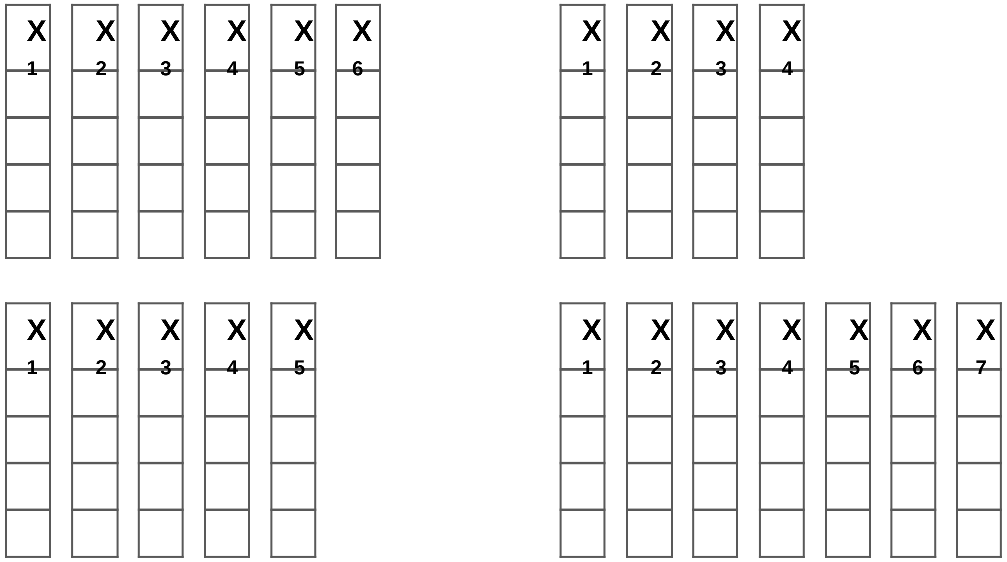


Data preparation for RAG

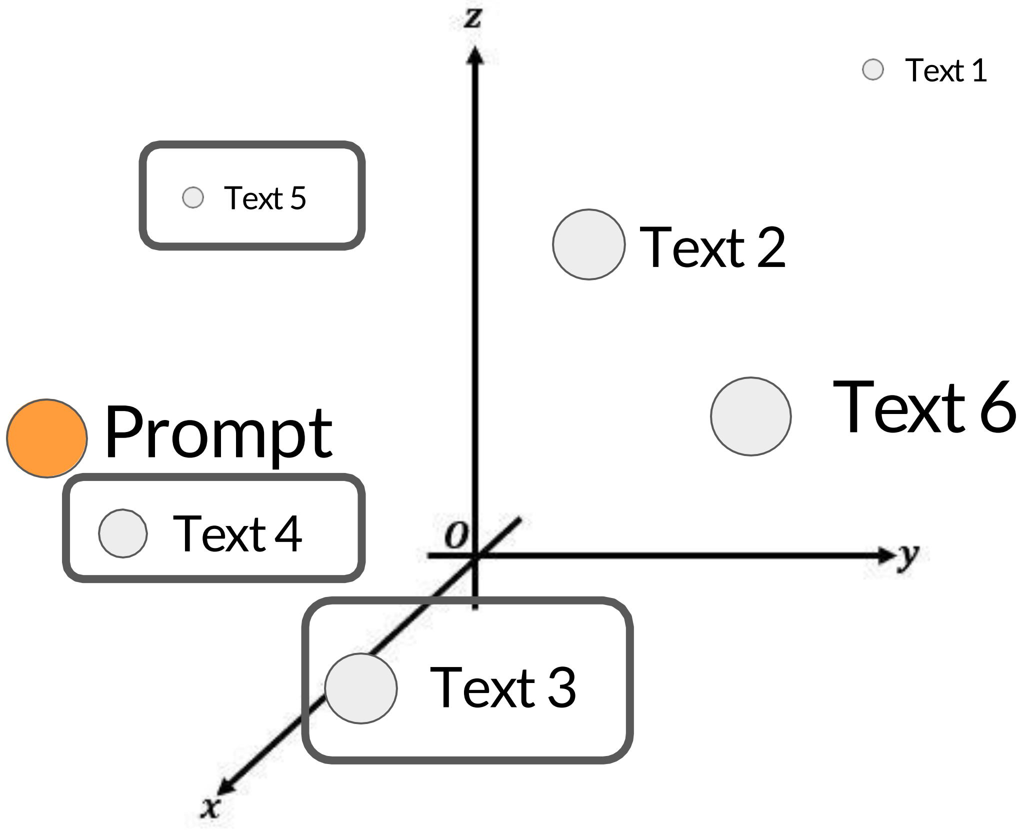
- Two considerations for using external data in RAG:
  - Data must fit inside context window
  - Data must be in format that allows its relevance to be assessed at inference time: **Embedding vectors**

Process each chunk with LLM
to produce embedding vectors





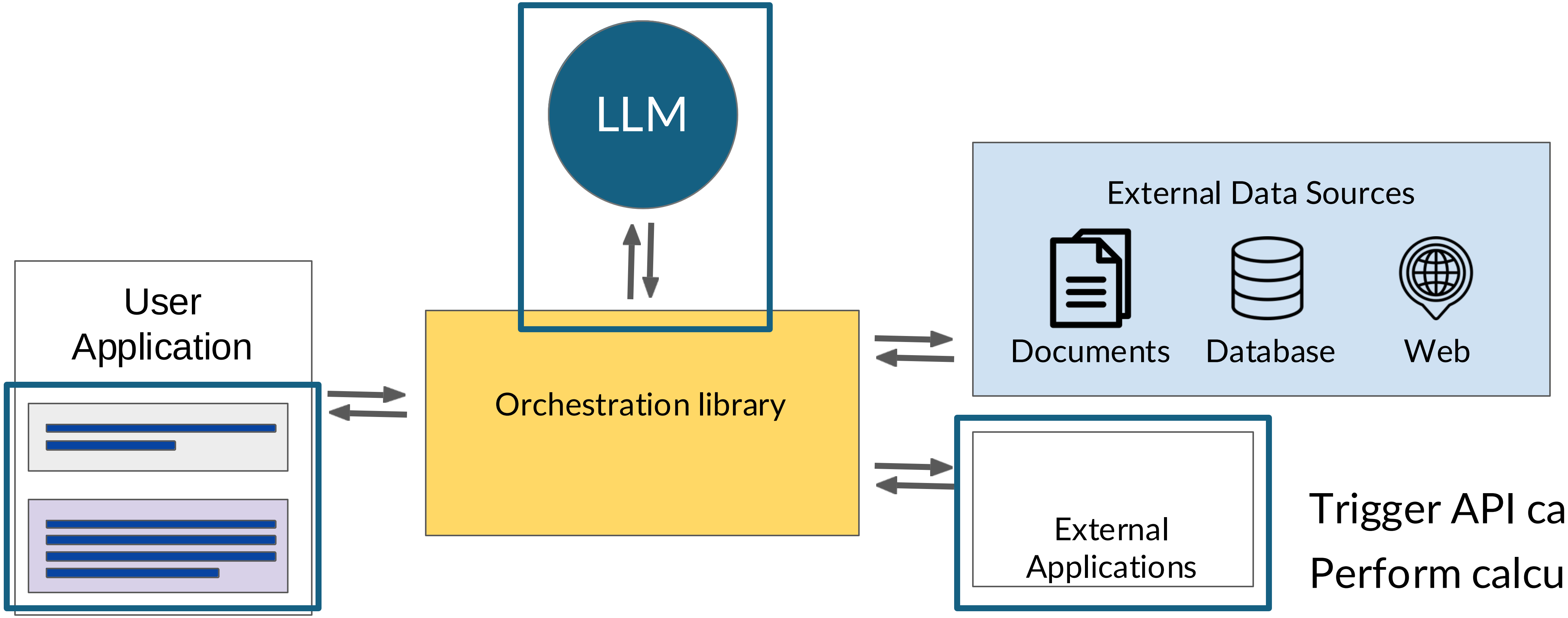
Vector database search



- Each text in vector store is identified by a key
- Enables a **citation** to be included in completion

Interact with external applications

LLM-powered applications



Trigger API call

Perform calculations

Requirements for using LLMs to power applications

Plan actions

Steps to process return:

Step 1: Check order ID

Step 2: Request label

Step 3: Verify user email

Step 4: Email user label

Requirements for using LLMs to power applications

Plan actions

Steps to process return:

Step 1: Check order ID

Step 2: Request label

Step 3: Verify user email

Step 4: Email user label

Format outputs

SQL Query:

SELECT COUNT(*)
FROM orders
WHERE order_id = 21104

Requirements for using LLMs to power applications

Plan actions

Steps to process return:

Step 1: Check order ID

Step 2: Request label

Step 3: Verify user email

Step 4: Email user label

Format outputs

SQL Query:

SELECT COUNT(*)
FROM orders
WHERE order_id = 21104

Validate actions

Collect required user information and make sure it is in the completion

User email:
vic.k@email.net

Requirements for using LLMs to power applications

Plan actions

Steps to process return:
Step 1: Check order ID
Step 2: Request label
Step 3: Verify user email
Step 4: Email user label

Format outputs

SQL Query:
SELECT COUNT(*)
FROM orders
WHERE order_id = 21104

Validate actions

Collect required user information and make sure it is in the completion

User email:
vic.k@email.net

Prompt structure is important!

Chain-of-Thought Prompting

LLMs can struggle with complex reasoning problems

Prompt

```
Q: Roger has 5 tennis balls.
He buys 2 more cans of tennis
balls. Each can has 3 tennis
balls. How many tennis balls
does he have now?

A: The answer is 11

Q: The cafeteria had 23
apples. If they used 20 to
make lunch and bought 6 more,
how many apples do they have?
```

Model

LLM

Completion

LLMs can struggle with complex reasoning problems

Prompt

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model

LLM

Completion

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

A: The answer is 27.

Humans take a step-by-step approach to solving complex problems

Humans take a step-by-step approach to solving complex problems

Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

Humans take a step-by-step approach to solving complex problems

Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

Start: Roger started with 5 balls.

Step 1: 2 cans of 3 tennis balls each is 6 tennis balls.

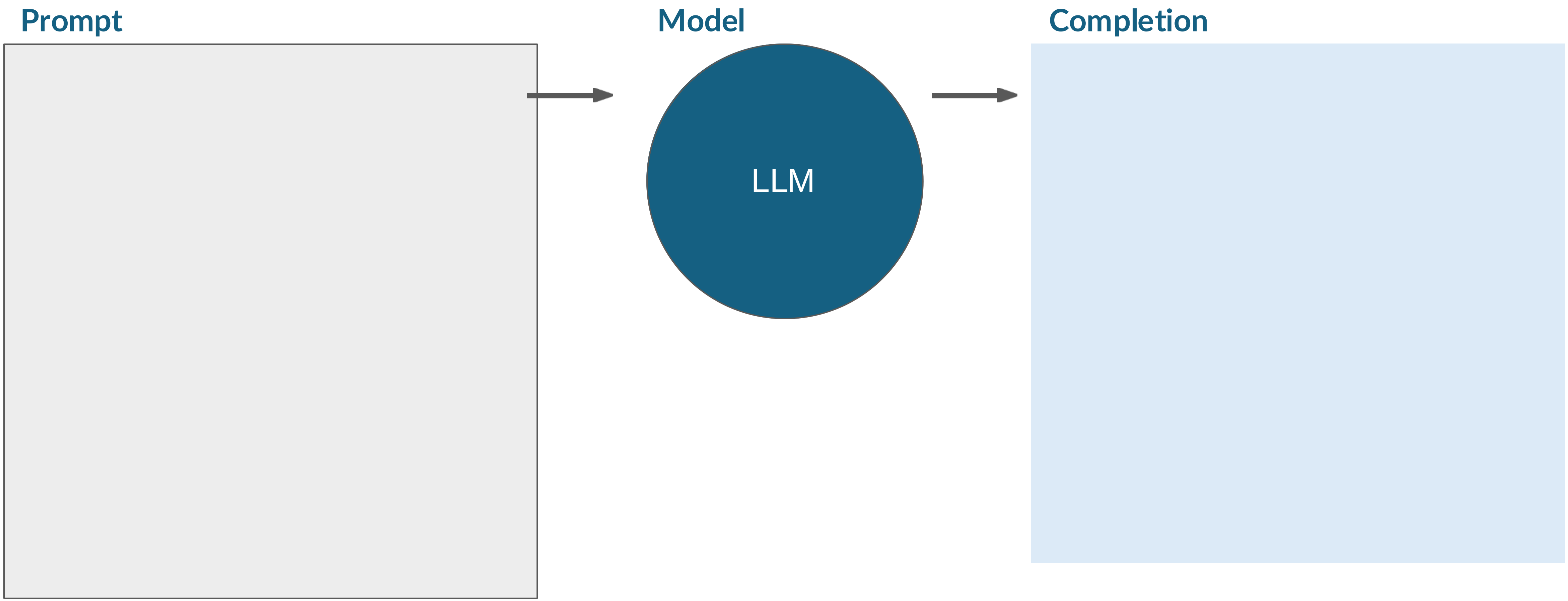
Step 2: 5 + 6 = 11

End: The answer is 11

Reasoning steps

“Chain of thought”

Chain-of-Thought Prompting can help LLMs reason



Source: Wei et al. 2022, “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models”

Chain-of-Thought Prompting can help LLMs reason

Prompt

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11.The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model

LLM

Completion

Q: Roger has 5 tennis balls.
...
...
...
how many apples do they have?

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23-20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9.

Source: Wei et al. 2022, "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models"

Chain-of-Thought Prompting can help LLMs reason

Prompt

```
Q: Yes or no: Would a pear
sink in water?

A: The density of a pear is
about 0.6 g/cm^3, which is
less than water. Thus, a pear
would float. So the answer is
no.

Q. Yes or no: Would a gold
ring sink to the bottom of a
swimming pool?
```

Model

LLM

Completion

```
Q: Yes or no: Would a pear
sink in water?
...
...
So the answer is no.

The density of gold is 19.3
g/cm^3, which is much greater
than the density of water.
Therefore, a gold ring would
sink to the bottom of a
swimming pool. So the answer
is yes.
```

Source: Wei et al. 2022, “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models”

Summary

Prompt

```
Q: summarize the exciting
future of generative AI
```

Model

LLM

Completion

```
Generative AI's future
promises to revolutionize
creativity,
personalization, and
healthcare. It will
enable tailored content,
automate tasks, and
enhance human
collaboration. As its
capabilities grow, focus
will be on ethical use
and ensuring fairness and
safety.
```

ChatGPT, November 2024

Thank you