Fall 2024 COS597R:
Deep Dive into Large Language Models

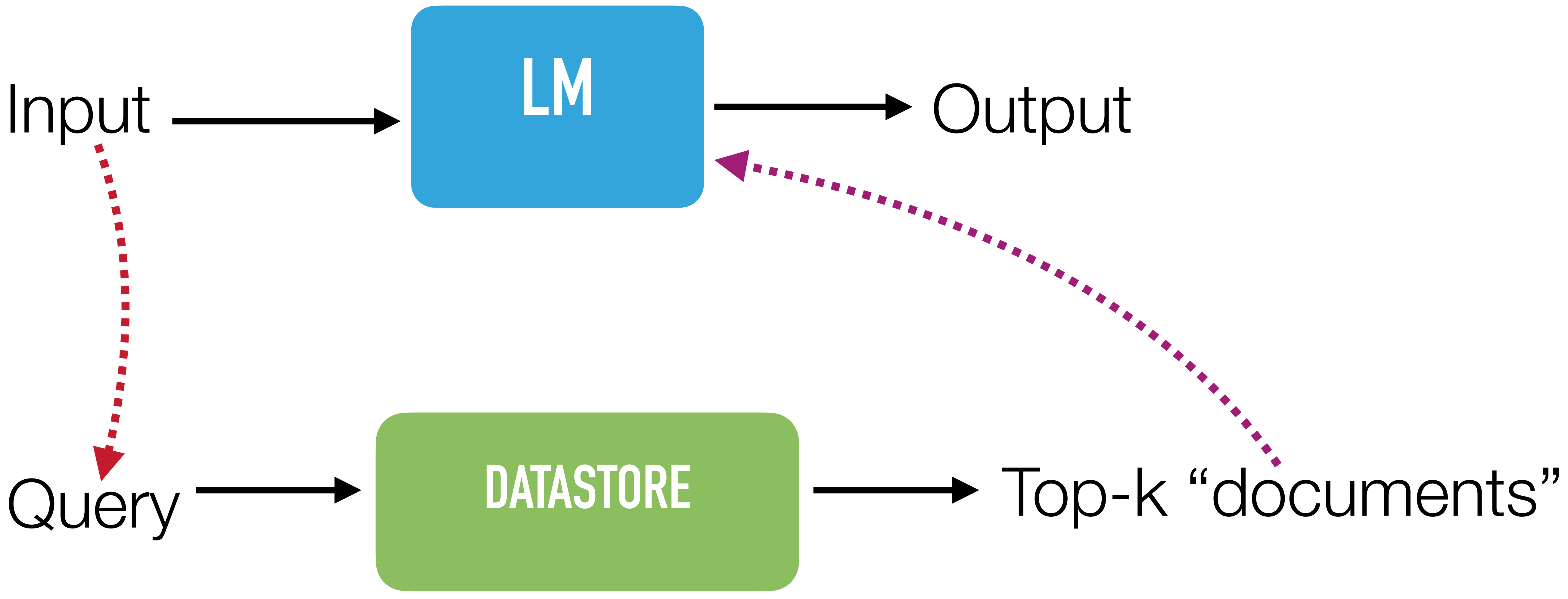
Danqi Chen, Sanjeev Arora

PLI PRINCETON UNIVERSITY

Lecture 18: Retrieval-augmented language models

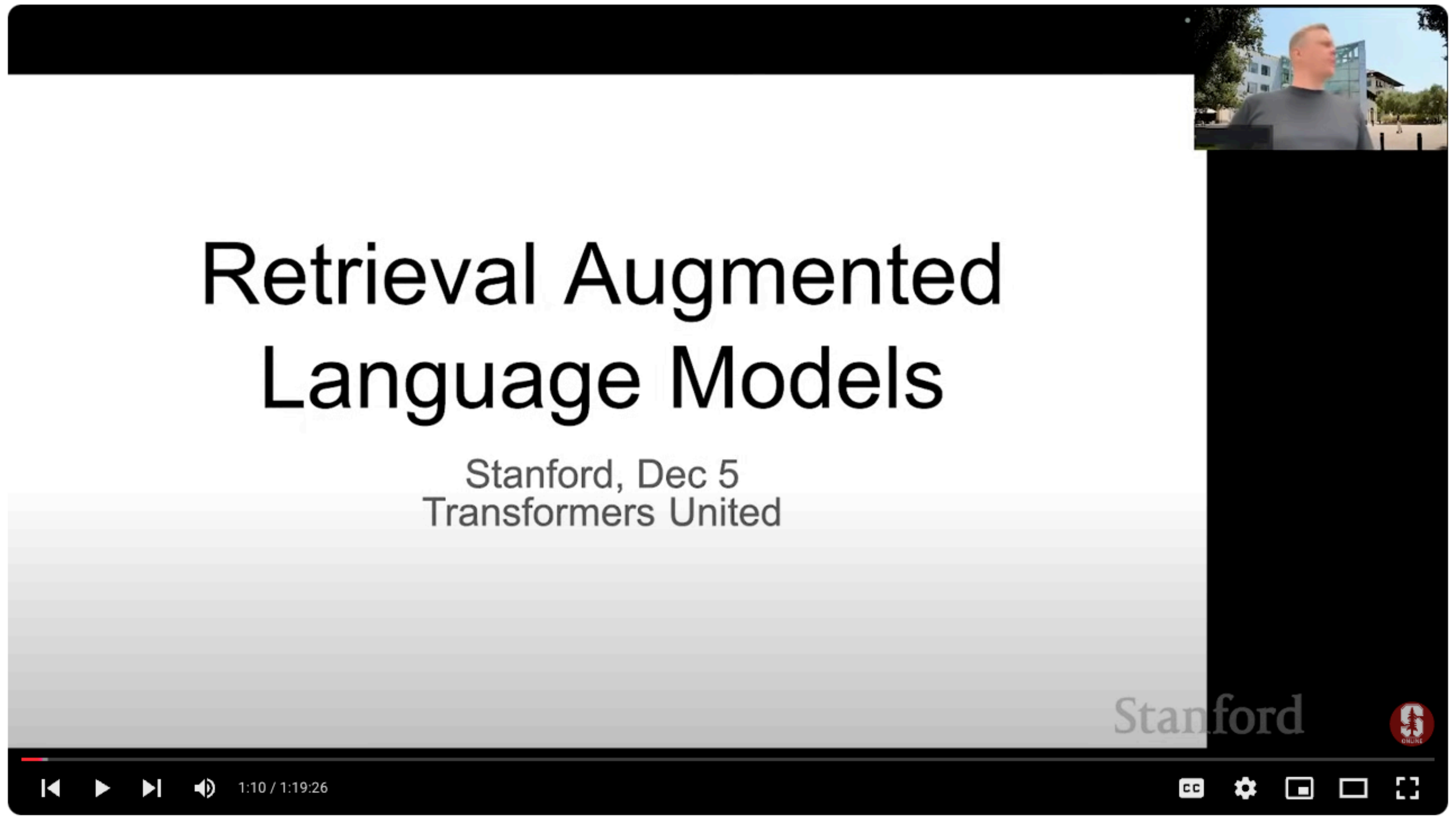
https://princeton-cos597r.github.io/

Retrieval-augmented LMs (RALMs)



- **Datastore**:
  - What should be **stored** in the datastore?
  - How to **index** the datastore (e.g., granularity)?
  - How to **search** top-k documents efficiently?
- How to **integrated** the retrieved outputs with LMs?
- You can also search the datastore in multiple rounds, and you can search the datastore using “output”!

Retrieval-augmented generation (RAG)



Stanford CS25: V3 I Retrieval Augmented Language Models

Douwe Kiela (Stanford CS25; 2023/12)

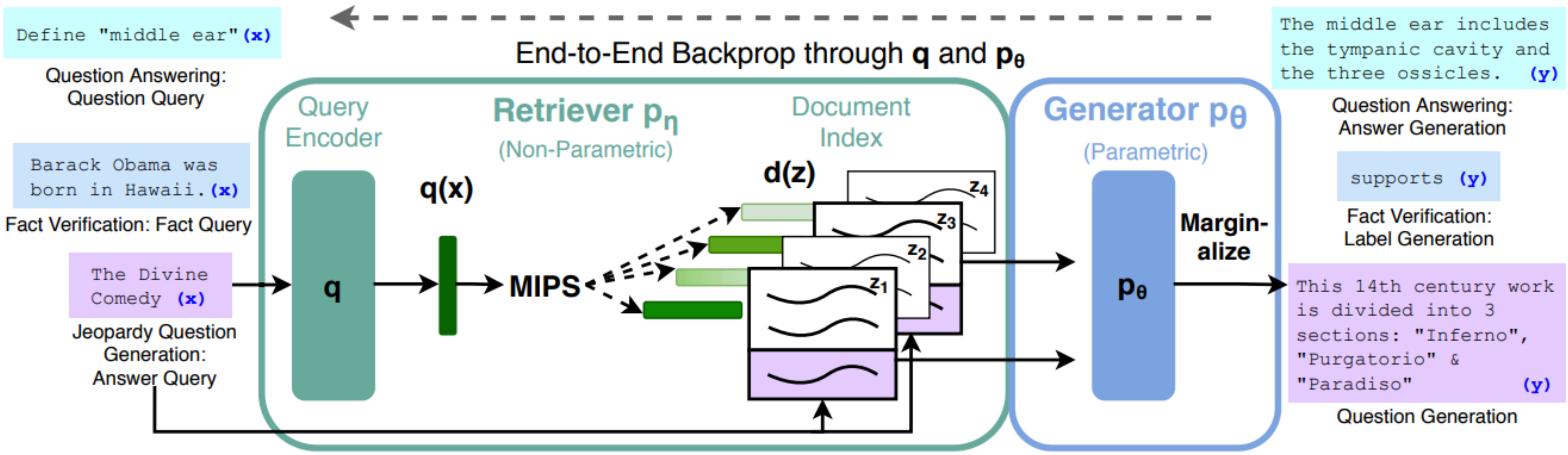
Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

Patrick Lewis[†‡], Ethan Perez[*],

Aleksandra Piktus[†], Fabio Petroni[†], Vladimir Karpukhin[†], Naman Goyal[†], Heinrich Küttler[†],

Mike Lewis[†], Wen-tau Yih[†], Tim Rocktäschel[†‡], Sebastian Riedel[†‡], Douwe Kiela[†]

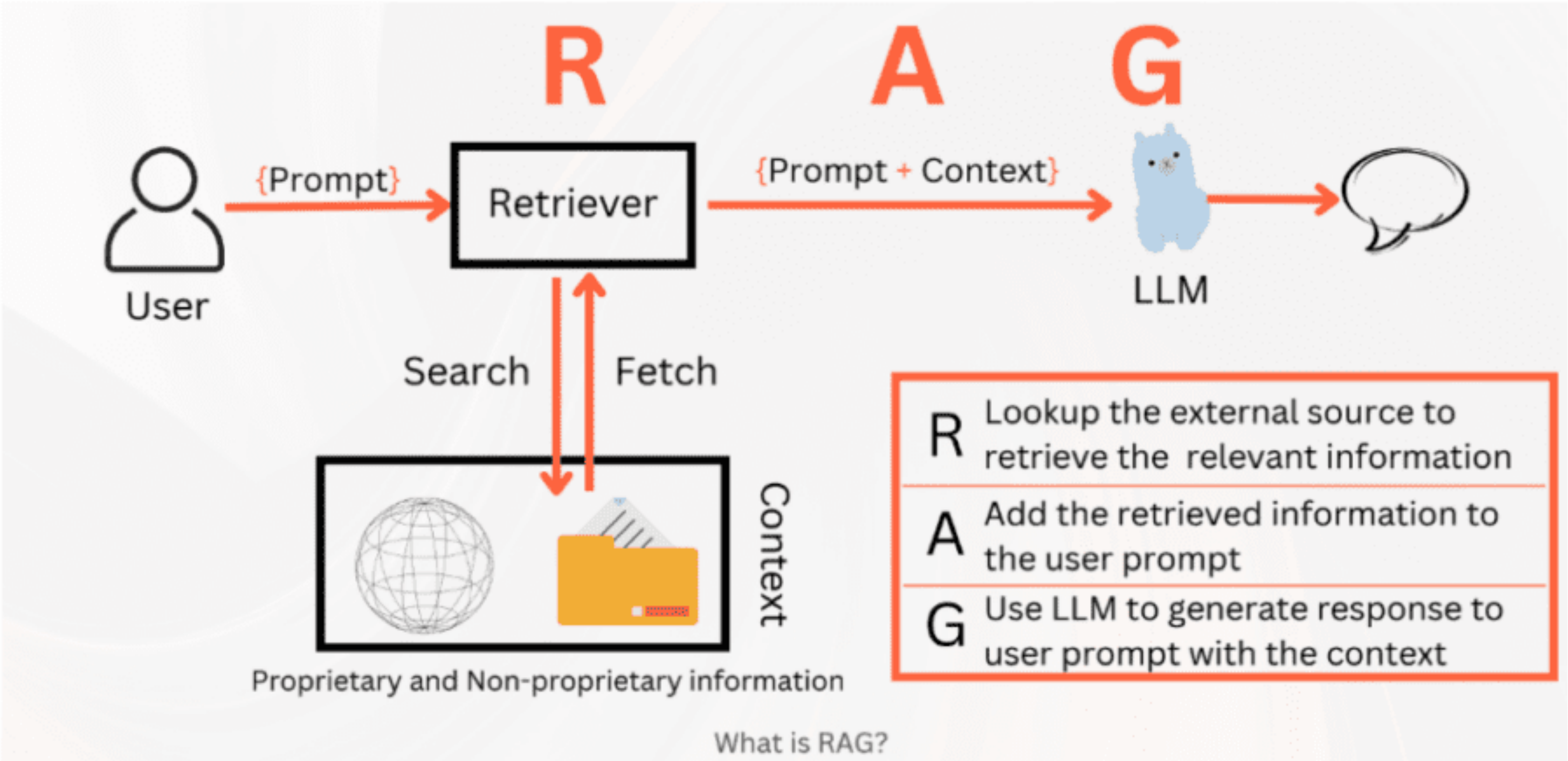
(Lewis et al., NeurIPS'20)



- An **encoder-decoder** model (BART)
- **Retriever**: DPR (for question answering)
- Fine-tuning on **individual** tasks (question answering, question generation, fact verification)

Retrieval-augmented generation (RAG)

Default RAG framework:



https://www.datasciencecentral.com/rag-and-its-evolution/

Retrieval-augmented LMs: two diverging paths

Path #1: Build a language model that has a built-in retrieval component

- You need to consider how to build the datastore/index as part of the model
- The optimal model architecture is still an ongoing exploration
- An alternative of scaling today's parametric (Transformer-based) LMs
- Lots of interesting technical challenges, not as successful as we hoped for 🥹

DeepMind

Improving language models by retrieving from trillions of tokens

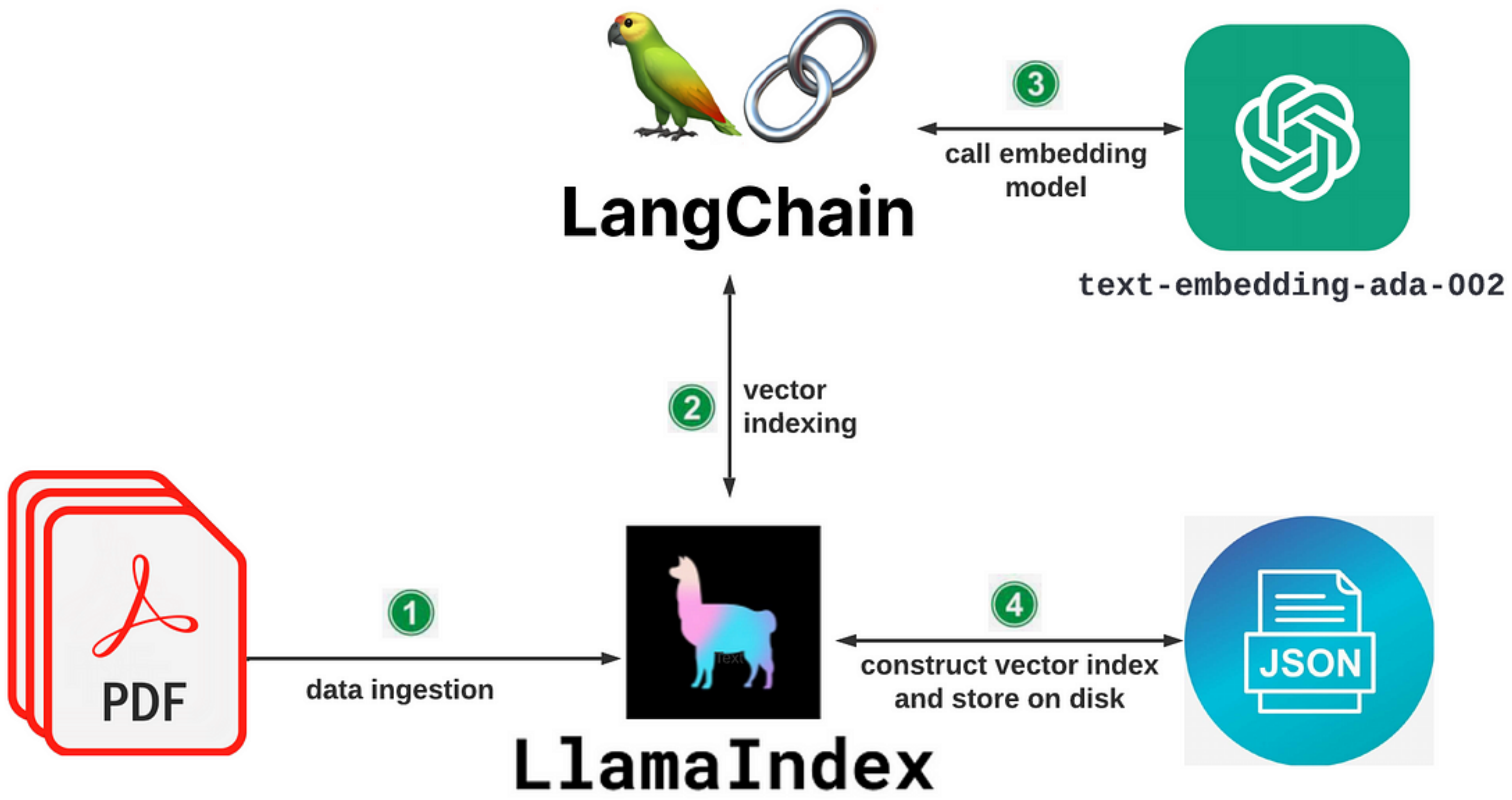
Sebastian Borgeaud[†], Arthur Mensch[†], Jordan Hoffmann[†], Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae[‡], Erich Elsen[‡] and Laurent Sifre[†,‡]

All authors from DeepMind, [†]Equal contributions, [‡]Equal senior authorship

Retrieval-augmented LMs: two diverging paths

Path #2: view retrieval as one of the “tools” that LMs learn how to use

- Assuming you already have a very powerful LM
- Retrieval can be viewed as an API or “black box” e.g., a search engine
- Research questions: when to call the retriever? how to take retrieved results in context?
- Very popular in developer community (“frozen RAG”)



The New England Journal of Medicine is a registered trademark of [QA(“Who is the publisher of The New England Journal of Medicine?”) → Massachusetts Medical Society] the MMS.

Out of 1400 participants, 400 (or [Calculator(400 / 1400) → 0.29] 29%) passed the test.

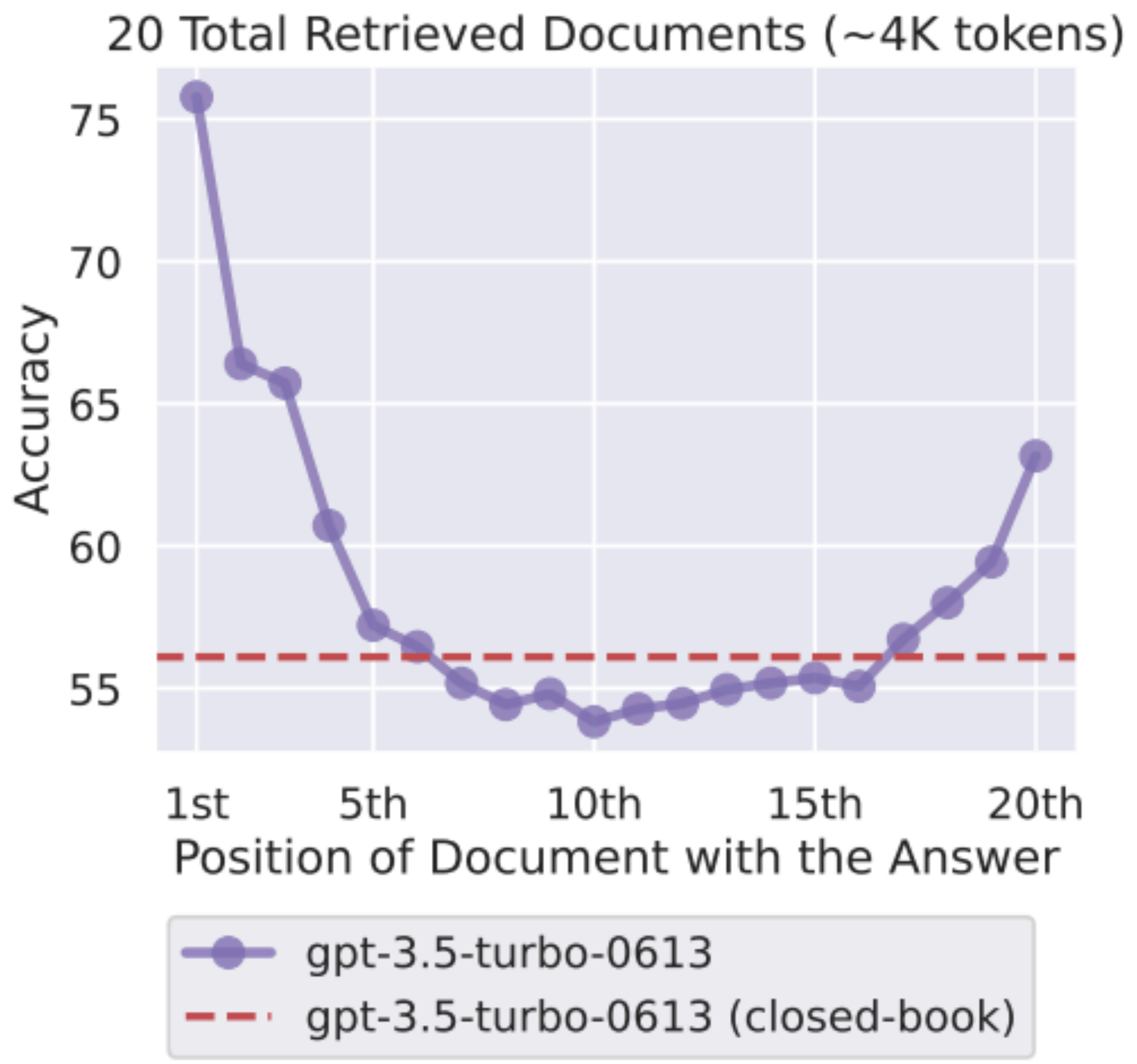
The name derives from “la tortuga”, the Spanish word for [MT(“tortuga”) → turtle] turtle.

The Brown Act is California’s law [WikiSearch(“Brown Act”) → The Ralph M. Brown Act is an act of the California State Legislature that guarantees the public's right to attend and participate in meetings of local legislative bodies.] that requires legislative bodies, like city councils, to hold their meetings open to the public.

Toolformer (Schick et al., NeurIPS’23)

Long-context LMs and RALMs

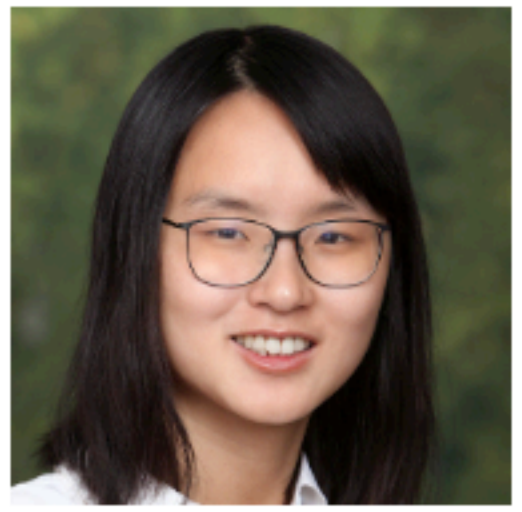
- Today's long-context LLMs support up to millions of tokens in their context window - Do we still need RAG?
- Long-context LMs support better RAG (more documents, more tokens)
- It puts less demand on retriever, but it can't really replace RAG (since the datastore is still much larger)
- Though there are still a lot of questions about whether long-context LMs can really support their contexts



(Liu et al., TACL 2023) "Lost in the Middle"

Recommended materials

ACL 2023 Tutorial:
Retrieval-based Language Models and Applications

Akari Asai[1], Sewon Min[1], Zexuan Zhong[2], Danqi Chen[2]

[1]University of Washington, [2]Princeton University

(2023/7)

https://acl2023-retrieval-lm.github.io/

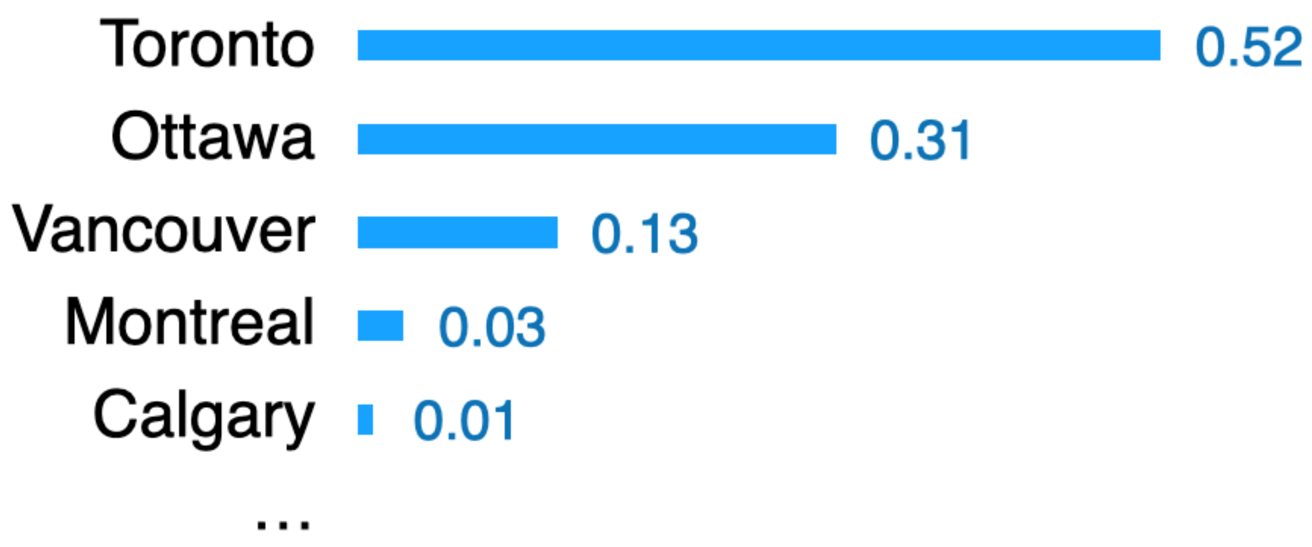
- Model architectures
- Training methods
- Applications and extensions (multi-lingual and multi-modal)

Why retrieval-augmented LMs?

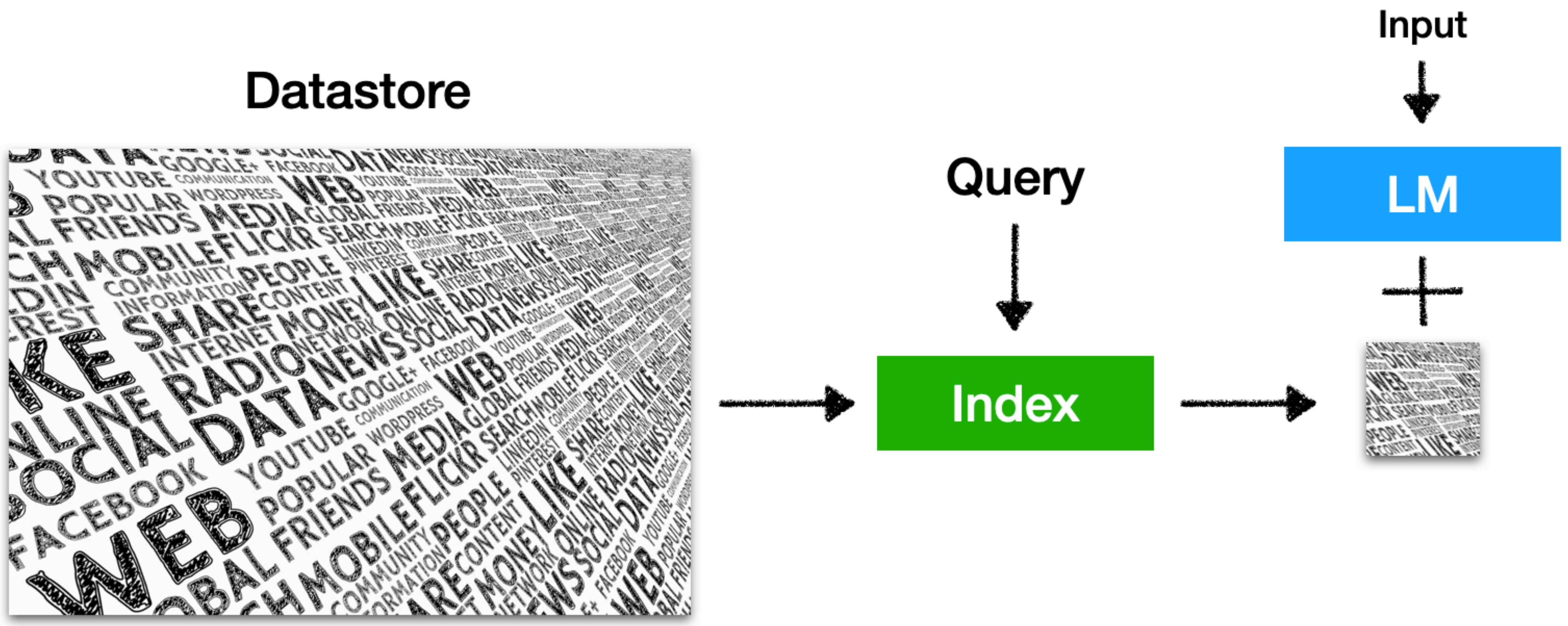
Retrieval-augmented LMs

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

The capital city of Ontario is ___



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



(Also referred to semiparametric and non-parametric models)

Can you cram all knowledge into 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**, ..

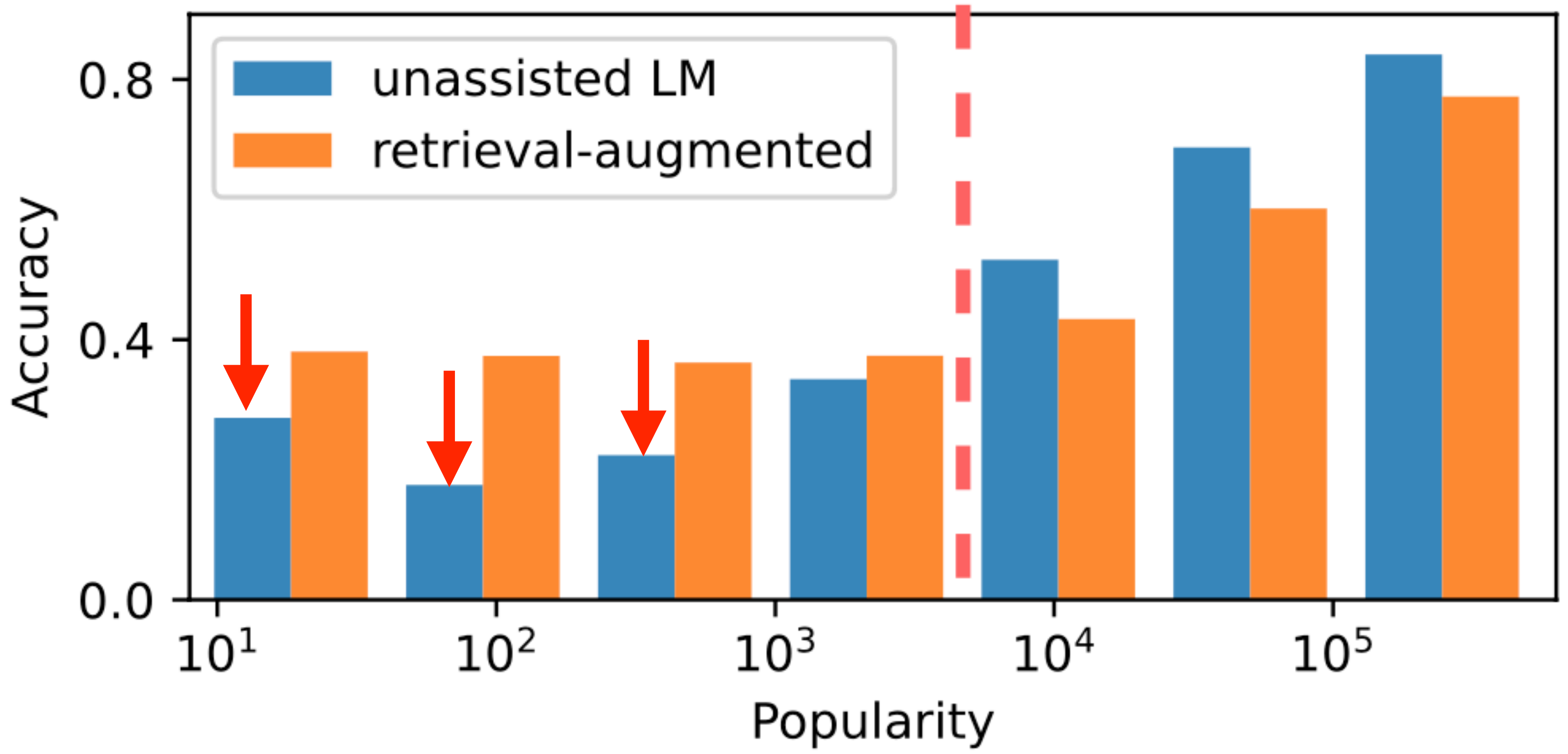
…

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,…

(From 2023/7)

What is Kathy Saltzman's occupation?



(Mallen et al., 2023)

GPT-3 davinci-003: 20%-30% accuracy

Can you cram all knowledge into parameters?

List 5 research papers authored by Danqi Chen

ChatGPT 4o

Here are five notable papers from her research:

1. ”A Fast and Accurate Dependency Parser using Neural Networks" (2014)
2. “Reading Wikipedia to Answer Open-Domain Questions" (2017)
3. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" (2018)
4. "Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context" (2019)
5. ”Stanford’s Graph-based Neural Dependency Parser at the CoNLL 2017 Shared Task" (2017)

*“Although primarily led by researchers at Google, **Danqi contributed to this landmark paper during her time as a visiting scientist at Google.**"*

(From 2024/9)

Can you cram all knowledge into parameters?

ChatGPT 4o

Share

DA

List 5 papers authored by Danqi Chen

Here are five notable papers authored by Danqi Chen:

1. **RoBERTa: A Robustly Optimized BERT Pretraining Approach** (2019)
   This paper presents RoBERTa, an optimized method for pretraining BERT models, achieving significant improvements in natural language understanding tasks. GOOGLE SCHOLAR
2. **Dense Passage Retrieval for Open-Domain Question Answering** (2020)
   In this work, Chen and colleagues introduce Dense Passage Retrieval (DPR), a technique that enhances the retrieval component of open-domain question-answering systems by learning dense representations for passages. GOOGLE SCHOLAR
3. **SimCSE: Simple Contrastive Learning of Sentence Embeddings** (2021)
   This paper proposes SimCSE, a method for learning sentence embeddings using contrastive learning, resulting in high-quality representations for various downstream tasks. GOOGLE SCHOLAR
4. **SpanBERT: Improving Pre-training by Representing and Predicting Spans** (2020)
   Chen and co-authors introduce SpanBERT, a pretraining approach that focuses on span-level representations, leading to better performance in tasks like question answering and coreference resolution. GOOGLE SCHOLAR
5. **Reading Wikipedia to Answer Open-Domain Questions** (2017)
   This work explores methods for leveraging Wikipedia as a knowledge source to develop systems capable of answering open-domain questions effectively. GOOGLE SCHOLAR

These publications highlight Danqi Chen's significant contributions to natural language processing and machine learning.

Citations

Google Scholar
Danqi Chen - Google Scholar
Danqi Chen. Princeton University. Verified email at cs.princeton.edu - Homepage. Natural Language...

Search Results

arXiv
LESS: Selecting Influential Data for Targeted Instruction Tuning
February 5, 2024 — LESS is an algorithm that selects influential data from large datasets for...

Princeton Computer Science
Danqi Chen's Homepage - Princeton University
Danqi Chen is a natural language processing and machine learning researcher who co-leads the...

Princeton Computer Science
Papers - Princeton University
Sewon Min, Jordan Boyd-Graber, Chris Alberti, Danqi Chen, Eunsol Choi, Michael Collins, Kelvin...

Princeton Collaboration
Danqi Chen - Princeton University
Calculated based on number of publications stored in Pure and citations from Scopus. 2011 2023....

Princeton Computer Science
Danqi Chen - Princeton University
Danqi Chen is a researcher and student at Stanford University, working on deep learning, natural...

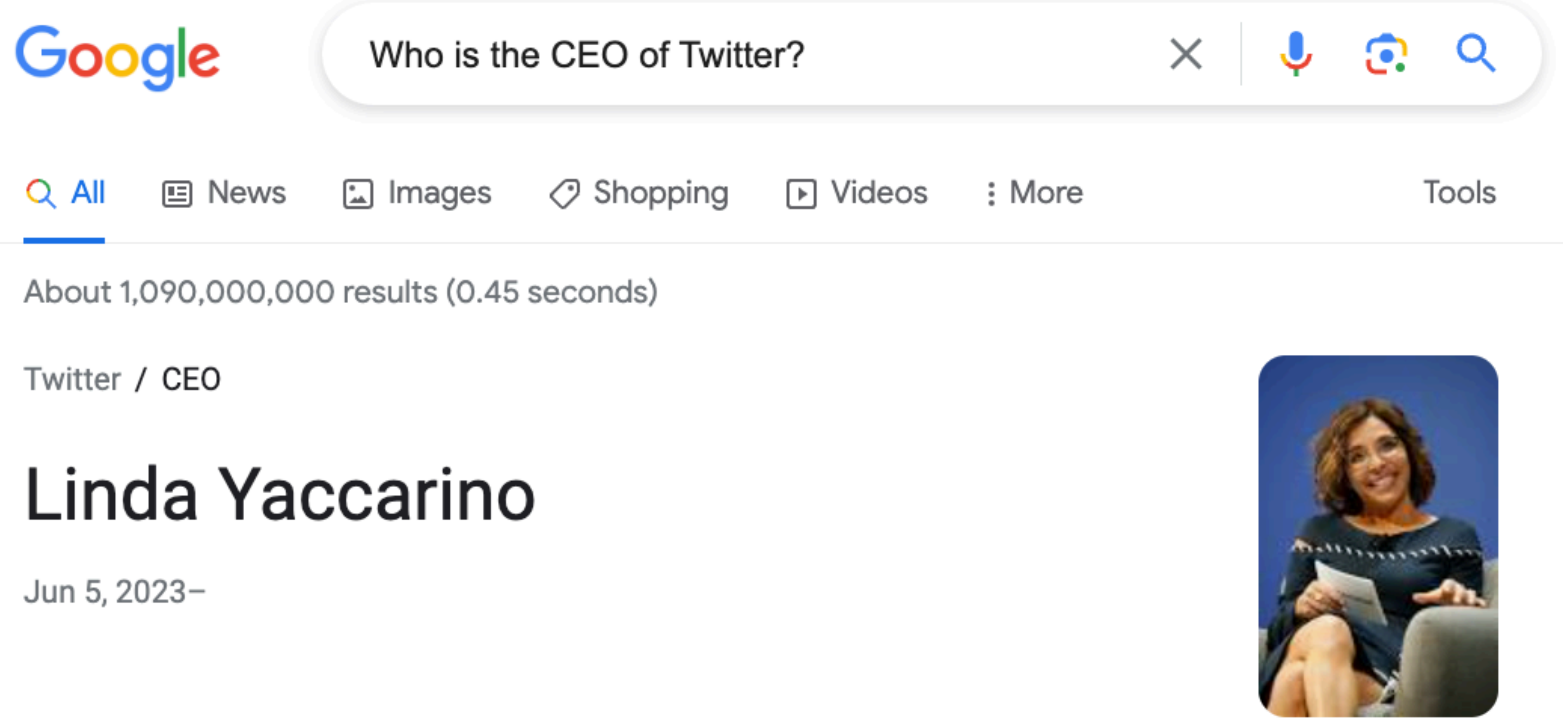
(From 2024/11)

LLMs knowledge is easily outdated

Who is the CEO of Twitter?

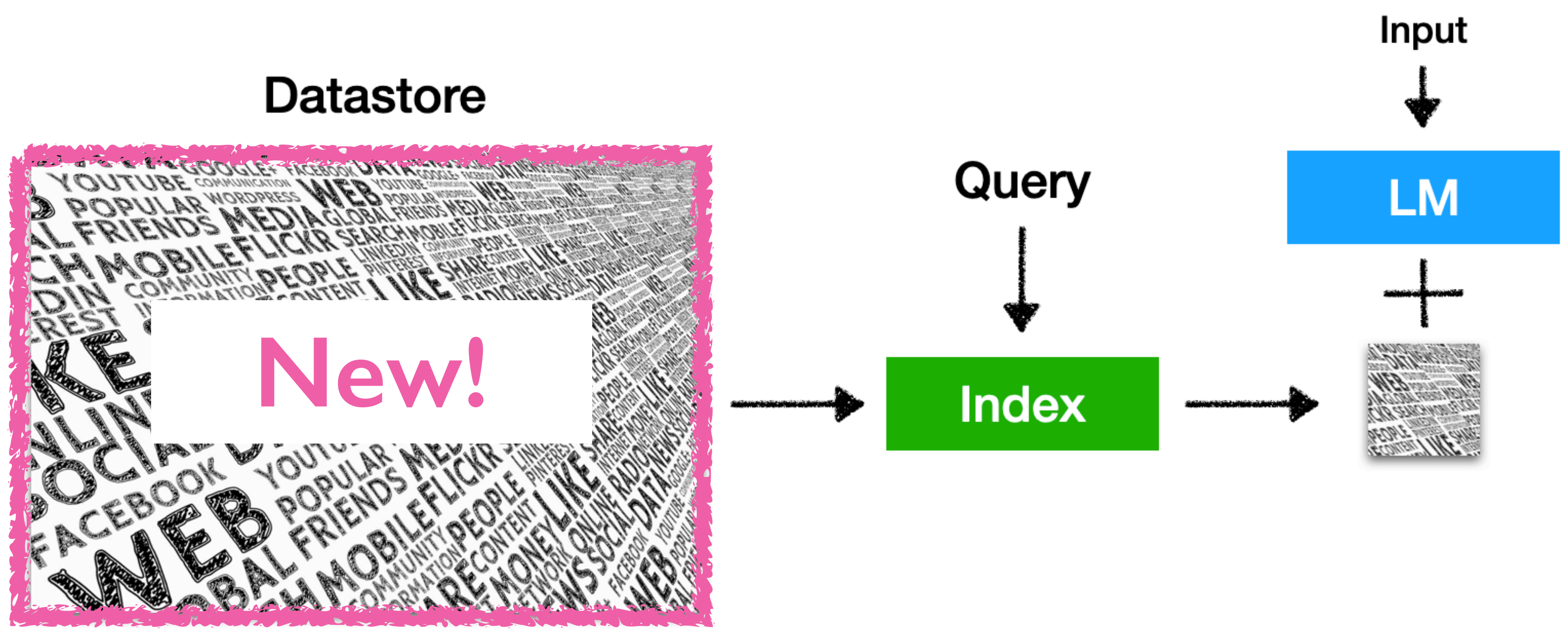
ChatGPT

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



- Existing **knowledge editing** methods are still NOT scalable (active research!)

- The datastore can be easily **updated** and **expanded** - even without retraining!



LLMs knowledge is easily outdated

DC Who is currently the prime minister of the UK?

Rishi Sunak is currently the Prime Minister of the United Kingdom, having taken office in October 2022 after Liz Truss's brief tenure. He is a member of the Conservative Party and is the UK's first British Asian Prime Minister.

Copy Retry

Claude can make mistakes. Please double-check responses.

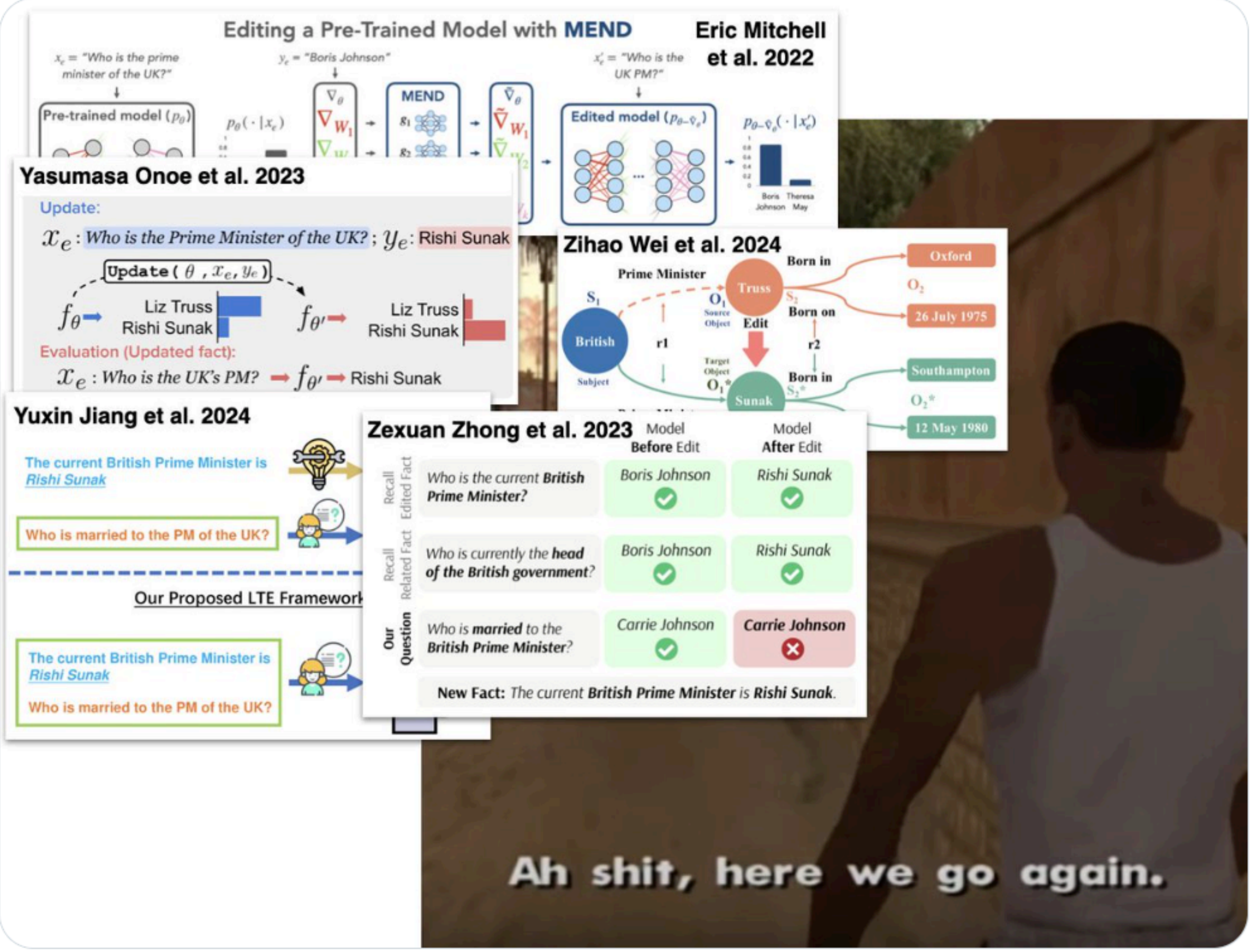
Who is currently the prime minister of the UK?

As of November 13, 2024, the Prime Minister of the United Kingdom is Sir Keir Starmer. He assumed office on July 5, 2024, following a significant victory by the Labour Party in the general election. ASSOCIATED PRESS Prior to his premiership, Starmer served as the Leader of the Labour Party from 2020 and was the Member of Parliament for Holborn and St Pancras since 2015. WIKIPEDIA His tenure has been marked by commitments to climate action, including a pledge to reduce UK carbon emissions by 81% by 2035. THE TIMES

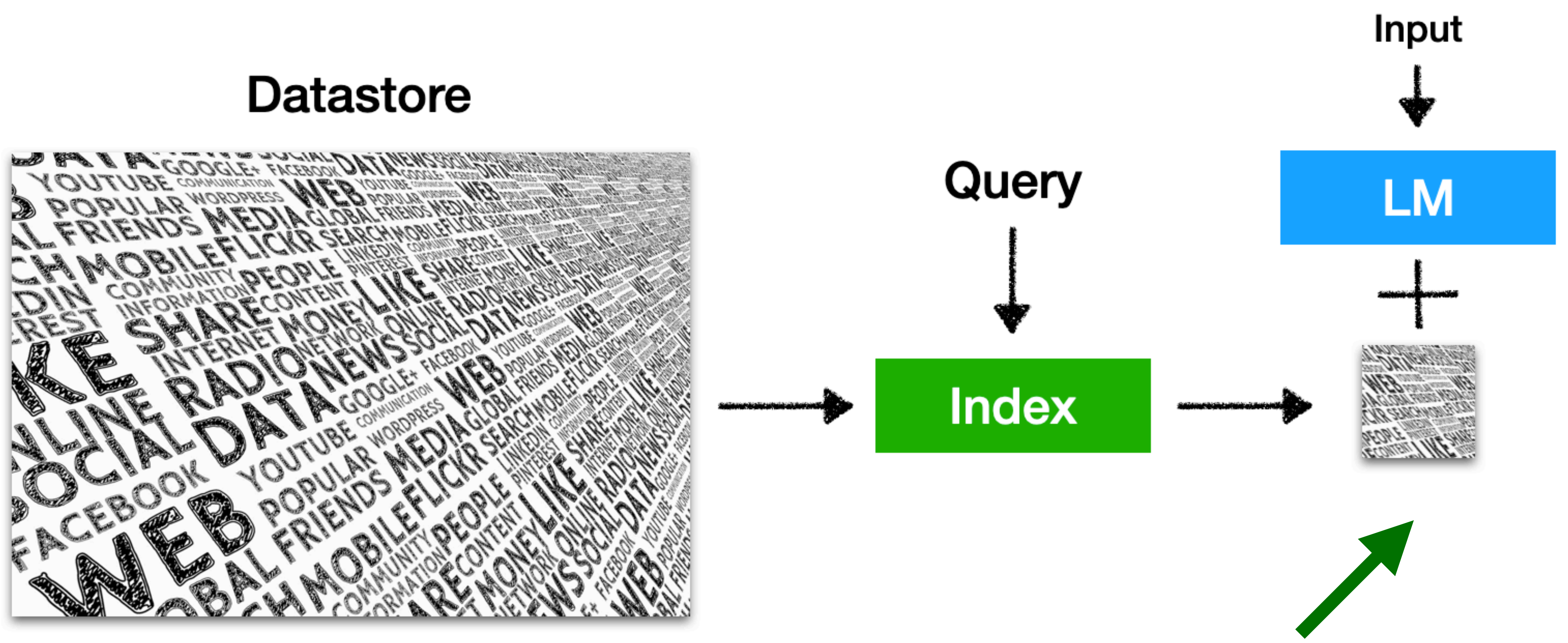
Sources

Greg Durrett
@gregd_nlp

new british pm elected
knowledge editing researchers:

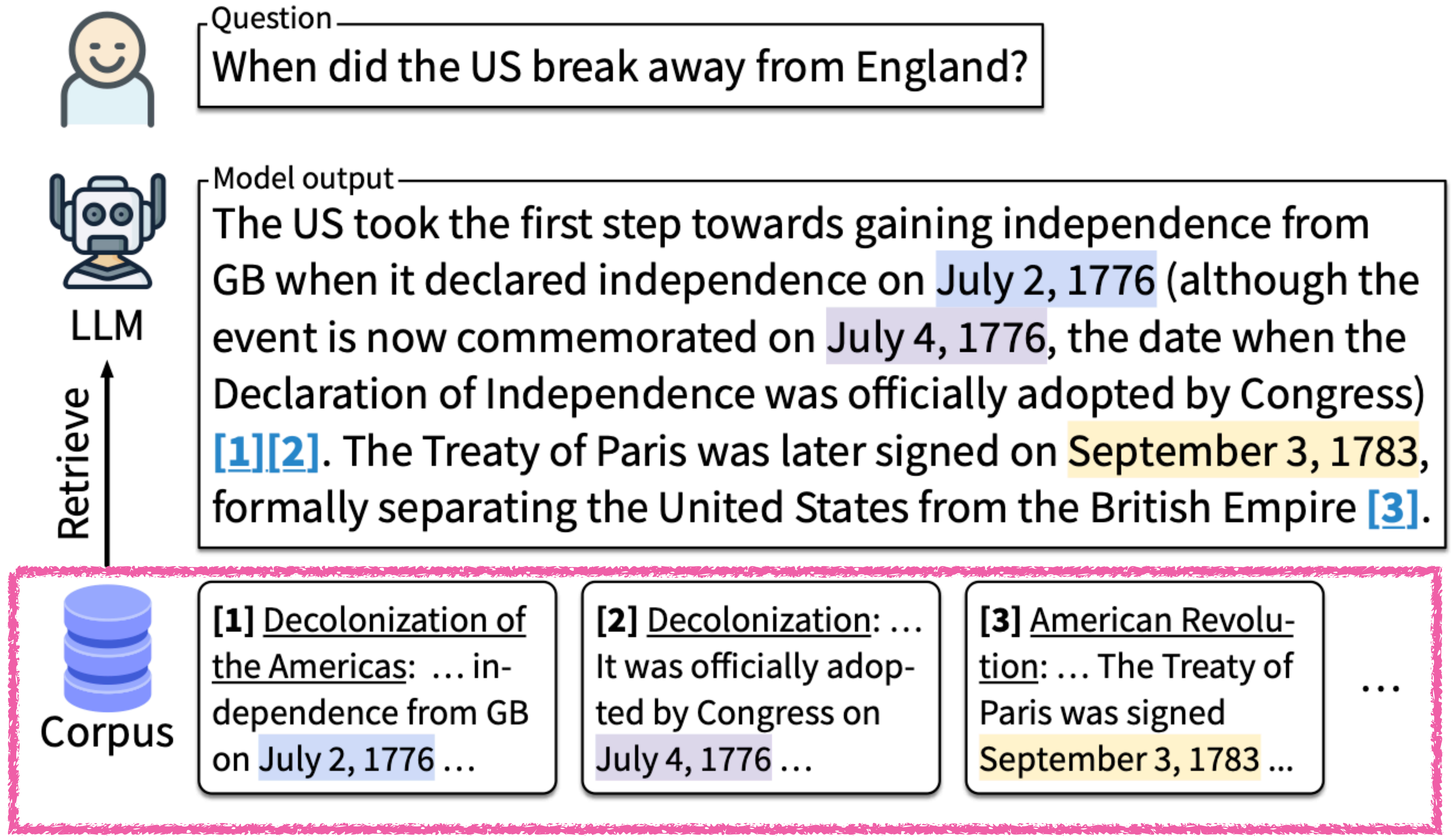


LLM outputs are hard to interpret and verify



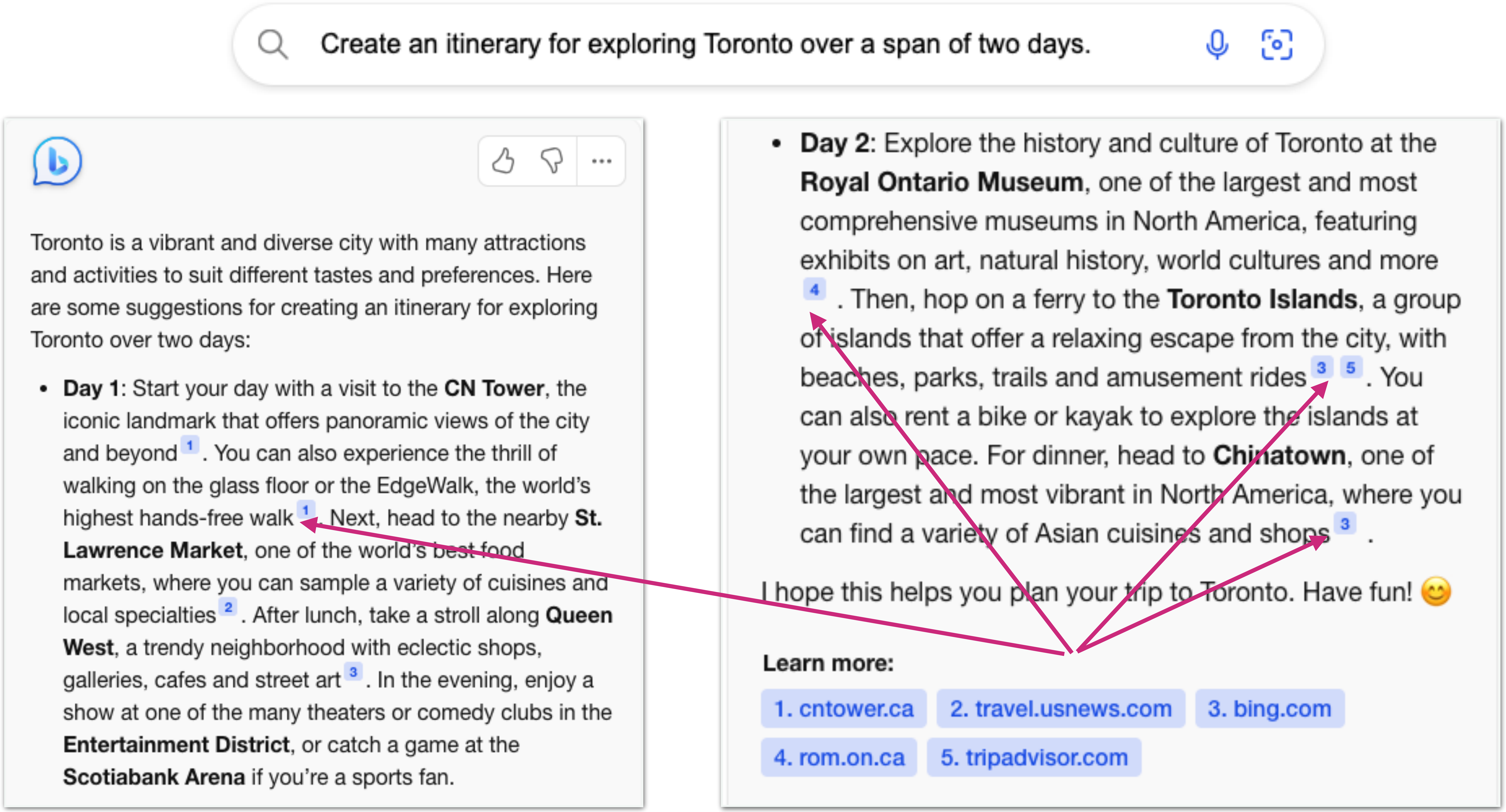
Can **trace knowledge source** from retrieval results - better **interpretability** & **control**

Generating text with citations

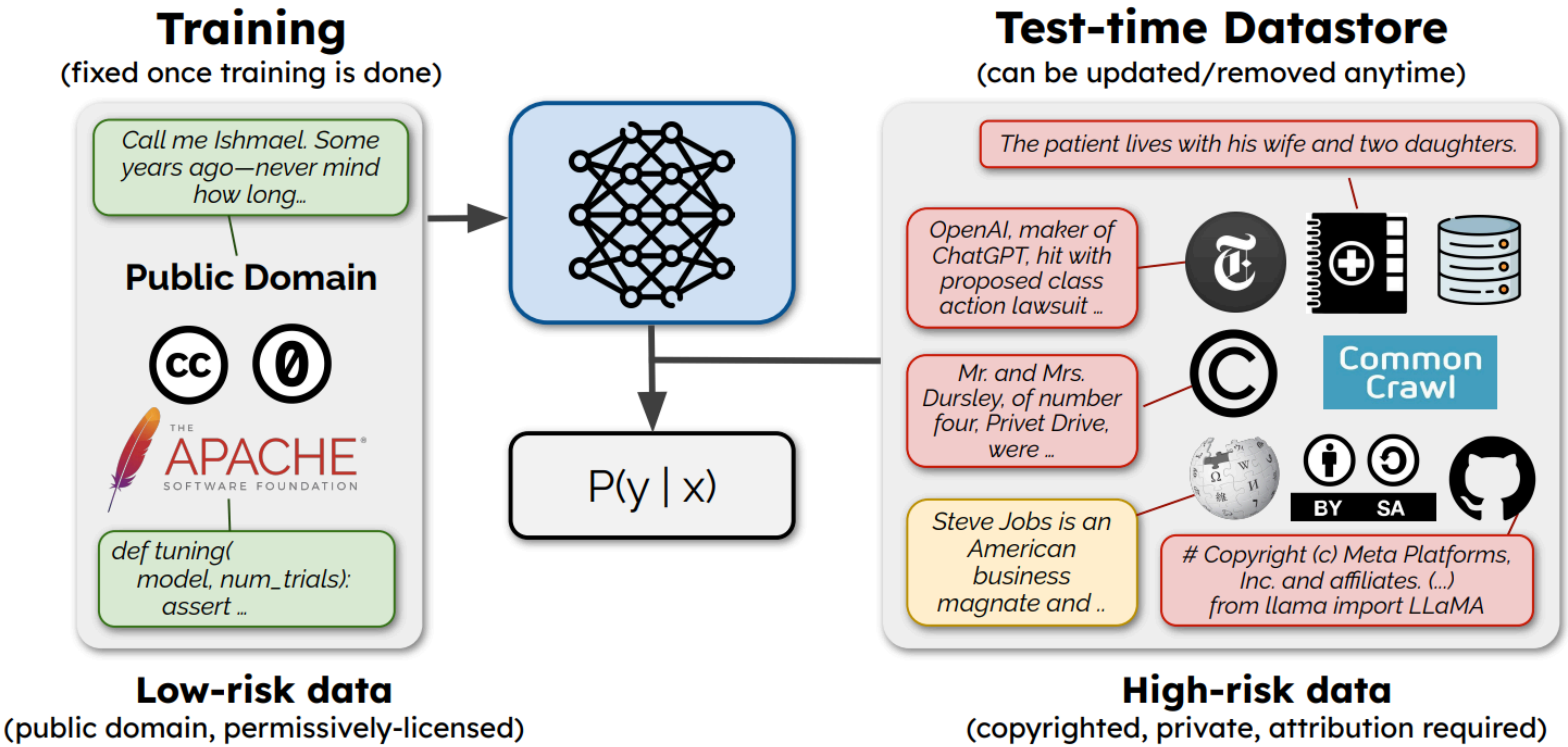


(Nakano et al. 2021; Menick et al., 2022; Gao et al., 2023)

LLM outputs are hard to interpret and verify



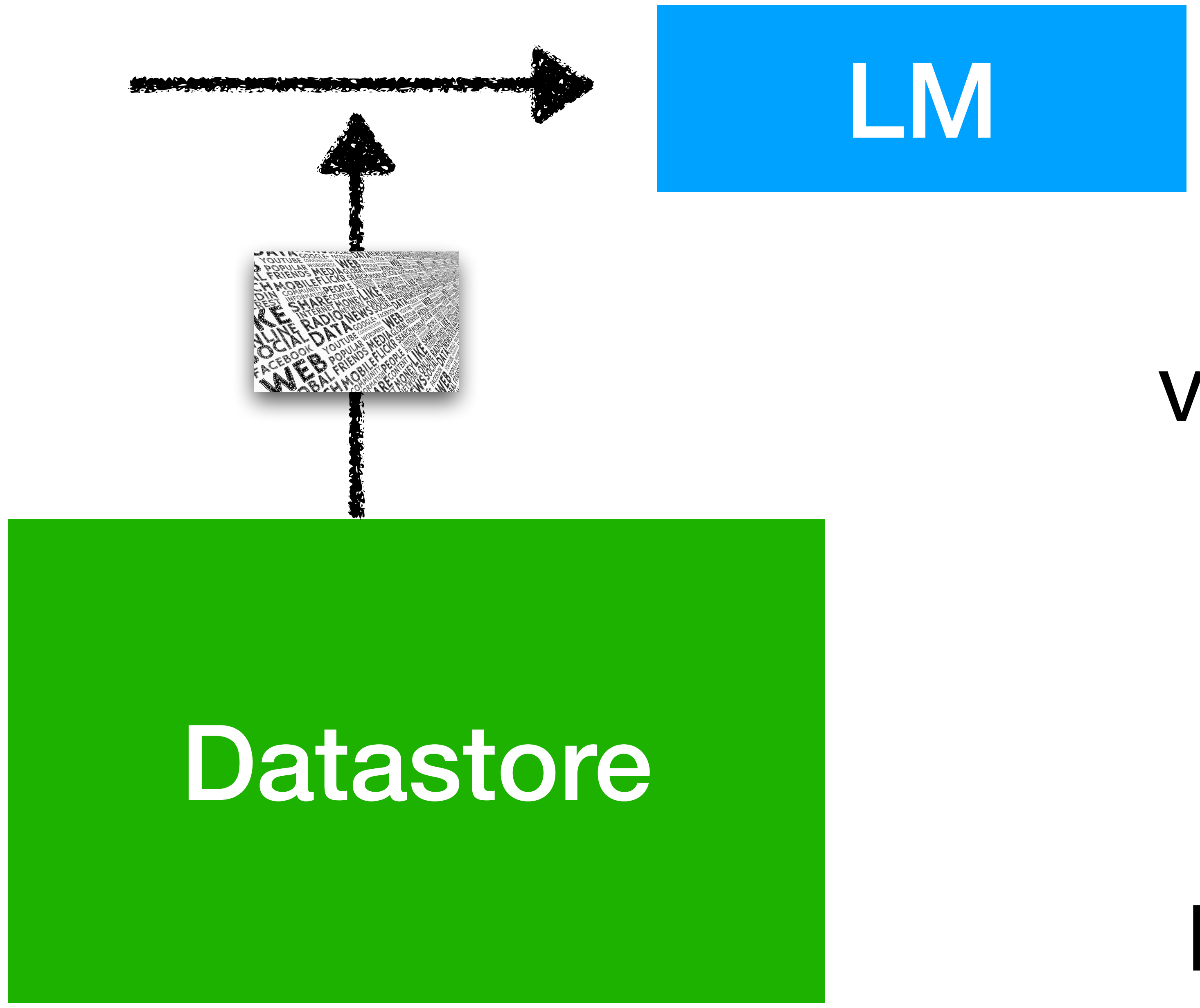
Private and copyrighted data for training LMs



SILO (Min et al., ICLR'24)

- **Machine unlearning** is a challenging problem too (active research!)

Can we scale LMs with (test-time) datastore?



vs.

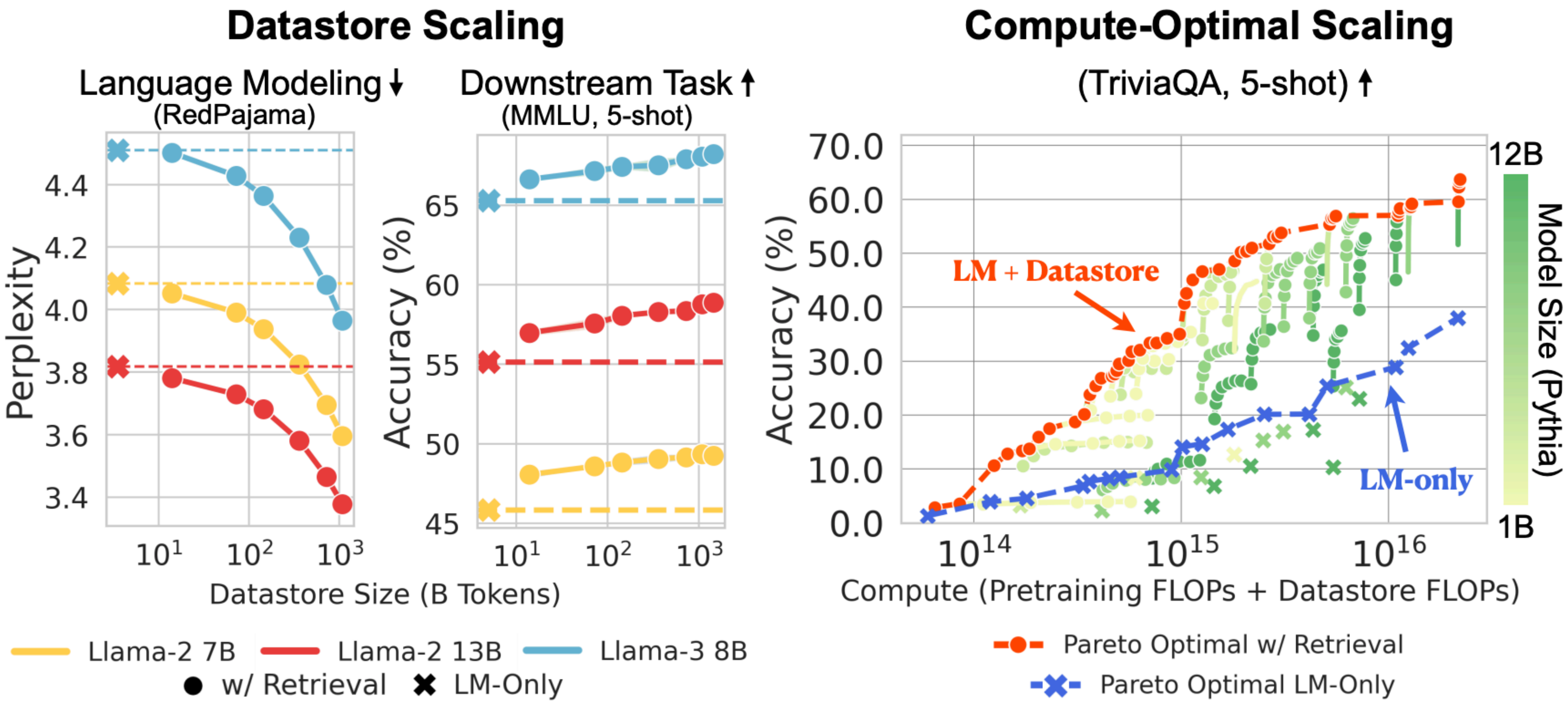
Long-term goal: can we possibly reduce the **training** and **inference costs,** and scale down the size of LLMs?

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

Can we scale LMs with (test-time) datastore?

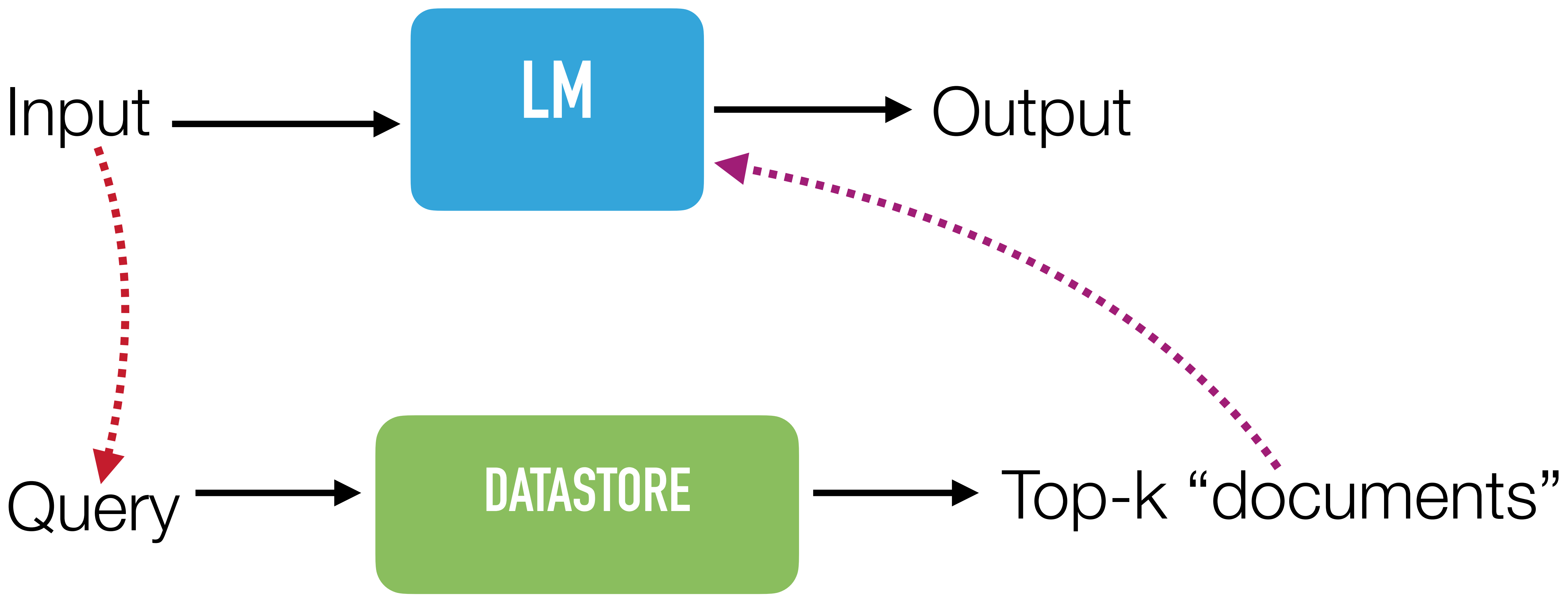
Scaling Retrieval-Based Language Models with a Trillion-Token Datastore

Rulin Shao[1] Jacqueline He[1] Akari Asai[1] Weijia Shi[1]
Tim Dettmers[1] Sewon Min[1] Luke Zettlemoyer[1] Pang Wei Koh[1,2]
[1]University of Washington [2]Allen Institute for AI
{rulins,jyyh,akari,swj0419,dettmers,sewon,lsz,pangwei}
@cs.washington.edu



RALMs: model architectures and training methods (path #1)

Roadmap



ARCHITECTURE

- How is the retriever represented? Granularity of datastore?
- How are retrieved contexts integrated with LMs?
- Sizes of LMs vs datastore?

LEARNING

- How are the LM and retriever trained together?
- Training from scratch? Fine-tuning?
- Pre-training or instruction tuning?

How is retrieval implemented?

- Sim: a similarity score between two pieces of text

Example $$\mathrm{sim}(i,j) = \underbrace{\mathrm{tf}_{i,j}}_{\text{\# of occurrences of } i \text{ in } j} \times \log \frac{N}{\mathrm{df}_i}$$

N: # of total docs

df_i: # of docs containing i

Sparse retrieval

Example $$\mathrm{sim}(i,j) = \mathrm{Encoder}(i) \cdot \mathrm{Encoder}(j)$$

Maps the text into an h-dimensional vector

Dense retrieval

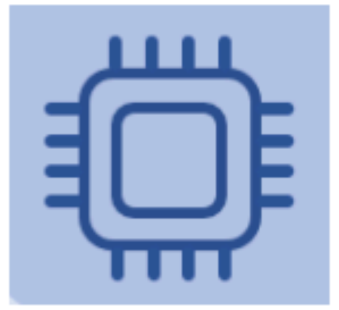
- An entire field of study on how to define or learn these similarity functions better
- There are efficient data structures/infrastructure for supporting fast and accurate search from a large datastore

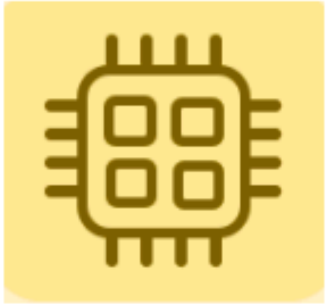
How is retrieval implemented?

Software: FAISS, Distributed FAISS, SCaNN, etc…

Method	Class name	`index_factory`	Main parameters	Bytes/vector	Exhaustive	Comments
Exact Search for L2	`IndexFlatL2`	`"Flat"`	`d`	`4*d`	yes	brute-force
Exact Search for Inner Product	`IndexFlatIP`	`"Flat"`	`d`	`4*d`	yes	also for cosine (normalize vectors beforehand)
Hierarchical Navigable Small World graph exploration	`IndexHNSWFlat`	`"HNSW,Flat"`	`d`, `M`	`4*d + x * M * 2 * 4`	no	
Inverted file with exact post-verification	`IndexIVFFlat`	`"IVFx,Flat"`	`quantizer`, `d`, `nlists`, `metric`	`4*d + 8`	no	Takes another index to assign vectors to inverted lists. The 8 additional bytes are the vector id that needs to be stored.
Locality-Sensitive Hashing (binary flat index)	`IndexLSH`	-	`d`, `nbits`	`ceil(nbits/8)`	yes	optimized by using random rotation instead of random projections
Scalar quantizer (SQ) in flat mode	`IndexScalarQuantizer`	`"SQ8"`	`d`	`d`	yes	4 and 6 bits per component are also implemented.
Product quantizer (PQ) in flat mode	`IndexPQ`	`"PQx"`, `"PQ"M"x"nbits`	`d`, `M`, `nbits`	`ceil(M * nbits / 8)`	yes	
IVF and scalar quantizer	`IndexIVFScalarQuantizer`	"IVFx,SQ4" "IVFx,SQ8"	`quantizer`, `d`, `nlists`, `qtype`	SQfp16: 2 * `d` + 8, SQ8: `d` + 8 or SQ4: `d/2` + 8	no	Same as the `IndexScalarQuantizer`
IVFADC (coarse quantizer+PQ on residuals)	`IndexIVFPQ`	`"IVFx,PQ"y"x"nbits`	`quantizer`, `d`, `nlists`, `M`, `nbits`	`ceil(M * nbits/8)+8`	no	
IVFADC+R (same as IVFADC with re-ranking based on codes)	`IndexIVFPQR`	`"IVFx,PQy+z"`	`quantizer`, `d`, `nlists`, `M`, `nbits`, `M_refine`, `nbits_refine`	`M+M_refine+8`	no	

Exact Search





CPU vs. GPU

Approximate Search
(Relatively easy to scale to ~1B elements)

Example: RETRO

DeepMind

Improving language models by retrieving from trillions of tokens

ARCHITECTURE

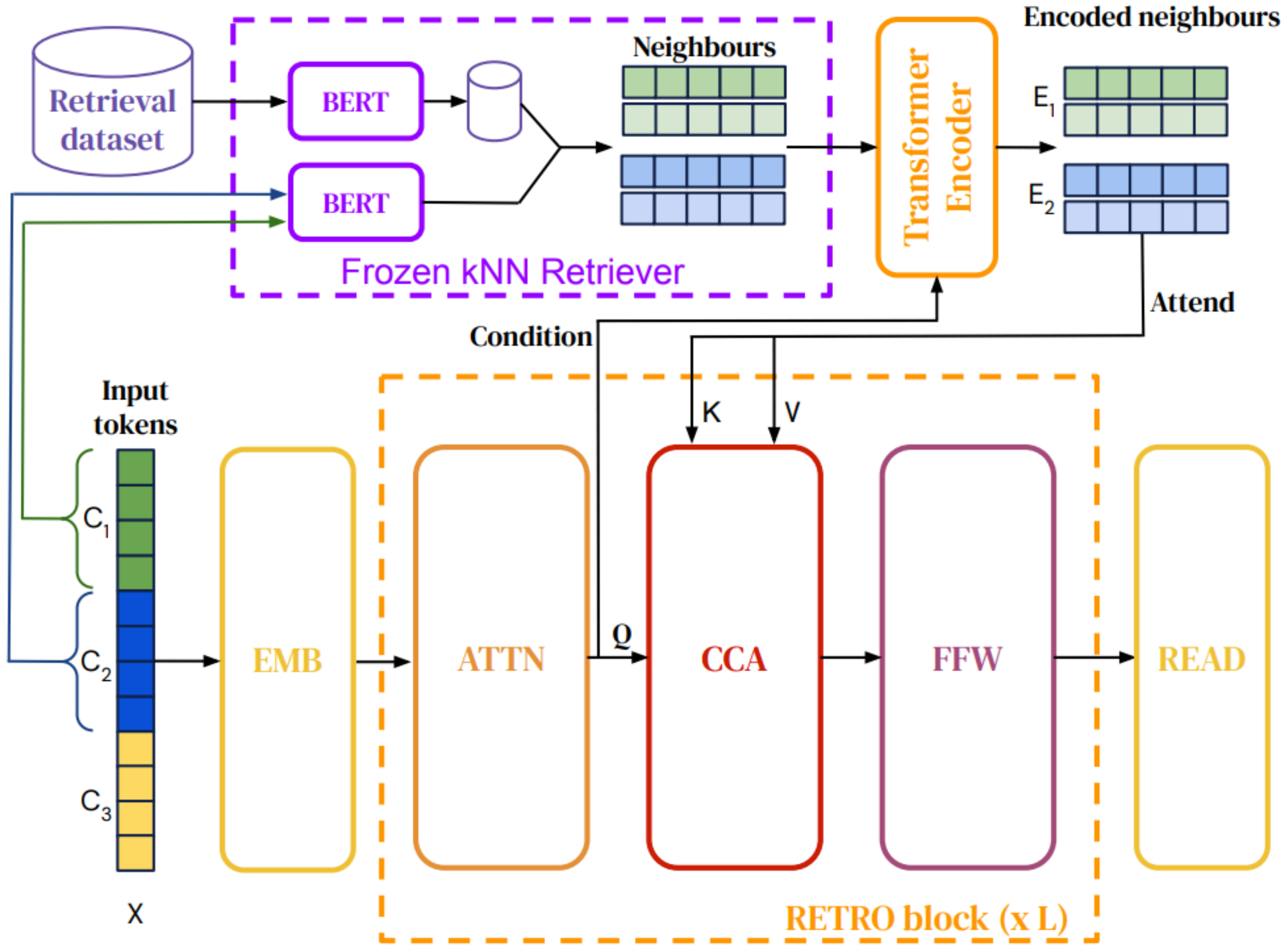
- How is the retriever represented? Granularity of datastore?
- Sizes of LMs vs datastore?
- How are retrieved contexts integrated with LMs?

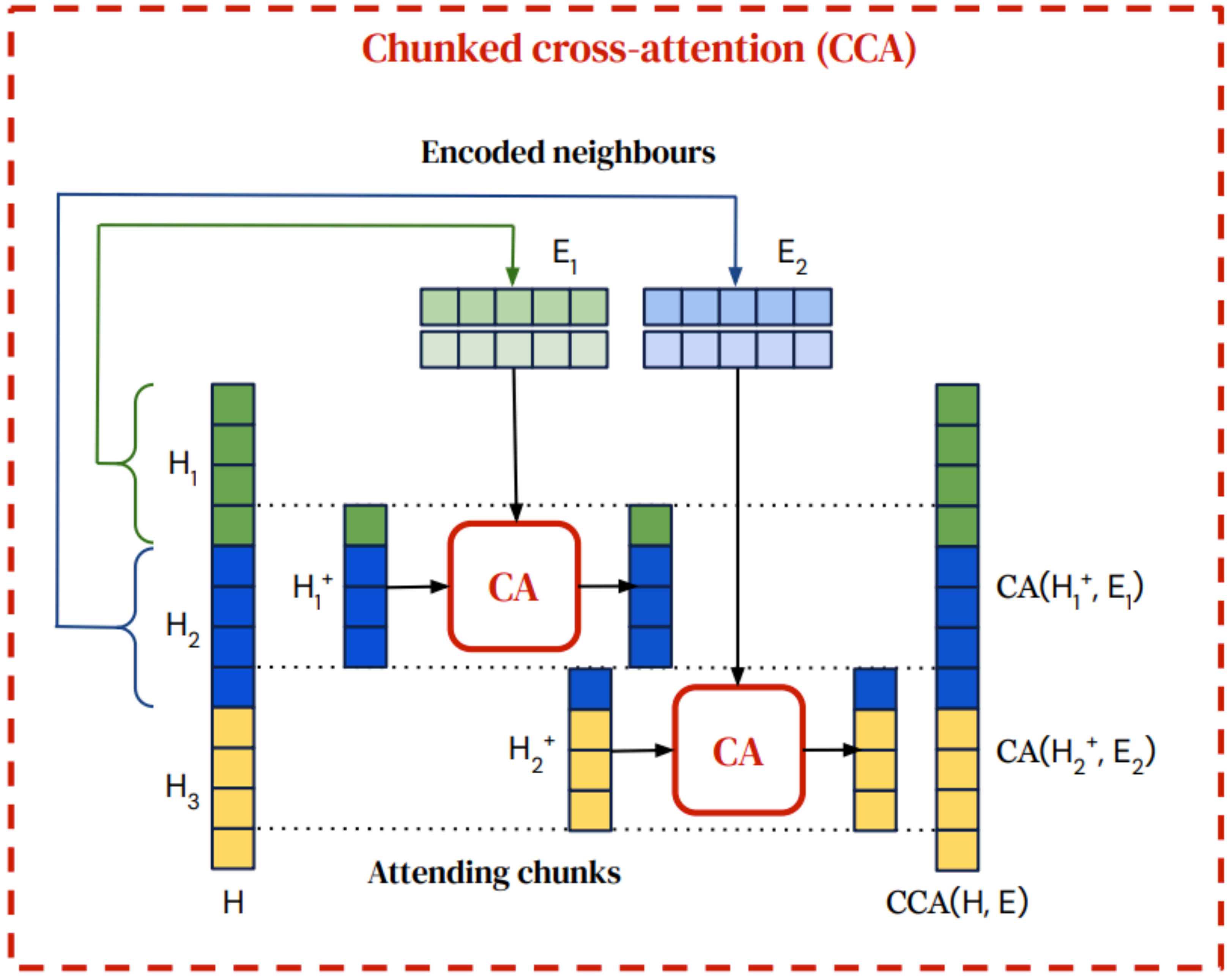
  - Granularity: chunks of 64 tokens
  - Representation: frozen BERT encoders (pre-trained but for different tasks)
  - LMs: 150M-7B parameters; datastore: up to 2T tokens (training tokens 600B tokens)

Example: RETRO

ARCHITECTURE

- How are retrieved contexts integrated with LMs?





The information is integrated into the intermediate layers of Transformers (with an extra encoder)

Example: RETRO

LEARNING

- How are the LM and retriever trained together?
- Training from scratch? Fine-tuning?
- Pre-training or instruction tuning?

- The retriever ("encoder") is pre-trained and not updated anymore
- Trained from scratch or fine-tuned ("retrofit")

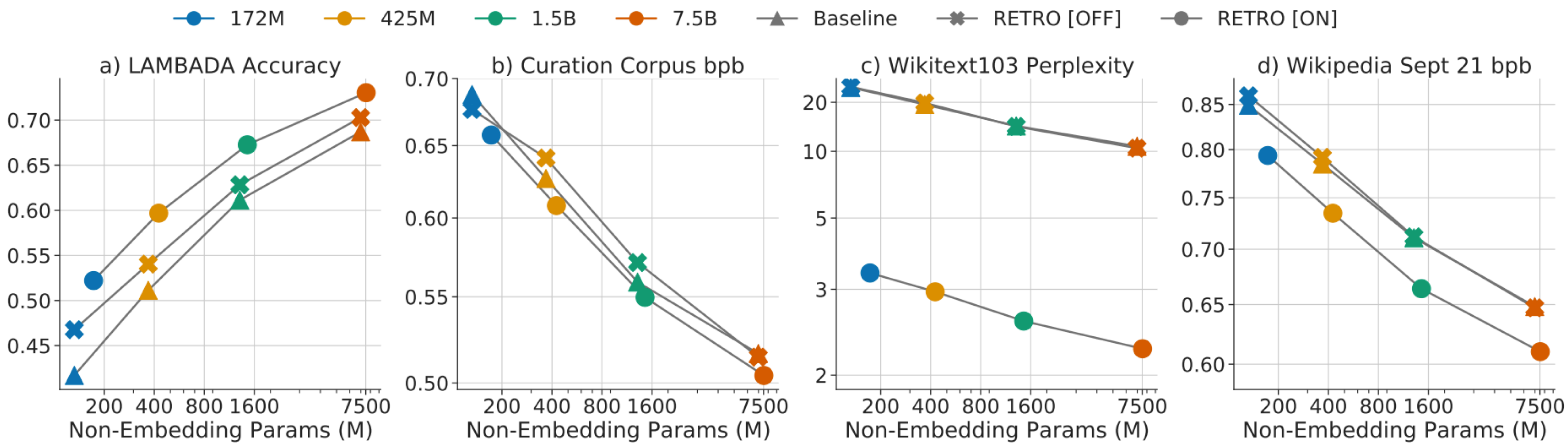
[Submitted on 11 Oct 2023 (v1), last revised 29 May 2024 (this version, v3)]

InstructRetro: Instruction Tuning post Retrieval-Augmented Pretraining

Boxin Wang, Wei Ping, Lawrence McAfee, Peng Xu, Bo Li, Mohammad Shoeybi, Bryan Catanzaro

RETRO: experiments

Perplexity



In restrospect, too many works focused on improving perplexity instead of downstream tasks at that time..

RETRO: experiments

Question answering

Model	Test Accuracy
REALM (Guu et al., 2020)	40.4
DPR (Karpukhin et al., 2020)	41.5
RAG (Lewis et al., 2020)	44.5
EMDR^2 (Sachan et al., 2021)	52.5
FID (Izacard and Grave, 2021)	51.4
FID + Distill. (Izacard et al., 2020)	**54.7**
Baseline 7B (closed book)	30.4
RETRO 7.5B (DPR retrieval)	45.5

It is not better than specialized QA models

No evals on any of in-context learnings tasks in GPT-3

"With a 2 trillion token database, our Retrieval-Enhanced Transformer (Retro) obtains ***comparable performance to GPT-3 and Jurassic-1 on the Pile, despite using 25× fewer parameters****."*

RETRO is not open-sourced… :(:(

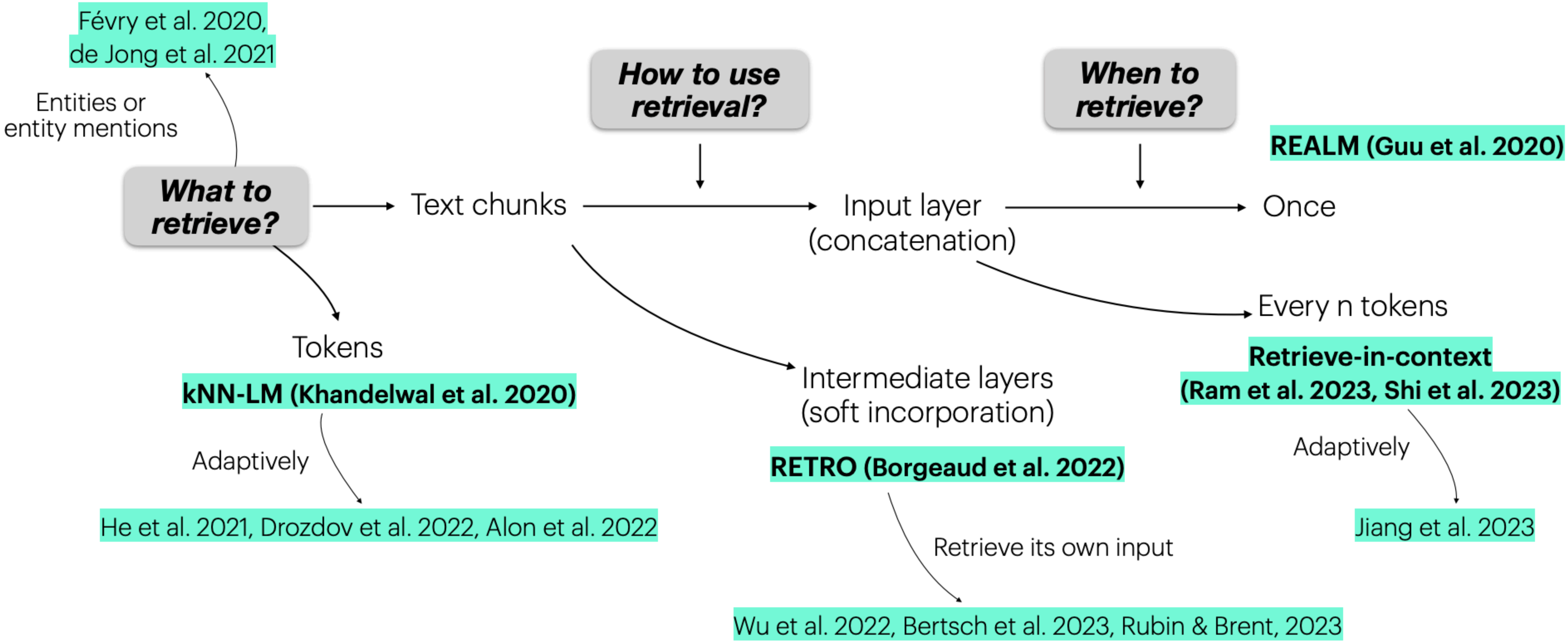
RETRO++ (NVIDIA)

Tasks	Small GPT	Small RETRO	Medium GPT	Medium RETRO	XL GPT	XL RETRO	XXL GPT	XXL RETRO
Knowledge-intensive Tasks								
HellaSwag	31.3	36.2 ↑4.9	43.2	46.2 ↑3.0	56.7	59.0 ↑2.3	72.3	70.6 ↓1.7
BoolQ	59.3	61.8 ↑2.5	57.4	57.2 ↓0.2	62.2	62.7 ↑0.5	67.3	70.7 ↑3.4
Knowledge-nonintensive Tasks								
Lambada	41.7	41.4 ↓0.3	54.1	55.0 ↑0.9	63.9	64.0 ↑0.1	73.9	72.7 ↓1.2
RACE	34.6	32.5 ↓2.1	37.3	37.3 ↑0.0	40.8	39.9 ↓0.9	44.3	43.2 ↓1.1
PiQA	64.3	64.8 ↑0.5	70.2	68.7 ↓1.5	73.7	74.1 ↑0.4	78.5	77.4 ↓1.1
WinoGrande	52.4	52.0 ↓0.4	53.8	55.2 ↑1.4	59.0	60.1 ↑1.1	68.5	65.8 ↓2.7
ANLI-R2	35.1	36.2 ↑1.1	33.5	33.3 ↓0.2	34.3	35.3 ↑1.0	32.2	35.5 ↑3.3
HANS	51.5	51.4 ↓0.1	50.5	50.5 ↑0.0	50.1	50.0 ↓0.1	50.8	56.5 ↑5.7
WiC	50.0	50.0 ↑0.0	50.2	50.0 ↓0.2	47.8	49.8 ↑2.0	52.4	52.4 ↑0.0
Avg. Acc. (↑)	46.7	47.4 ↑0.7	50.0	50.4 ↑0.4	54.3	55.0 ↑0.7	60.0	60.5 ↑0.5

Table 6: Accuracy (Acc.) on nine downstream tasks evaluated in the zero-shot setting for pretrained LMs with different parameter sizes.

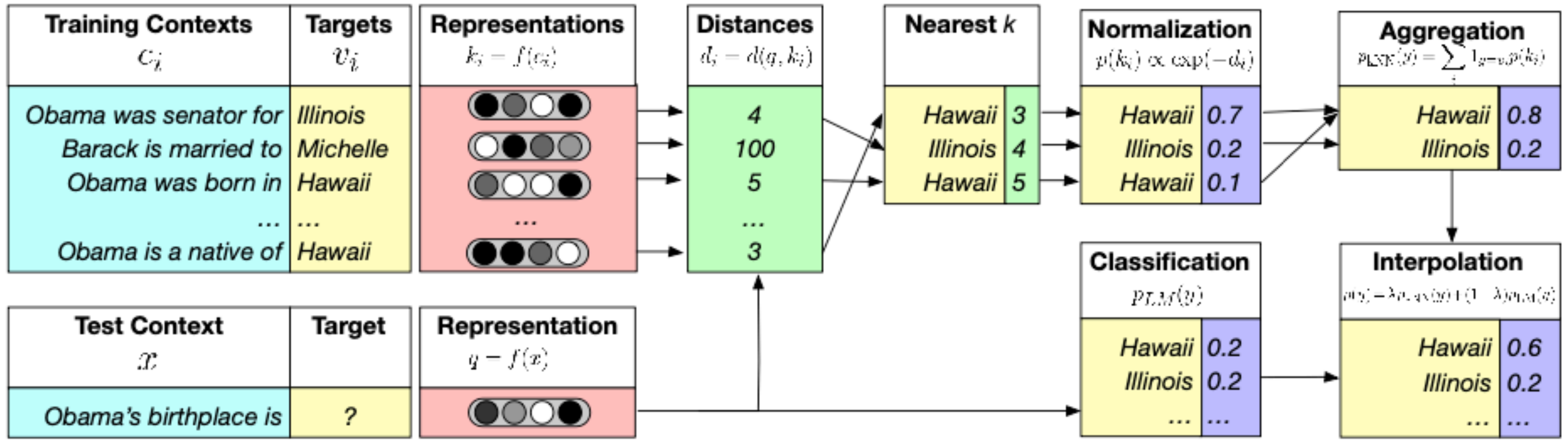
RETRO++ (Wang et al., ENMNLP'23)

Other architectures for RALMs



kNN-LMs

(Retrieval can be added at the output layer)



ARCHITECTURE

- **Token**-level datastore
- Representation = input to last FFN layer
- Integration only at the **output layer**

$$p_{\mathrm{kNN}}(w \mid c_t) \propto \sum_{\boxed{(c,x)\in\mathcal{D}}} \mathbb{I}(x = w) \exp(-\|f_\theta(c) - f_\theta(c_t)\|^2)$$

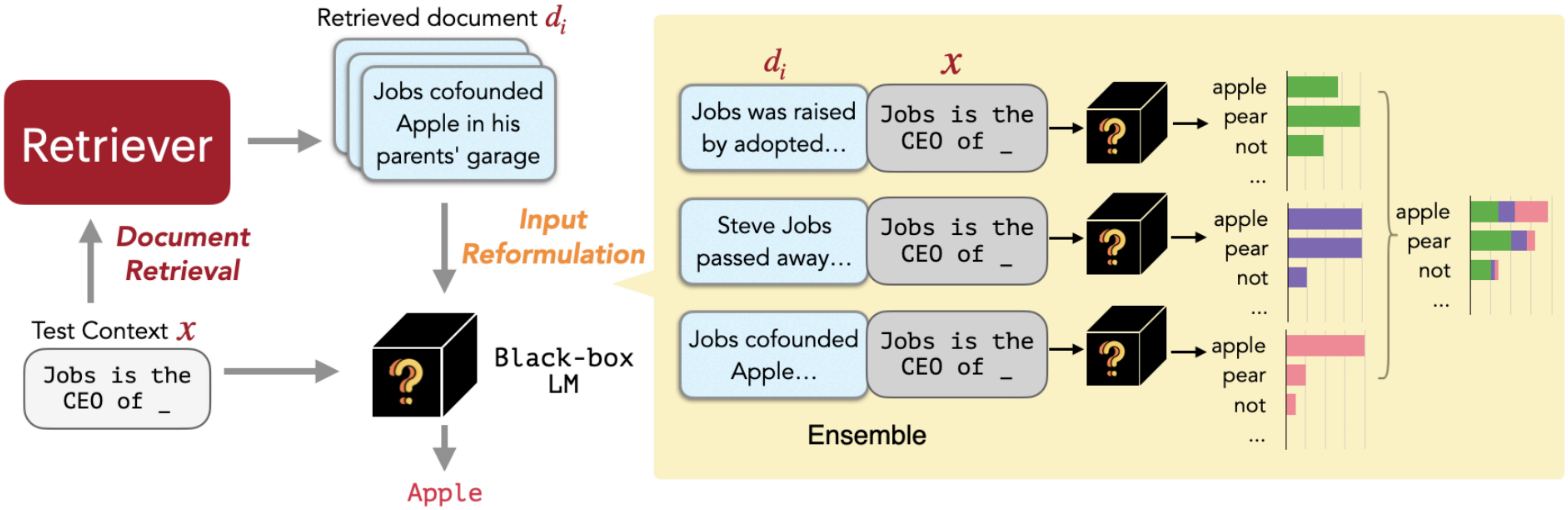
→ only keep top-K after NN search

$$P(w \mid c_t) = \lambda P_{\mathrm{LM}}(w \mid c_t) + (1-\lambda) P_{\mathrm{kNN}}(w \mid c_t)$$ (Linear interpolation)

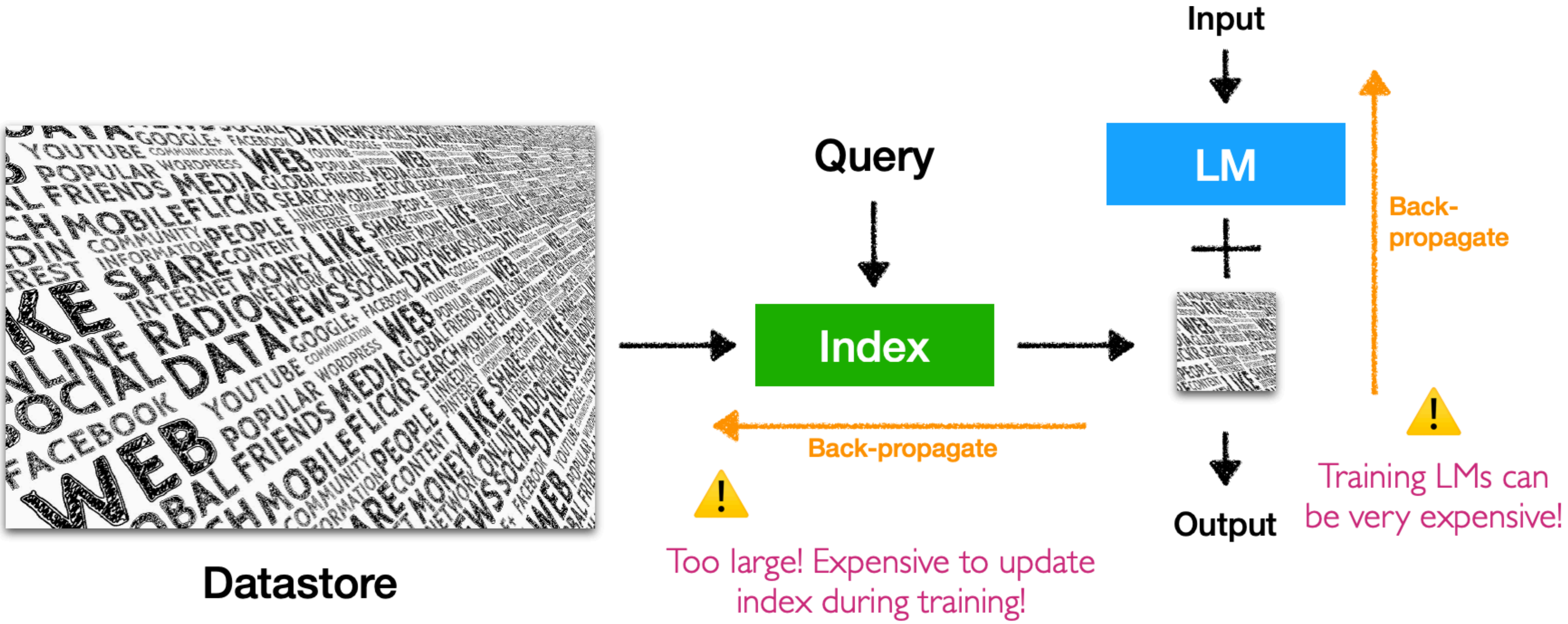
kNN-LM (Khandelwal et al., ICLR'20)

REPLUG

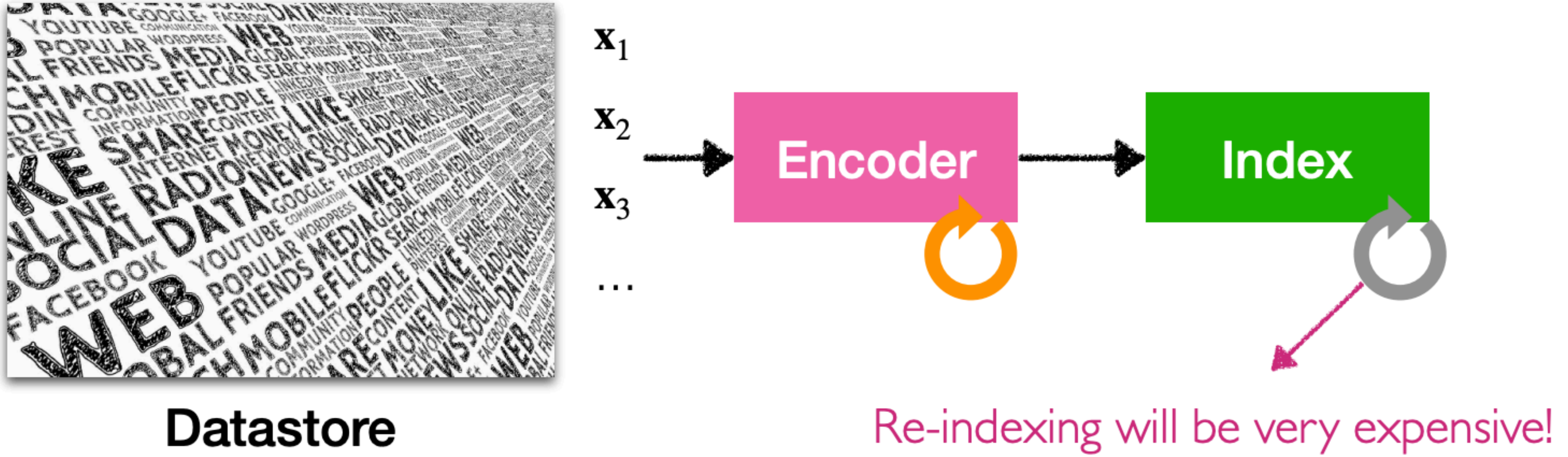
(Retrieval can be added at the input layer)



Why is training so hard for RALMs?



Why is training so hard for RALMs?

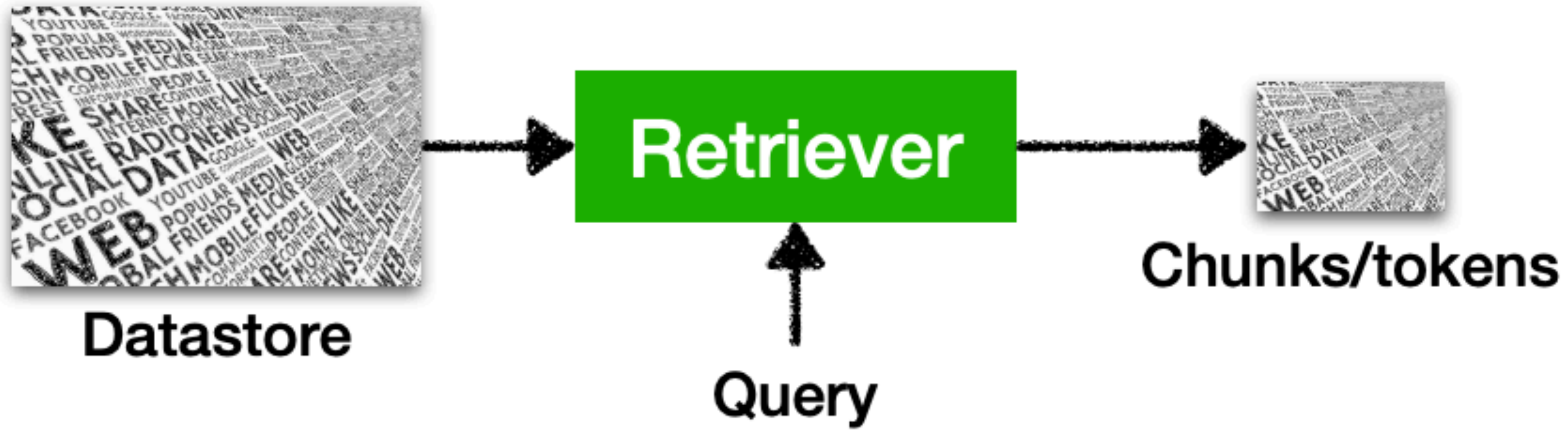


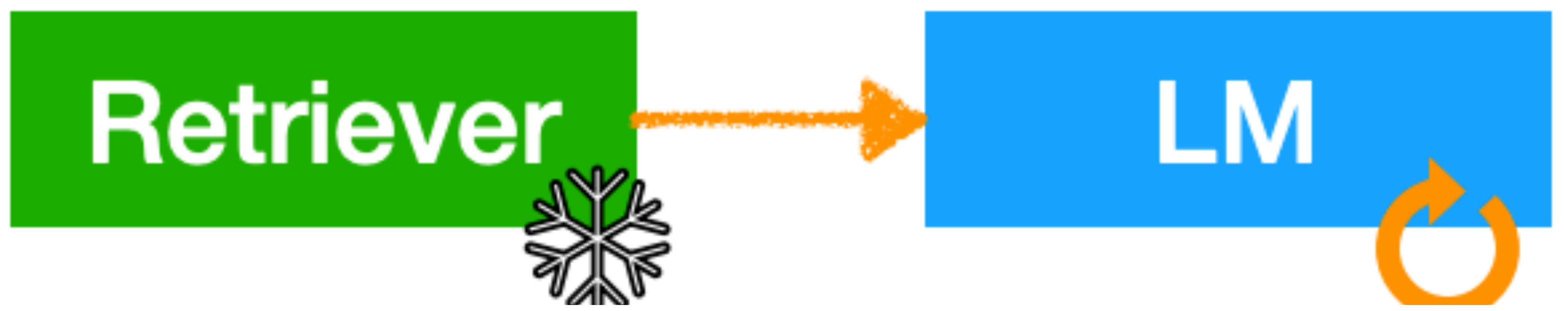
Different training methods

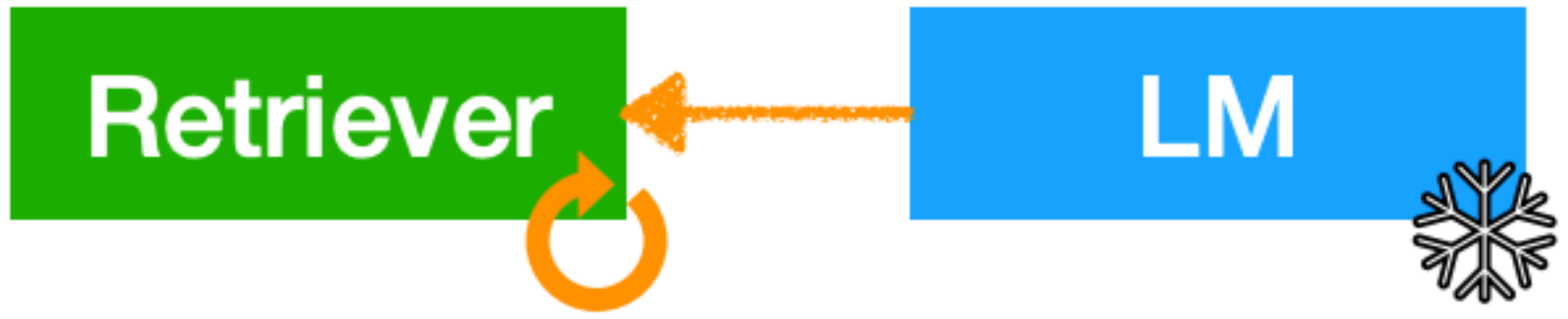
- Independent training
- Sequential training
- Joint training w/ asynchronous index update
- Joint training w/ in-batch approximation

- Training language models

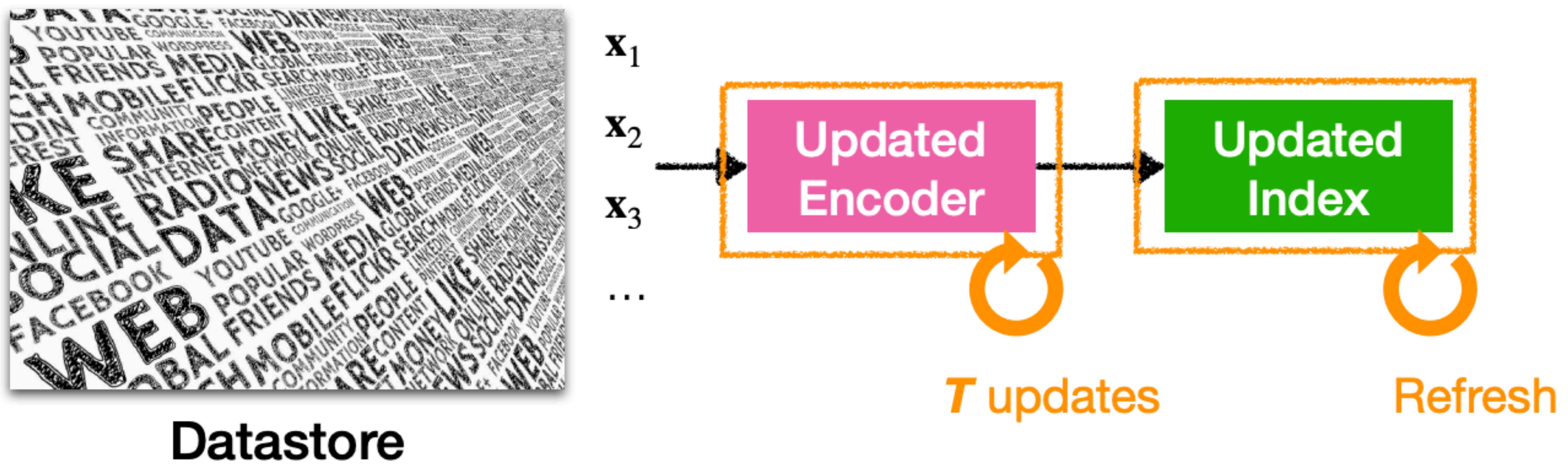
- Training retrieval models







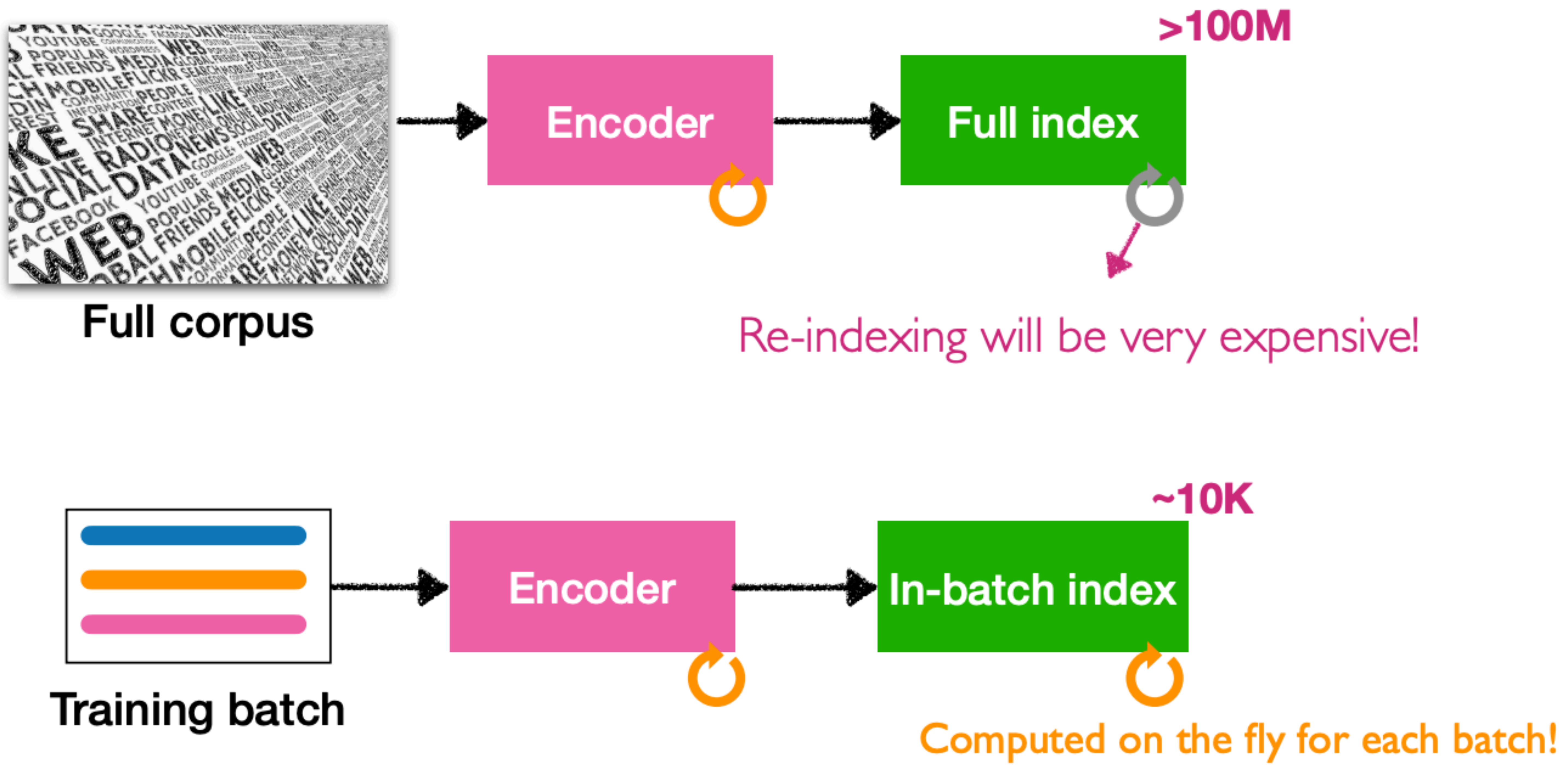
Training with asynchronous index updates



Examples: REALM (Guu et al., ICML'20), ATLAS (Izacard et al., JMLR'23)

Training with in-batch approximations

In-batch approximation



Training with in-batch approximations

Training Language Models with Memory Augmentation

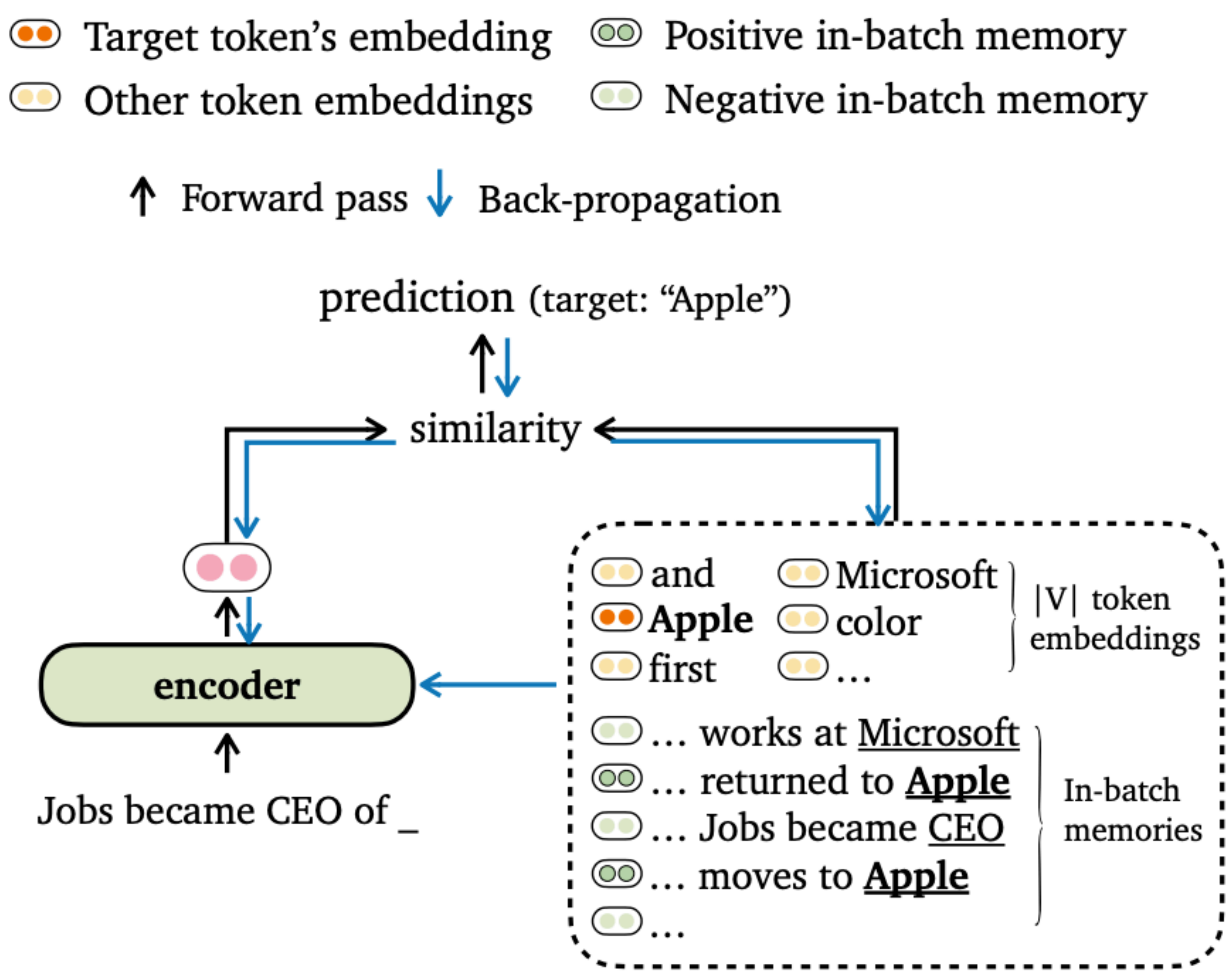
Zexuan Zhong[†] **Tao Lei**[*] **Danqi Chen**[†]
[†]Princeton University
{zzhong, danqic}@cs.princeton.edu, taole@google.com

TRIME (Zhong et al., EMNLP'22)

Nonparametric Masked Language Modeling

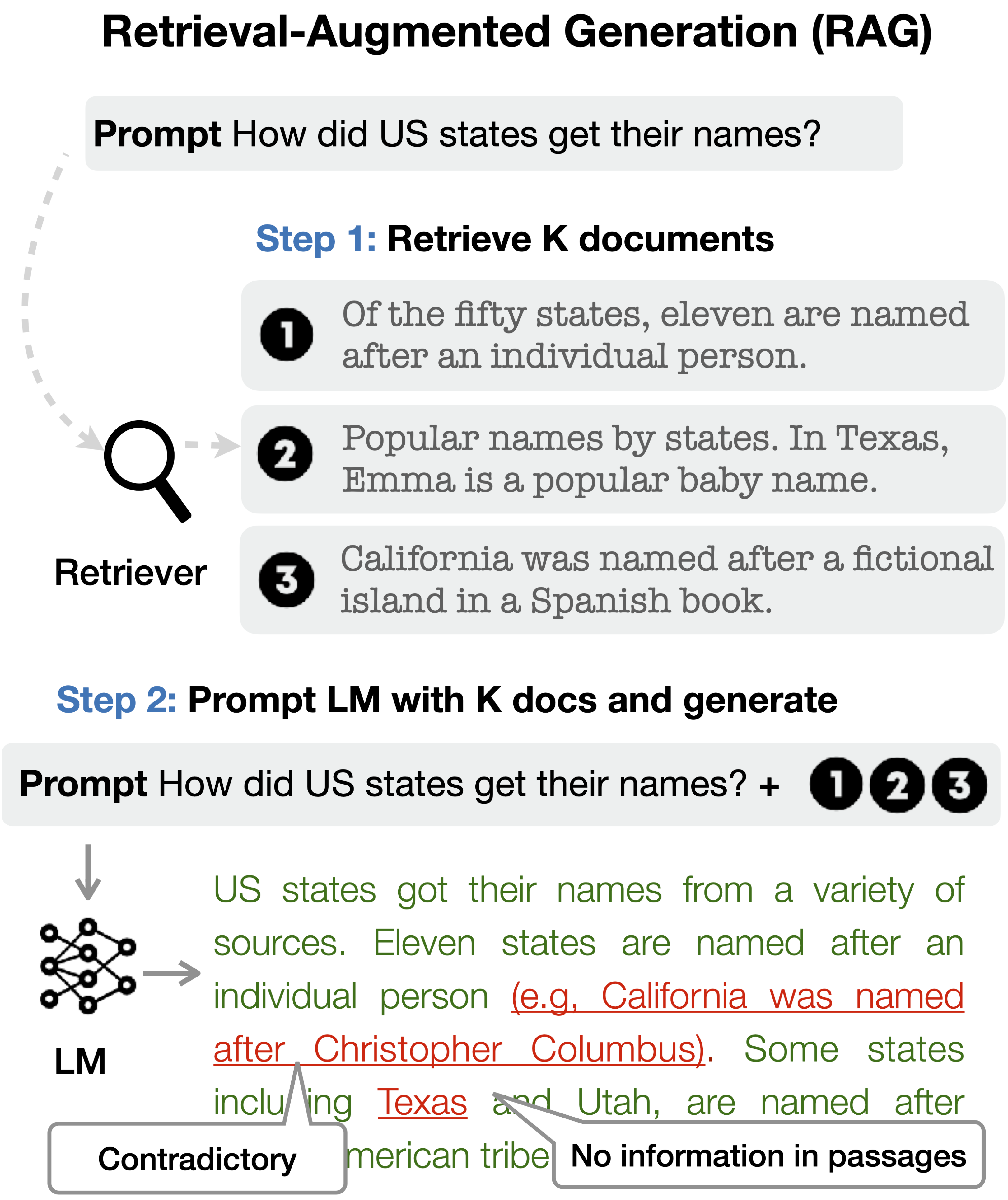
Sewon Min[1,2] **Weijia Shi**[1,2] **Mike Lewis**[2] **Xilun Chen**[2]
Wen-tau Yih[2] **Hannaneh Hajishirzi**[1,3] **Luke Zettlemoyer**[1,2]
[1]University of Washington [2]Meta AI [3]Allen Institute for AI
{sewon,swj0419,hannaneh,lsz}@cs.washington.edu
{mikelewis,xilun,scottyih}@meta.com

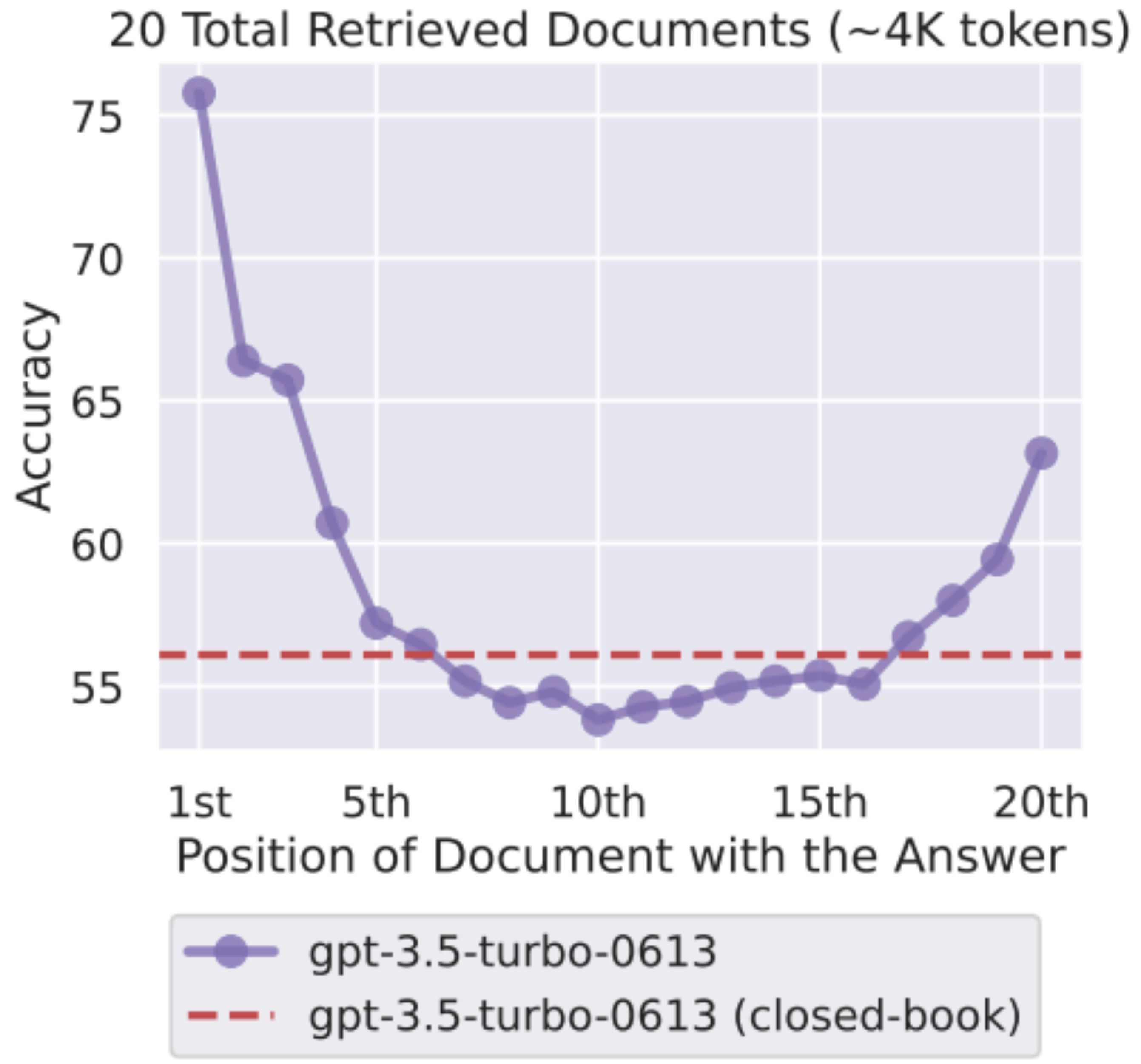
NPM (Min et al., ACL'23 Findings)



Advanced “frozen” RAG frameworks (path #2)

Key challenges



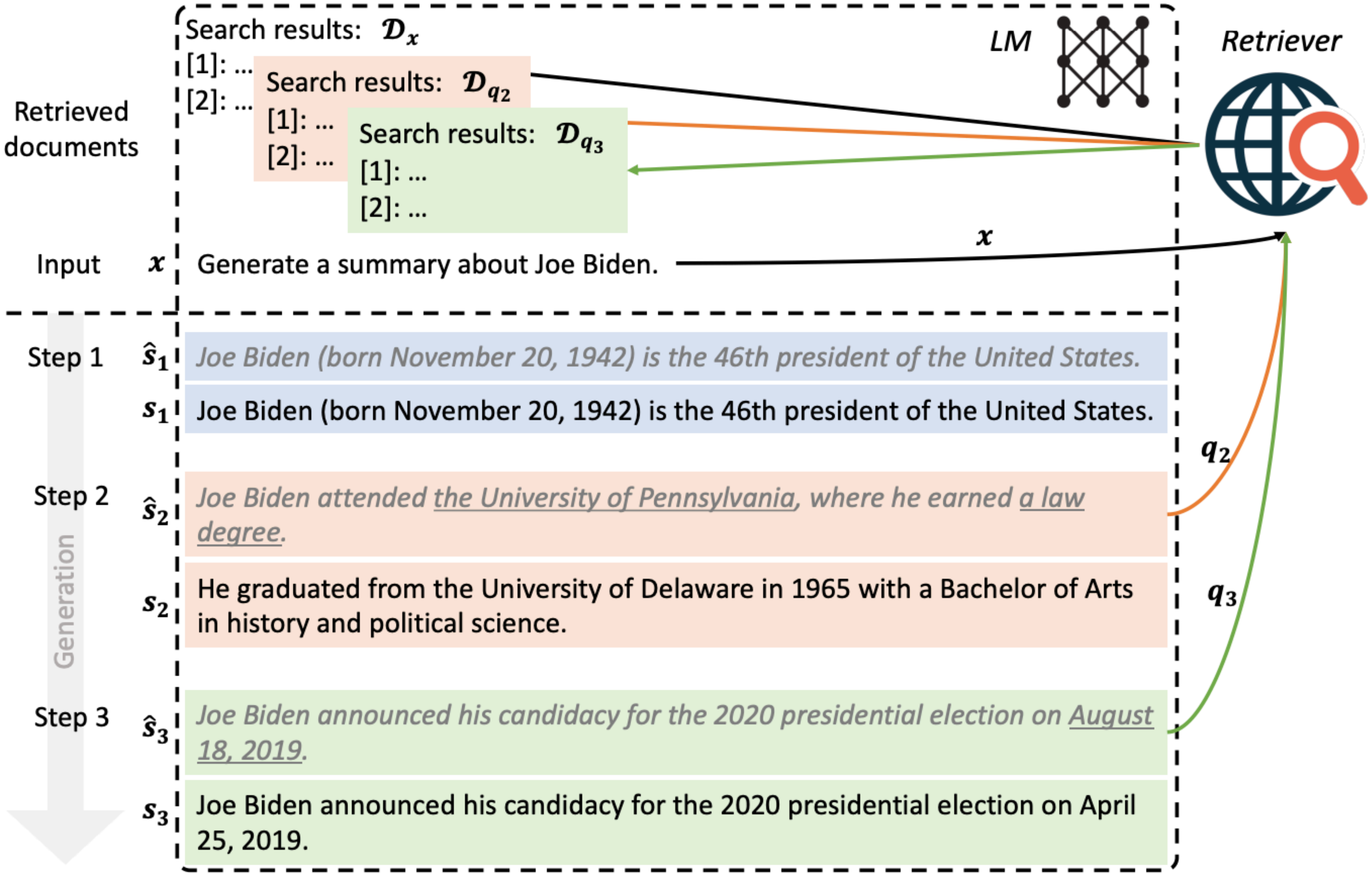


#1. Can LMs **discern** relevant and irrelevant passages?

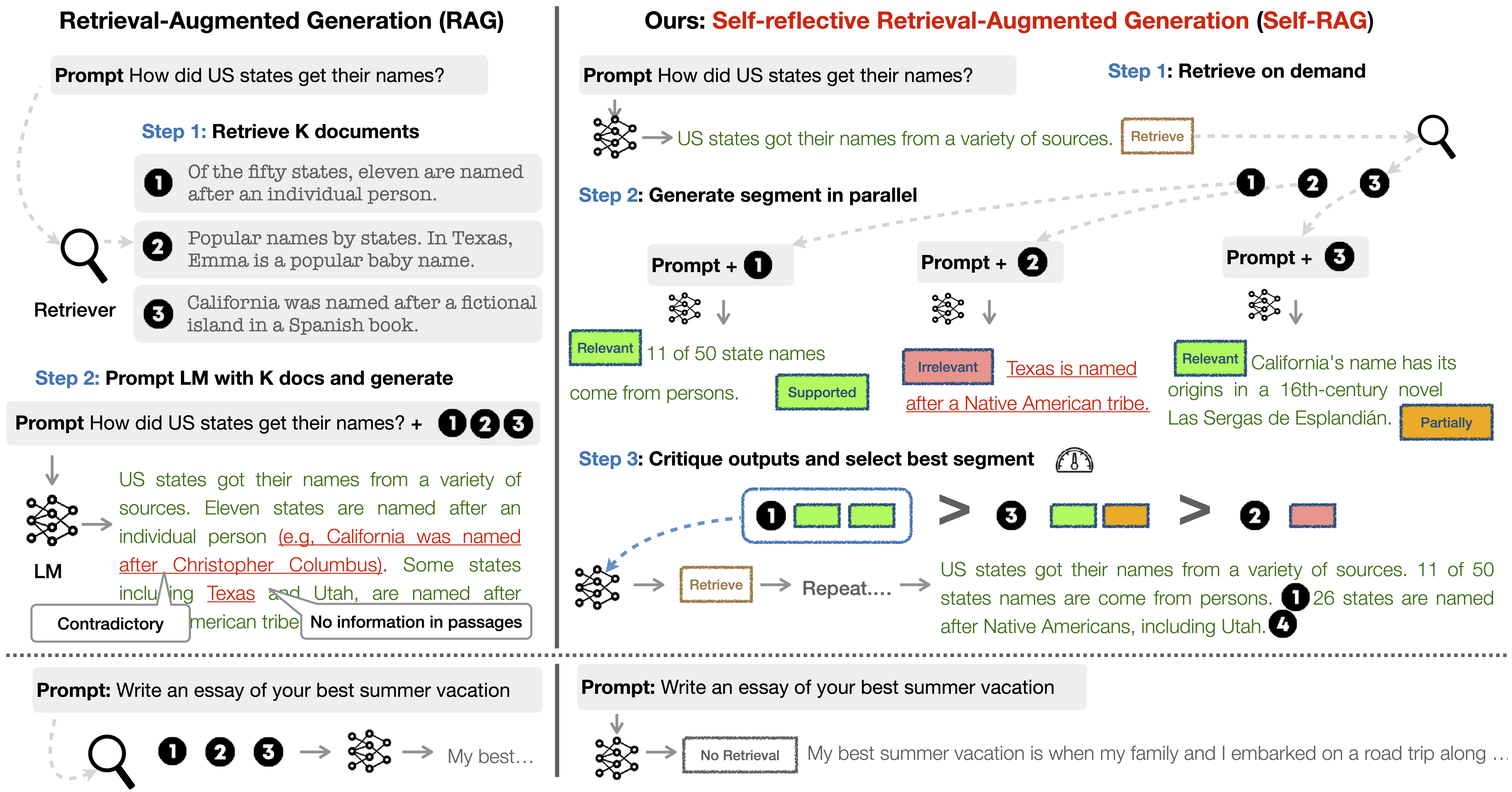
#2. Can LMs leverage all the contexts **effectively**?

#3. Can LMs **synthesize** information from different passages just in context?

Active Retrieval Augmented Generation (FLARE)



Self-Reflective Retrieval-Augmented Generation (Self-RAG)



Self-RAG (Asai et al., ICLR'24)