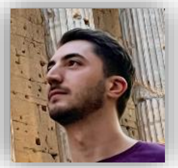
NLP and the Web – WS 2024/2025

Lecture 12
Neural Language Modeling 4


Dr. Thomas Arnold
Hovhannes Tamoyan
Kexin Wang

Ubiquitous Knowledge Processing Lab
Technische Universität Darmstadt

Syllabus (tentative)

Nr.	Lecture
01	Introduction / NLP basics
02	Foundations of Text Classification
03	IR – Introduction, Evaluation
04	IR – Word Representation
05	IR – Transformer/BERT
06	IR – Dense Retrieval
07	IR – Neural Re-Ranking
08	LLM – Language Modeling Foundations, Tokenization
09	LLM – Neural LLM
10	LLM – Adaptation
11	LLM – Prompting, Alignment, Instruction Tuning
12	**LLM – Long Contexts, RAG**
13	LLM – Scaling, Computation Cost
14	Review & Preparation for the Exam

Outline

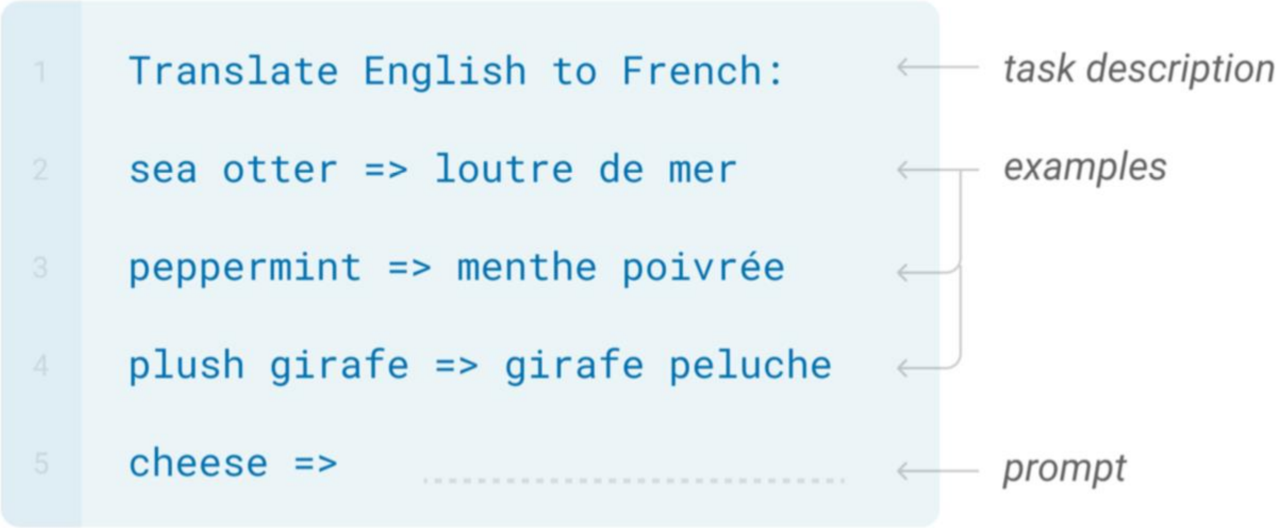
Recap

Reinforcement Learning

Long Context

Retrieval-based LMs

In-Context Learning



Language Modeling != Following Human Instructions

TECHNISCHE UNIVERSITÄT DARMSTADT

PROMPT	*Explain the moon landing to a 6 year old in a few sentences.*
COMPLETION	GPT-3

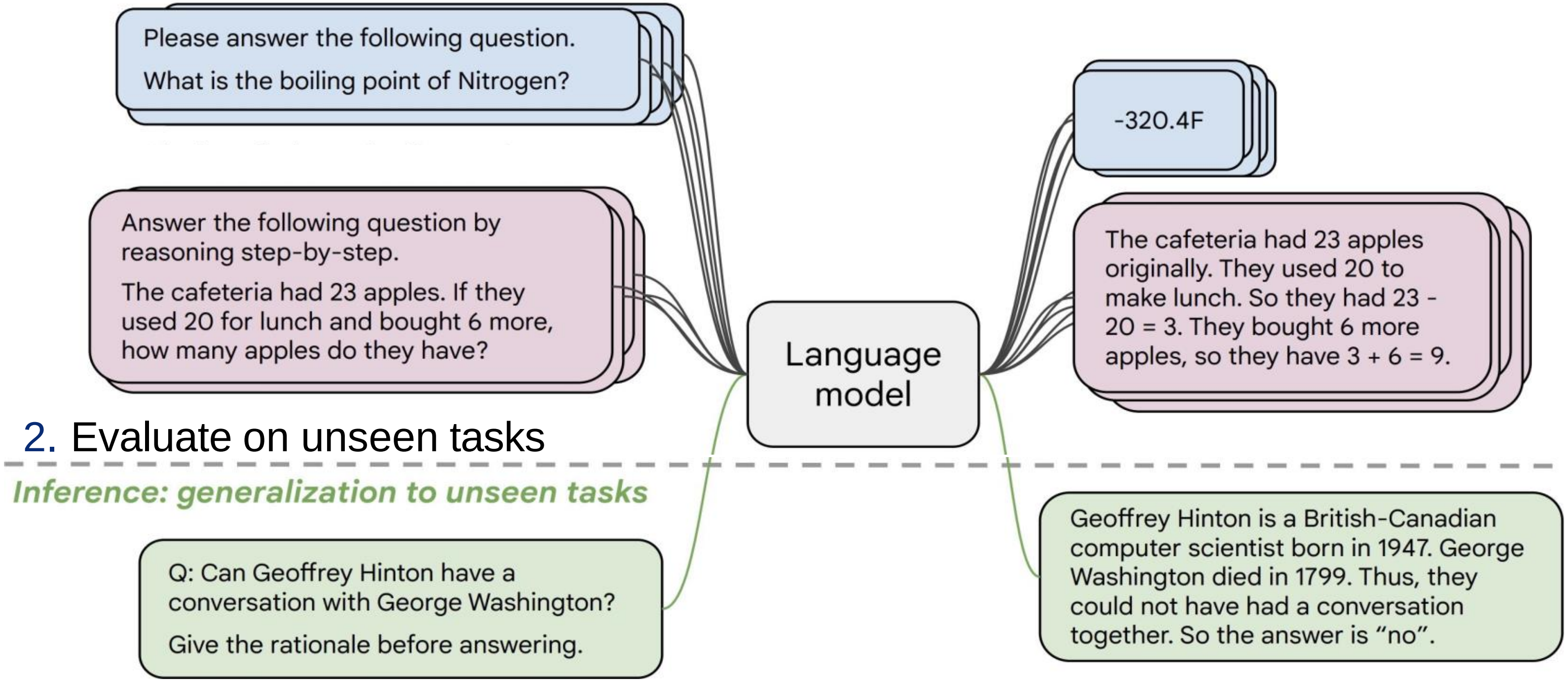
```
Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.
```

There is a mismatch between LLM pre-training and user intents.

Instruction-Tuning

[Weller et al. 2020. Mishra et al. 2021; Wang et al. 2022, Sanh et al. 2022; Wei et al., 2022, Chung et al. 2022, many others]

1. Collect examples of (instruction, output) pairs across many tasks and finetune an LM



2. Evaluate on unseen tasks

Outline

Recap

Reinforcement Learning

Long Context

Retrieval-based LMs

Why Reinforcement Learning?

- Remember the limits of Instruction-tuning?

  1. Difficult to collect diverse labeled data

  2. Rote learning (token by token) —
     - limited creativity

  3. Agnostic to model's knowledge —
     - may encourage hallucinations

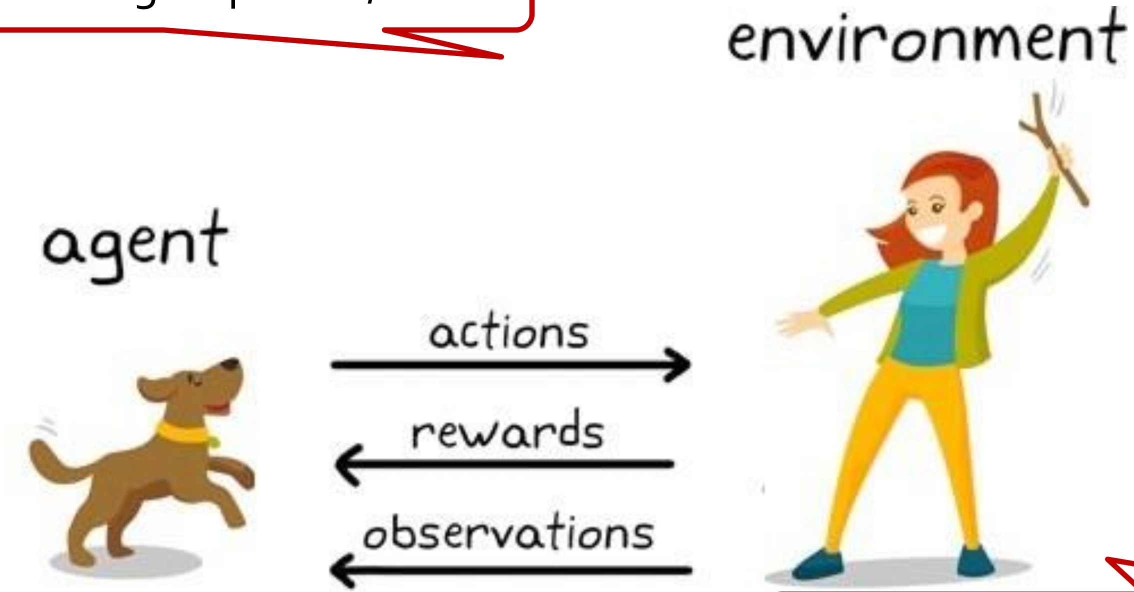
Limited/sparse feedback—usually considered a curse, but now a blessing.

"don't give a man fish rather teach him how to fish by himself"

The model itself should be involved in the alignment loop.

Reinforcement Learning: Intuition

Action here: generating responses/token

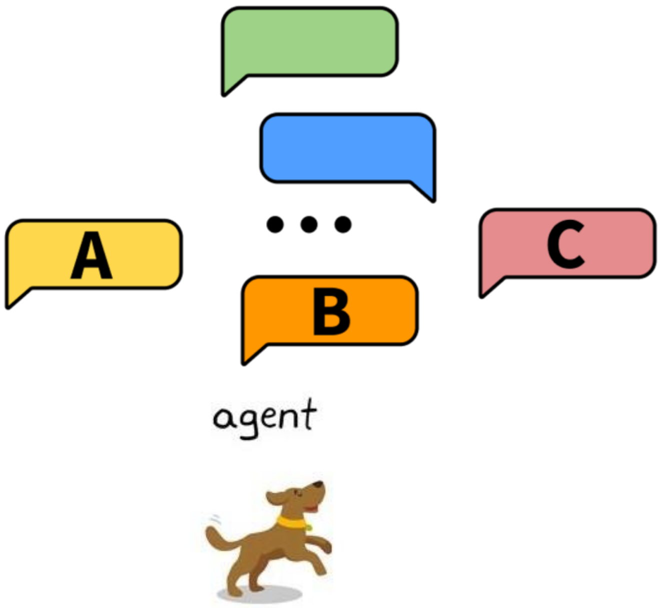


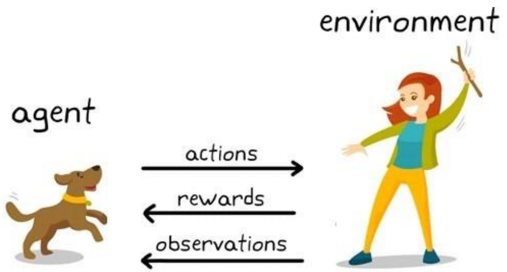
[figure credit]

Reward here: whether humans liked the generation (sequence of actions=tokens)

Intuition

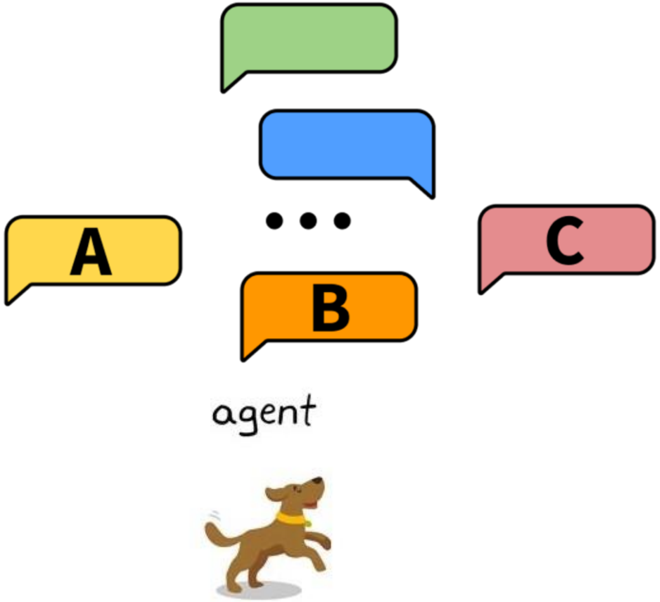
Task: choose the better next message in a conversation

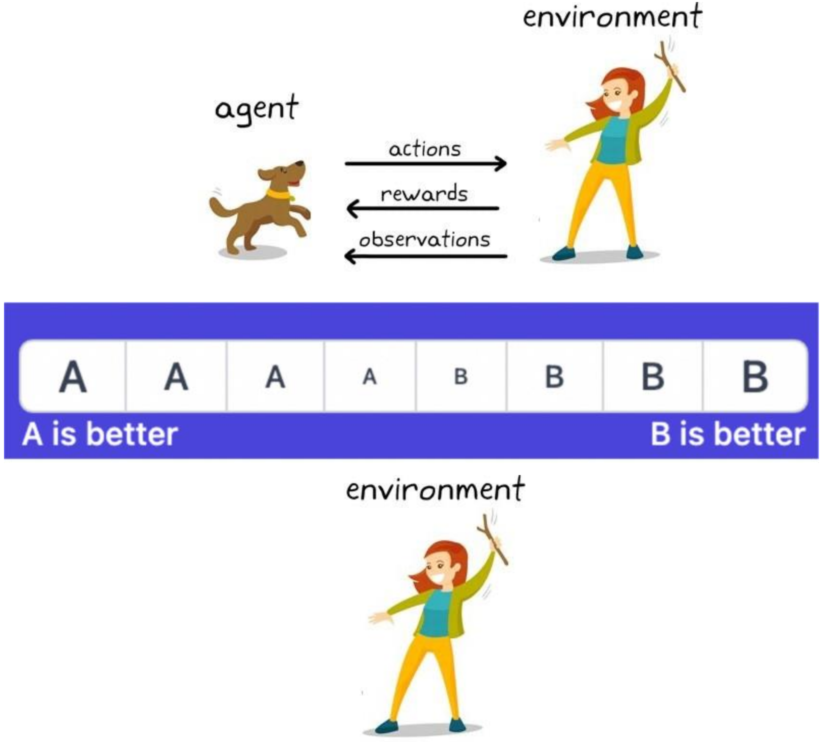




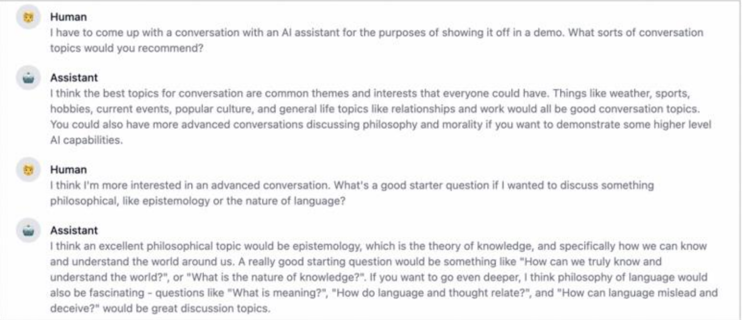
Intuition

Scoring interface: Likert scale or rankings





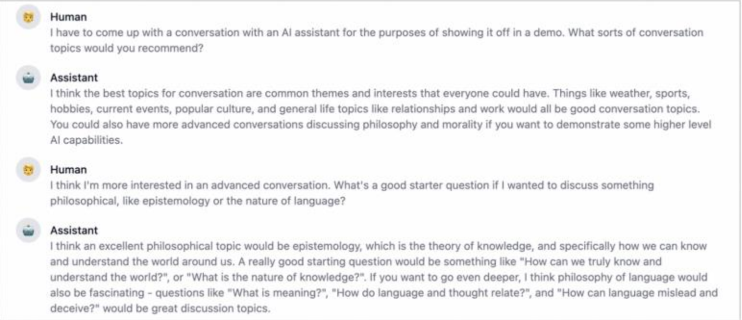
Intuition

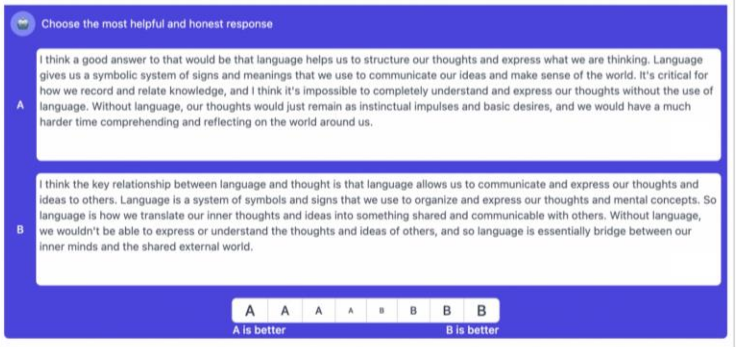


human has conversation with the LLM

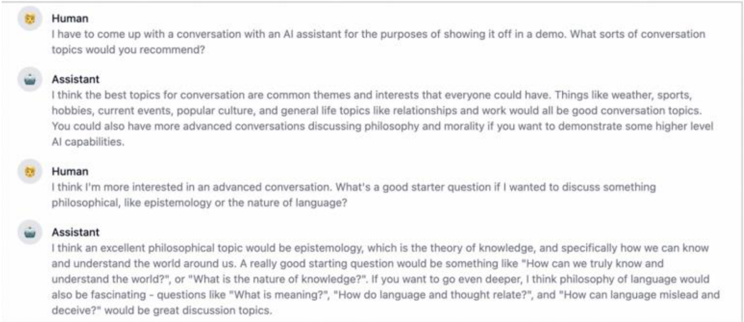
Intuition

LLM provides two options for next responses

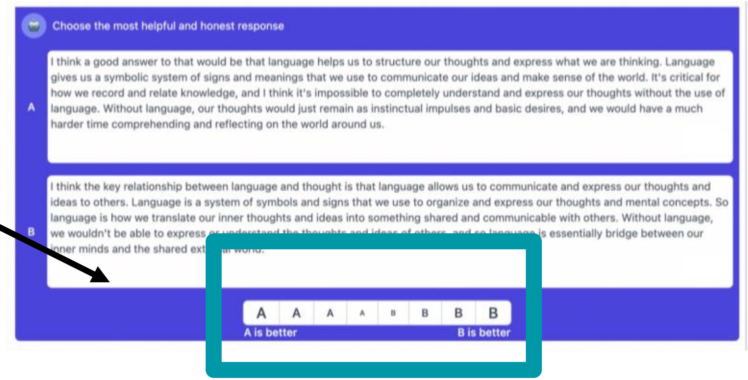




Intuition



human rates better response

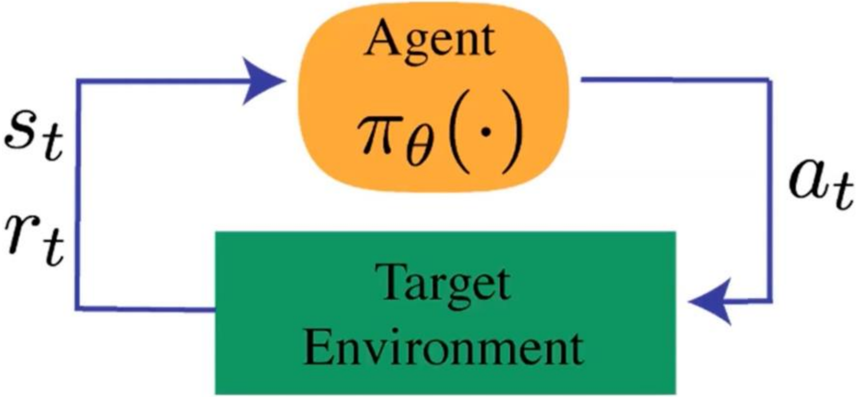


Reinforcement Learning: Abridged History

- The field of reinforcement learning (RL) has studied these (and related) problems for many years now [Williams, 1992; Sutton and Barto, 1998]
- Circa 2013: resurgence of interest in RL applied to deep learning, game-playing [Mnih et al., 2013]
- But there is a renewed interest in applying RL. Why?
  - RL w/ LMs has commonly been viewed as very hard to get right (still is!)
  - We have found successful RL variants that work for language models (e.g., PPO; [Schulman et al., 2017])

Reinforcement Learning: Formalism

- An agent interacts with an environment by taking actions
- The environment returns a reward for the action and a new state (representation of the world at that moment).
- Agent uses a policy function to choose an action at a given state.
- We need to figure out: (1) reward function and (2) the policy function



Some notation:

s_t : state
r_t : reward
a_t : action
$a_t \sim \pi_\theta(s_t)$: policy

Reinforcement Learning from Human Feedback

- Imagine a reward function: $R(s; \text{prompt}) \in \mathbb{R}$ for any output s to a prompt.
- The reward is higher when humans prefer the output.
- Good generation is equivalent to finding reward-maximizing outputs:

Expected reward over the course of sampling from our policy (generative model)

$$\mathbb{E}_{\hat{s} \sim p_\theta}[R(\hat{s}; \text{prompt})]$$

$p_\theta(s)$ is a pre-trained model with params θ we would like to optimize (policy function)

[Slide credit: Jesse Mu]

Reinforcement Learning from Human Feedback

- Imagine a reward function: $R(s; \text{prompt}) \in \mathbb{R}$ for any output s to a prompt.
- The reward is higher when humans prefer the output.
- Good generation is equivalent to finding reward-maximizing outputs:

Expected reward over the course of sampling from our policy (generative model)

$$\mathbb{E}_{\hat{s} \sim p_\theta}[R(\hat{s}; \text{prompt})]$$

$p_\theta(s)$ is a pre-trained model with params θ we would like to optimize (policy function)

- On the notation:
  - "$\mathbb{E}$" here is an empirical expectation (i.e., average).
  - "~" indicates sampling from a given distribution.

Reinforcement Learning from Human Feedback

- Imagine a reward function: $R(s; \text{prompt}) \in \mathbb{R}$ for any output s to a prompt.
- The reward is higher when humans prefer the output
- Good generation is equivalent to finding reward-maximizing outputs:

$$\mathbb{E}_{\hat{s} \sim p_\theta}[R(\hat{s}; \text{prompt})]$$

- What we need to do:
  - (1) Estimate the reward function $R(s; \text{prompt})$.
  - (2) Find the best generative model p_θ that maximizes the expected reward:

$$\hat{\theta} = \text{argmax}_\theta \mathbb{E}_{\hat{s} \sim p_\theta}[R(\hat{s}; \text{prompt})]$$

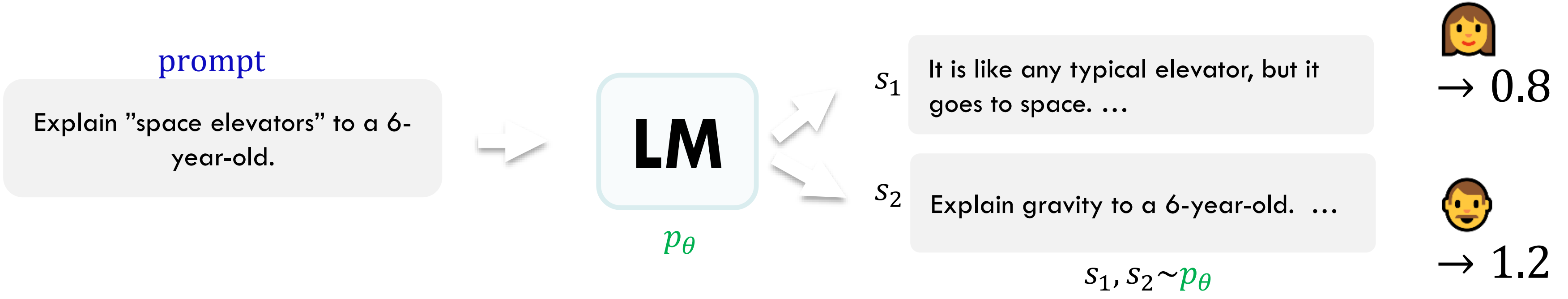
Estimating the Reward R

- Obviously, we don't want to use human feedback directly since that could be 💰💰💰
- Alternatively, we can build a model to mimic their preferences [Knox and Stone, 2009]

Estimating the Reward R

- Obviously, we don't want to use human feedback directly since that could be 💰💰💰
- Alternatively, we can build a model to mimic their preferences [Knox and Stone, 2009]
- Approach 1: get humans to provide absolute scores for each output

Challenge: human judgments on different instances and by different people can be noisy and mis-calibrated!

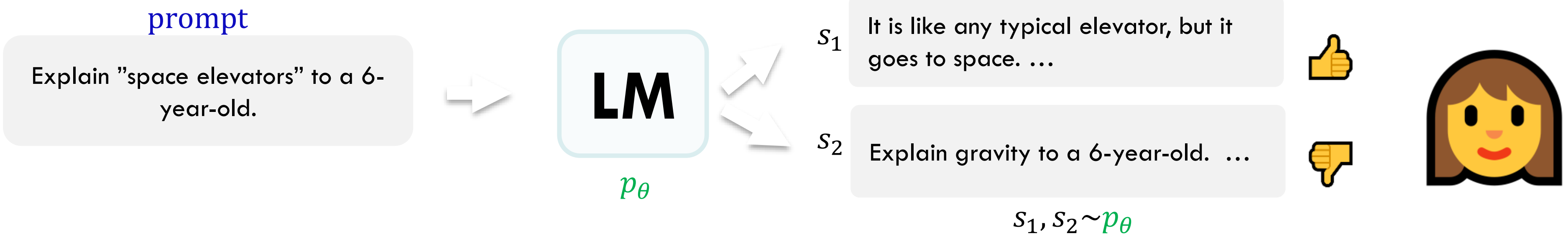


Estimating the Reward R

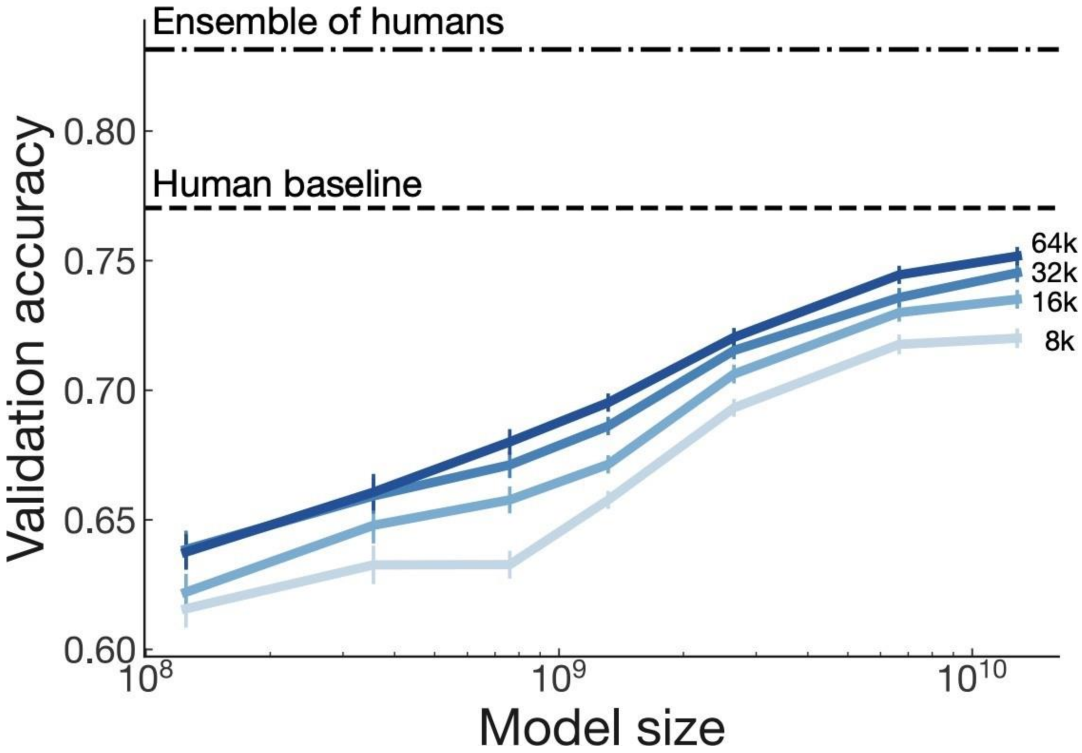
- Obviously, we don't want to use human feedback directly since that could be 💰💰💰
- Alternatively, we can build a model to mimic their preferences [Knox and Stone, 2009]
- Approach 2: ask for pairwise comparisons [Phelps et al. 2015; Clark et al. 2018]

Bradley-Terry [1952] paired comparison model

Pairwise comparison of multiple provides which can be more reliable



Scaling Reward Models

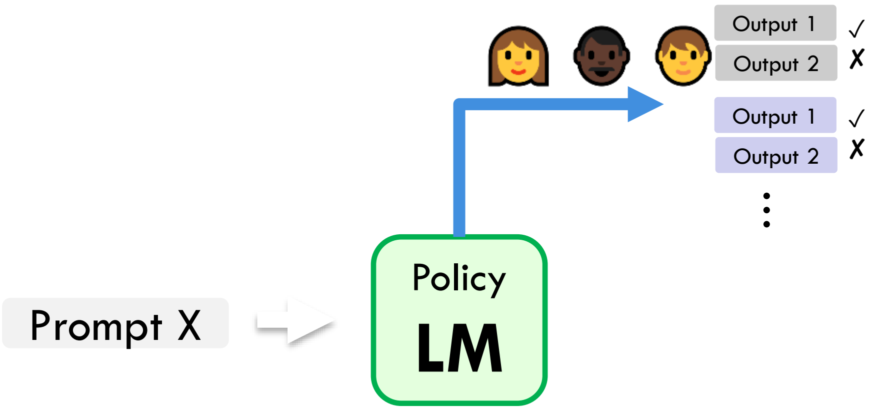


Large enough reward trained on large enough data approaching human performance.

[Stiennon et al., 2020]

Putting it Together

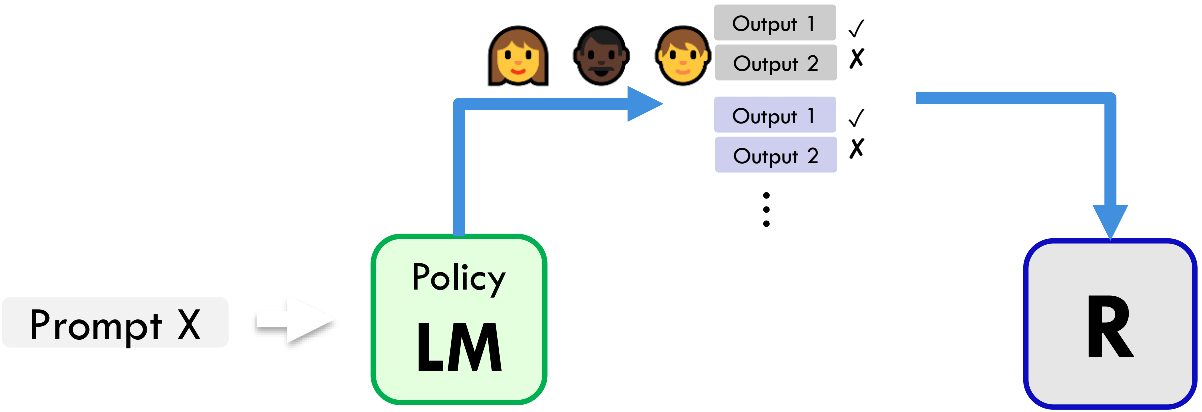
- First collect a dataset of human preferences
  - Present multiple outputs to human annotators and ask them to rank the output based on preferability



Human annotators specify their preferences

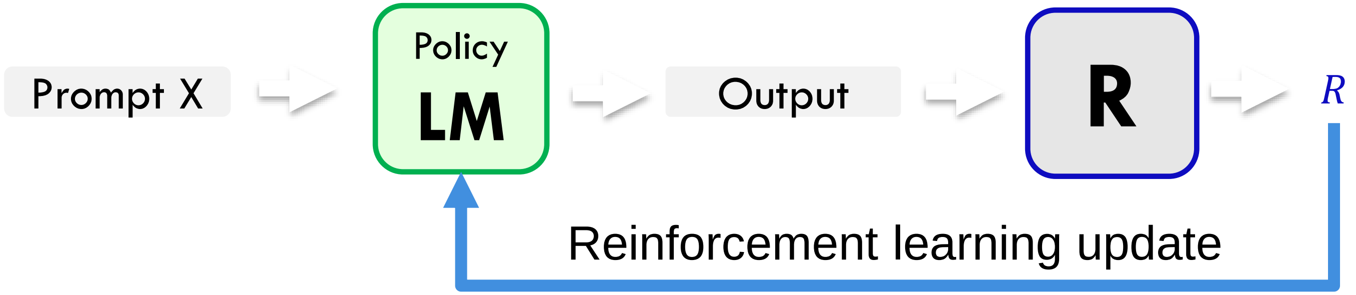
Putting it Together (2)

- Using this data, we can train a reward model
  - The reward model returns a scalar reward which should numerically represent the human preference.



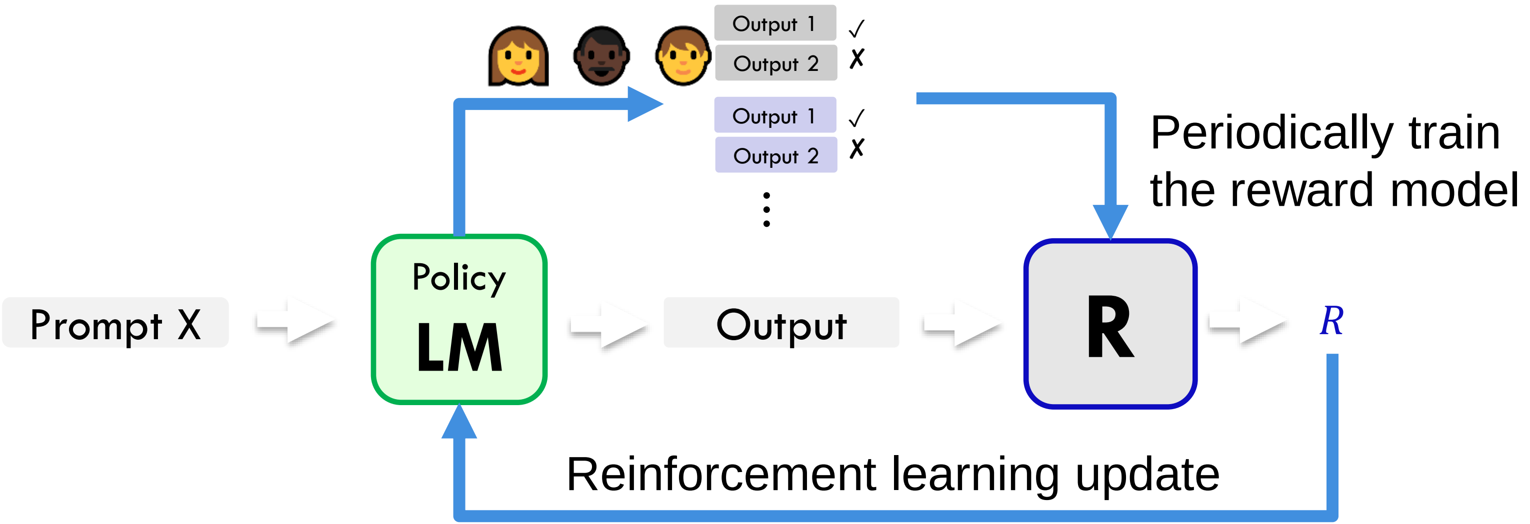
Putting it Together (3)

- We want to learn a policy (a Language Model) that optimizes against the reward model



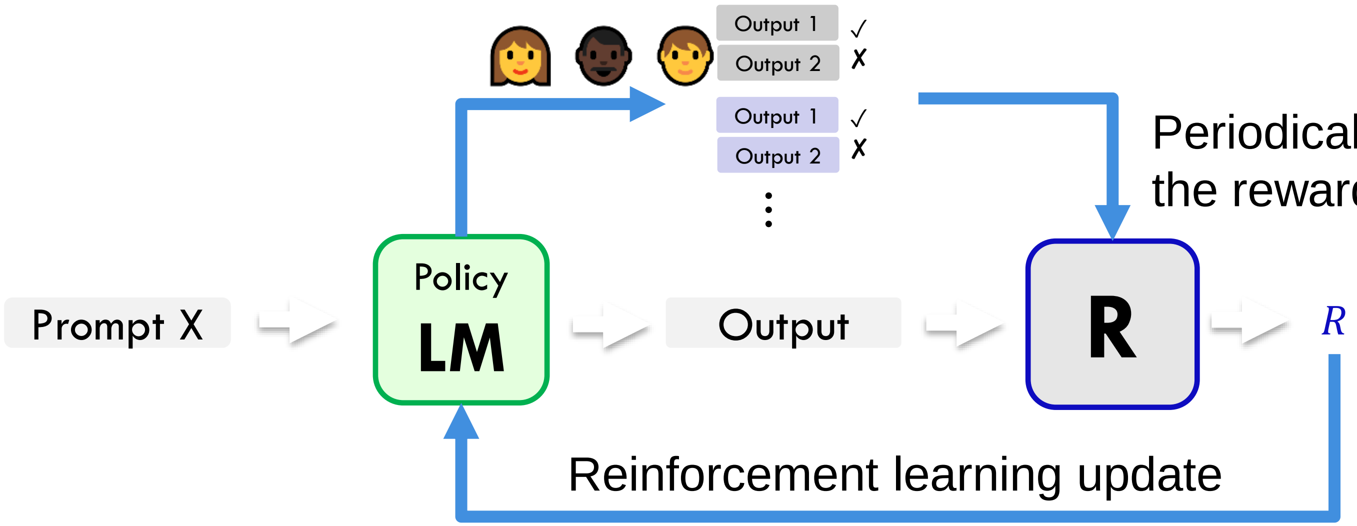
Putting it Together (4)

- Periodically train the reward model with more samples and human feedback



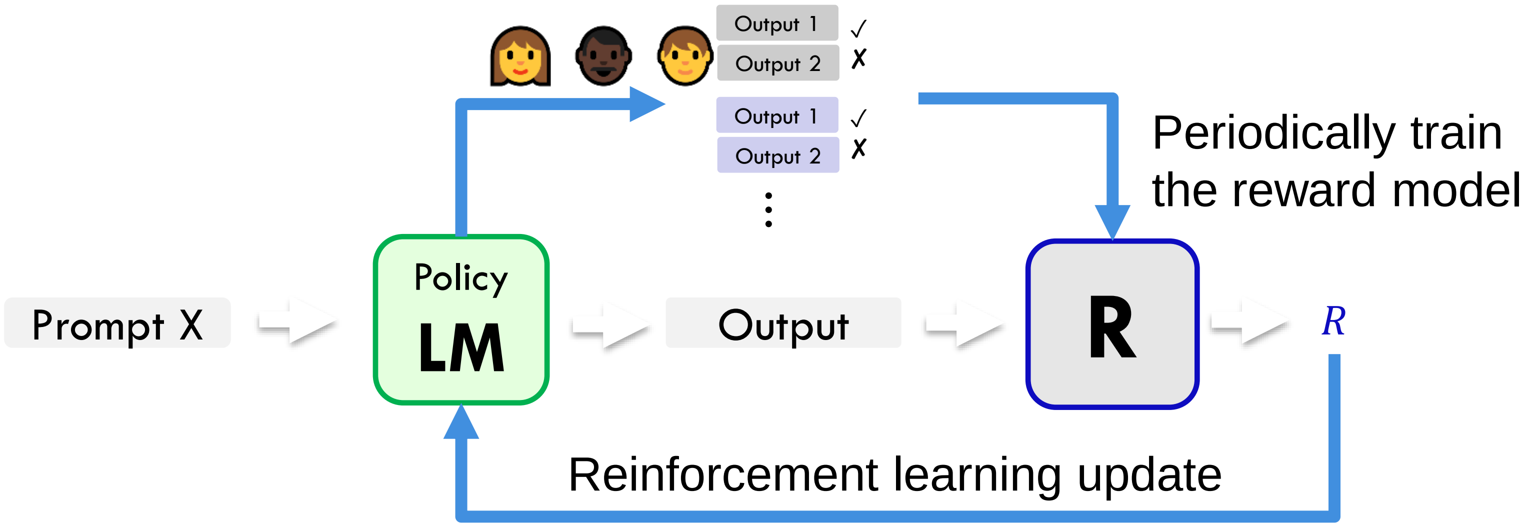
One missing ingredient

- It turns out that this approach doesn't quite work. The policy will learn to "cheat".



One missing ingredient

- Will learn to produce an output that would get a **high** reward but might be **gibberish** or **irrelevant** to the prompt.
- Note, since R is trained on natural inputs, it may not generalize to unnatural inputs.



Regularizing with Pre-trained Model

- **Solution:** add a penalty term that penalizes too much deviations from the distribution of the pre-trained LM.

$$\hat{R}(s;p) \coloneqq R(s;p) - \beta \log\left(\frac{p^{RL}(s)}{p^{PT}(s)}\right)$$

pay a price when $p^{RL}(s) < p^{PT}(s)$

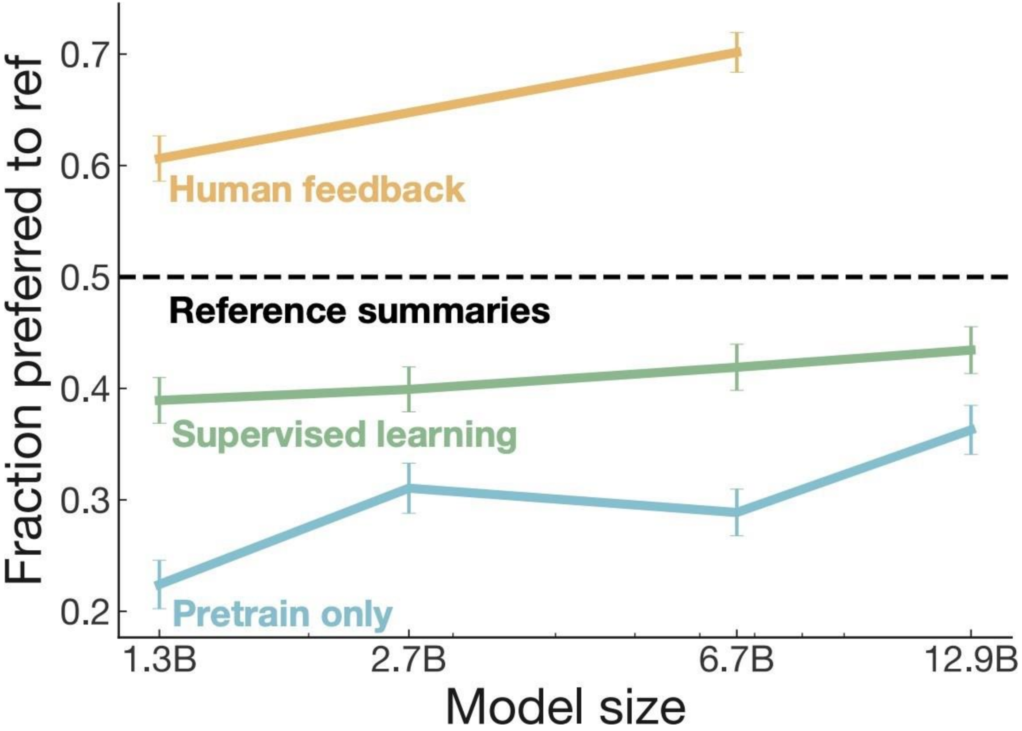
- This prevents the policy model from diverging too far from the pretrained model.

The overall recipe

Pretrain → Align (instruct-tune) → Align (RLHF)

Reinforcement-Learning from Human Feedback

RLHF Gains over Instruction-Tuning



[Stiennon et al., 2020]

GPT3 vs. InstructGPT3 (RLHF-ed)

PROMPT *Explain the moon landing to a 6 year old in a few sentences.*

COMPLETION GPT-3

```
Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.
```

InstructGPT

```
People went to the moon, and they took pictures of what they saw, and
sent them back to the earth so we could all see them.
```

JOHNS HOPKINS WHITING SCHOOL of ENGINEERING

GPT3 vs. InstructGPT3 (RLHF-ed)

PROMPT *Write a short poem about a wise frog.*

COMPLETION

GPT-3

```
Write a short story in which a character has two different names.

Write a short story in which you try to get something back that you
have lost.

Write a short story in which a character has a bad dream.
```

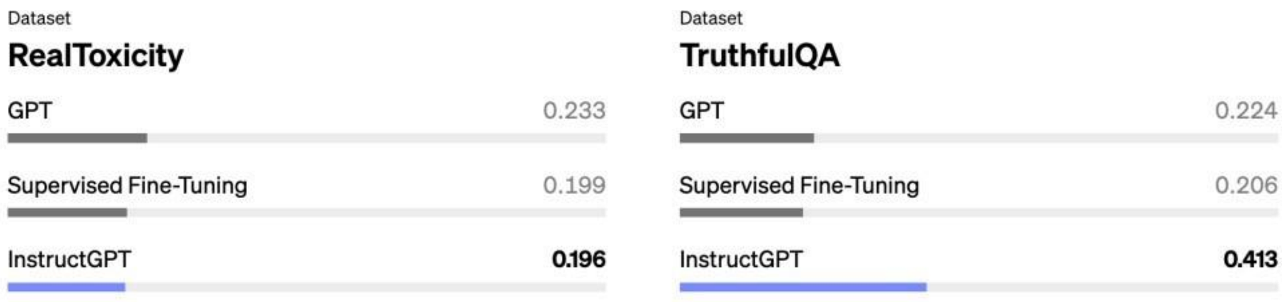
InstructGPT

```
The frog is a symbol of wisdom
He knows all the secrets of the world
He is a master of disguise
And a great teacher of life
He is a symbol of transformation
And the bringer of change
He is the frog who has seen it all
And knows the meaning of it all
```

Can Help with Toxicity and Truthfulness

- Note, reward model can be used to induce any desired behavior as needed:
  - Avoiding bias
  - Avoiding responses outside its scope
  - Avoiding toxicity
  - …

Lower is better



Higher is better

Summary Thus Far

- Reinforcement learning can help mitigate some of the problems with supervised instruction tuning

- Reward model is trained via ranking feedback of humans.
- Regularization to restraint the model from deviating too far from the pre-trained policy

- Limitations:
  - RL can be tricky to get right
  - Training a good reward may require a lot of annotations

Outline

Recap

Reinforcement Learning

Long Context

Retrieval-based LMs

Feeding Lots of Things to LM

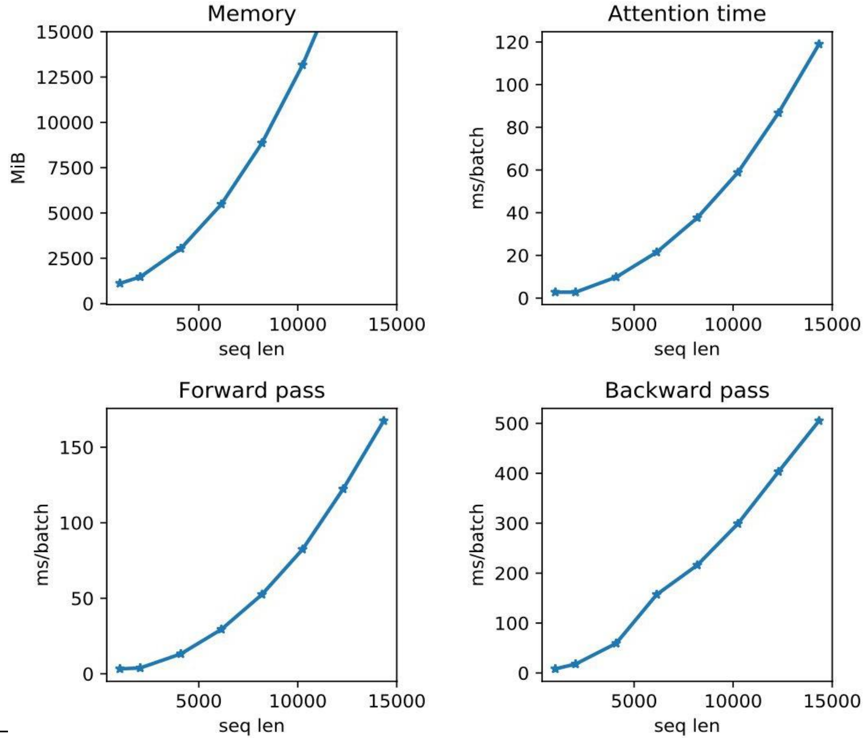
- Books, scientific articles, government reports, videos, your daily experience, etc. they all are much longer than 2k tokens!!

- How do you enable language models process massive amounts of data?

- One approach: just scale up your model—train it on a much longer context window size.
  - The bottleneck: memory usage and number of operations in Self-Attention increases quadratically.

Transformer LMs and Long Inputs

- **Length generalization:** Do Transformers work accurately on long inputs?
- **Efficiency considerations:** How efficient are LMs are long inputs?



Transformer LMs and Long Inputs

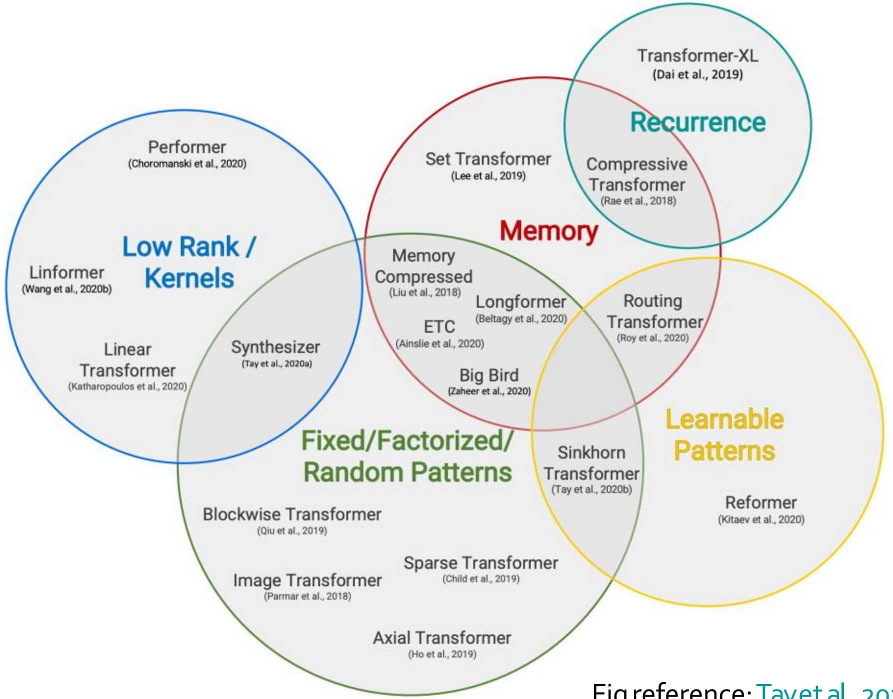
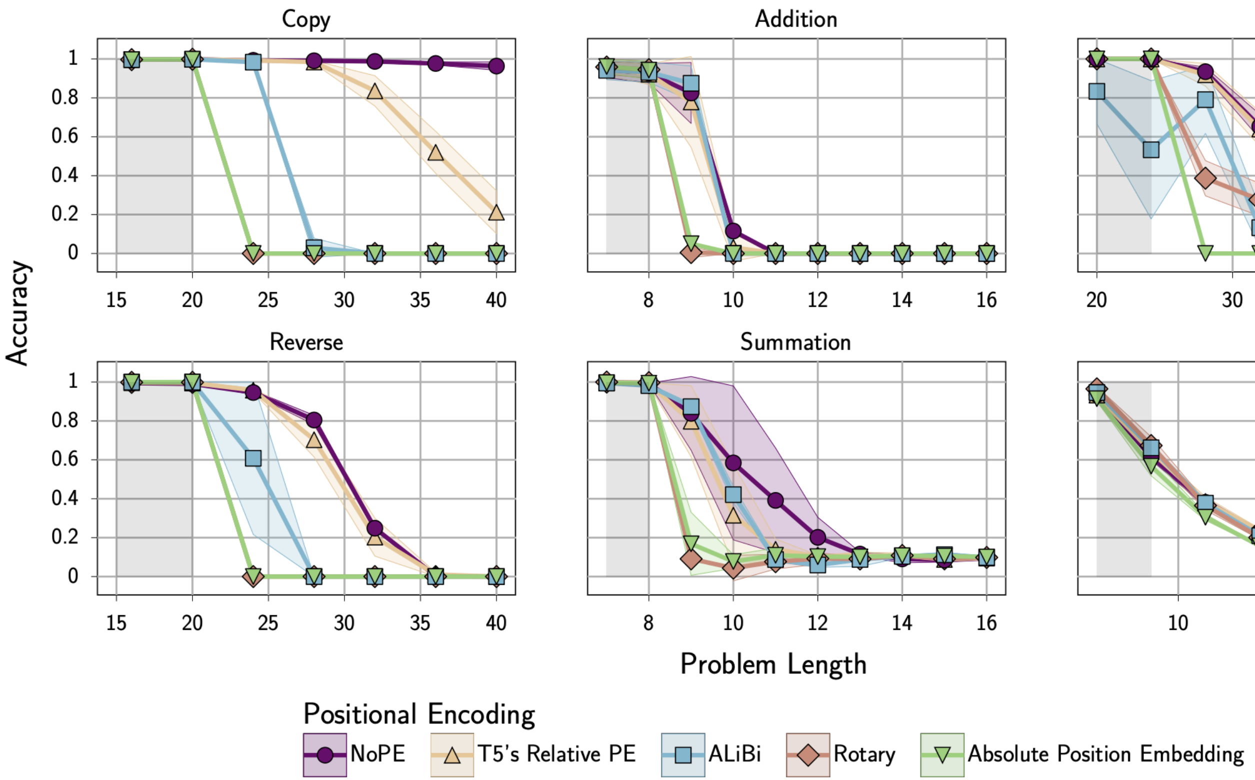


Fig reference: Tay et al., 2020

Length Generalization

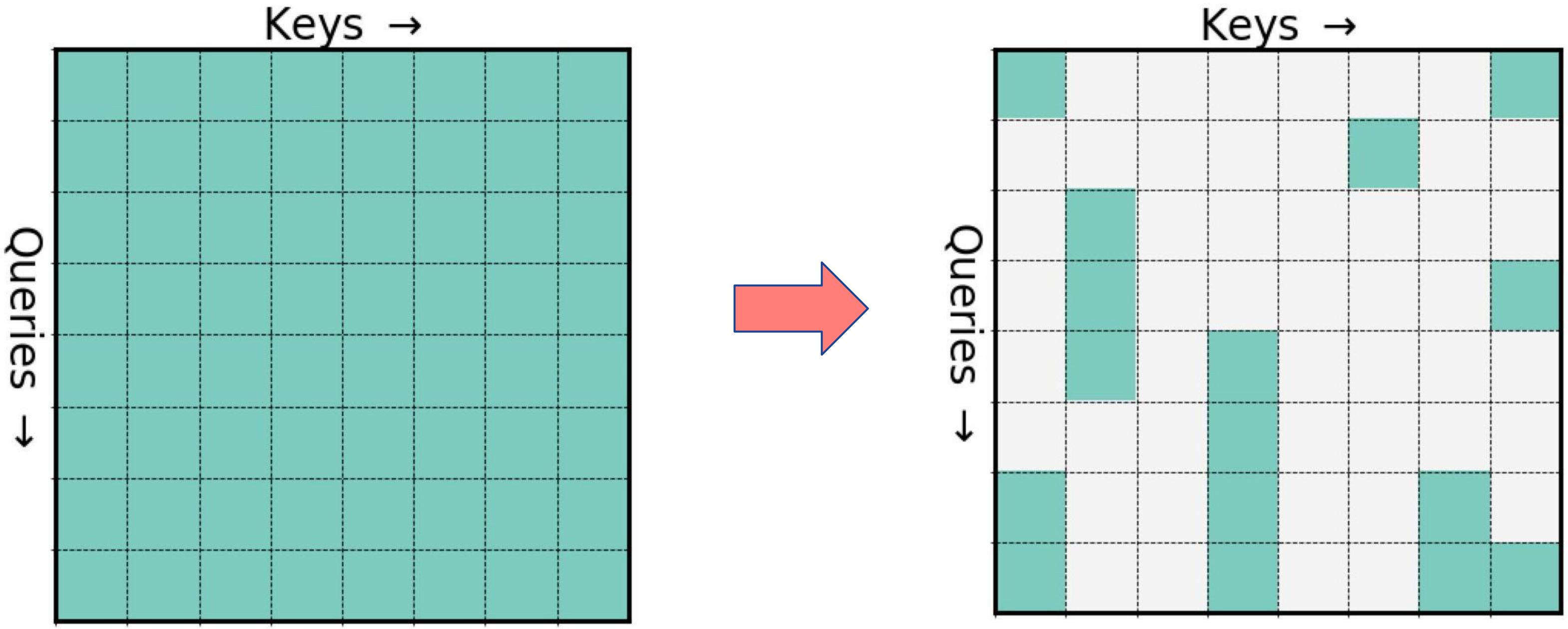


Efficiency consideration:

Sparse Attention Patterns

Sparse Attention Patterns

- The idea is to make the attention operation sparse

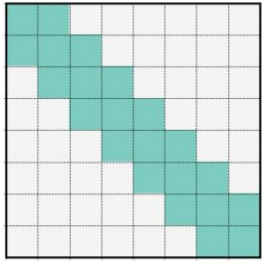


[NAACL 2021 Tutorial Beltagy, Cohan, Hajishirzi, Min, and Peters]

Pre-specified Sparsity Patterns

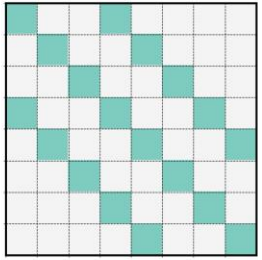
- A variety of patterns has been explored in the past work
  - Longformer (Beltagy et al., 2020), Sparse Transformer (Child et al., 2019), …

Sliding window



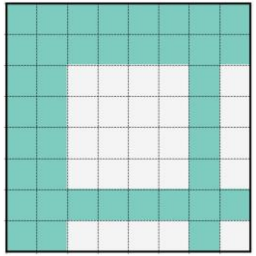
Sparse Transformer
Longformer

Dilated



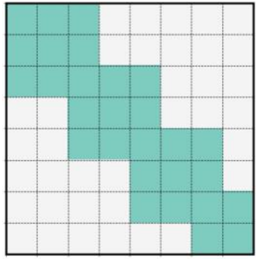
Longformer

Global



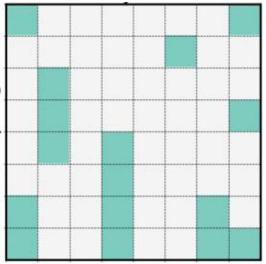
Big Bird

Blocked



Big Bird
Sinkhorn

Random



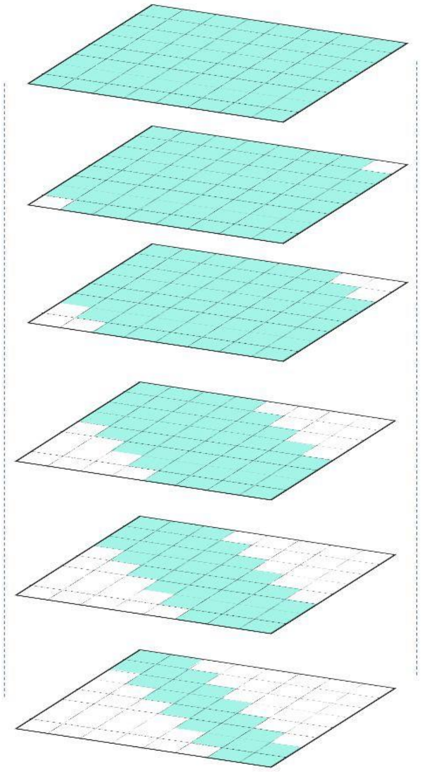
Big Bird

[NAACL 2021 Tutorial Beltagy, Cohan, Hajishirzi, Min, and Peters]

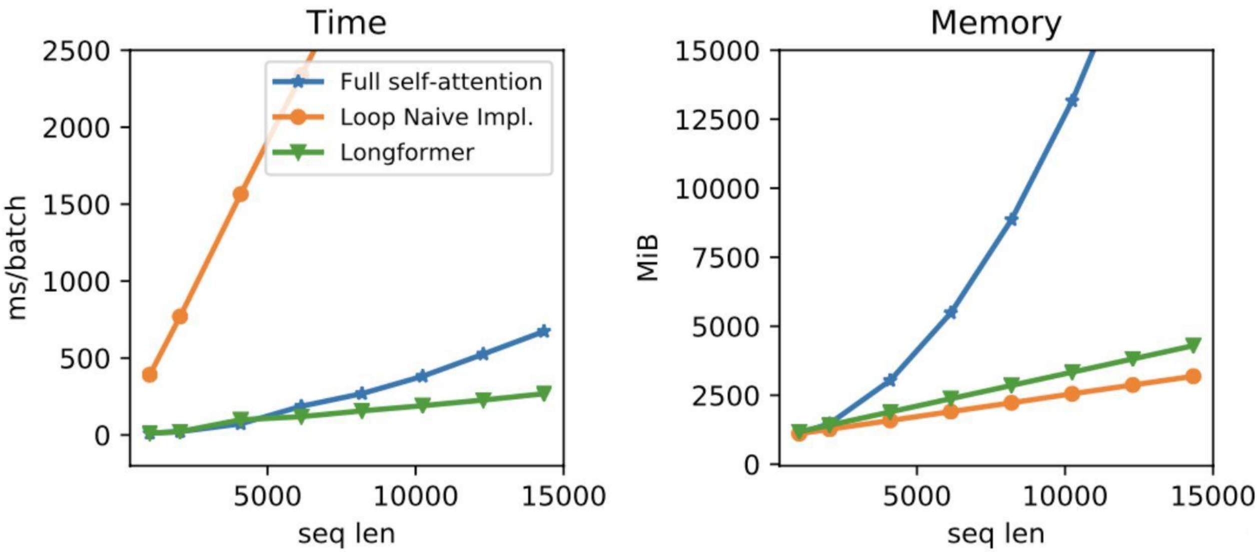
Pre-specified Sparsity Patterns

- Different layers and attention heads can follow different patterns

- A common setup is to have earlier layers with sparser attention pattern.
  - Longformer (Beltagy et al., 2020)



Pre-specified Sparsity Patterns: Computations



[Longformer (Beltagy et al., 2020)]

A Notable Adoption: GPT-3

- Sparse patterns also used in GPT-3 (Brown et al., 2020)

2.1 Model and Architectures

We use the same model and architecture as GPT-2 [RWC+19], including the modified initialization, pre-normalization, and reversible tokenization described therein, with the exception that we use alternating dense and locally banded sparse attention patterns in the layers of the transformer, similar to the Sparse Transformer [CGRS19]. To study the dependence of ML performance on model size, we train 8 different sizes of model, ranging over three orders of magnitude from 125 million parameters to 175 billion parameters, with the last being the model we call GPT-3. Previous work [KMH+20]

Outline

Recap

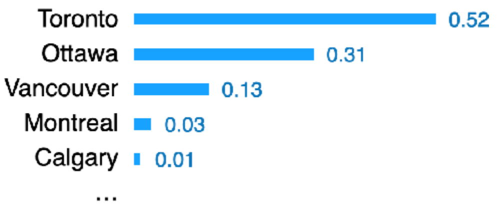
Long Context

Retrieval-based LMs

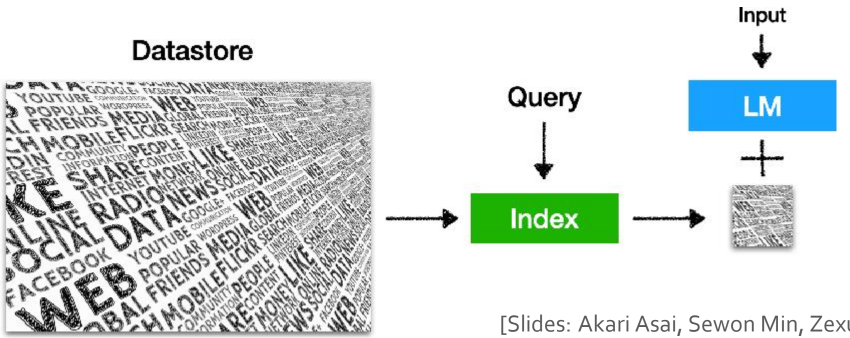
Retrieval-based Language Models

- It is a **language model** $P(x_n | x_1, x_2, \cdots, x_{n-1})$

The capital city of Ontario is ___



- It retrieves from an **external datastore** (at least during inference time)



[Slides: Akari Asai, Sewon Min, Zexuan Zhong, Danqi Chen]

Why Retrieval-based LMs?

LLMs can't memorize all (long-tail) knowledge in their parameters

List 5 important papers authored by Geoffrey Hinton

ChatGPT

Geoffrey Hinton is a renowned computer scientist …
Here are five important papers authored by him:

1. "**Learning Internal Representations by Error Propagation**" (with **D. E. Rumelhart** and **R. J. Williams**) - This paper, published in **1986**, ..

2. "**Deep Boltzmann Machines**" (with **R. Salakhutdinov**) - Published in **2009**, ..

…

4. "**Deep Learning**" (with Y. Bengio and A. Courville) - Published as a book in **2016**,…

5. "**Attention Is All You Need**" (with V. Vaswani, N. Shazeer, et al.) - Published in **2017**, this paper introduced the Transformer model,…

Why Retrieval-based LMs?

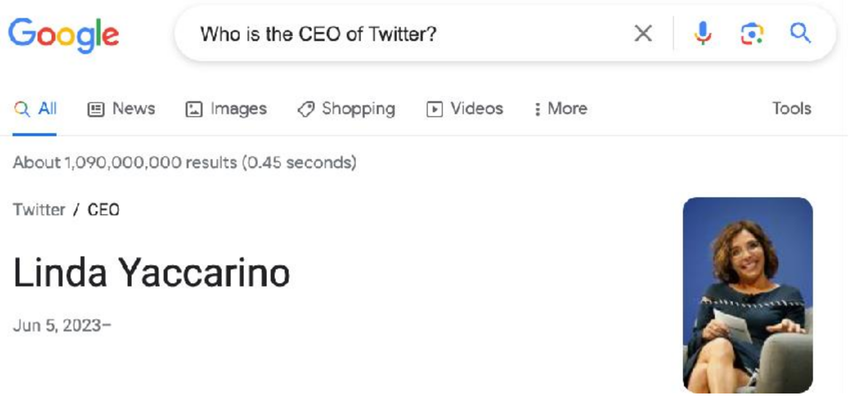
TECHNISCHE UNIVERSITÄT DARMSTADT

LLMs’ knowledge is easily outdated and hard to update

Who is the CEO of Twitter?

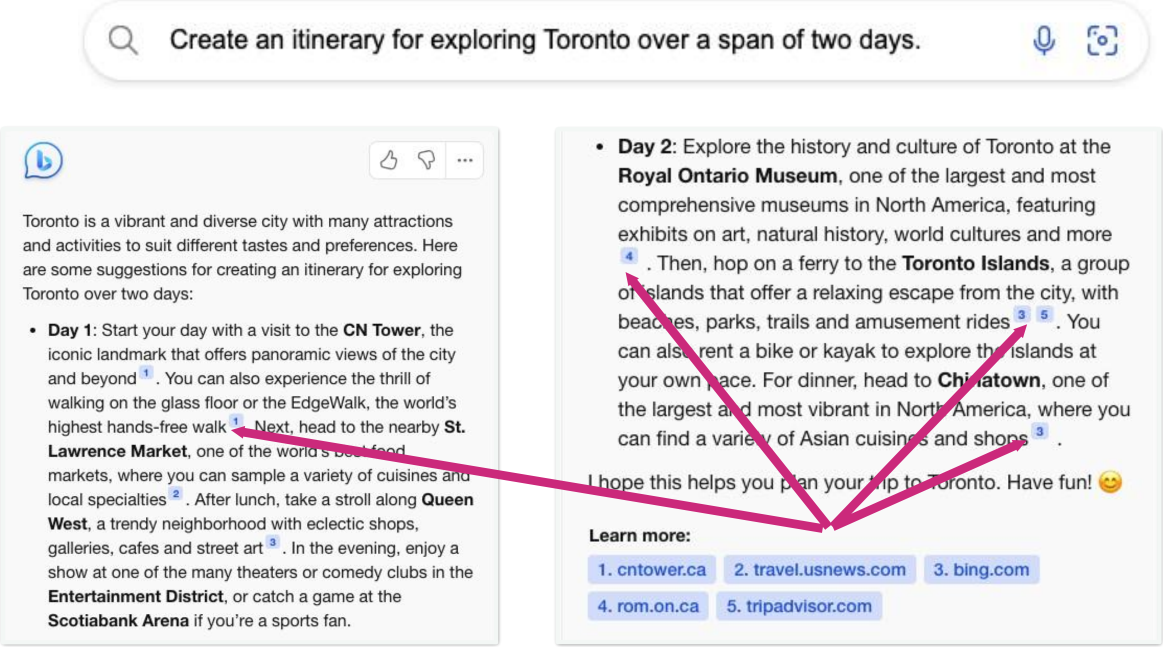
ChatGPT

As of my **knowledge cutoff in September 2021**, the CEO of Twitter is **Jack Dorsey**….



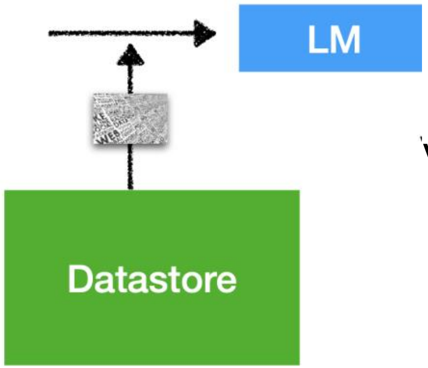
Why Retrieval-based LMs?

LLMs’ output is challenging to interpret and verify



Why Retrieval-based LMs?

LLMs are *large* and expensive to train and run



vs.

Long-term goal: can we possibly reduce the **training** and **inference costs,** and scale down the size of LLMs?

e.g., RETRO (Borgeaud et al., 2021): “obtains comparable performance to GPT-3 on the Pile, despite using **25x fewer parameters**”

What are the Key Design Questions?

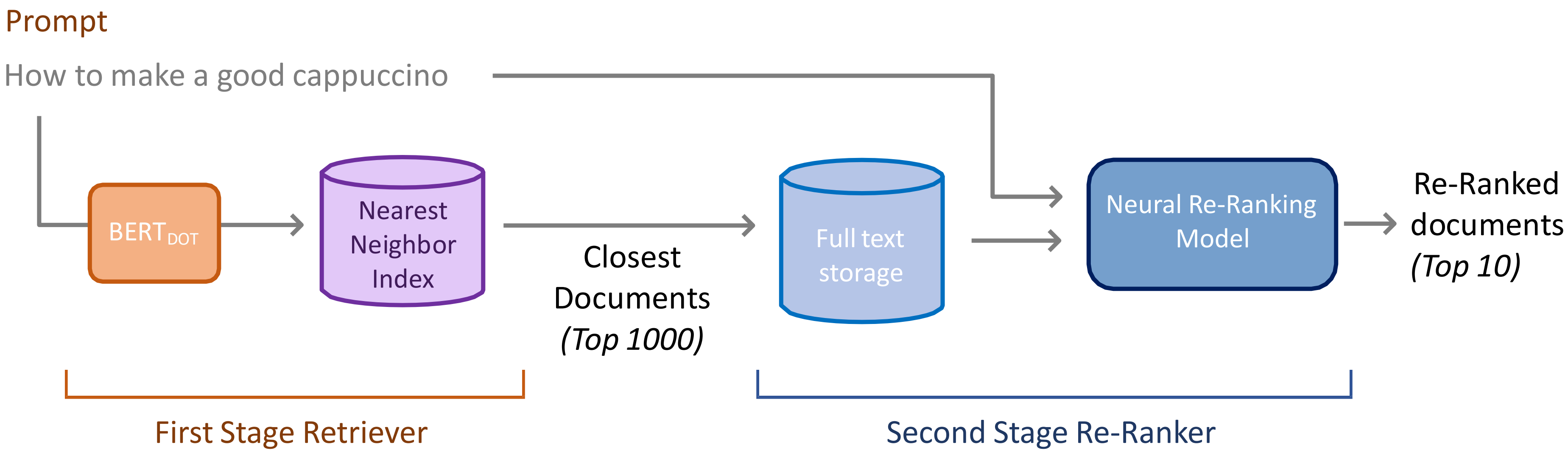
- What are your memories?
  - Documents, database records, training examples, etc.

- How to retrieve memories?
  - Use an off-the-shelf search engine (e.g. Google, StackOverflow).
  - How to train your own memory retriever.

- How to use retrieved memories?
  - "Text fusion"
  - Common failure modes:
    - Underutilization: model ignores retrieved memories.
    - Overreliance: model depends too much on memories!

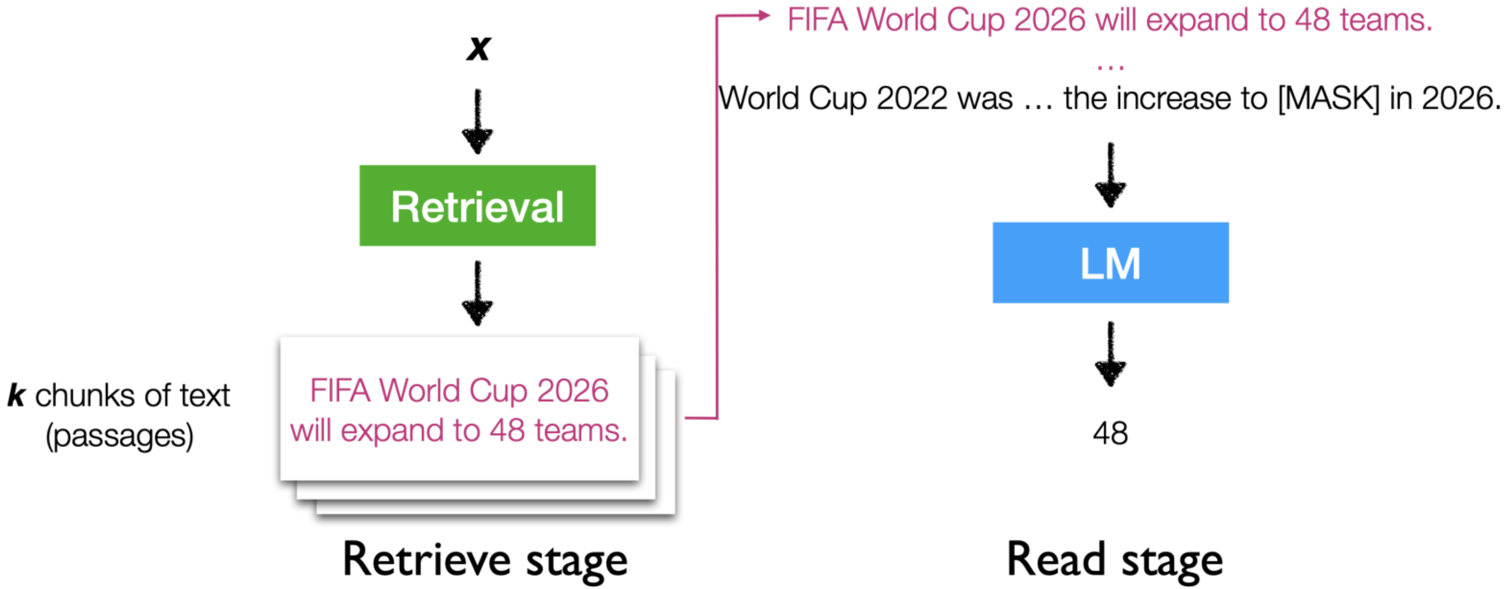
Anatomy of a Neural Retreiver

Remember our IR lectures? (especially neural retrieval and approximate NN)



Retrieval-Augmented LM

- ***x*** = World Cup 2022 was the last before the increase to [MASK] in the 2026 tournament.

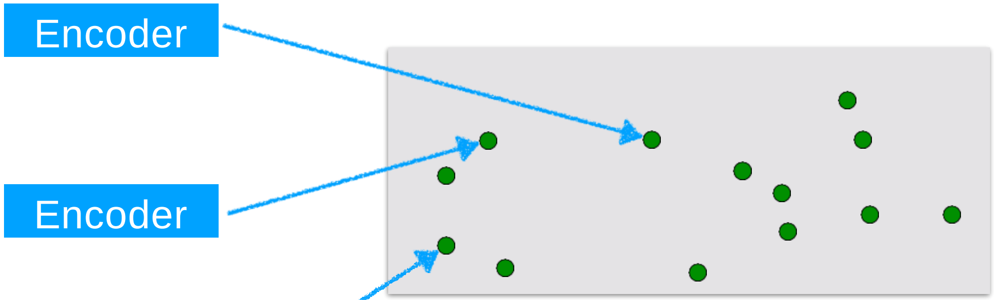


Retrieval-Augmented LM

FIFA World Cup 2026 will expand to 48 teams.

In 2022, the 32 national teams involved in the tournament.

Team USA celebrated after winning its match against Iran …



$$\mathbf{z} = \text{Encoder}(z)$$

Wikipedia
13M chunks (passages)
(called *documents* in the paper)

Retrieval-Augmented LM

$\boldsymbol{x}$ = World Cup 2022 was … the increase to [MASK] in 2026.

FIFA World Cup 2026 will expand to 48 teams.

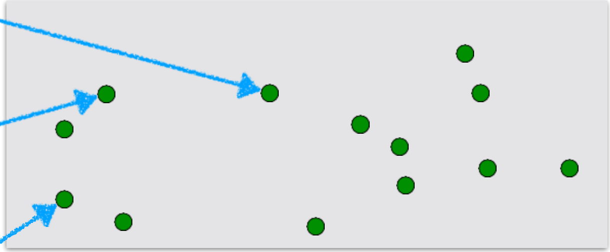
Encoder

In 2022, the 32 national teams involved in the tournament.

Encoder

Team USA celebrated after winning its match against Iran …

Encoder



$\mathbf{z} = \mathrm{Encoder}(z)$

Wikipedia
13M chunks (passages)
(called *documents* in the paper)

Retrieval-Augmented LM

x = World Cup 2022 was … the increase to [MASK] in 2026.

Encoder

FIFA World Cup 2026 will expand to 48 teams.

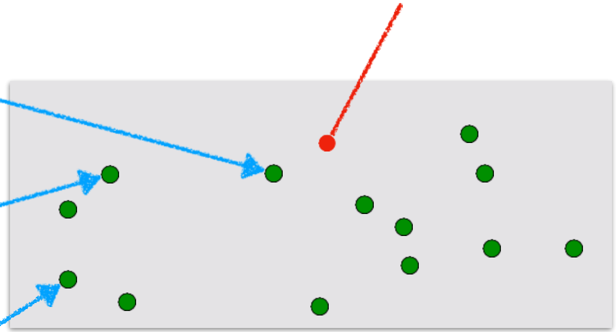
Encoder

In 2022, the 32 national teams involved in the tournament.

Encoder

Team USA celebrated after winning its match against Iran …

Encoder



$\mathbf{z} = \text{Encoder}(z)$

$\mathbf{x} = \text{Encoder}(x)$

Wikipedia
13M chunks (passages)
(called *documents* in the paper)

Retrieval-Augmented LM

TECHNISCHE UNIVERSITÄT DARMSTADT

x = World Cup 2022 was … the increase to [MASK] in 2026.

Encoder

FIFA World Cup 2026 will expand to 48 teams.

Encoder

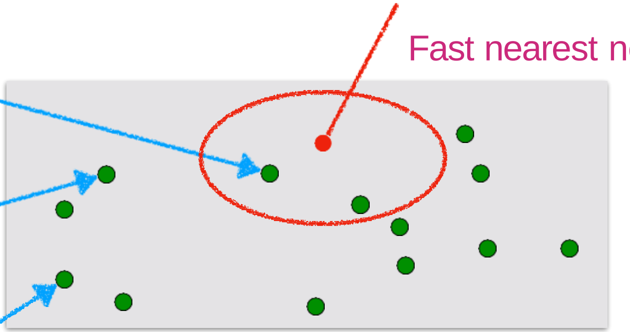
In 2022, the 32 national teams involved in the tournament.

Encoder

Team USA celebrated after winning its match against Iran …

Encoder

Fast nearest neighbor search



$\mathbf{z} = \text{Encoder}(z)$

$\mathbf{x} = \text{Encoder}(x)$

Wikipedia

13M chunks (passages)

(called *documents* in the paper)

Retrieval-Augmented LM

x = World Cup 2022 was … the increase to [MASK] in 2026.

Encoder

FIFA World Cup 2026 will expand to 48 teams.

Encoder

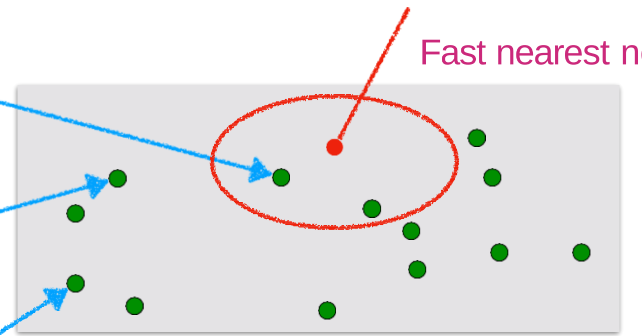
In 2022, the 32 national teams involved in the tournament.

Encoder

Team USA celebrated after winning its match against Iran …

Encoder

Fast nearest neighbor search



$$\mathbf{z} = \text{Encoder}(z)$$

$$\mathbf{x} = \text{Encoder}(x)$$

$$z_1, \ldots, z_k = \text{argTop-}k\,(\mathbf{x} \cdot \mathbf{z})$$

k retrieved chunks

Wikipedia
13M chunks (passages)
(called *documents* in the paper)

Retrieval-Augmented LM: Common Variant

TECHNISCHE UNIVERSITÄT DARMSTADT

What to retrieve?

- **Chunks**
- Tokens
- Others

How to use retrieval?

- **Input layer**
- Intermediate layers
- Output layer

When to retrieve?

- **Once**
- Every n tokens (n>1)
- Every token

Retrieval-Augmented LM: Example Variation

What to retrieve?

- **Chunks**
- Tokens
- Others

How to use retrieval?

- Input layer
- **Intermediate layers**
- Output layer

When to retrieve?

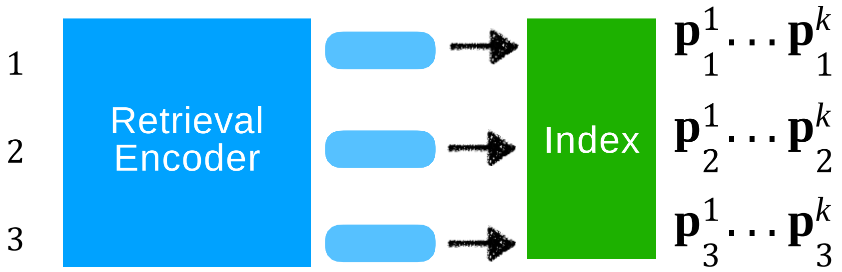
- Once
- **Every *n* tokens (*n*>1)**
- Every token

IR in the Middle of LM

x = World Cup 2022 was the last with 32 teams, before the increase to

1 2 3

(k chunks of text per split)



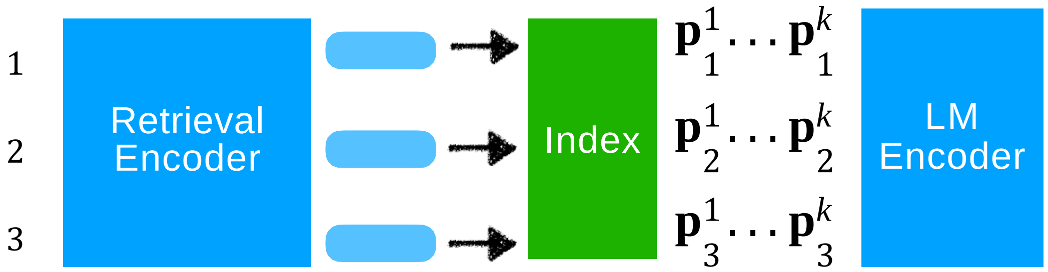
Borgeaud et al. 2021. “Improving language models by retrieving from trillions of tokens”

IR in the Middle of LM

$\boldsymbol{x}$ = World Cup 2022 was the last with 32 teams, before the increase to

1 2 3

(k chunks of text per split)

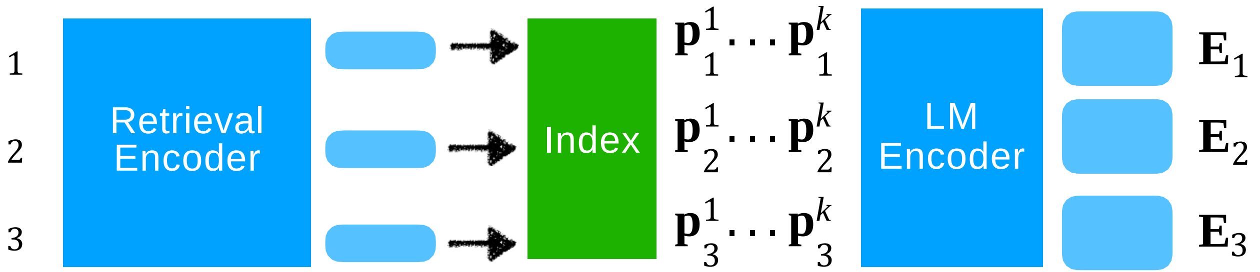


IR in the Middle of LM

x = World Cup 2022 was / the last with 32 teams, / before the increase to

1 2 3

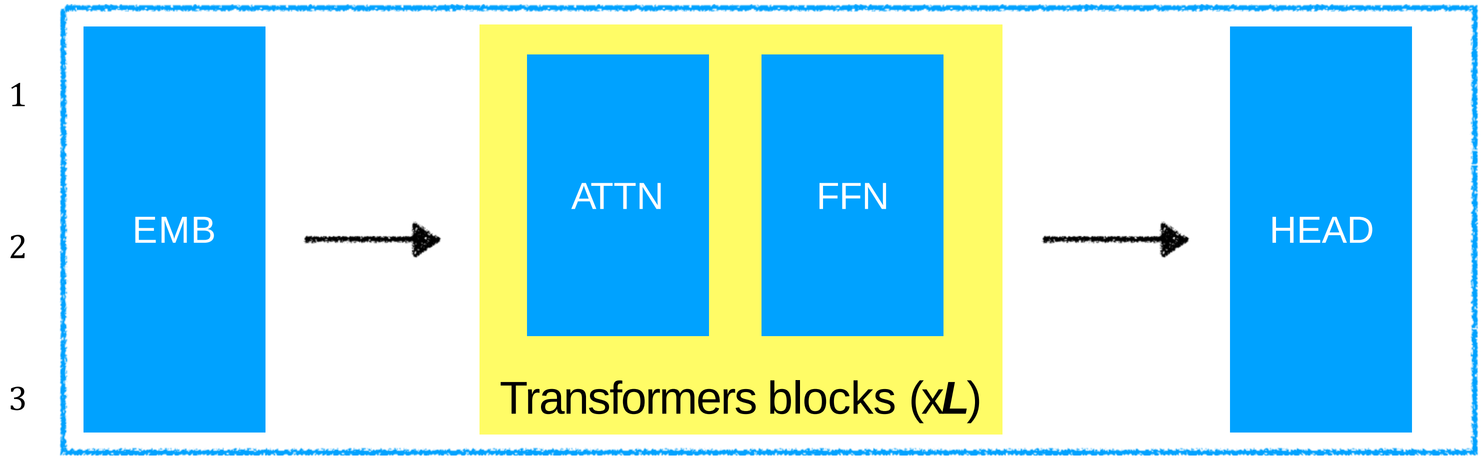
(k chunks of text per split)



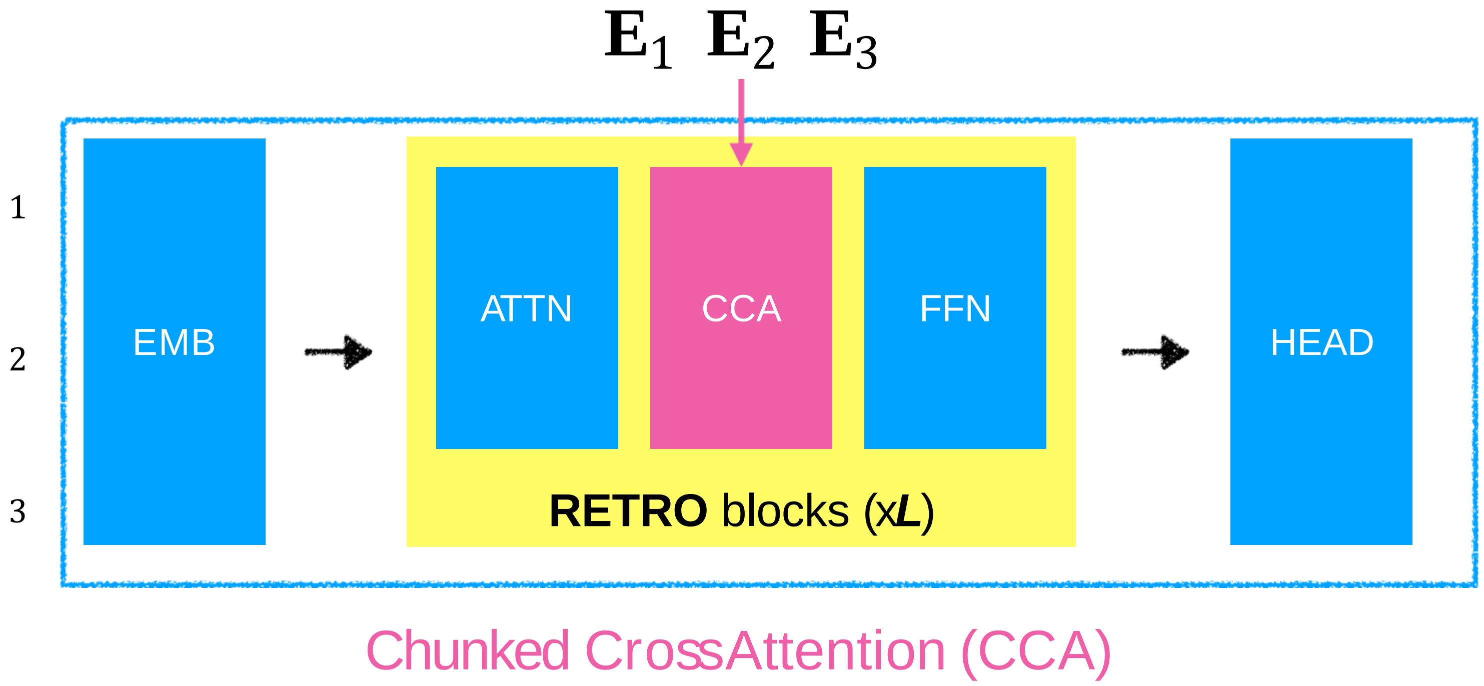
(A $r \times k \times d$ matrix)

(r = # tokens per text chunk)
(d = hidden dimension)
(k = # retrieved chunks per split)

Regular Decoder

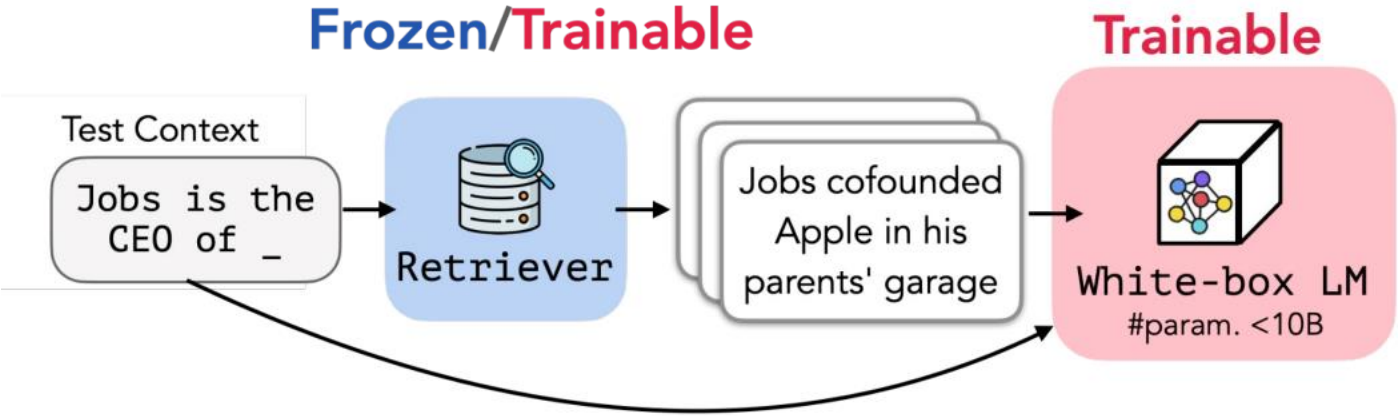


Regular Decoder with IR Embeddings



Training: End-to-end

- There are various ideas in the literature for how to train these models efficiently and in an end-to-end fashion.



Main Takeaways

- How do we enable LMs to utilize external knowledge?
    - Retrieval-augmented language models

- A retriever is a function, f(input, memory) → score

- What we did not discuss:
    - Attribution: Tracing decisions to the source knowledge
    - How to modify the knowledge
    - Conflicting knowledge
    - Editing knowledge
    - More efficient scaling
    - ....