Retrieval-based LMs-II

Large Language Models: Introduction and Recent Advances

ELL881 · AIL821

Yatin Nandwani
Research Scientist, IBM Research

Closed Book vs Open Book Exams

Parametric LLMs

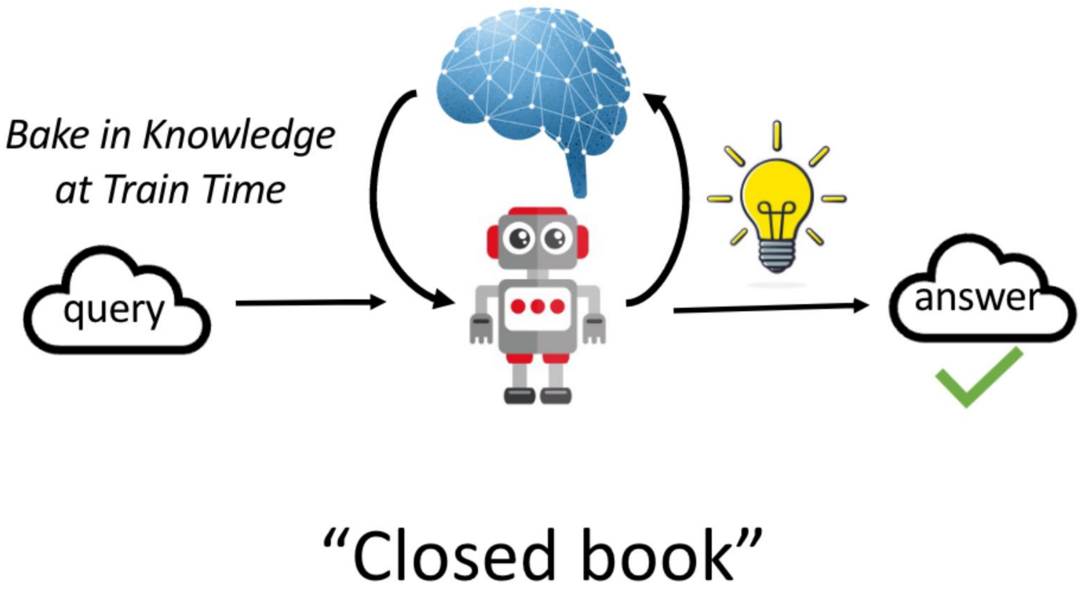


Image source: http://arxiv.org/abs/2403.10131

Closed Book vs Open Book Exams

Parametric LLMs

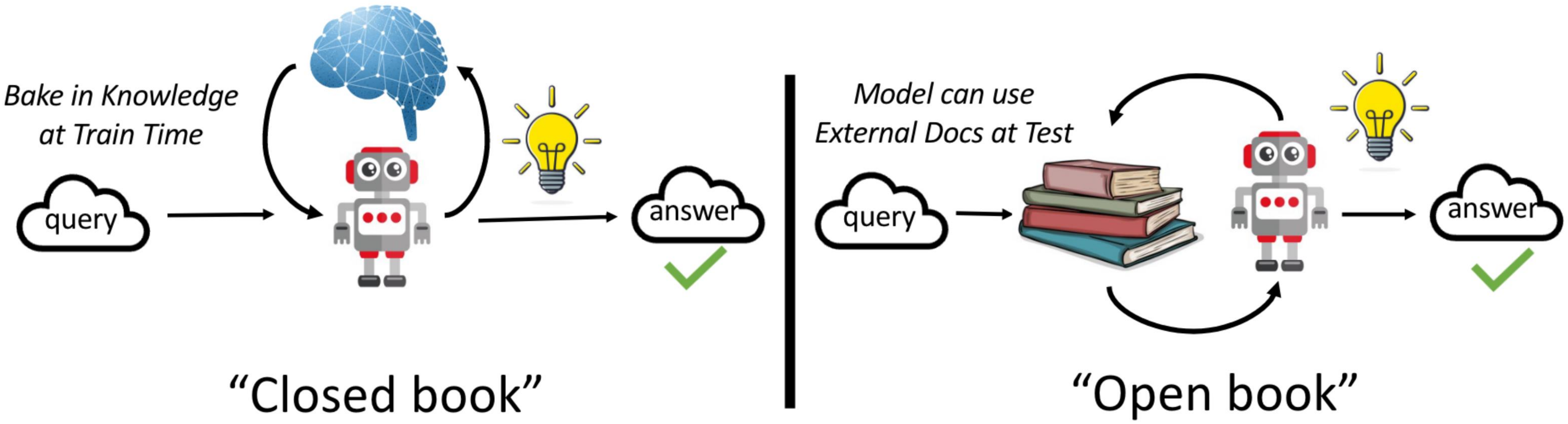


Image source: http://arxiv.org/abs/2403.10131

Closed Book vs Open Book Exams

Parametric LLMs

Retrieval-based LLMs

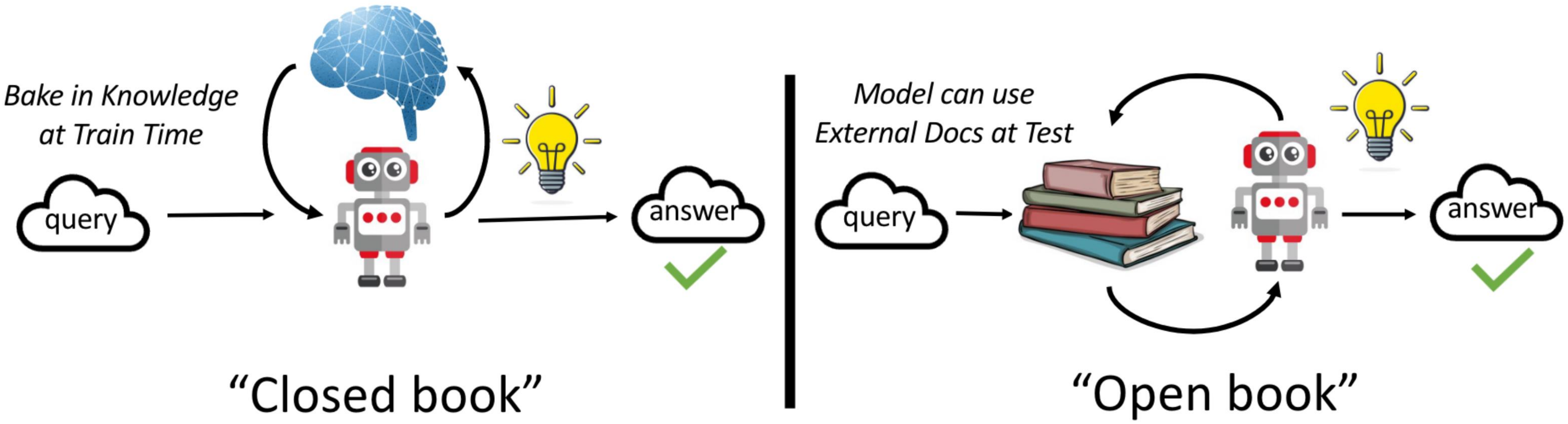
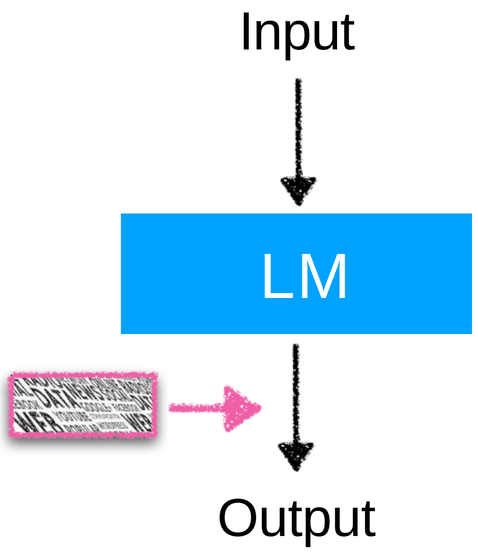


Image source: http://arxiv.org/abs/2403.10131

How to use the Book?

- Output interpolations - After solving the question yourself?

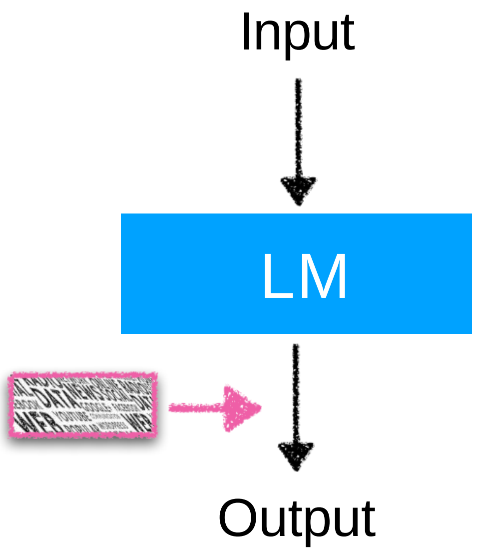


Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

How to use the Book?

- Output interpolations - After solving the question yourself?

kNN LMs

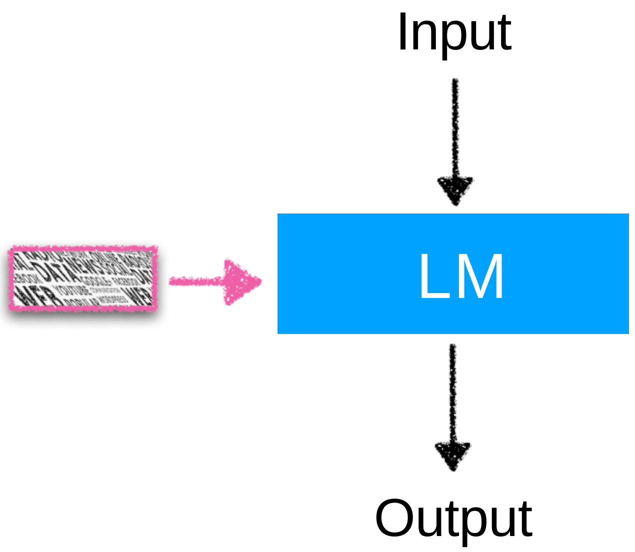


Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

How to use the Book?

- Output interpolations - After solving the question yourself?

- Intermediate fusion – modify the LM architecture to be aware of the book?



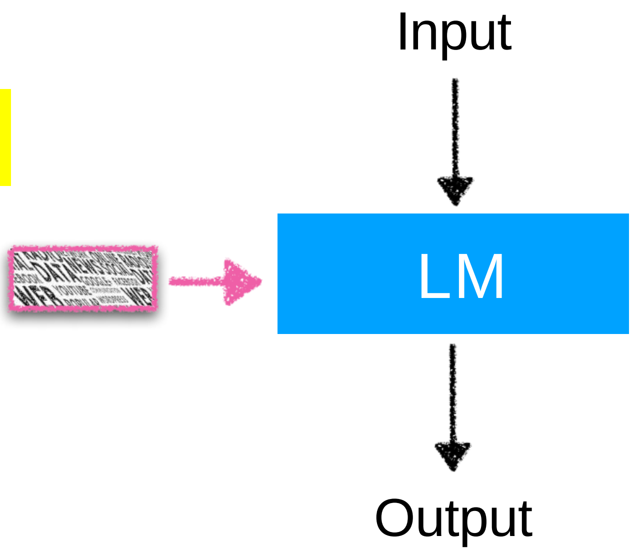
Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

How to use the Book?

- Output interpolations - After solving the question yourself?

- Intermediate fusion – modify the LM architecture to be aware of the book?

RETRO



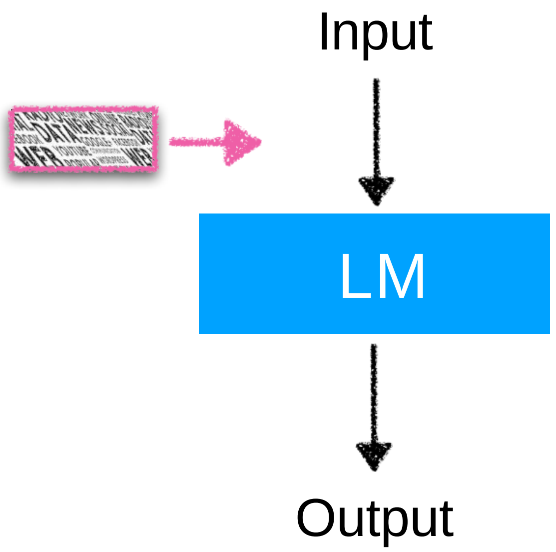
Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

How to use the Book?

- Output interpolations - After solving the question yourself?

- Intermediate fusion – modify the LM architecture to be aware of the book?

- Input augmentation (RAG) - Before you start solving?



Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

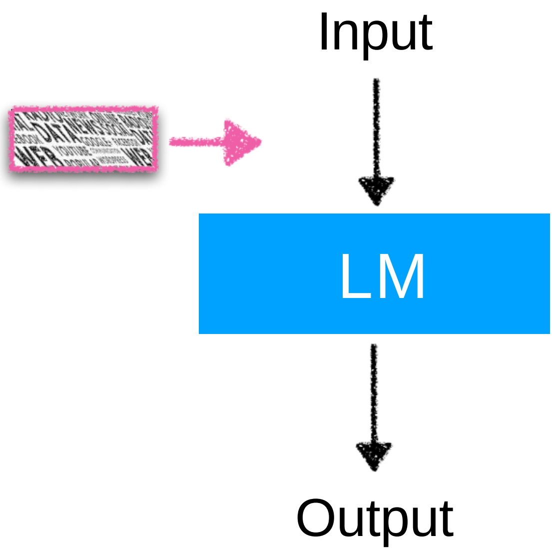
How to use the Book?

- Output interpolations - After solving the question yourself?

- Intermediate fusion – modify the LM architecture to be aware of the book?

- Input augmentation (RAG) - Before you start solving?

RAG,
REALM



Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

How to use the Book?

- Output interpolations - After solving the question yourself?

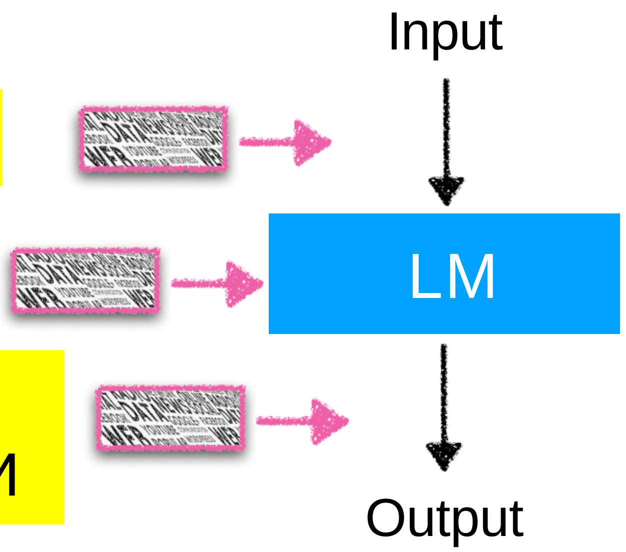
kNN LMs

- Intermediate fusion – modify the LM architecture to be aware of the book?

RETRO

- Input augmentation (RAG) - Before you start solving?

RAG, REALM



Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

How to read (fetch information from) the book?

Outline

- Motivation
  - Drawbacks of Parametric LLMs – *hallucination, verification …*
  - Motivating Retrieval-based LLMs – *close book vs open book*
- Retrieval Methods – *sparse, dense, reranking, black-box*
- kNN, RETRO, REALM, RAG – *seminal works*
- Overview of Training Techniques – *independent, sequential, joint training …*
- Limitations – *lost in the middle, still hallucinating, retriever failures …*

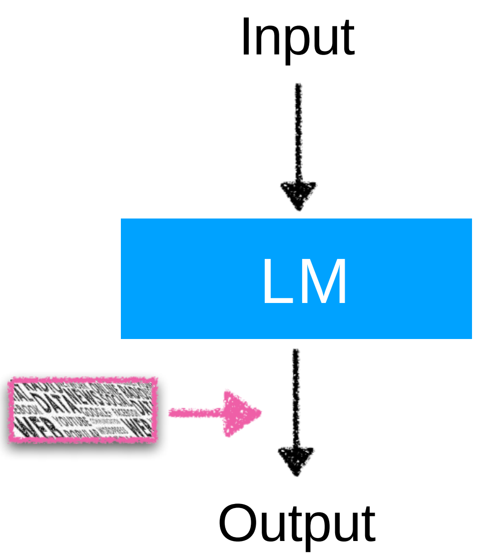
Outline

- Motivation
  - Drawbacks of Parametric LLMs – *hallucination, verification …*
  - Motivating Retrieval-based LLMs – *close book vs open book*
- Retrieval Methods – *sparse, dense, reranking, black-box*
- kNN, RETRO, REALM, RAG – *seminal works*
- Overview of Training Techniques – *independent, sequential, joint training …*
- Limitations – *lost in the middle, still hallucinating, retriever failures …*

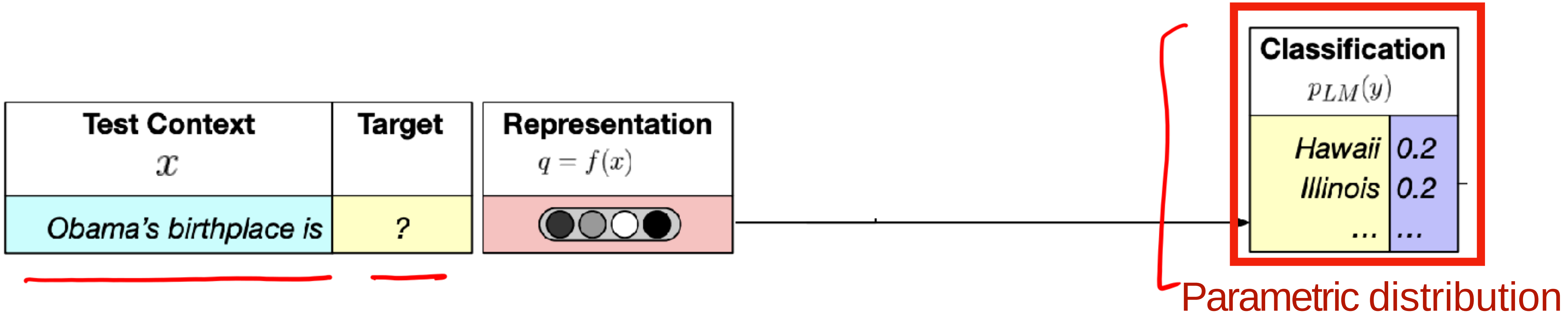
How to use the Book?

- Output interpolations - After solving the question yourself?

kNN LMs



kNN-LM (Khandelwal et al. 2020)

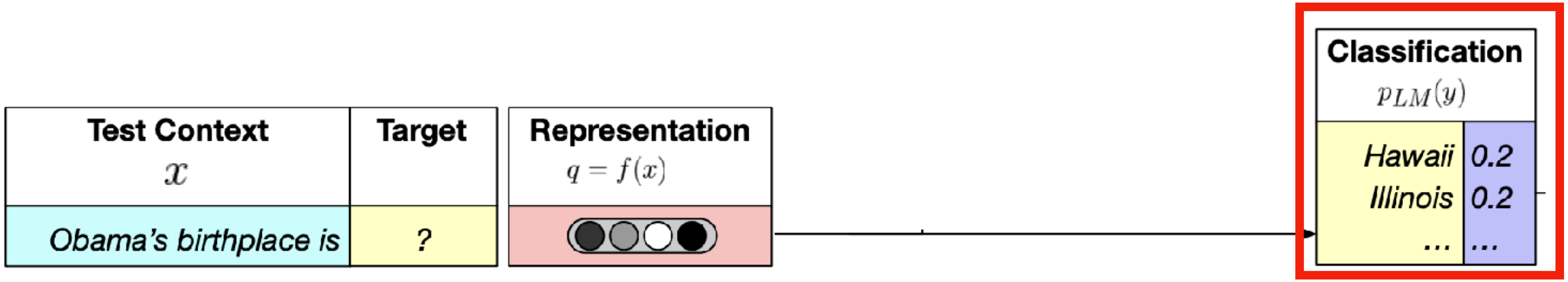


Khandelwal et al. Generalization through Memorization: Nearest Neighbor Language Models. ICLR 2020.

Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

kNN-LM (Khandelwal et al. 2020)

… Obama was born in Hawaii, and graduated from Columbia University. … Obama is a native of Hawaii, …. … Obama was senator for Illinois from 1997 to 2005, …. Barack is Married to Michelle and their first daughter,



Parametric distribution

Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

kNN-LM (Khandelwal et al. 2020)

Indexed Contexts (keys)	Values
Obama	*was*
Obama was	*senator*
Obama was born	*in*
Obama was born in	*Hawaii*
Obama was born in Hawaii	*and*
Obama was born in Hawaii and	*graduated*

… Obama was born in Hawaii, and graduated from Columbia University. … Obama is a native of Hawaii, ….
… Obama was senator for Illinois from 1997 to 2005,
…. Barack is Married to Michelle and their first daughter,

Test Context x	Target	Representation $q = f(x)$
Obama's birthplace is	?	

Classification $p_{LM}(y)$

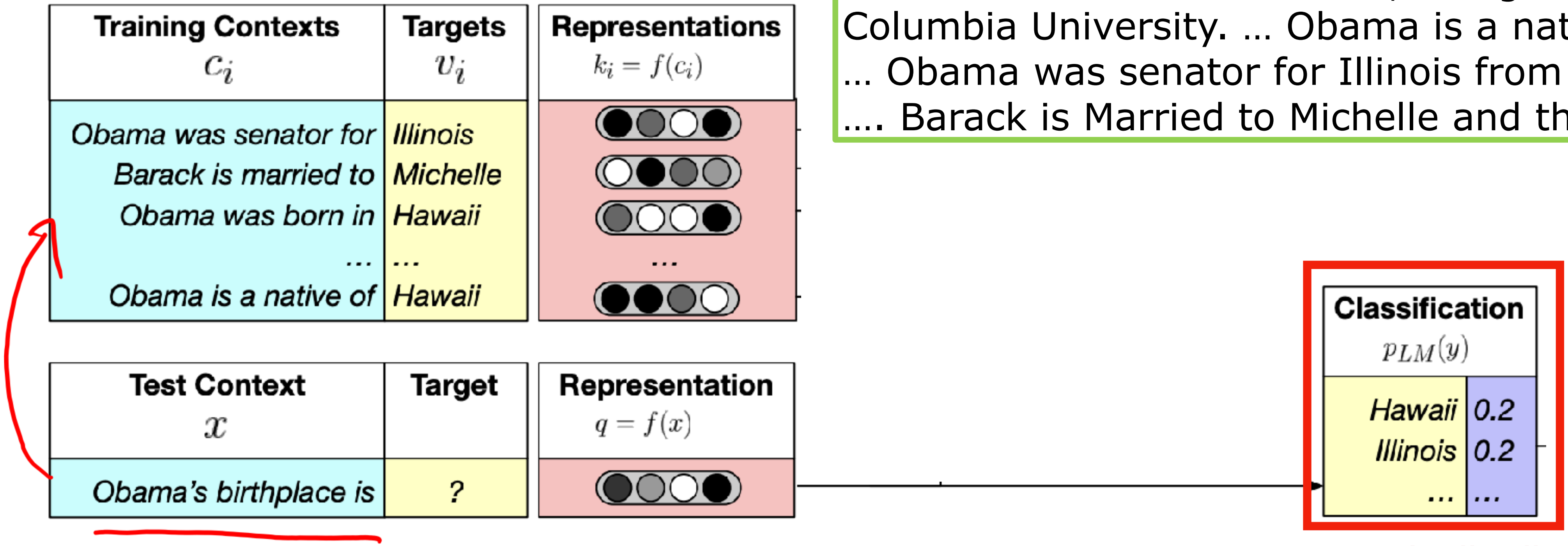
Hawaii	0.2
Illinois	0.2
…	…

Parametric distribution

Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

kNN-LM (Khandelwal et al. 2020)

… Obama was born in Hawaii, and graduated from Columbia University. … Obama is a native of Hawaii, …. … Obama was senator for Illinois from 1997 to 2005, …. Barack is Married to Michelle and their first daughter,



Parametric distribution

Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

kNN-LM (Khandelwal et al. 2020)

The size of the datastore =
# of tokens in the corpus (>1B)

Training Contexts c_i	Targets v_i	Representations $k_i = f(c_i)$
Obama was senator for	*Illinois*	
Barack is married to	*Michelle*	
Obama was born in	*Hawaii*	
...	...	...
Obama is a native of	*Hawaii*	

Test Context x	Target	Representation $q = f(x)$
Obama's birthplace is	?	

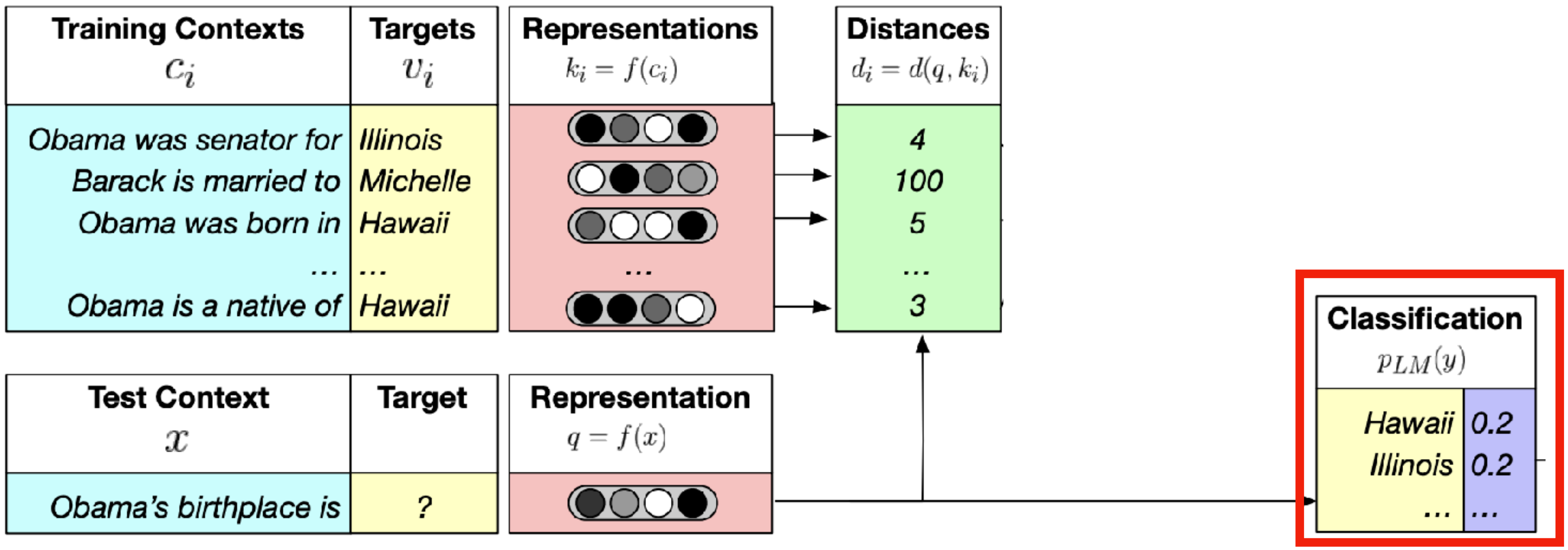
… Obama was born in Hawaii, and graduated from Columbia University. … Obama is a native of Hawaii, …. … Obama was senator for Illinois from 1997 to 2005, …. Barack is Married to Michelle and their first daughter,

Classification $p_{LM}(y)$	
Hawaii	0.2
Illinois	0.2
...	...

Parametric distribution

Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

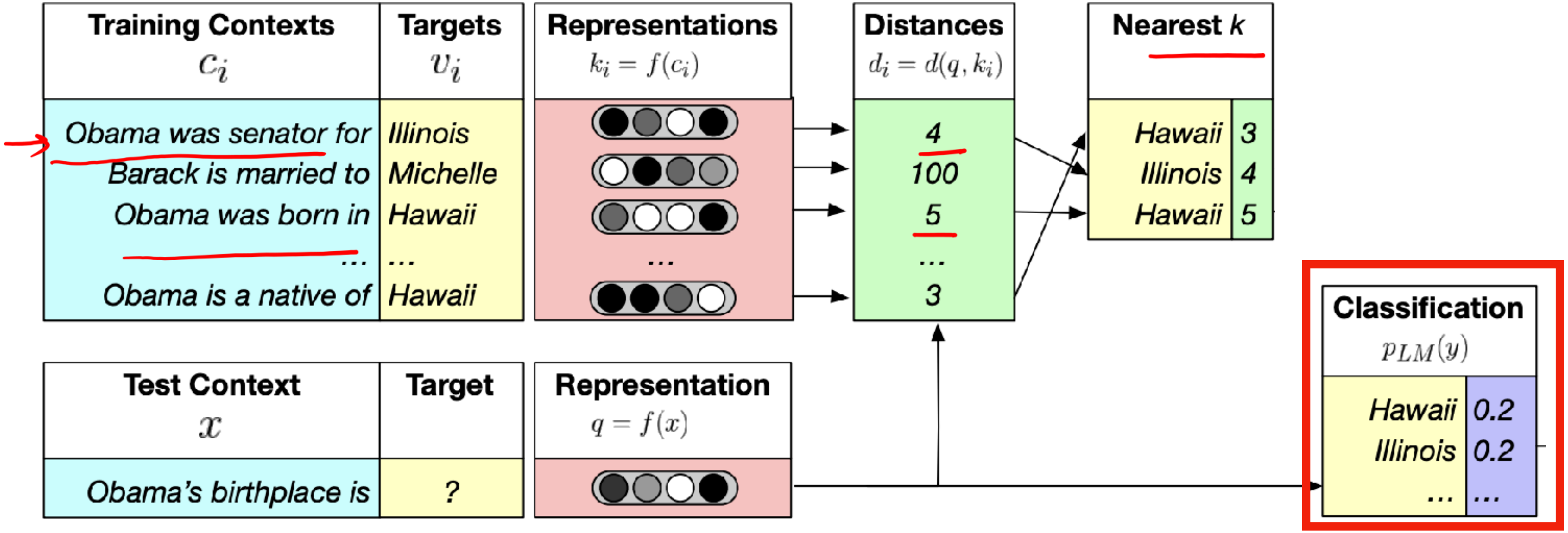
kNN-LM (Khandelwal et al. 2020)



Parametric distribution

Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

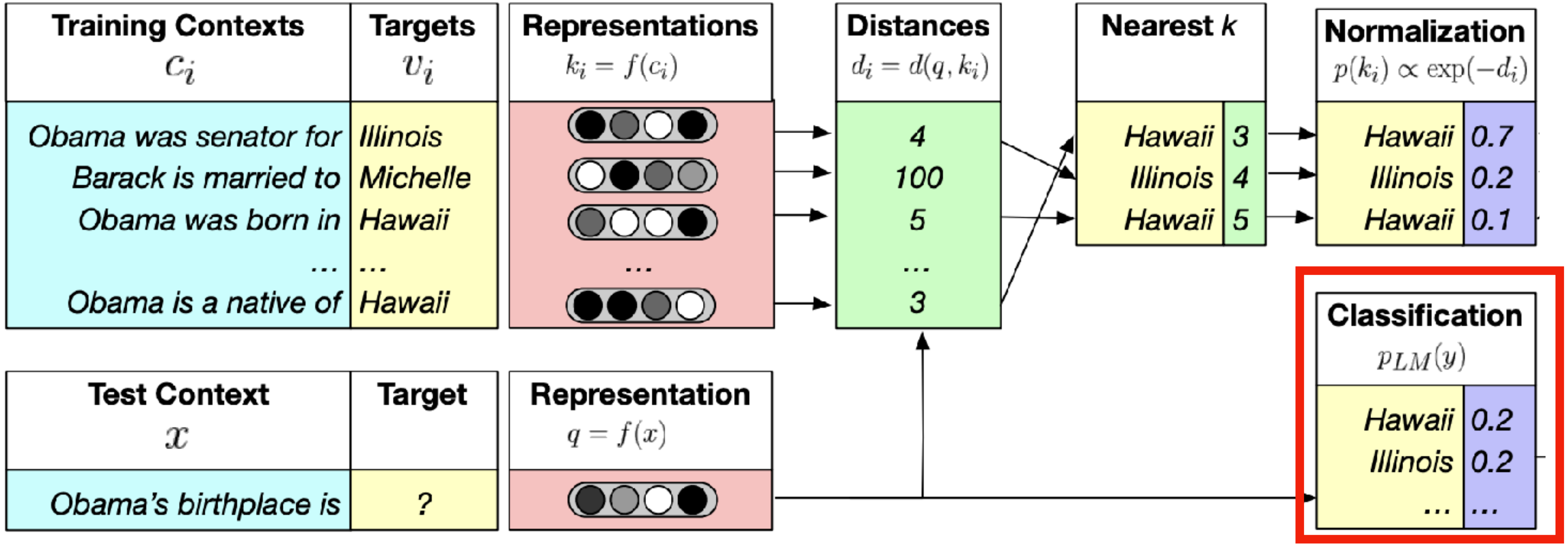
kNN-LM (Khandelwal et al.2020)



Parametric distribution

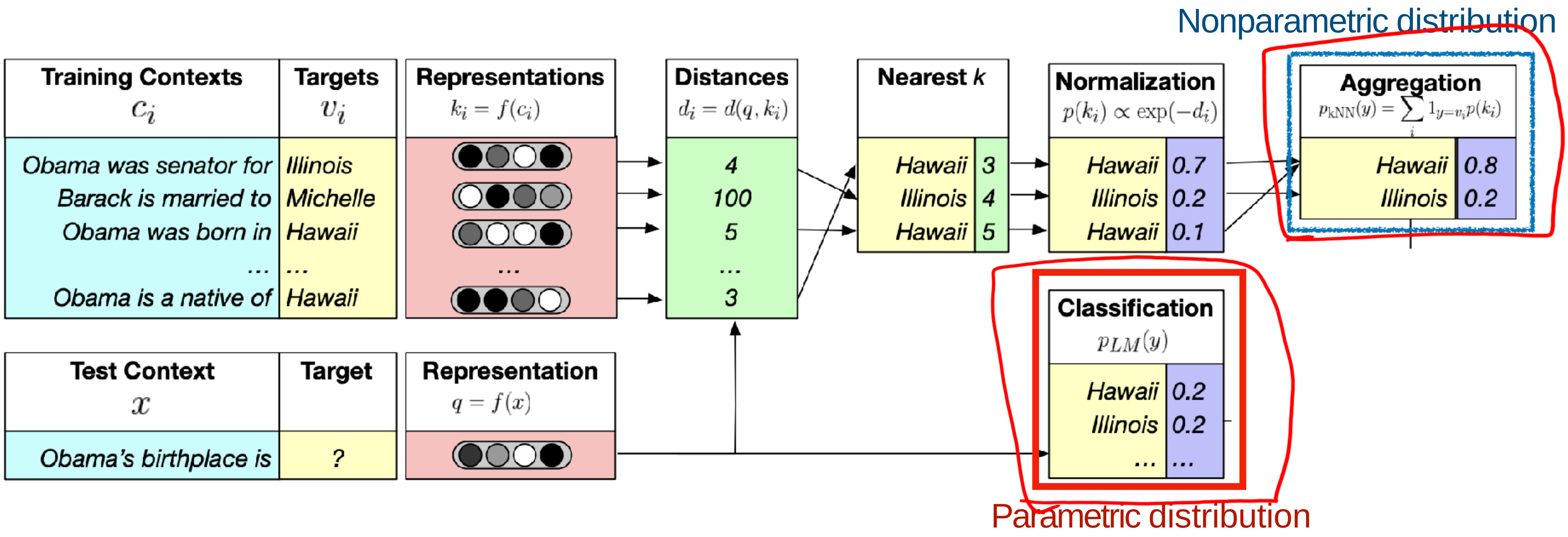
Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

kNN-LM (Khandelwal et al. 2020)

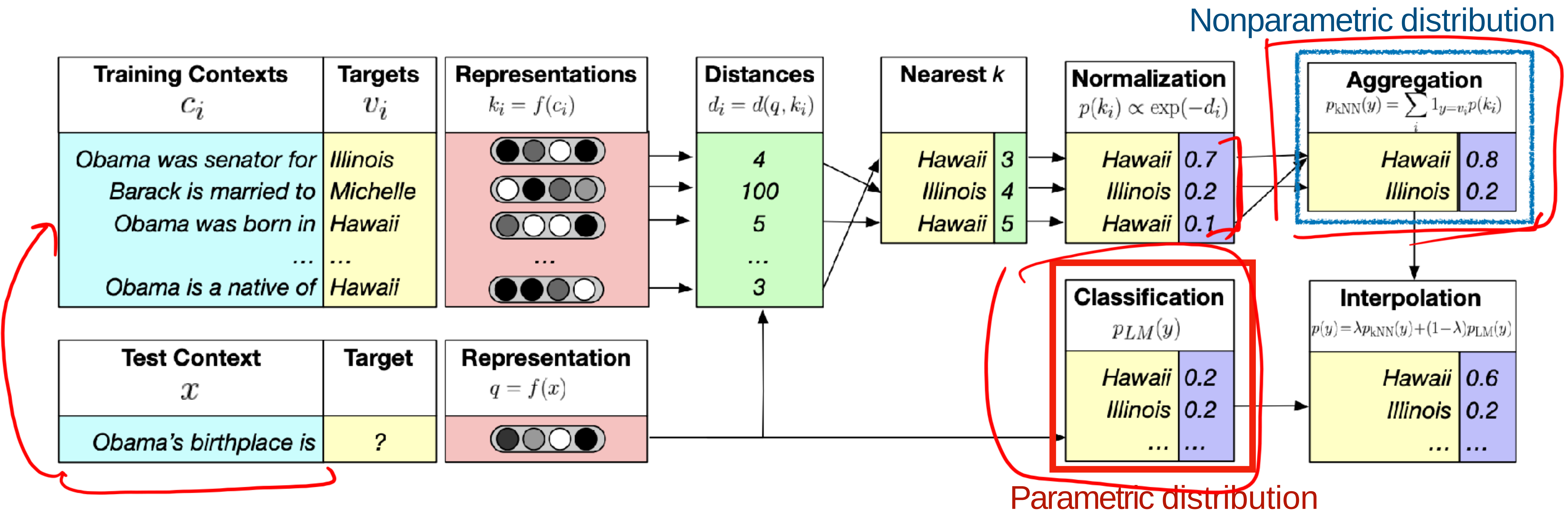


Parametric distribution

kNN-LM (Khandelwal et al. 2020)



kNN-LM (Khandelwal et al. 2020)



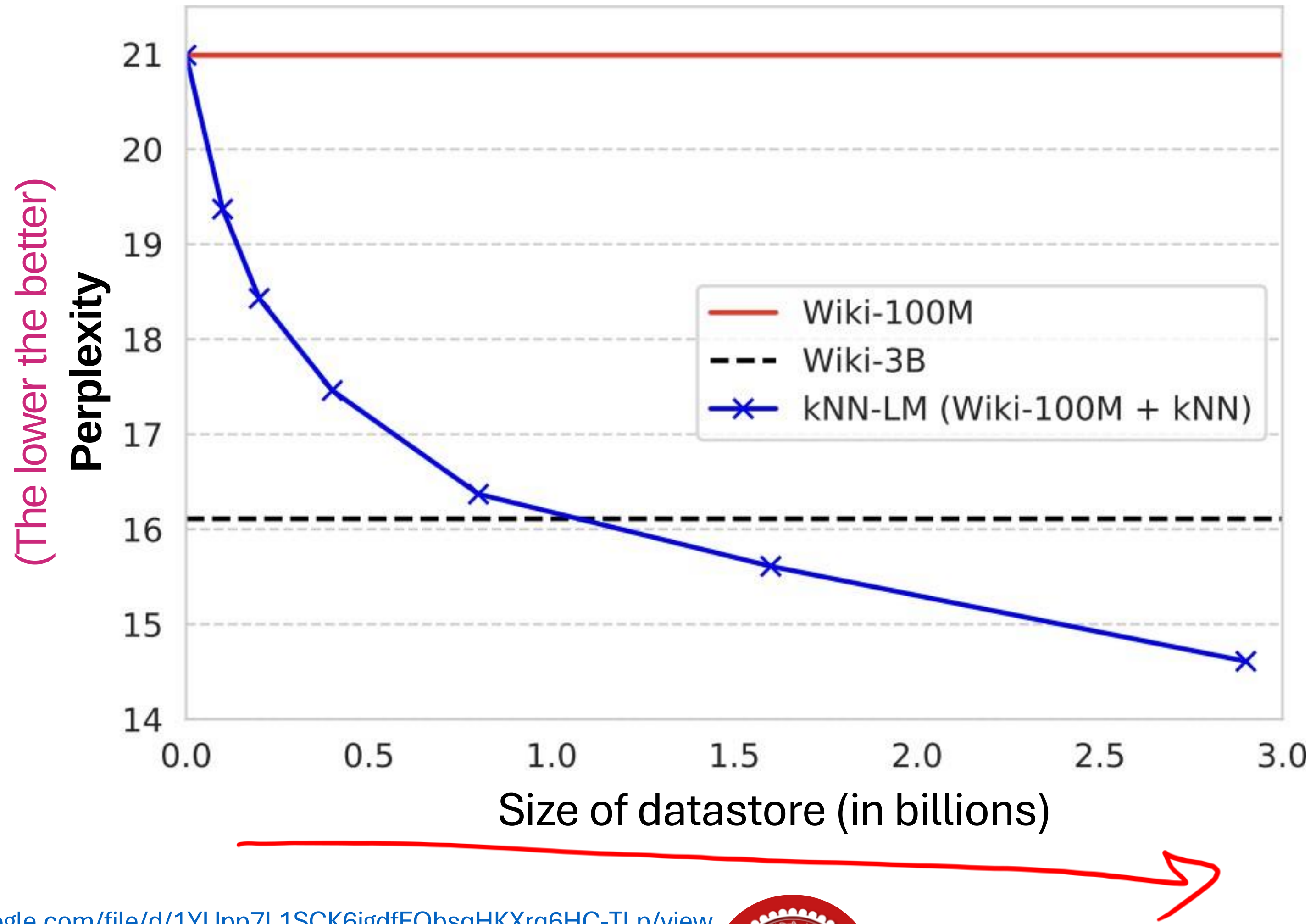
λ : hyperparameter

$$P_{k\mathrm{NN-LM}}(y|x) = (1-\lambda)P_{\mathrm{LM}}(y|x) + \lambda P_{k\mathrm{NN}}(y|x)$$

kNN-LM – Computational Cost

- **Key embedding computation**
  - Single forward pass over the data – fraction of cost of training for one epoch
- **Building cache using FAISS index**
  - 103M – 2 hours on a single CPU
- **Inference overhead**
  - 250K tokens: 25 minutes with $k = 1024$

kNN-LM - results



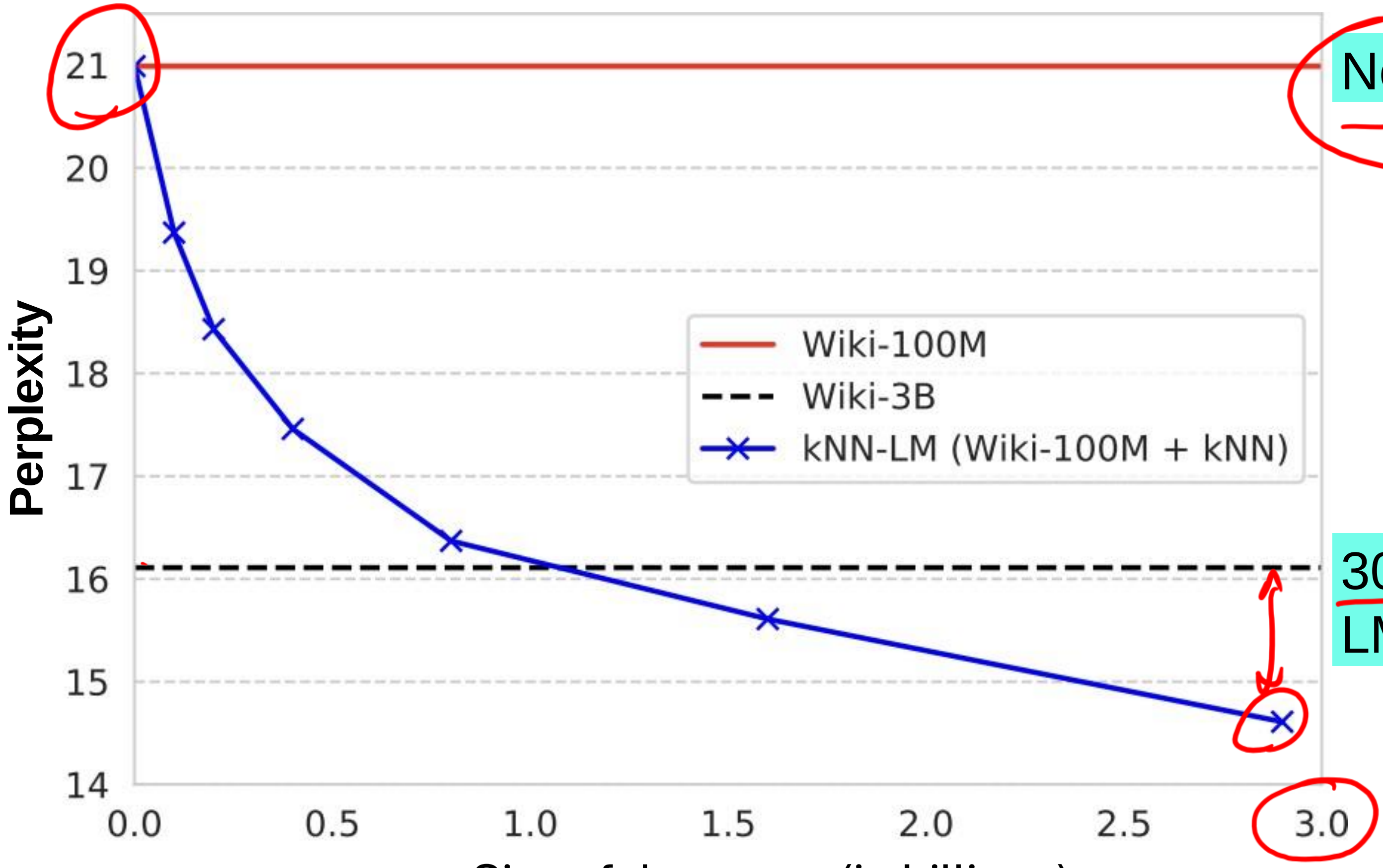
Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

kNN-LM - results

103M

No-retrieval LM

(The lower the better)

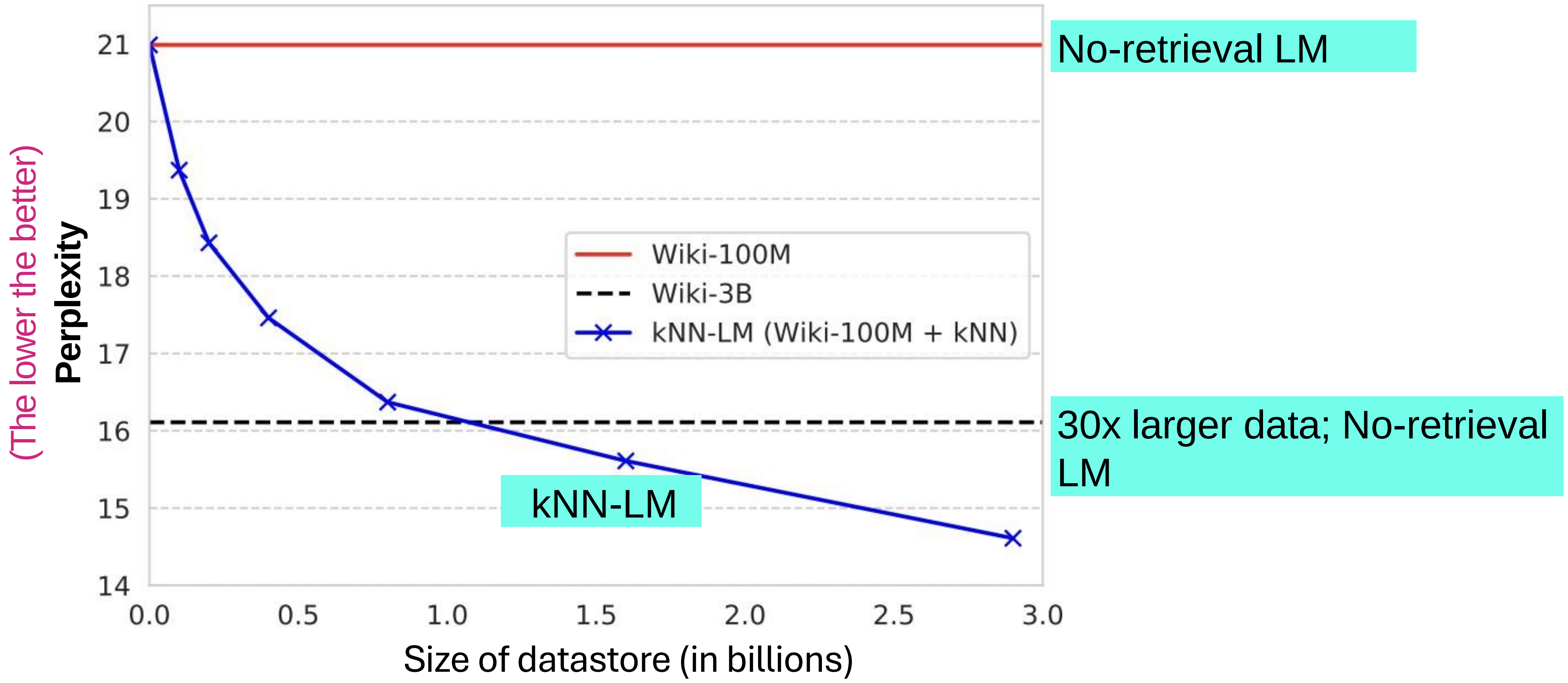


Size of datastore (in billions)

30x larger data; No-retrieval LM

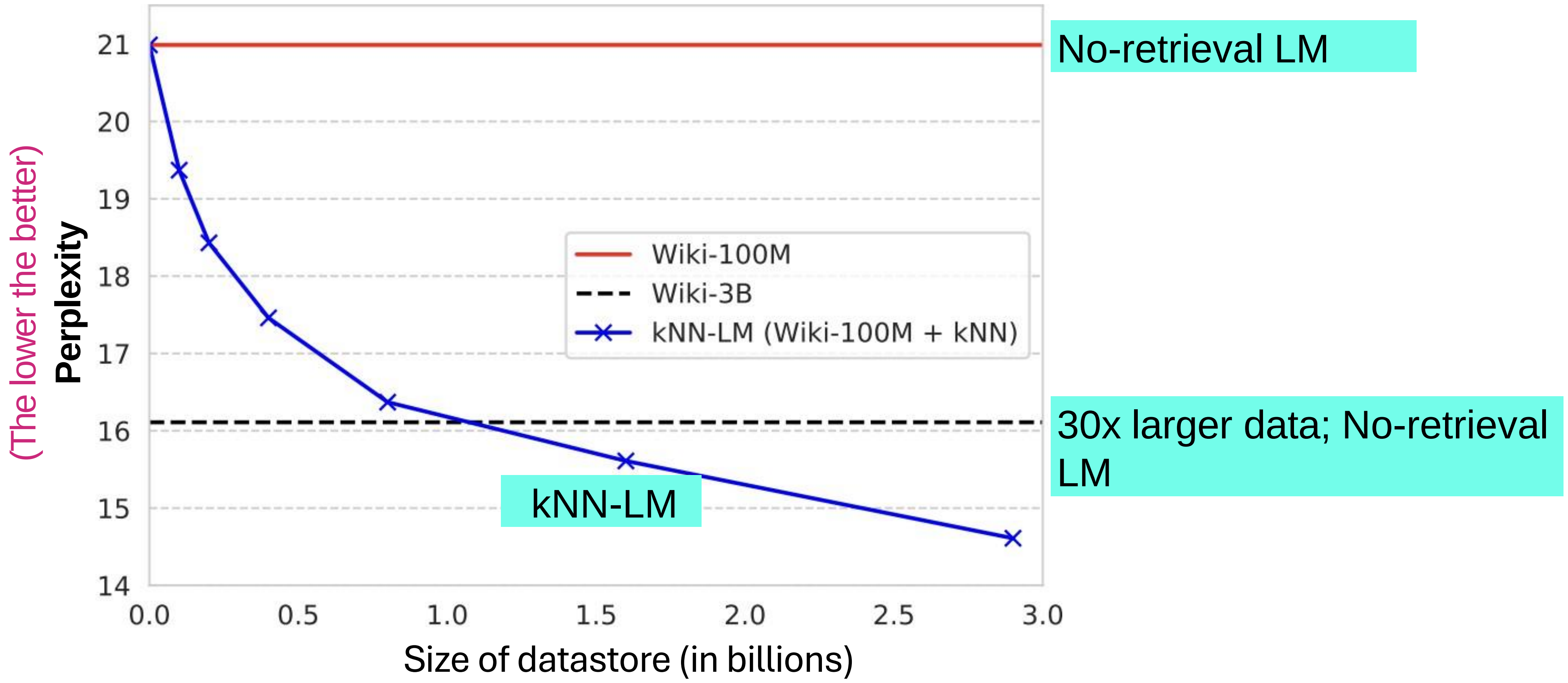
Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

kNN-LM - results



Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

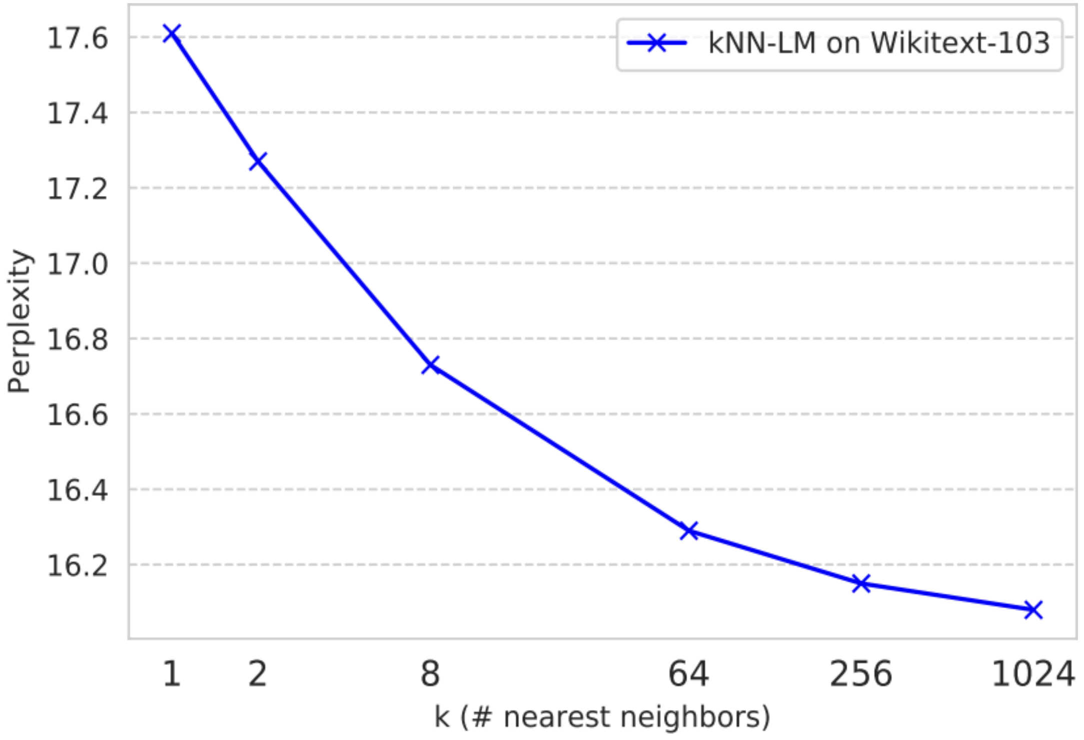
kNN-LM - results



Outperforms no-retrieval LM

Better with bigger datastore

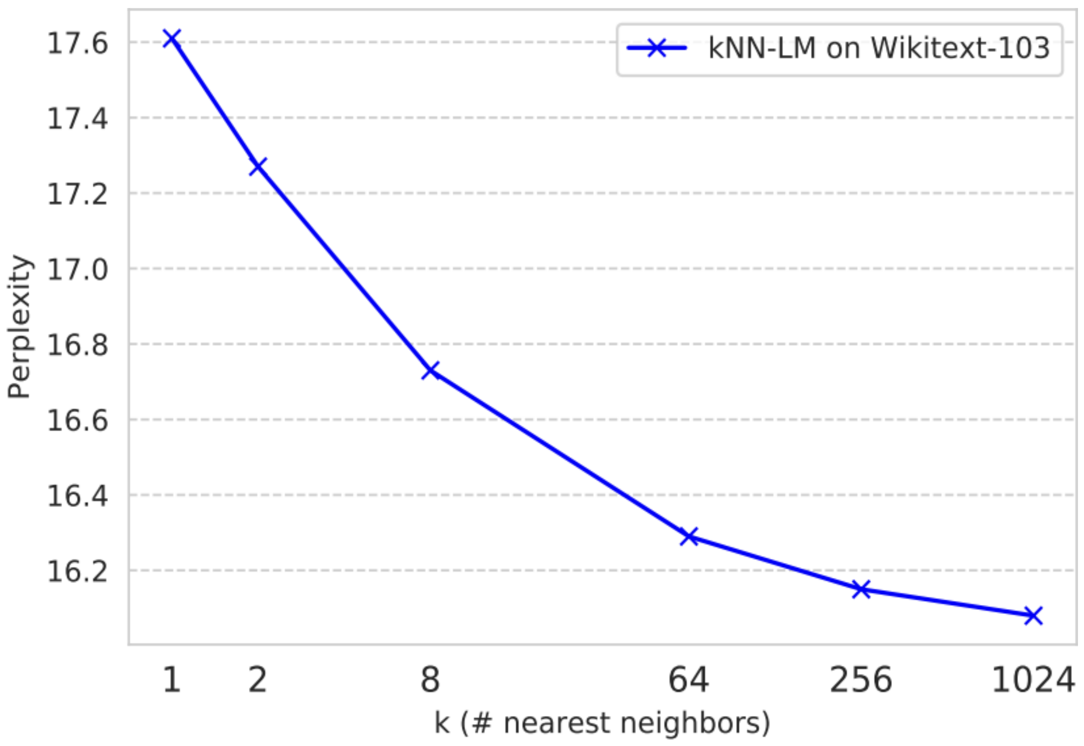
kNN-LM - results

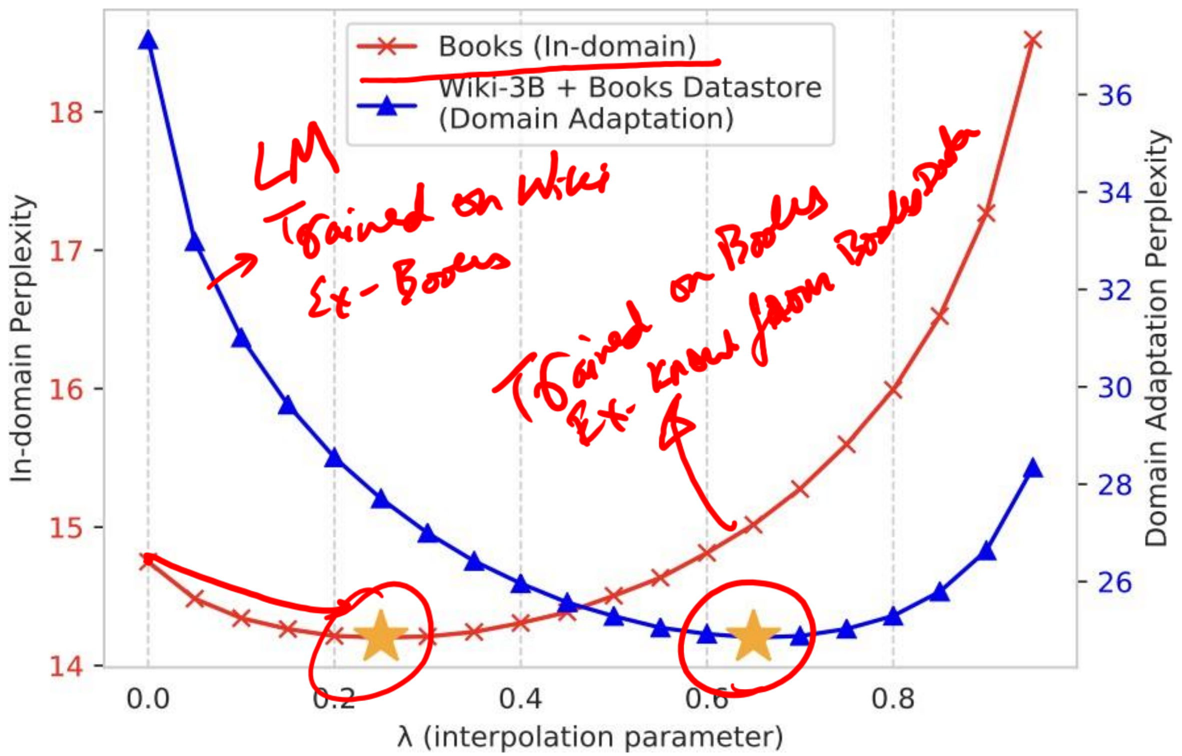


Better with bigger ***k***

kNN-LM - results

LM → Wiki 103M → Training
Books ⇔ Test





Better with bigger ***k***

Helps more out-of-domain

kNN-LM: how to finetuning on downstream tasks?

- In LM task, "input" is a sub-sentence, and "output" is the "next word".
- kNN-LM organizes the "unstructured knowledge" as "input-output" pairs.

Indexed Contexts (keys)	Values
Obama	*was*
Obama was	*senator*
Obama was born	*in*
Obama was born in	*Hawaii*
Obama was born in Hawaii	*and*
Obama was born in Hawaii and	*graduated*

… Obama was born in Hawaii, and graduated from Columbia University. … Obama is a native of Hawaii, …. … Obama was senator for Illinois from 1997 to 2005, …. Barack is Married to Michelle and their first daughter,

kNN-LM: how to finetuning on downstream tasks?

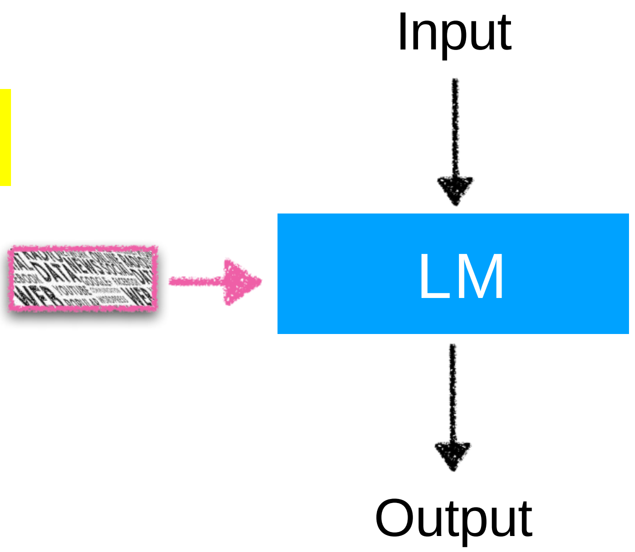
- In LM task, “input” is a sub-sentence, and “output” is the “next word”.
- kNN-LM organizes the “unstructured knowledge” as “input-output” pairs.
- We search for the most similar “input” (sub-sentence) in the corpus, and use its corresponding “output”.
- How to fine-tune for a downstream task?
- Would need the most similar “input”
  - i.e., need examples labeled for the target task
  - Not clear how to organize unstructured text as input-output pairs for the desired task

How to use the Book?

- Output interpolations - After solving the question yourself?

- Intermediate fusion – modify the LM architecture to be aware of the book?

RETRO



Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

RETRO - Retrieval-Enhanced Transformer

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

RETRO - Retrieval-Enhanced Transformer

Indexed Contexts (keys)	Values
Obama	*was*
Obama was	*senator*
Obama was born	*in*
Obama was born in	*Hawaii*

… Obama was born in Hawaii, and graduated from Columbia University. Obama is a native of Hawaii, Obama was senator for Illinois from 1997 to 2005. Barack is Married to Michelle and their first daughter…

In kNN LM

- We have an indexed key in kNN LM for each token.

RETRO - Retrieval-Enhanced Transformer

Indexed Contexts (keys)	Values
Obama	*was*
Obama was	*senator*
Obama was born	*in*
Obama was born in	*Hawaii*

… Obama was born in Hawaii, and graduated from Columbia University. Obama is a native of Hawaii, Obama was senator for Illinois from 1997 to 2005. Barack is Married to Michelle and their first daughter…

In kNN LM

- We have an indexed key in kNN LM for each token. This causes two issues:

RETRO - Retrieval-Enhanced Transformer

Indexed Contexts (keys)	Values
Obama	*was*
Obama was	*senator*
Obama was born	*in*
Obama was born in	*Hawaii*

… Obama was born in Hawaii, and graduated from Columbia University. Obama is a native of Hawaii, Obama was senator for Illinois from 1997 to 2005. Barack is Married to Michelle and their first daughter…

In kNN LM

- We have an indexed key in kNN LM for each token. This causes two issues:
  - Restricts the size of corpus that can be indexed
  - k-neighbors returns only k tokens

RETRO - Retrieval-Enhanced Transformer

Indexed Contexts (keys)	Values
Obama	*was*
Obama was	*senator*
Obama was born	*in*
Obama was born in	*Hawaii*

… Obama was born in Hawaii, and graduated from Columbia University. Obama is a native of Hawaii, Obama was senator for Illinois from 1997 to 2005. Barack is Married to Michelle and their first daughter…

In kNN LM

- We have an indexed key in kNN LM for each token. This causes two issues:
  - Restricts the size of corpus that can be indexed
  - k-neighbors returns only k tokens
- What if we retrieve the entire continuation instead of just one token?

RETRO - Retrieval-Enhanced Transformer

Indexed Contexts (keys)	Values
Obama	*was*
Obama was	*senator*
Obama was born	*in*
Obama was born in	*Hawaii*

… Obama was born in Hawaii, and graduated from Columbia University. Obama is a native of Hawaii, Obama was senator for Illinois from 1997 to 2005. Barack is Married to Michelle and their first daughter…

Indexed Keys (**N**)	**Values (N,F)**
Obama was born in Hawaii and	*Obama was born in Hawaii and graduated from Columbia University.*
and graduated from Columbia University.	*and graduated from Columbia University. Obama is a native of Hawaii,*

In kNN LM

- We have an indexed key in kNN LM for each token. This causes two issues:
  - Restricts the size of corpus that can be indexed
  - k-neighbors returns only k tokens
- What if we retrieve the entire continuation instead of just one token?

RETRO - Retrieval-Enhanced Transformer

Indexed Contexts (keys)	Values
Obama	*was*
Obama was	*senator*
Obama was born	*in*
Obama was born in	*Hawaii*

… Obama was born in Hawaii, and graduated from Columbia University. Obama is a native of Hawaii, Obama was senator for Illinois from 1997 to 2005. Barack is Married to Michelle and their first daughter…

Indexed Keys (**N**)	**Values (N,F)**
Obama was born in Hawaii and	*Obama was born in Hawaii and graduated from Columbia University.*
and graduated from Columbia University.	*and graduated from Columbia University. Obama is a native of Hawaii,*

What if we retrieve the entire continuation instead of just one token? Two advantages:

- For same corpus, # of indexed keys reduce by a fraction of |N| = size of each chunk.
- Each search returns k*(|N| + |F|) tokens

RETRO - Retrieval-Enhanced Transformer

Indexed Contexts (keys)	Values
Obama	*was*
Obama was	*senator*
Obama was born	*in*
Obama was born in	*Hawaii*

... Obama was born in Hawaii, and graduated from Columbia University. Obama is a native of Hawaii, Obama was senator for Illinois from 1997 to 2005. Barack is Married to Michelle and their first daughter...

Indexed Keys (**N**)	**Values (N,F)**
Obama was born in Hawaii and	*Obama was born in Hawaii and graduated from Columbia University.*
and graduated from Columbia University.	*and graduated from Columbia University. Obama is a native of Hawaii,*

What if we retrieve the entire continuation instead of just one token? Two advantages:

- For same corpus, # of indexed keys reduce by a fraction of |N| = size of each chunk.
- Each search returns k*(|N| + |F|) tokens

How to use the k retrieved chunks?

RETRO – How to use k retrieved chunks?

- Split the input also into smaller chunks.

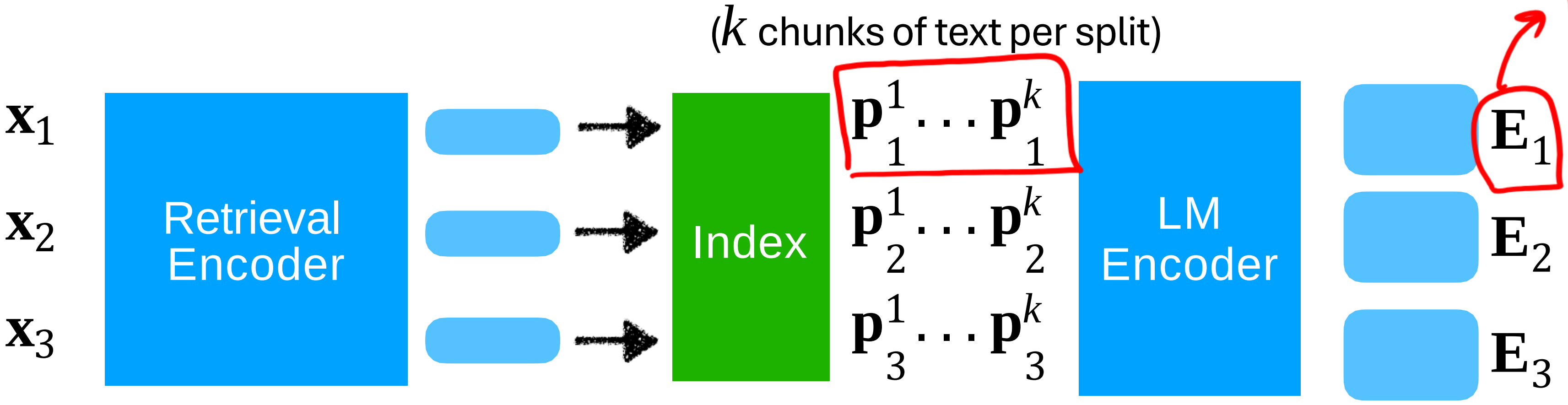
$$\boldsymbol{x} = \underbrace{\text{World Cup 2022 was}}_{\mathbf{x}_1}\ \underbrace{\text{the last with 32 teams,}}_{\mathbf{x}_2}\ \underbrace{\text{before the increase to 48 in 2026.}}_{\mathbf{x}_3}$$

- At the end of each input chunk,
  - retrieve "k" chunks similar to the input chunk
  - Note that the retrieved value contains the continuation as well.

RETRO – How to use k retrieved chunks?

$\boldsymbol{x}$ = World Cup 2022 was the last with 32 teams, before the increase to 48 in 2026.

$\mathbf{x}_1$ $\mathbf{x}_2$ $\mathbf{x}_3$



r d k

$k \times r \times d$

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

(r = # tokens per text chunk)

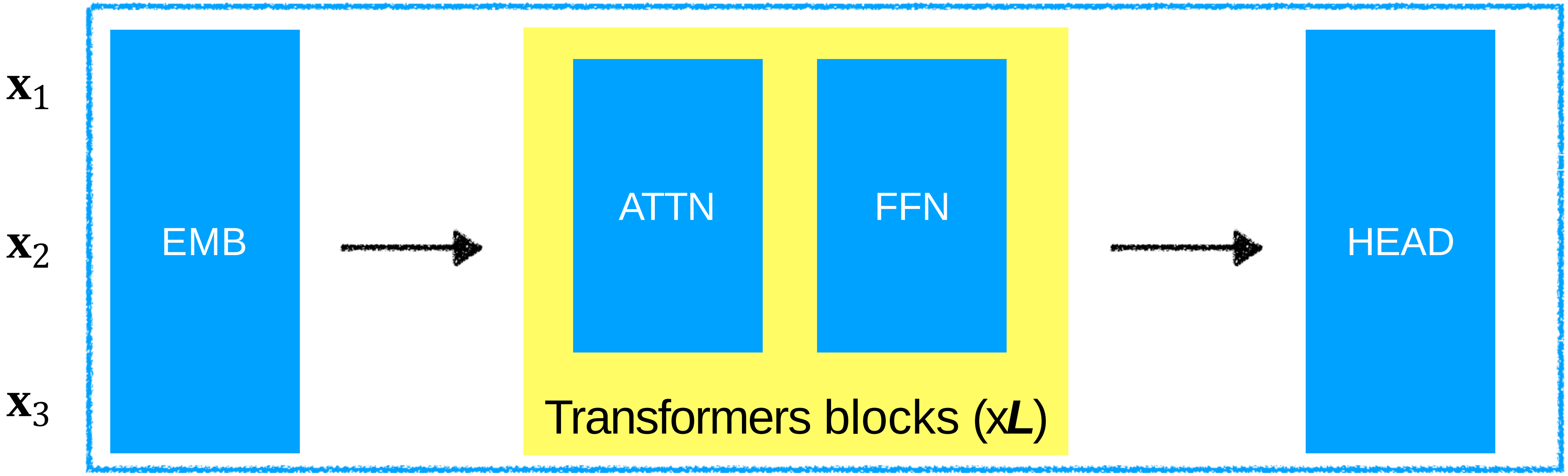
(d = hidden dimension)

(k = # retrieved chunks per split)

Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

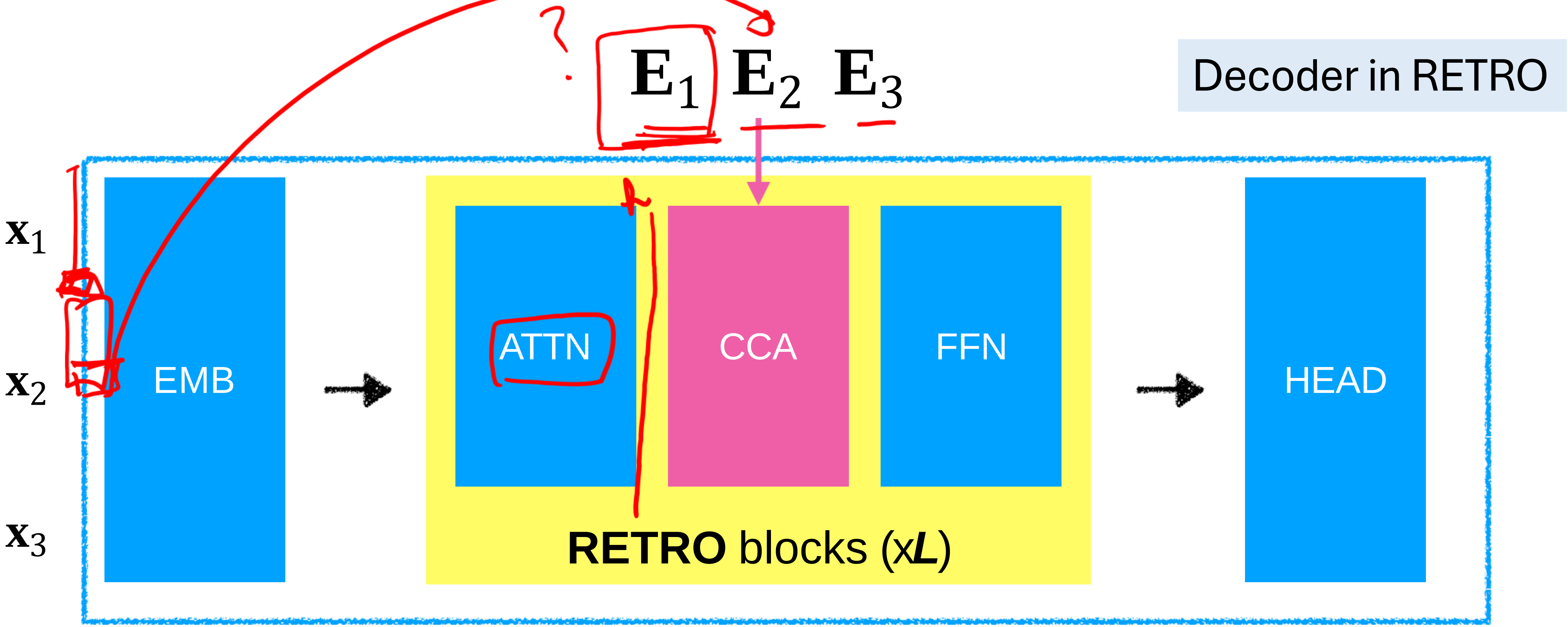
RETRO – How to use k retrieved chunks?

Regular Decoder



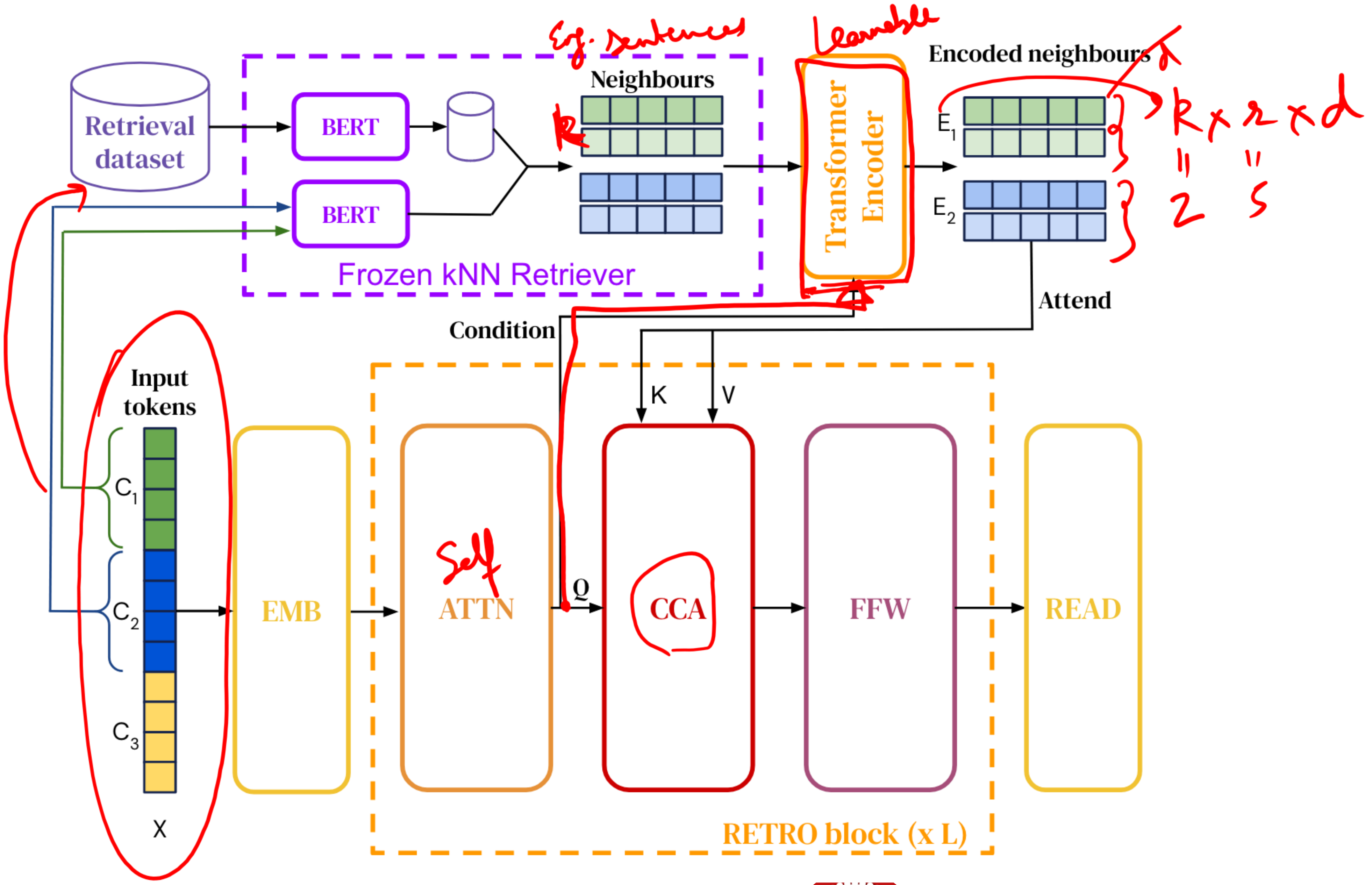
Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

RETRO – How to use k retrieved chunks?



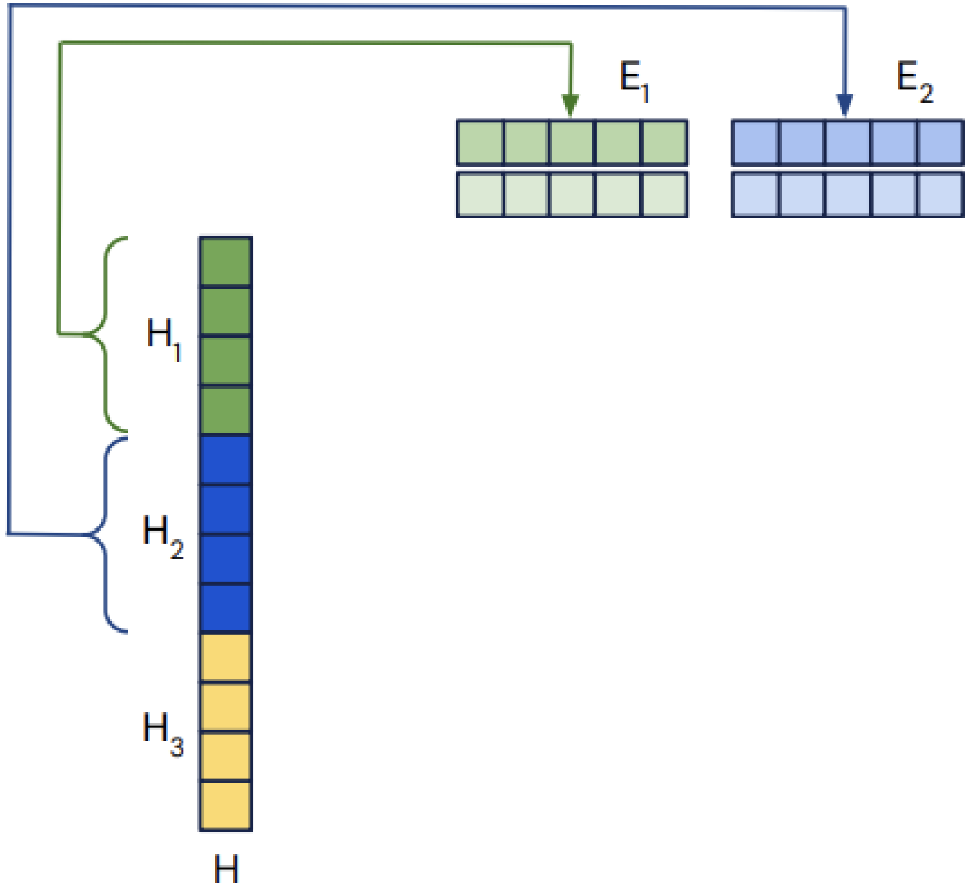
Chunked Cross Attention (CCA)

Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view



Eg. Sentences
Learnable
Encoded neighbours
Retrieval dataset
BERT
BERT
Neighbours
Transformer Encoder
$k \times r \times d$
$= 2$ $= 5$
E_1
E_2
Frozen kNN Retriever
Attend
Condition
Input tokens
C_1
C_2
C_3
X
EMB
Self ATTN
Q
K
V
CCA
FFW
READ
RETRO block (x L)

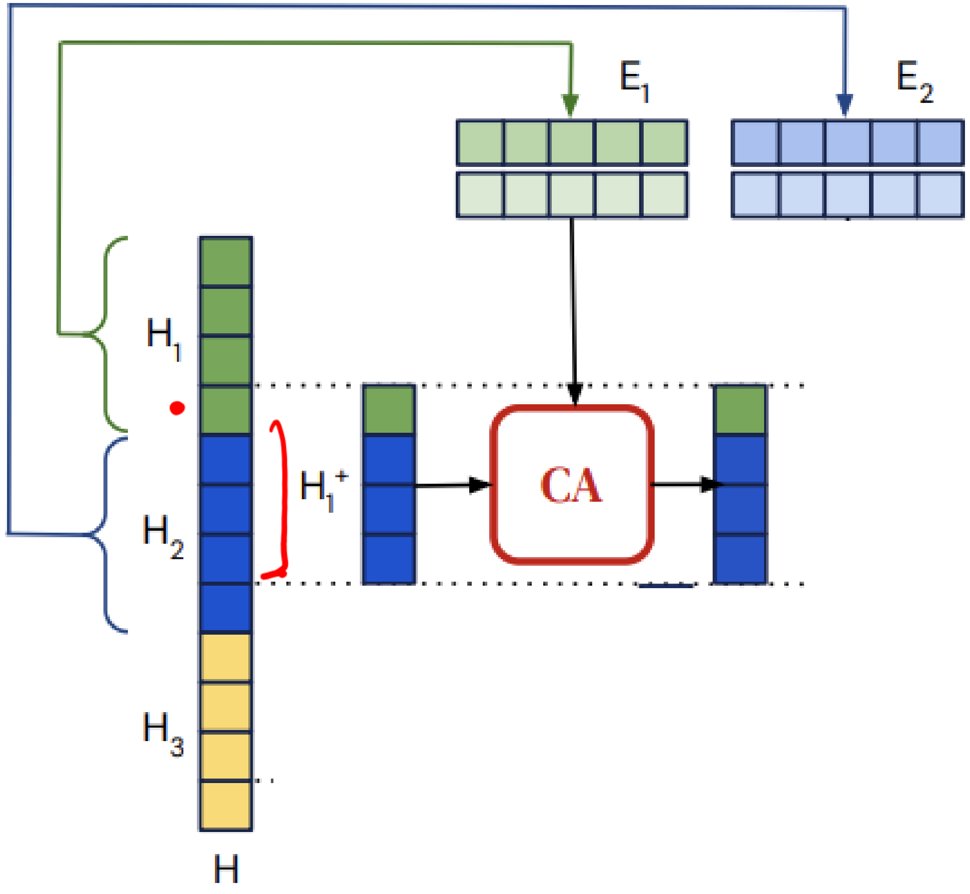
RETRO – How to use k retrieved chunks?



Chunked Cross Attention (CCA)

Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

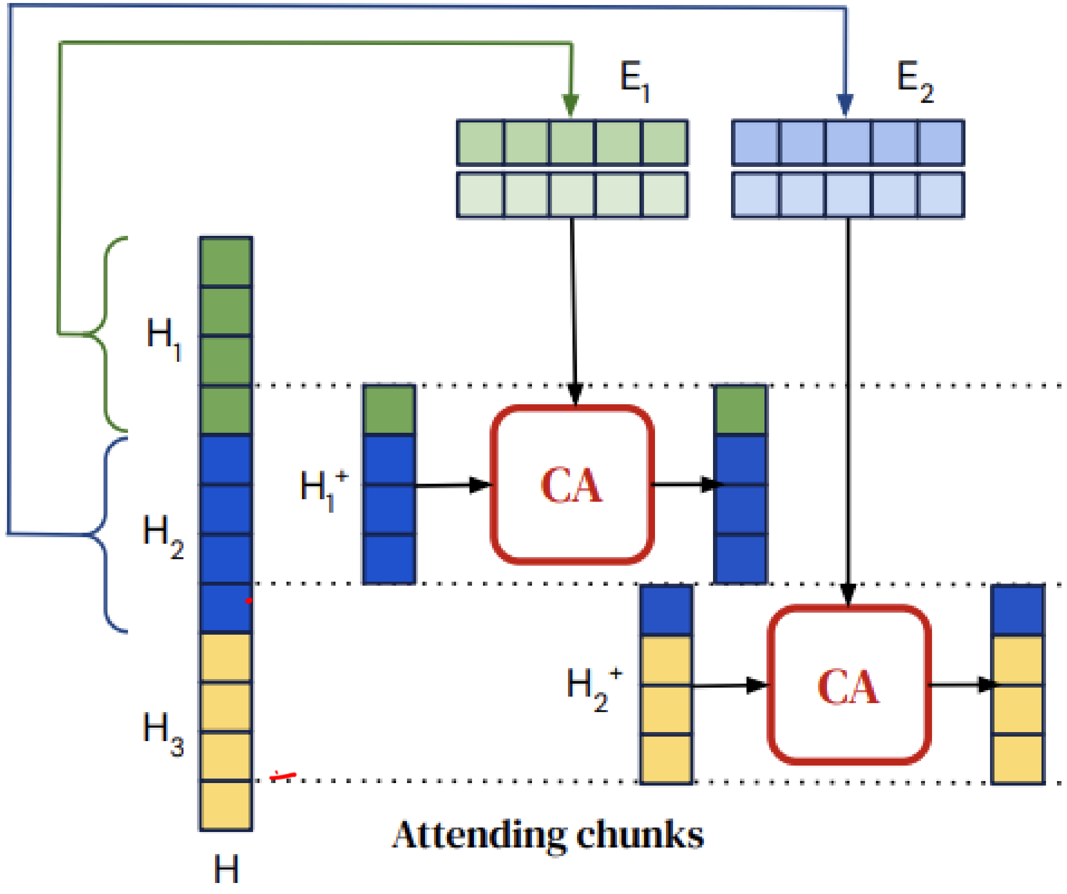
RETRO – How to use k retrieved chunks?



Chunked Cross Attention (CCA)

Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

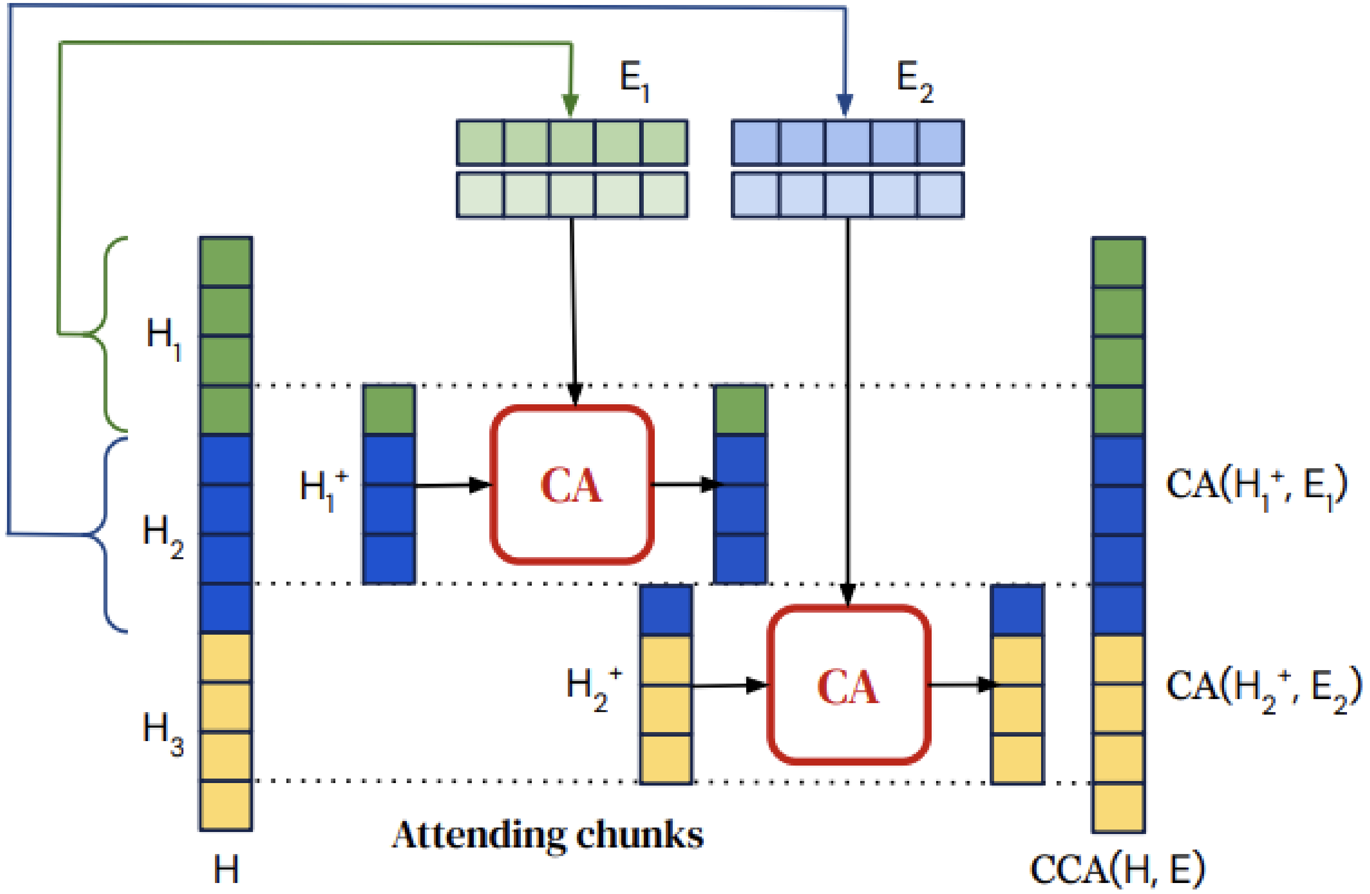
RETRO – How to use k retrieved chunks?



Chunked Cross Attention (CCA)

Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

RETRO – How to use k retrieved chunks?



Chunked Cross Attention (CCA)

Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

RETRO Results

Model	Retrieval Set	#Database tokens	#Database keys	Perplexity Valid	Perplexity Test
Adaptive Inputs (Baevski and Auli, 2019)	-	-	-	17.96	18.65
SPALM (Yogatama et al., 2021)	Wikipedia	3B	3B	17.20	17.60
kNN-LM (Khandelwal et al., 2020)	Wikipedia	3B	3B	16.06	16.12
Megatron (Shoeybi et al., 2019)	-	-	-	-	10.81
Baseline transformer (ours)	-	-	-	21.53	22.96
kNN-LM (ours)	Wikipedia	4B	4B	18.52	19.54
RETRO	Wikipedia	4B	0.06B	18.46	18.97
RETRO	C4	174B	2.9B	12.87	10.23
RETRO	MassiveText (1%)	18B	0.8B	18.92	20.33
RETRO	MassiveText (10%)	179B	4B	13.54	14.95
RETRO	MassiveText (100%)	1792B	28B	**3.21**	**3.92**

Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

RETRO Results

Model	Retrieval Set	#Database tokens	#Database keys	Perplexity Valid	Test
Adaptive Inputs (Baevski and Auli, 2019)	-	-	-	17.96	18.65
SPALM (Yogatama et al., 2021)	Wikipedia	3B	3B	17.20	17.60
*k*NN-LM (Khandelwal et al., 2020)	Wikipedia	3B	3B	16.06	16.12
Megatron (Shoeybi et al., 2019)	-	-	-	-	10.81
Baseline transformer (ours)	-	-	-	21.53	22.96
*k*NN-LM (ours)	Wikipedia	4B	4B	18.52	19.54
RETRO	Wikipedia	4B	0.06B	18.46	18.97
RETRO	C4	174B	2.9B	12.87	10.23
RETRO	MassiveText (1%)	18B	0.8B	18.92	20.33
RETRO	MassiveText (10%)	179B	4B	13.54	14.95
RETRO	MassiveText (100%)	1792B	28B	**3.21**	**3.92**

Content credit: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

{answer}" and left pad the data such that "answer:" coincides with the end of the first chunk of 64 tokens and thus aligns with the first retrieving chunk.

How to fine-tune RETRO for downstream task?

- Fine-tune on NQ
- Format the data as: "`[PAD] question: {question} \n answer: {answer}`"

LM to generate

[PAD] to ensure that "**`answer`**" coincides with end of first chunk

How to fine-tune RETRO for downstream task?

- Fine-tune on NQ
- Format the data as: "`[PAD] question: {question} \n answer: {answer}`"

$\underbrace{\texttt{[PAD] question: \{question\} \textbackslash n answer:}}_{\text{64 tokens}}$

64 tokens

How to fine-tune RETRO for downstream task?

- Fine-tune on NQ
- Format the data as: "`[PAD] question: {question} \n answer: {answer}`"
- Fine-tune 7.5B model using 25k steps with 20 retrieved passages for each sample

Results on NQ

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

- Performance similar to other methods, except for FiD
- Increasing # retrieved passages beyond 20 doesn't help

Frequency of calling retriever

- RETRO Triggers retriever after every *L* tokens
- Can we trigger it on demand?
  - Generate a special token that triggers retriever call [Toolformer [Schick et al. 23]]
  - Call it when LM itself is uncertain about the prediction. [Jiang et al. 2023]
  - RIG – Retriever Interleaved Generation [Radhakrishnan et al. 2024]

Frequency of calling retriever

- RETRO Triggers retriever after every *L* tokens
- Can we trigger it on demand?
  - Generate a special token that triggers retriever call [Toolformer [Schick et al. 23]]

Frequency of calling retriever

Toolformer: Language Models Can Teach Themselves to Use Tools

Timo Schick **Jane Dwivedi-Yu** **Roberto Dessì**[†] **Roberta Raileanu**
Maria Lomeli **Luke Zettlemoyer** **Nicola Cancedda** **Thomas Scialom**

Meta AI Research [†]Universitat Pompeu Fabra

Abstract

Language models (LMs) exhibit remarkable abilities to solve new tasks from just a few examples or textual instructions, especially at

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.

Toolformer [Schick et al. 23]

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.

Generate tokens that trigger retriever or other tools

Frequency of calling retriever

- RETRO Triggers retriever after every *L* tokens
- Can we trigger it on demand?
  - Generate a special token that triggers retriever call [Toolformer [Schick et al. 23]]
  - Call it when LM itself is uncertain about the prediction. [Jiang et al. 2023]

Triggering Retrieval w/ Uncertainty

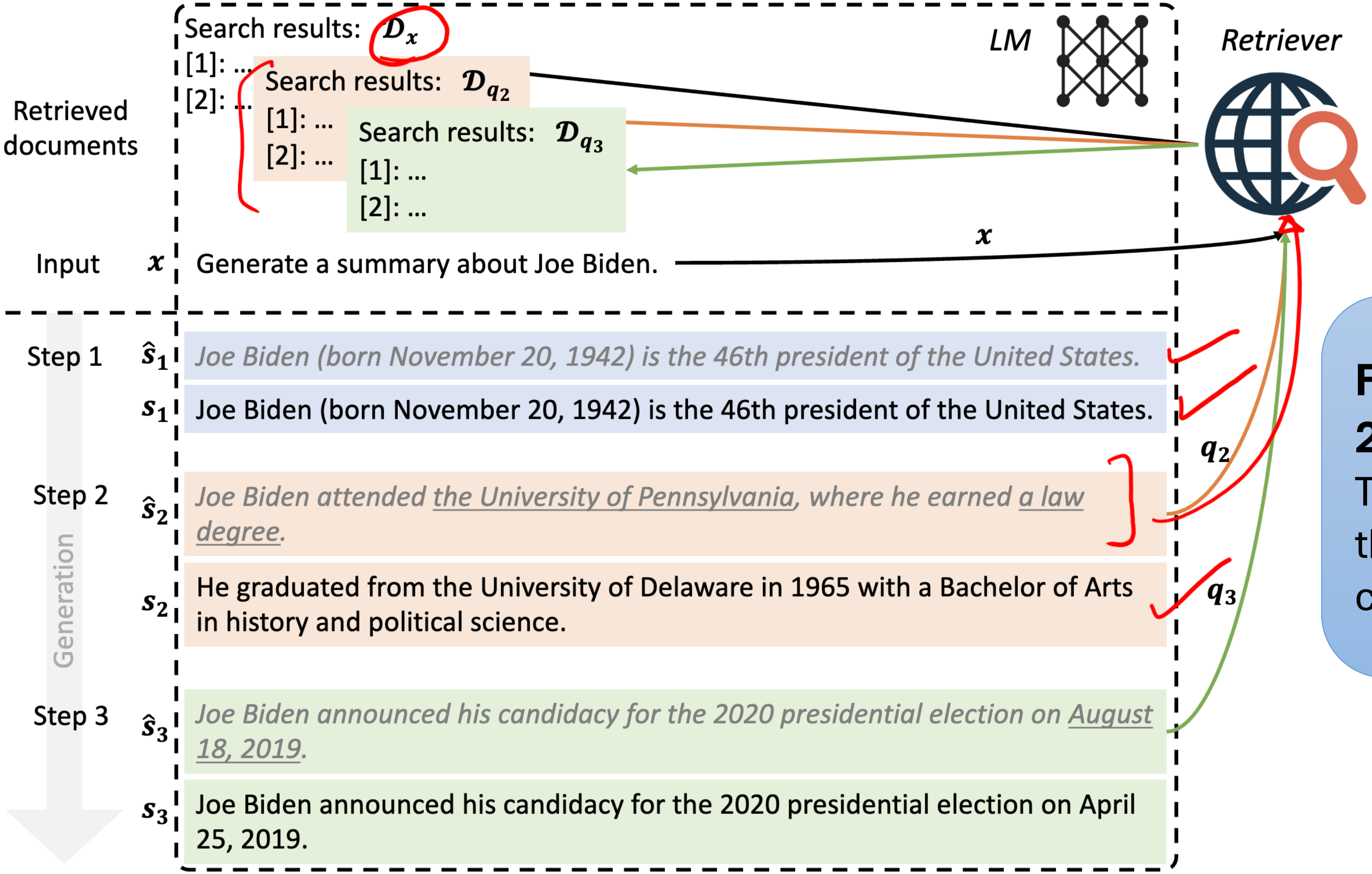
Active Retrieval Augmented Generation

Zhengbao Jiang[1*] **Frank F. Xu**[1*] **Luyu Gao**[1*] **Zhiqing Sun**[1*] **Qian Liu**[2]
Jane Dwivedi-Yu[3] **Yiming Yang**[1] **Jamie Callan**[1] **Graham Neubig**[1]
[1]Language Technologies Institute, Carnegie Mellon University
[2]Sea AI Lab [3]FAIR, Meta
{zhengbaj,fangzhex,luyug,zhiqings,gneubig}@cs.cmu.edu

Abstract

Despite the remarkable ability of large language models (LMs) to comprehend and generate language, they have a tendency to hallucinate and create factually inaccurate out-

hallucinate and create imaginary content (Maynez et al., 2020; Zhou et al., 2021). Augmenting LMs with retrieval components that look up relevant information from external knowledge resources is a promising direction to address hallucination (Khan-



FLARE (Jiang et al. 2023)

Tries to generate content, then does retrieval if LM certainty is low

Frequency of calling retriever

- RETRO Triggers retriever after every *L* tokens
- Can we trigger it on demand?
  - Generate a special token that triggers retriever call [Toolformer [Schick et al. 23]]
  - Call it when LM itself is uncertain about the prediction. [Jiang et al. 2023]
  - RIG – Retriever Interleaved Generation [Radhakrishnan et al. 2024]

Retrieve from Data Commons

Knowing When to Ask - Bridging Large Language Models and Data

Authors: Prashanth Radhakrishnan[1*], Jennifer Chen[1*], Bo Xu[1*], Prem Ramaswami[1*‡], Hannah Pho[1*], Adriana Olmos[1*], James Manyika[1], R. V. Guha[1*]

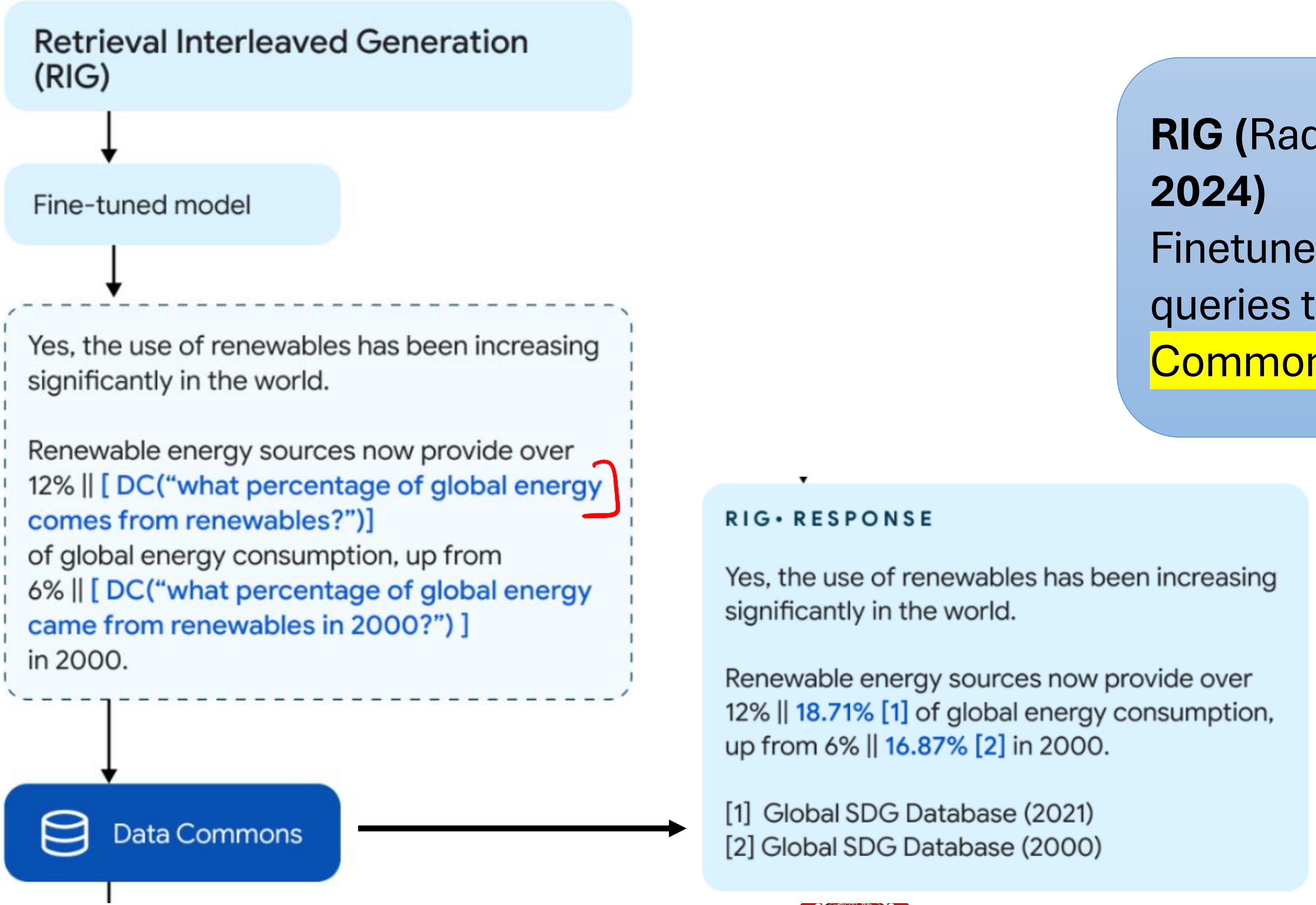
September 12, 2024

[1] Google, Inc. 1600 Amphitheatre Parkway, Mountain View, California, 94043
* Indicates that these authors contributed equally
‡ Indicates the corresponding authors

RIG (Radhakrishnan **et al. 2024)**
Finetune LLM to generate queries to retrieve from Data Commons

Abstract

Large Language Models (LLMs) are prone to generating factually incorrect information when responding to queries that involve numerical and statistical data or other timely facts. In this paper, we present an approach for enhancing the accuracy of LLMs by integrating them with Data Commons, a vast, open-source repository of public statistics from trusted organizations like the United Nations (UN), Center for Disease Control and Prevention (CDC) and global



RIG (Radhakrishnan **et al. 2024)**
Finetune LLM to generate queries to retrieve from Data Commons

Data Commons – open source initiative by Google

- Aims to organize the world's public datasets in a Knowledge Graph
- Encompasses a large range of statistical data from public sources such as
  - the United Nations, national census bureaus,
  - health ministries,
  - environmental agencies,
  - economic departments,
  - NGOs
  - …
- Includes more than 250 billion data points and over 2.5 trillion triples from hundreds of global sources.

How to use the Book?

- Output interpolations - After solving the question yourself?

kNN LMs

- Intermediate fusion – modify the LM architecture to be aware of the book?

RETRO

How to use the Book?

- Output interpolations - After solving the question yourself?

kNN LMs

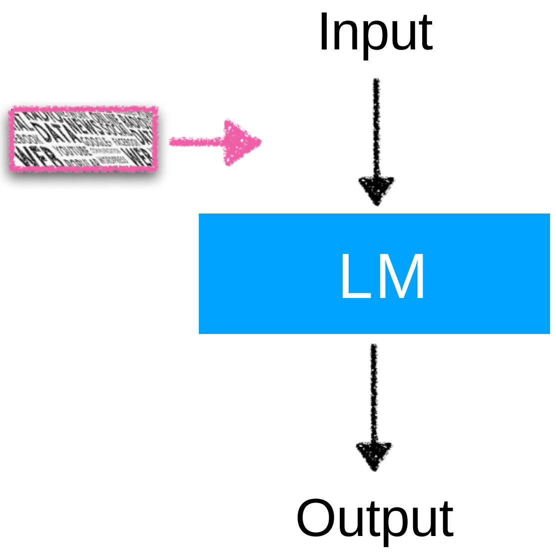
- Intermediate fusion – modify the LM architecture to be aware of the book?

RETRO

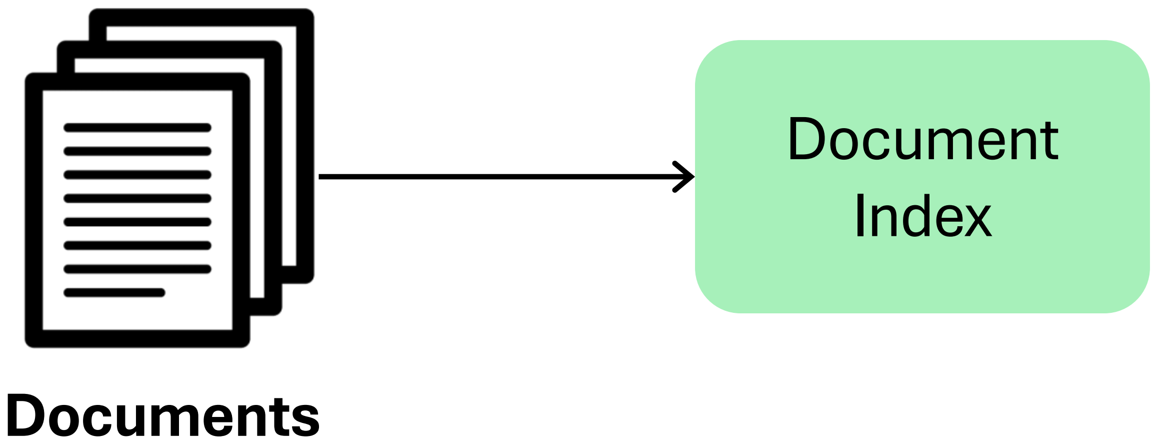
- Both the works mainly focused on reducing perplexity, and not on solving downstream tasks.
- Not the most popular methods for incorporating external knowledge

How to use the Book?

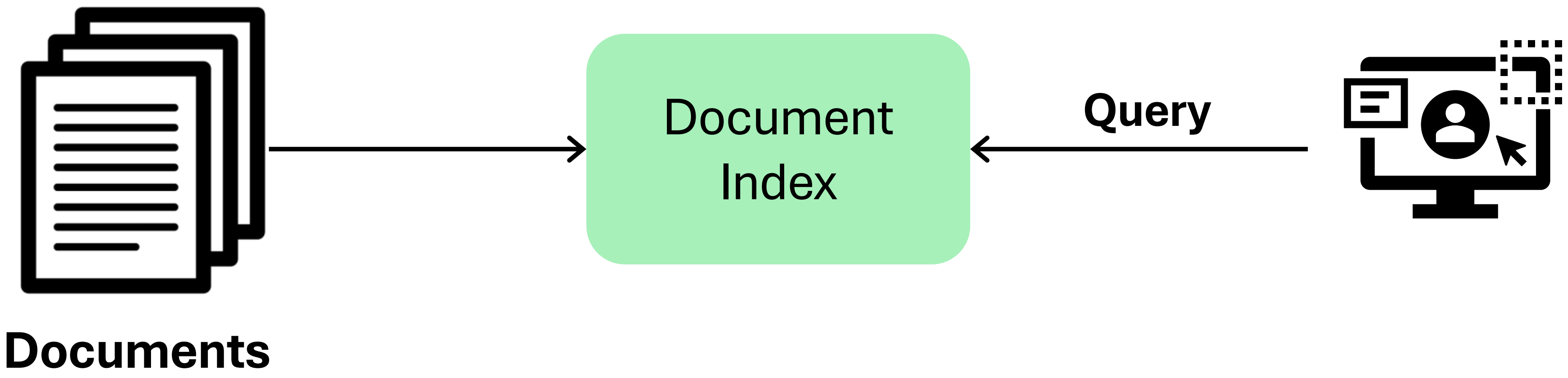
- Output interpolations - After solving the question yourself?
- Intermediate fusion – modify the LM architecture to be aware of the book?
- Input augmentation (RAG) - Before you start solving?



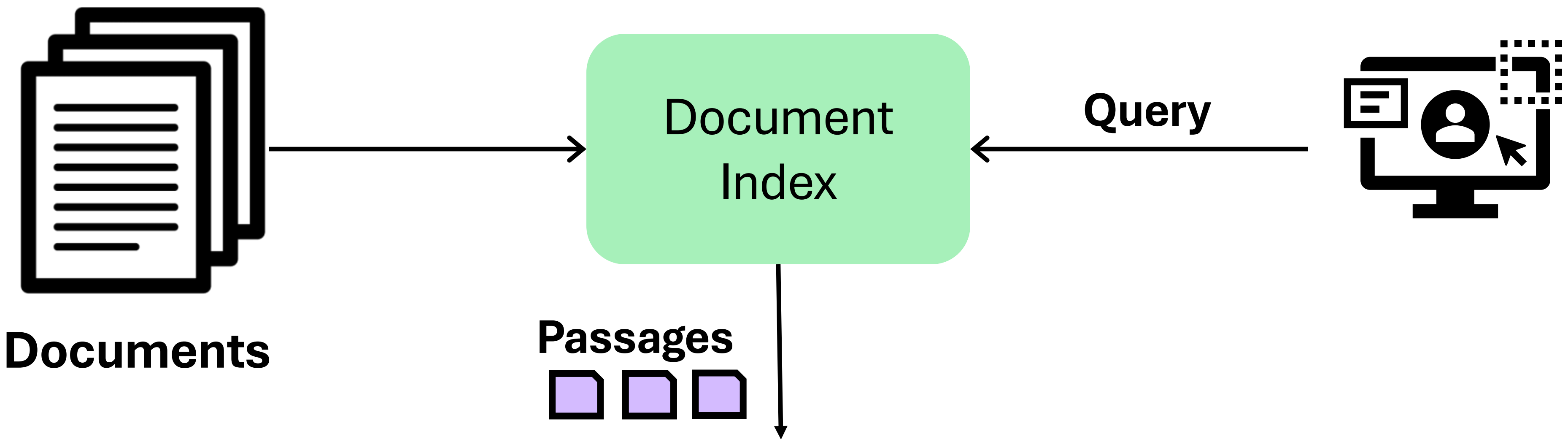
RAG - Architecture



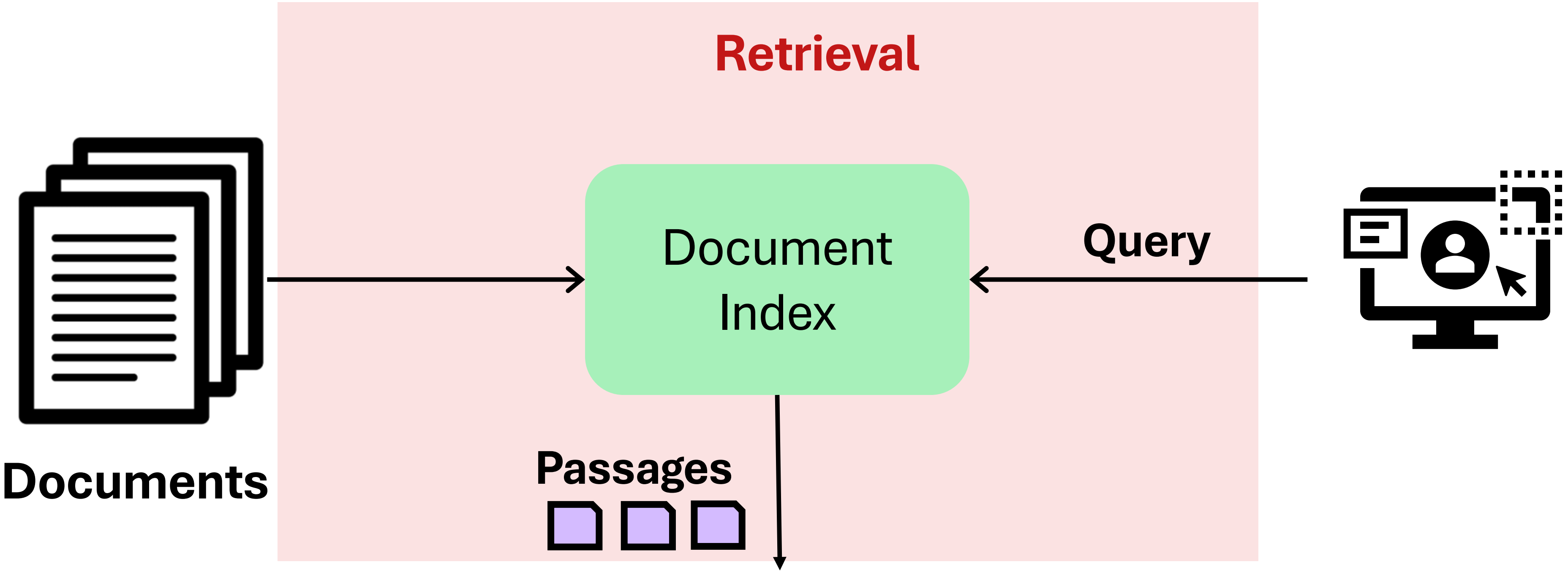
RAG - Architecture



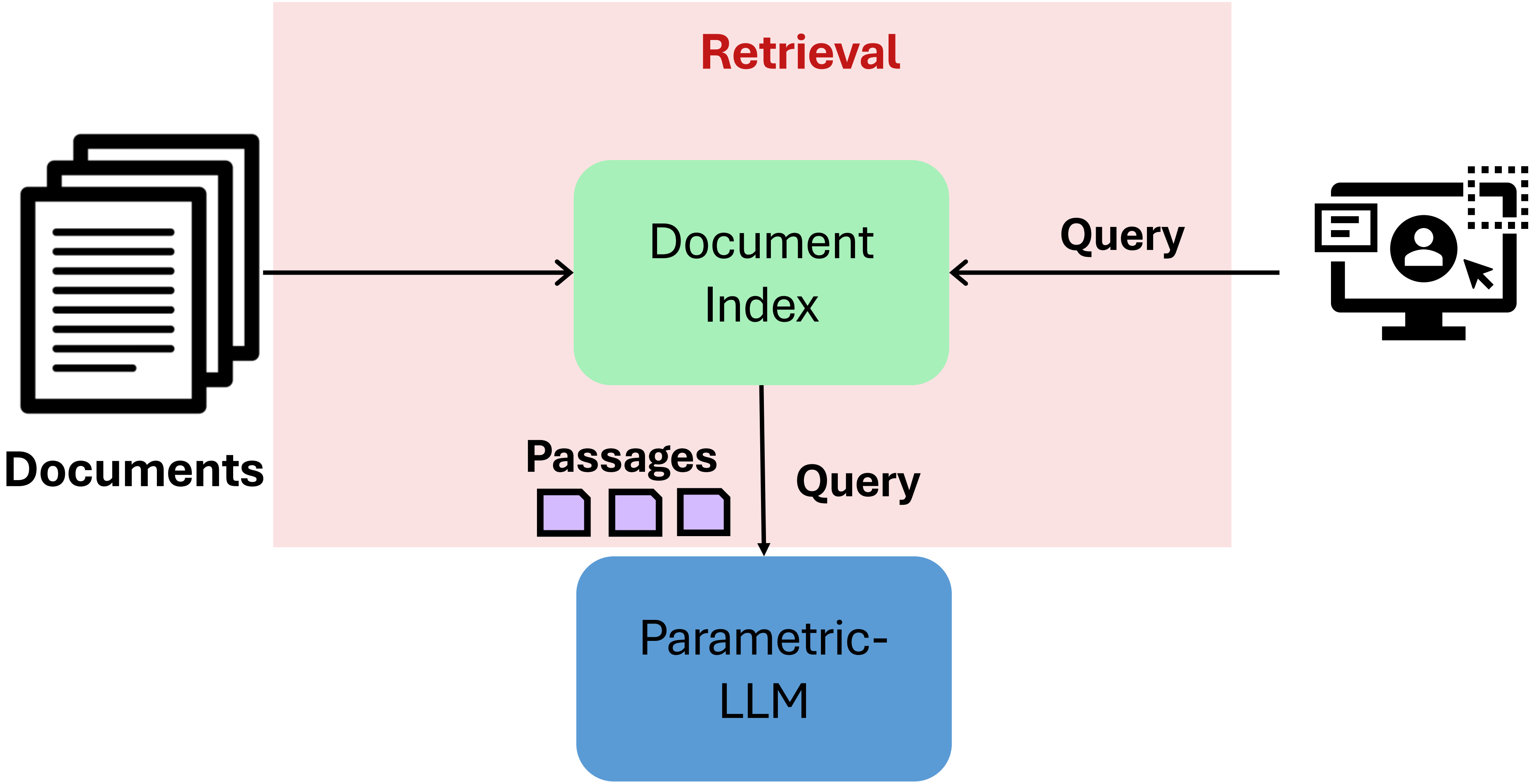
RAG - Architecture



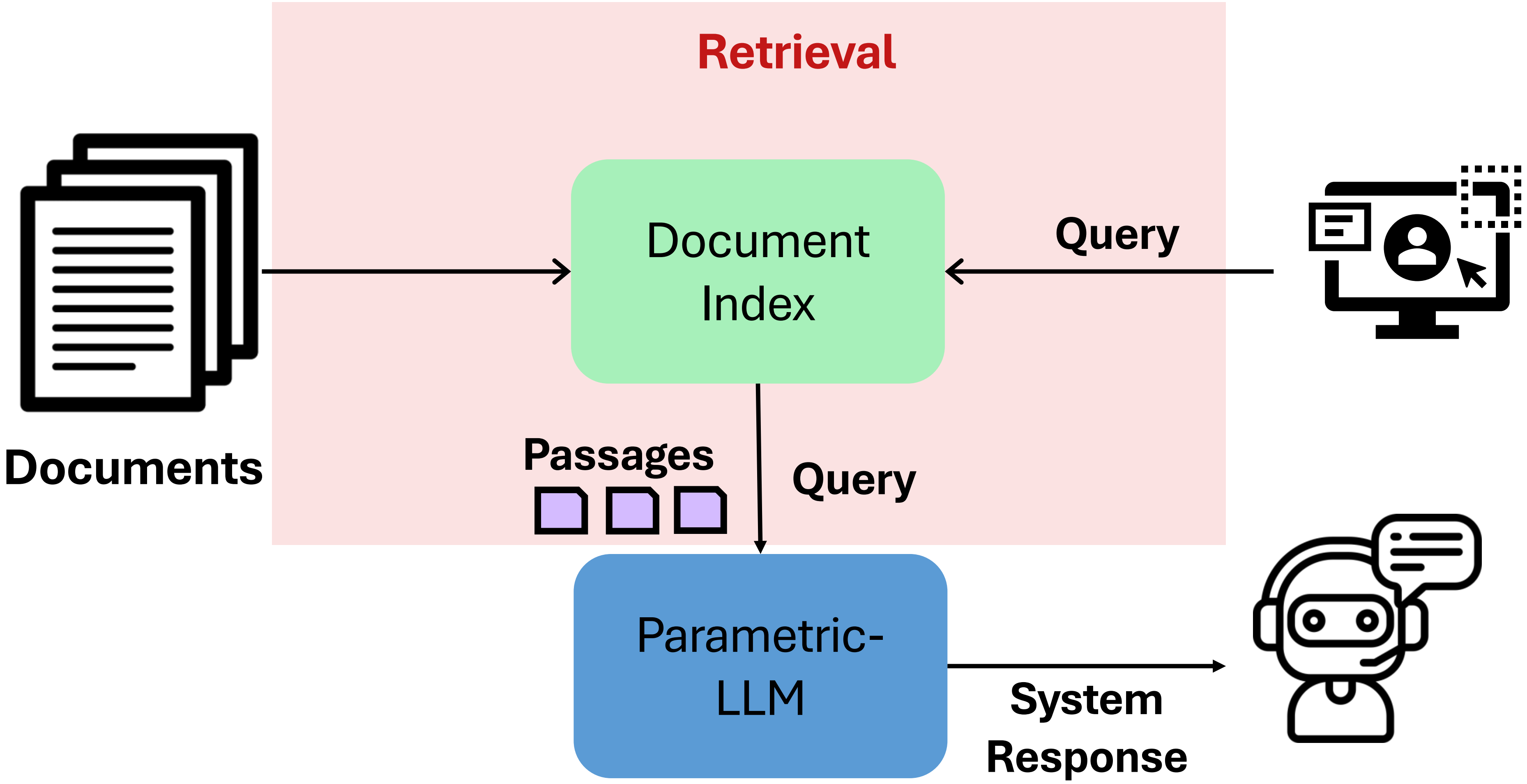
RAG - Architecture



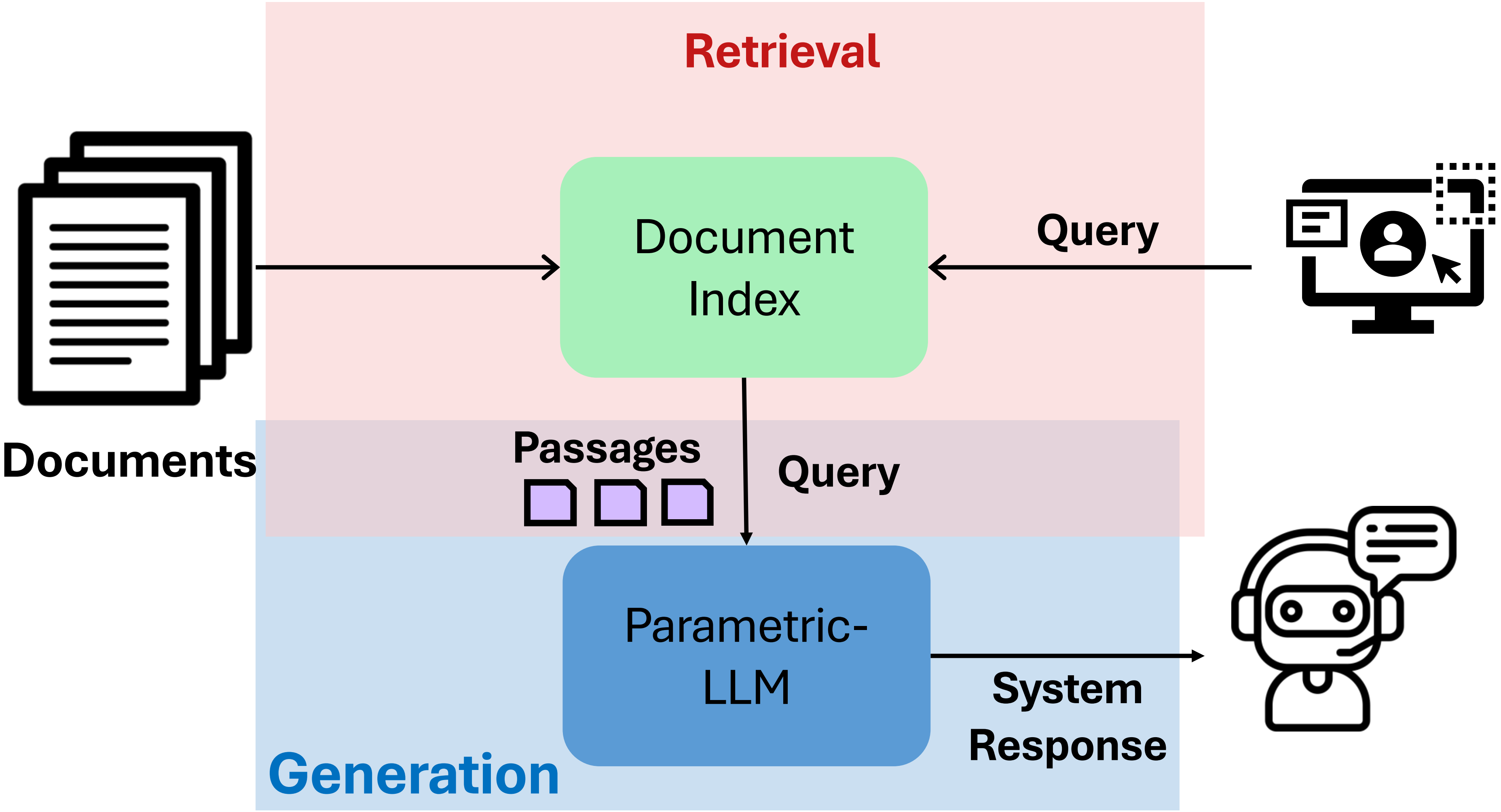
RAG - Architecture



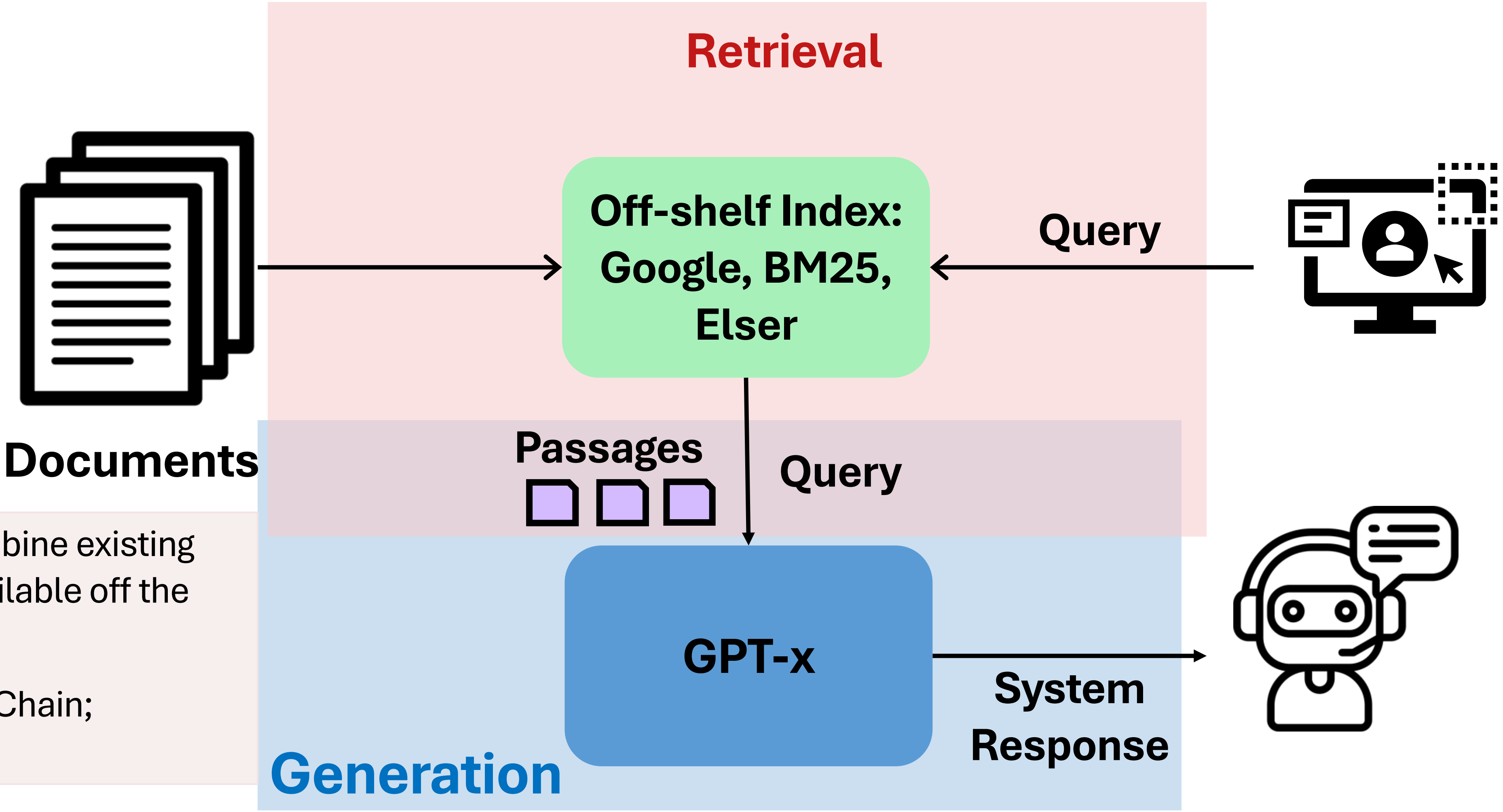
RAG - Architecture



Retrieval Based LLMs - Architecture



RAG - Architecture



- ✓ Simply combine existing models available off the shelf!
- ✓ Tools: LangChain; LlamaIndex

$q \rightarrow$ Ret.

Generator
LLM

Retrieval Based LLMs - Architecture

- REALM (Guu et al 2020): Retrieval-Augmented Language Model Pre-Training ICML 2020

- RAG (Lewis et al 2020): Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

$q \rightarrow$ En $\rightarrow$ e_q

Denc.

index

REALM (Guu et al 2020)

x = World Cup 2022 was the last with 32 teams before the increase to [MASK] in 2026.

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

↓

LM

↓

48

Guu et al. REALM: Retrieval-Augmented Language Model Pre-Training. ICML 2020.

Slide source: https://drive.google.com/file/d/1YUpp7L1SCK6jgdfFObsqHKXrq6HC-TLp/view

REALM (Guu et al 2020)

x = World Cup 2022 was the last with 32 teams before the increase to [MASK] in 2026.

***q* (=*x*)**

Retrieval

FIFA World Cup 2026 will expand to 48 teams.

k chunks of text (passages)

Retrieve stage

Chunk 1

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

LM

REALM (Guu et al 2020)

x = World Cup 2022 was the last with 32 teams before the increase to [MASK] in 2026.

***q* (=*x*)**

Retrieval

k chunks of text (passages)

FIFA World Cup 2026 will expand to 48 teams.

Retrieve stage

FIFA World Cup 2026 will expand to 48 teams.

…

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

LM

48

Read stage

REALM (Guu et al 2020)

MLM task: obtained from the embedding of the MASK token

[CLS] z_1 [SEP] x → LM → $P(y|x, z_1)$

[CLS] z_2 [SEP] x → LM → $P(y|x, z_2)$

⋮

[CLS] z_k [SEP] x → LM → $P(y|x, z_k)$

Weighted average

$$P(y|x) = \sum_{z \in \mathcal{D}} \underbrace{P(z|x)}_{\text{from the retrieve stage}} \underbrace{P(y|x, z)}_{\text{from the read stage}}$$

REALM (Guu et al 2020)

MLM task: obtained from the embedding of the MASK token

[CLS] z_1 [SEP] x → LM θ → $P(y|x, z_1)$

[CLS] z_2 [SEP] x → LM → $P(y|x, z_2)$

⋮

[CLS] z_k [SEP] x → LM → $P(y|x, z_k)$

Weighted average

η θ L

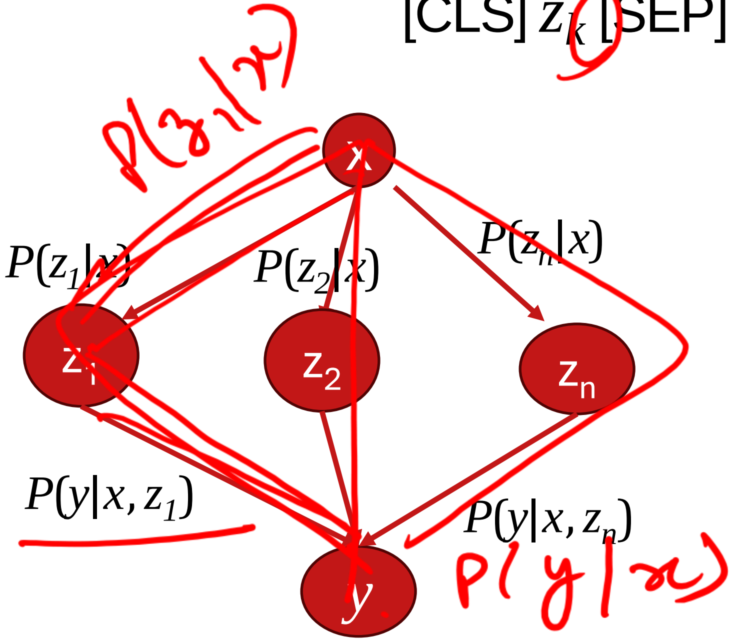
ϕ

maximize

$$P(y|x) = \sum_{z \in \mathcal{D}} \underbrace{P(z|x)}_{\eta} \underbrace{P(y|x,z)}_{\theta}$$

from the retrieve stage

from the read stage



REALM (Guu et al 2020)

[CLS] z_1 [SEP] x → LM → $P(y|x, z_1)$

[CLS] z_2 [SEP] x → LM → $P(y|x, z_2)$

⋮

[CLS] z_k [SEP] x → LM → $P(y|x, z_k)$

Weighted average

0 if not one of top *k*

$$\sum_{\cancel{z \in D}} \underbrace{P(z|x)}_{\text{from the retrieve stage}} \underbrace{P(y|x, z)}_{\text{from the read stage}}$$

$z \in top_k(q)$

Need to approximate
Consider top *k* chunks only

REALM: Joint Training

Trainable components

- Retriever
  - Document Encoder ✓
  - Query Encoder ✓
- Reader: LM ✓

REALM: Pre-Training

$$\text{Maximize} \sum_{z \,\in\, topk(p_\eta(\cdot|x))} p_\eta(z \mid x) \boxed{p_\theta\left(y_{[MASK]} \mid x, z\right)}$$

$e_x = Enc_q(x)$

$e_z = Enc_d(z)$

$P_\eta(z|x) \propto sim.(e_x, e_z)$

REALM: Pre-Training

$$\text{Maximize} \sum_{z \in topk(p_\eta(\cdot|x))} p_\eta(z \mid x)\, p_\theta\left(y_{[MASK]} \mid x, z\right)$$

Retriever

q (=x)

Index

top-K retrieved chunks

The pyramidion on top allows for less material higher up the pyramid.

$P_\eta(z|x)$

REALM: Pre-Training

$$\text{Maximize} \sum_{z \in topk(p_\eta(\cdot|x))} p_\eta(z \mid x)\, p_\theta\left(y_{[MASK]} \mid x, z\right)$$

Retriever

Reader

q (=x)

Index

top-K retrieved chunks

The pyramidion on top allows for less material higher up the pyramid.

$P_\eta(z|x)$

[CLS] The pyramidion on top … the pyramid**.** **[SEP]** The [MASK] at the top of the pyramid.

LM

pyramidion

$P_\theta(y|x,z)$

REALM: Pre-Training

$$\text{Maximize} \sum_{z \in topk(p_\eta(\cdot|x))} p_\eta(z \mid x)\, p_\theta(y_{[MASK]} \mid x, z)$$

Retriever Reader

η_q η_d

Index

$$p_\eta(z|x) \propto \exp(\mathbf{d}(z)^\top \mathbf{q}(x)) \quad \mathbf{d}(z) = \text{enc}_d(z), \quad \mathbf{q}(x) = \text{enc}_q(x)$$

REALM: Training Approximations

- Freeze top-k documents
- Freeze index (document embeddings), but search top-k documents
- Update index every T steps

REALM: Fine-Training

x_1 y

Maximize $\sum_{z \in topk(p_\eta(\cdot|x))} p_\eta(z \mid x) \underbrace{\cancel{p_\theta(y_{[MASK]} \mid x, z)}}_{LM}$ $p_\theta(y \mid x, z)$

[CLS] The internal angle of an equilateral triangle are equal 60 degrees. [SEP] What's the angle of an equilateral triangle?

BERT

$S(z, y)$ = set of spans matching y in z.

$$p(y \mid z, x) \propto \sum_{s \in S(z,y)} \exp\left(\mathtt{MLP}\left(\left[h_{\mathtt{START(s)}}; h_{\mathtt{END(s)}}\right]\right)\right)$$

$$h_{\mathtt{START(s)}} = \mathtt{BERT}_{\mathtt{START(s)}}(\mathtt{join}_{\mathtt{BERT}}(x, z_{\text{body}})),$$

$$h_{\mathtt{END(s)}} = \mathtt{BERT}_{\mathtt{END(s)}}(\mathtt{join}_{\mathtt{BERT}}(x, z_{\text{body}})),$$

REALM: Fine-Training

MLP [s.e]

60 degrees

Z θ

$$\text{Maximize} \sum_{z \in topk(p_\eta(\cdot|x))} p_\eta(z \mid x)\, p_\theta(y \mid x, z)$$

Retriever

Reader

MLP (e_{q_i} e_p) → score

q (=x)

Index

top-K retrieved chunks

The internal angle of an equilateral triangle are equal, 60 degrees.

$P_\eta(z|x)$

[CLS] The internal angle of an equilateral triangle are equal, 60 degrees. **[SEP]** What's the angle of an equilateral triangle?

LM

60 degrees

$P_\theta(y|x,z)$

Cold Start Problem

- **Reader:** MLM pretraining
- **Inverse Cloze Task:** used to pretrain retriever embeddings

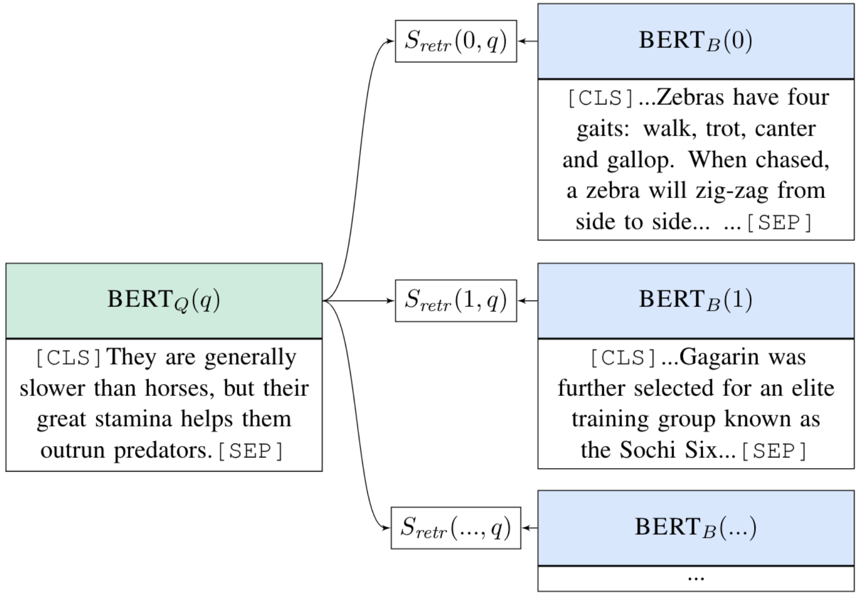


Figure 2: Example of the Inverse Cloze Task (ICT), used for retrieval pre-training. A random sentence (pseudo-query) and its context (pseudo evidence text) are derived from the text snippet: "*...Zebras have four gaits: walk, trot, canter and gallop.* ***They are generally slower than horses, but their great stamina helps them outrun predators.*** *When chased, a zebra will zig-zag from side to side...*" The objective is to select the true context among candidates in the batch.

q

Image source: Lee et al 2019, Latent Retrieval for Weakly Supervised Open Domain Question Answering

REALM – Results

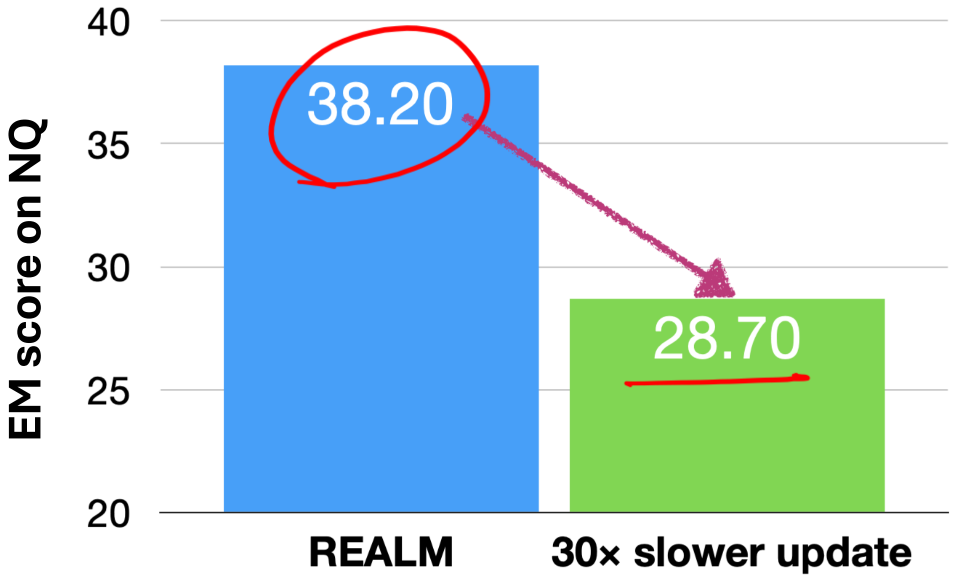
Name	Architectures	Pre-training	NQ (79k/4k)	WQ (3k/2k)	CT (1k /1k)	# params
		Baselines with Frozen retriever + reranking				
DrQA (Chen et al., 2017)	Sparse Retr.+DocReader	N/A	-	20.7	25.7	34m
HardEM (Min et al., 2019a)	Sparse Retr.+Transformer	BERT	28.1	-	-	110m
GraphRetriever (Min et al., 2019b)	GraphRetriever+Transformer	BERT	31.8	31.6	-	110m
PathRetriever (Asai et al., 2019)	PathRetriever+Transformer	MLM	32.6	-	-	110m
ORQA (Lee et al., 2019)	Dense Retr.+Transformer	ICT+BERT	33.3	36.4	30.1	330m
		REALM				
Ours ($\mathcal{X}$ = CC-News, $\mathcal{Z}$ = Wikipedia)	Dense Retr.+Transformer	REALM	**40.4**	**40.7**	42.9	330m

ORQA = REALM – joint pre-training with retriever

REALM: Index update rate

How often should we update the retrieval index?

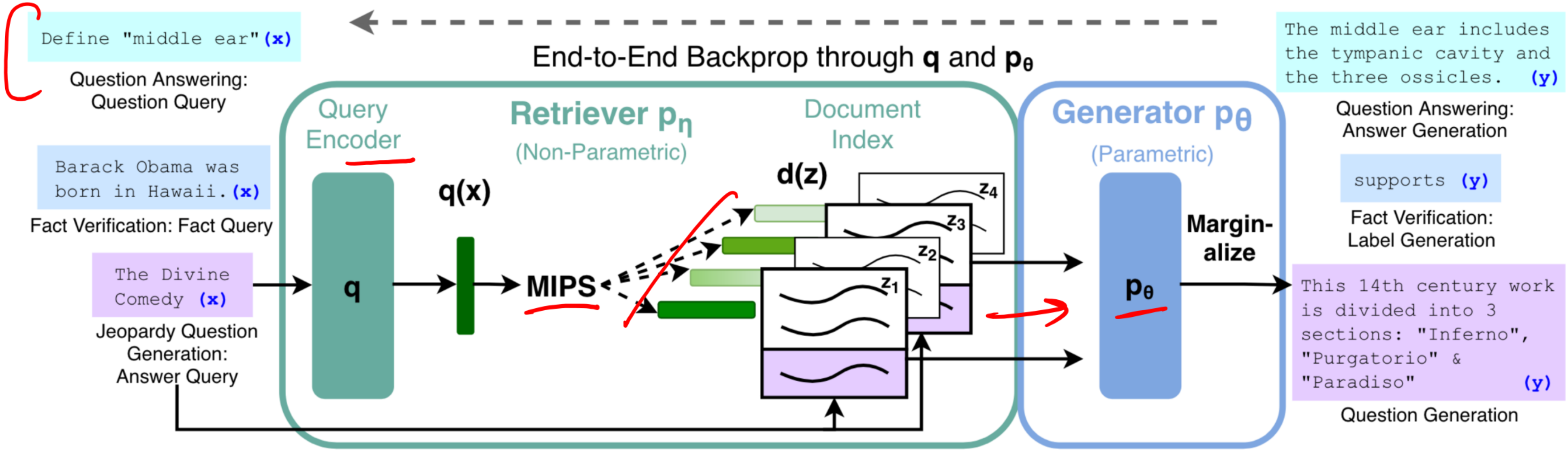
- Frequency too high: expensive
- Frequency too slow: out-dated



REALM: updating the index every 500 training steps

Guu et al. REALM: Retrieval-Augmented Language Model Pre-Training. ICML 2020.

RAG: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks (Lewis et al. 2020)



RAG: Joint Training Equation (Lewis et al. 2020)

Maximize $$\sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) \boxed{p_\theta(y|x, z)}$$

Same as REALM fine-tuning

RAG: Joint Training Equation (Lewis et al. 2020)

Maximize $$\sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y|x, z) = \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x)$$

Same as REALM fine-tuning

RAG: Joint Training Equation (Lewis et al. 2020)

RAG-Sequence Model

Maximize $$\sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y|x,z) = \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) \underbrace{\prod_{i}^{N} p_\theta(y_i|x,z,y_{1:i-1})}_{\text{Factorize}}$$

- Given a retrieved document, generate the entire sequence y
- Marginalize over all the retrieved documents
- Can we generate one token given all documents, and then proceed to the next token?

RAG: Joint Training Equation (Lewis et al. 2020)

RAG-Sequence Model

Maximize $$\sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y|x,z) = \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) \prod_i^N p_\theta(y_i|x,z,y_{1:i-1})$$

RAG: Joint Training Equation (Lewis et al. 2020)

RAG-Sequence Model

Maximize $$\sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y|x, z) = \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) \prod_{i}^{N} p_\theta(y_i|x, z, y_{1:i-1})$$

Maximize $$\prod_{i}^{N} \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y_i|x, z, y_{1:i-1})$$

RAG: Joint Training Equation (Lewis et al. 2020)

RAG-Sequence Model

Maximize $$\sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y|x,z) = \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) \prod_{i}^{N} p_\theta(y_i|x,z,y_{1:i-1})$$

dist. over $|V| \rightarrow k$

Maximize $$\sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y_i|x,z,y_{1:i-1})$$

$\rightarrow$ Prob. $(y_i | x, z_1 - z_k, y_1 - y_{i-1})$

Probability of decoding y_i given document z

RAG: Joint Training Equation (Lewis et al. 2020)

RAG-Sequence Model

Maximize $$\sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y|x,z) = \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) \prod_i^N p_\theta(y_i|x,z,y_{1:i-1})$$

Maximize $$\sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y_i|x,z,y_{1:i-1})$$

Probability of decoding y_i given document z

Marginalize over all documents

RAG: Joint Training Equation (Lewis et al. 2020)

RAG-Sequence Model

Maximize $$\sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y|x,z) = \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) \prod_{i}^{N} p_\theta(y_i|x,z,y_{1:i-1})$$

Maximize $$\prod_{i}^{N} \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y_i|x,z,y_{1:i-1})$$

Product over all N tokens

RAG: Joint Training Equation (Lewis et al. 2020)

RAG-Sequence Model

Maximize $$\sum_{z\in\text{top-}k(p(\cdot|x))} p_\eta(z|x)p_\theta(y|x,z) = \sum_{z\in\text{top-}k(p(\cdot|x))} p_\eta(z|x)\prod_i^N p_\theta(y_i|x,z,y_{1:i-1})$$

RAG-Token Model

Maximize $$\prod_i^N \sum_{z\in\text{top-}k(p(\cdot|x))} p_\eta(z|x)p_\theta(y_i|x,z,y_{1:i-1})$$

RAG: Results

Table 1: Open-Domain QA Test Scores. For TQA, left column uses the standard test set for Open-Domain QA, right column uses the TQA-Wiki test set. See Appendix D for further details.

	Model	NQ	TQA	WQ	CT
Closed Book	T5-11B [52]	34.5	- /50.1	37.4	-
	T5-11B+SSM[52]	36.6	- /60.5	44.7	-
Open Book	REALM [20]	40.4	- / -	40.7	46.8
	DPR [26]	41.5	**57.9**/ -	41.1	50.6
	RAG-Token	44.1	55.2/66.1	**45.5**	50.0
	RAG-Seq.	**44.5**	56.8/**68.0**	45.2	**52.2**

Outline

- Motivation
  - Drawbacks of Parametric LLMs – *hallucination, verification …*
  - Motivating Retrieval-based LLMs – *close book vs open book*
- Major components of Retrieval-based LLMs – *index, retrieve, read …*
- Retrieval Methods – *sparse, dense, reranking, black-box*
- REALM, RAG – *seminal works*
- Overview of Training Techniques – *independent, sequential, joint training …*
- Limitations – *lost in the middle, still hallucinating, retriever failures …*

Training methods for retrieval-augmented LMs

- Independent training
- Sequential training
- Joint training

Training methods for retrieval-augmented LMs

- **Independent training**
- Sequential training
- Joint training

Independent Training

Retrieval models and language models are trained **independently**

- Training language models

Input → LM → Output

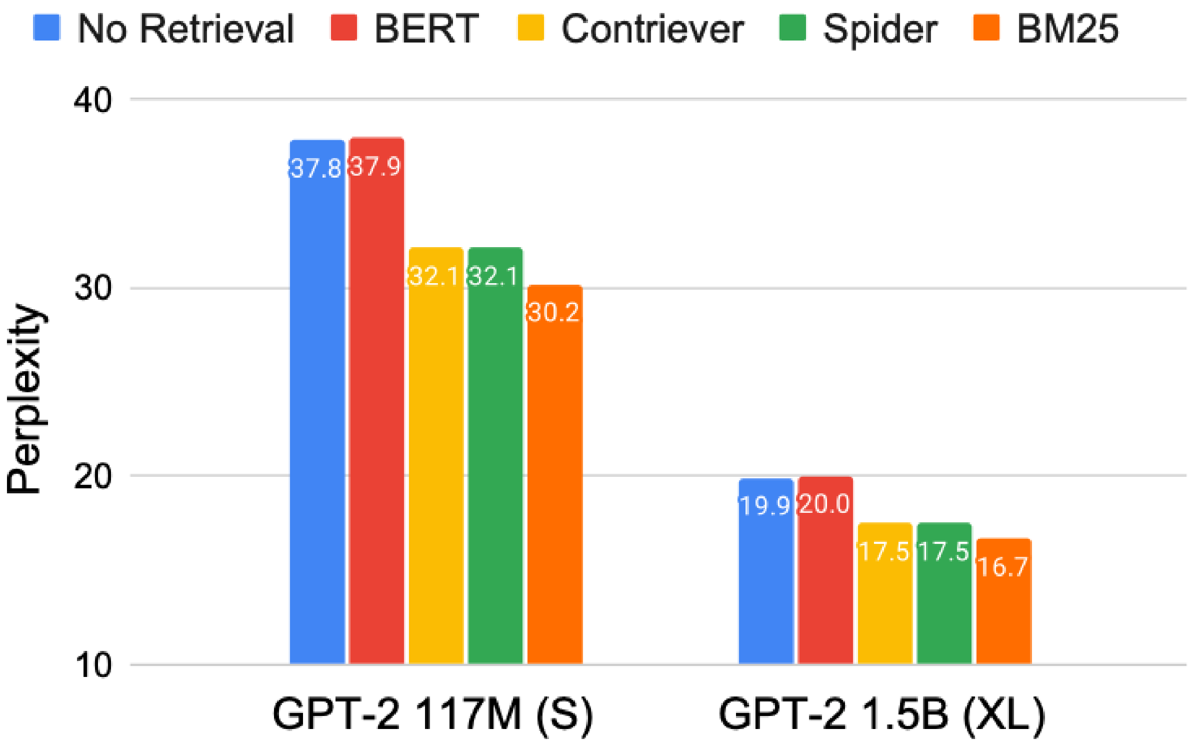
- Training retrieval models

Datastore → Retriever → 

Chunks/tokens

Query → Retriever

RAG with LMs using different retrievers



Ram et al. In-Context Retrieval-Augmented Language Models. TACL 2023.

Better retrieval model

Better base LMs

→ Better **retrieval-based LMs**

Each component can be improved separately

Independent Training

- 👍 Work with off-the-shelf models (no extra training required)
- 👍 Each part can be improved independently

Independent Training

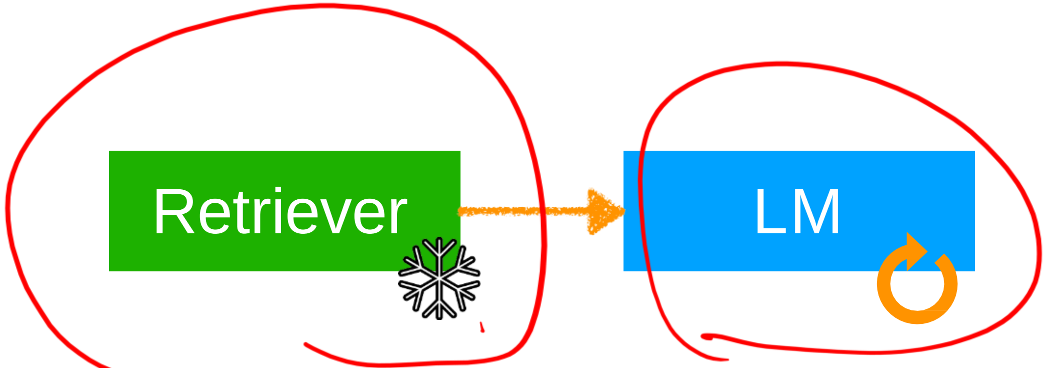
- Work with off-the-shelf models (no extra training required)
- Each part can be improved independently
- LMs are not trained to leverage retrieval
- Retrieval models are not optimized for LM tasks/domains

Training methods for retrieval-augmented LMs

- Independent training
- **Sequential training**
- Joint training

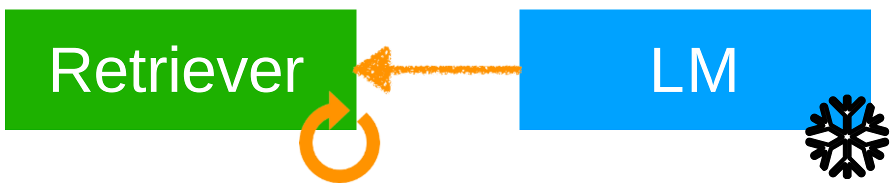
Sequential Training

- One component is first trained independently and then fixed
- The other component is trained with an objective that depends on the first one



RETRO (Borgeaud et al., 2021)

"Improving language models by retrieving from trillions of tokens"



REPLUG (Shi et al., 2023)

REPLUG: Retrieval-Augmented Black-Box Language Models

Sequential Training

- 👍 Work with off-the-shelf components (either a large index or a powerful LM)
- 👍 LMs are trained to effectively leverage retrieval results.
- 👍 Retrievers are trained to provide text that helps LMs the most.
- 👎 One component is still fixed and not trained.

Sequential Training

- Work with off-the-shelf components (either a large index or a powerful LM)
- LMs are trained to effectively leverage retrieval results.
- Retrievers are trained to provide text that helps LMs the most.
- One component is still fixed and not trained.

Let's jointly train retrieval models and LMs!

Training methods for retrieval-augmented LMs

- Independent training
- Sequential training
- **Joint training**

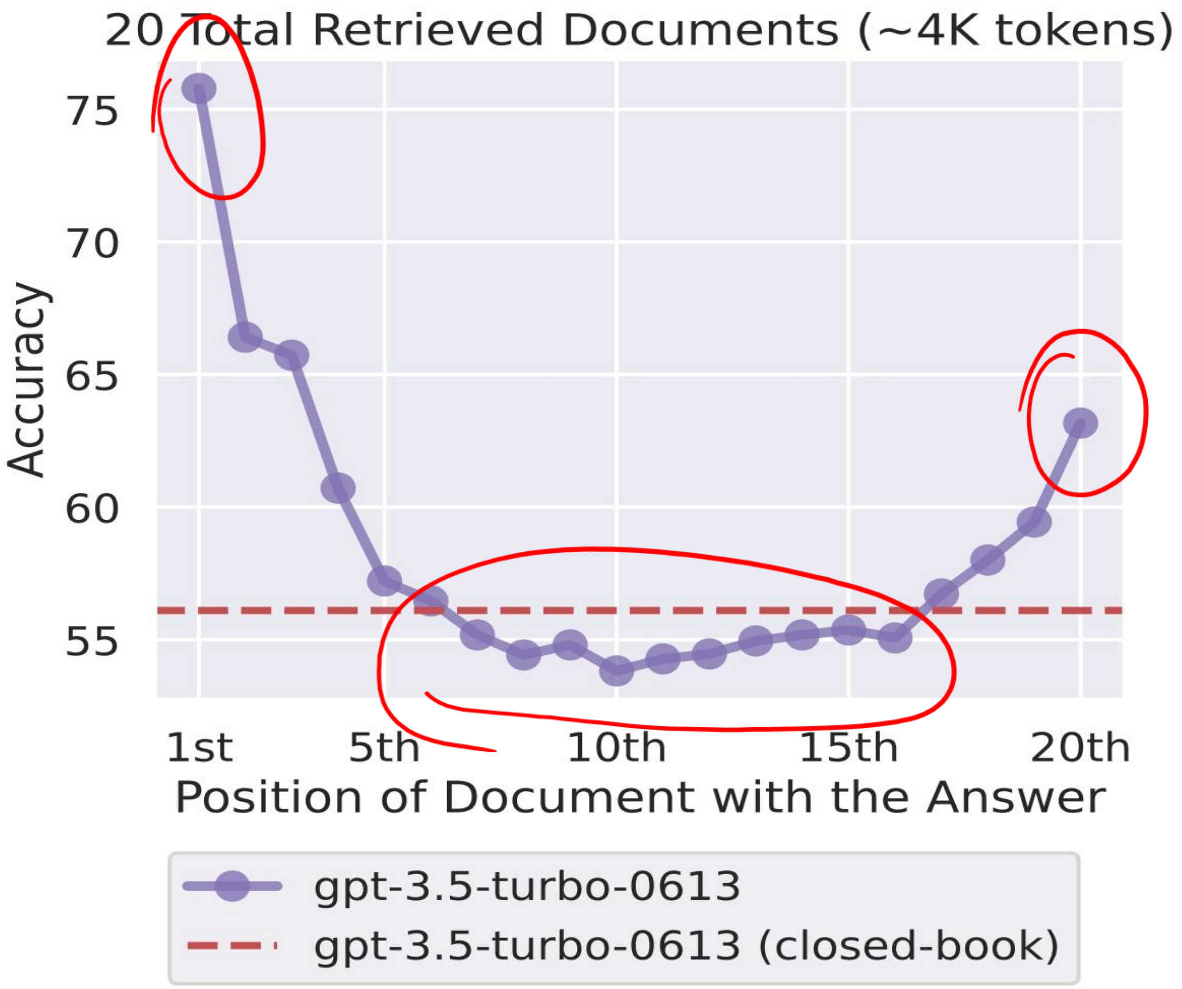
Joint Training

- End-to-end trained — each component is optimized
- Good performance
- Training is more complicated
  (async update, overhead, data batching, etc)
- Train-test discrepancy still remains

Outline

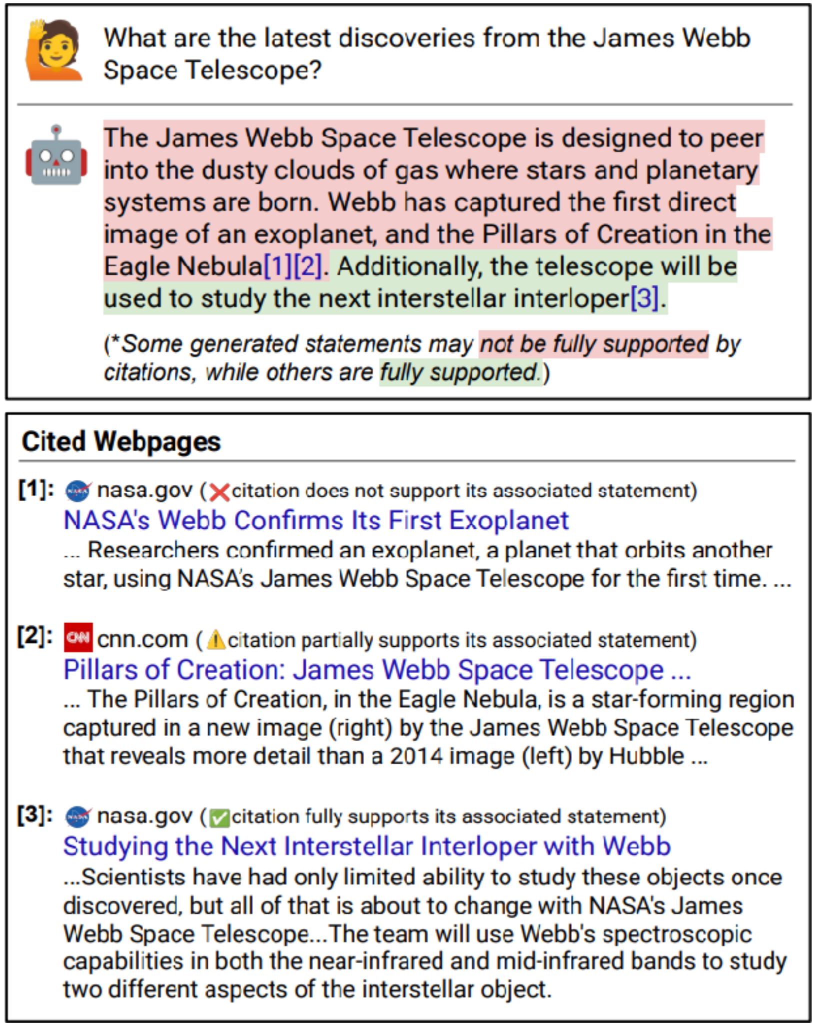
- Motivation
  - Drawbacks of Parametric LLMs – *hallucination, verification …*
  - Motivating Retrieval-based LLMs – *close book vs open book*
- Major components of Retrieval-based LLMs – *index, retrieve, read …*
- Retrieval Methods – *sparse, dense, reranking, black-box*
- REALM, RAG – *seminal works*
- Overview of Training Techniques – *independent, sequential, joint training …*
- Limitations – *lost in the middle, still hallucinating, retriever failures …*

Lost in the Middle!



- As Context Increases, Models Miss Relevant Info
- “lost-in-the- middle” (Liu et al. 2023) demonstrates that models pay less attention to things in the middle of context windows

Retrieval-augmented LMs can still hallucinate



Liu et al. Evaluating Verifiability in Generative Search Engines. Findings of EMNLP 2023.

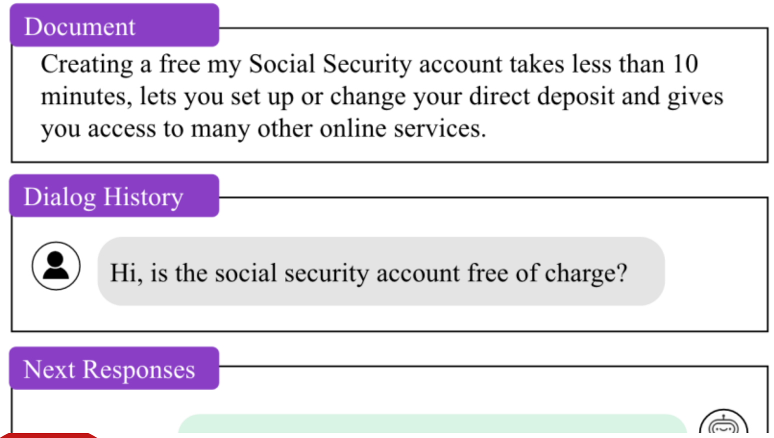
Quantifying Hallucination

Pointwise Mutual Information Based Metric and Decoding Strategy for Faithful Generation in Document Grounded Dialogs

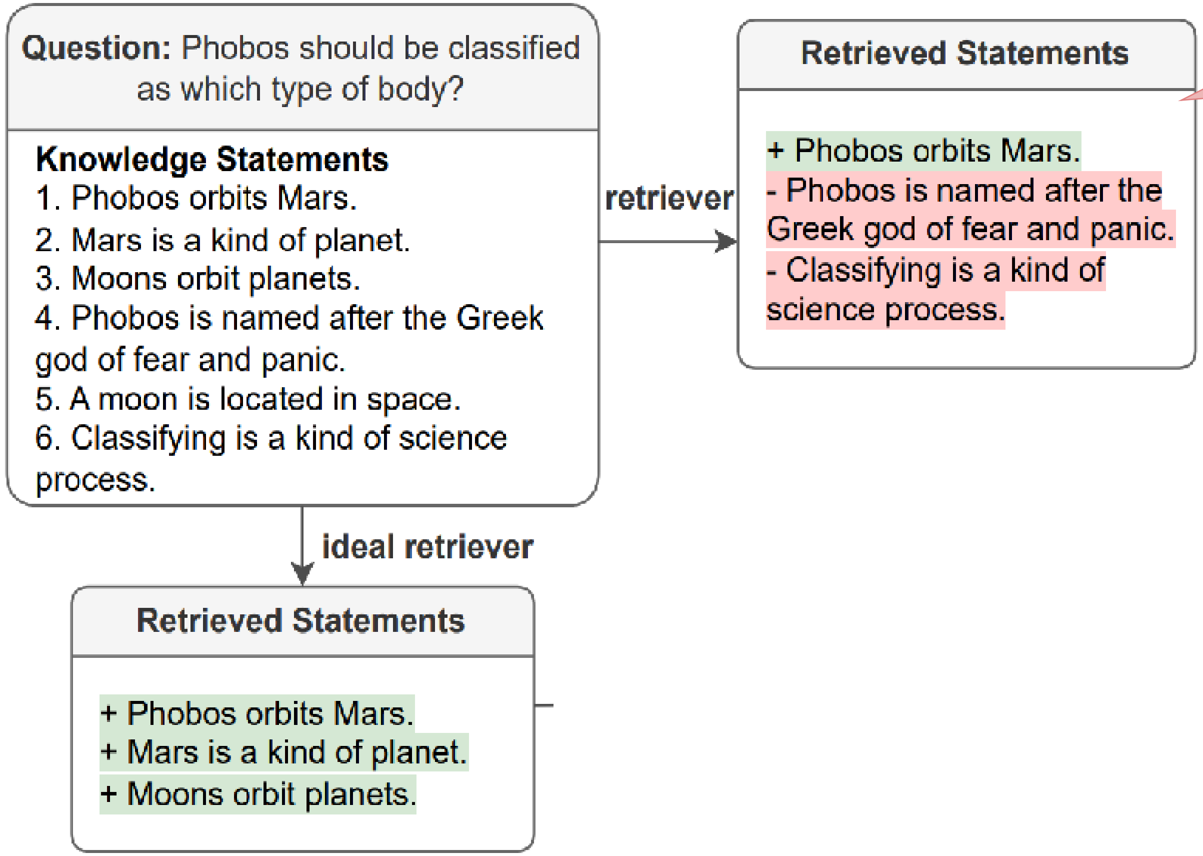
Yatin Nandwani, Vineet Kumar, Dinesh Raghu, Sachindra Joshi and **Luis A. Lastras**
IBM Research, AI
{yatin.nandwani@, vineeku6@in, diraghu1@in, jsachind@in, lastrasl@us}.ibm.com

Abstract

A major concern in using deep learning based generative models for document-grounded dialogs is the potential generation of responses that are not *faithful* to the underlying document. Existing automated metrics used for evaluating the faithfulness of response with respect to the grounding document measure the degree of similarity between the generated response and the document's content. However, these automated



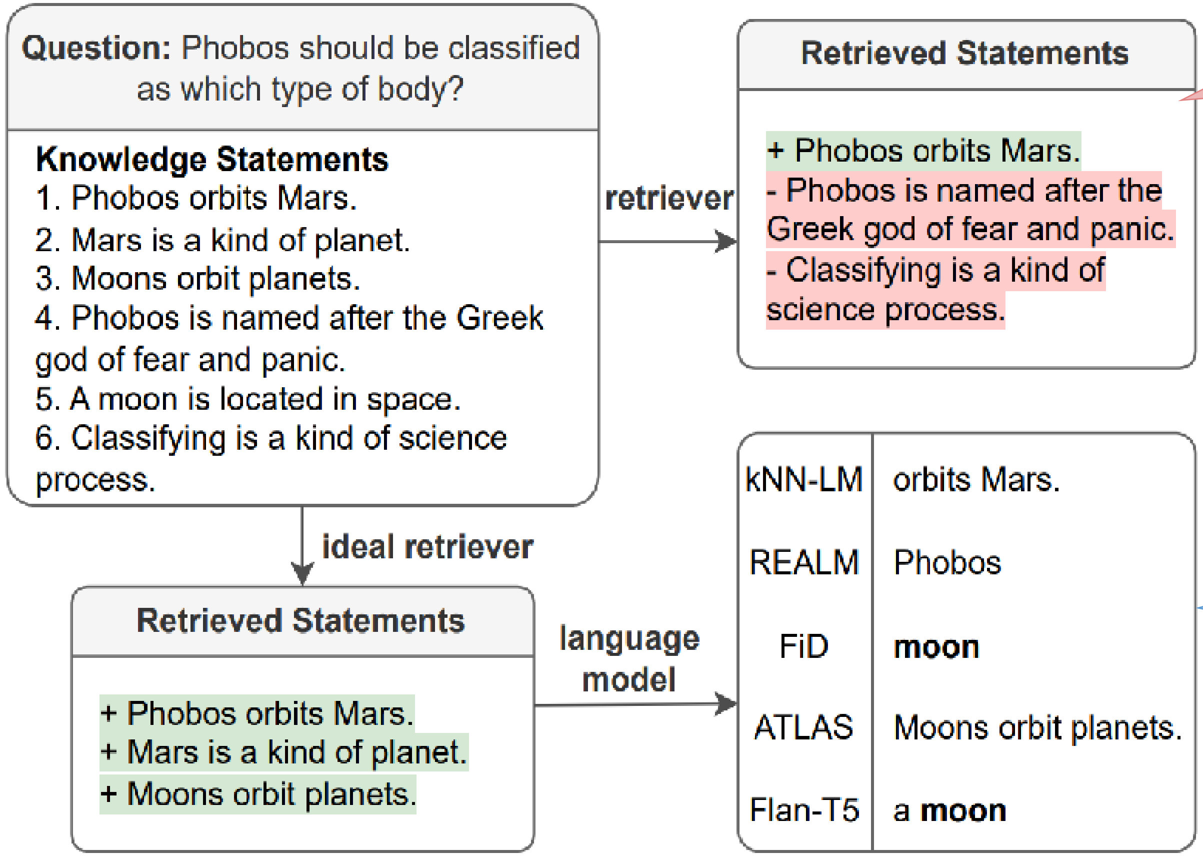
Retrieval Failures



- Retrieval fails to fetch correct information.

BehnamGhader et al. Can Retriever-Augmented Language Models Reason? The Blame Game Between the Retriever and the Language Model. EMNLP Findings 2023.

Reasoning Failures



- Retrieval fails to fetch correct information.

- Even with ideal retriever, LM fails to give right answer.

BehnamGhader et al. Can Retriever-Augmented Language Models Reason? The Blame Game Between the Retriever and the Language Model. EMNLP Findings 2023.

Adapt LM to Domain Corpus?

RAFT: Adapting Language Model to Domain Specific RAG

Tianjun Zhang Shishir G. Patil Naman Jain Sheng Shen Matei Zaharia Ion Stoica Joseph E. Gonzalez

`tianjunz@berkeley.edu, shishirpatil@berkeley.edu`

UC Berkeley

Abstract

Pretraining Large Language Models (LLMs) on large corpora of textual data is now a standard paradigm. When using these LLMs for many downstream applications, it is common to additionally bake in new knowledge (e.g., time-critical news, or private domain knowledge) into the pretrained model either through RAG-based-prompting, or finetuning. However, the optimal methodology for the model to gain such new knowledge remains an open question. In this pa-

ments). In these settings, general knowledge reasoning is less critical but instead, the primary goal is to maximize accuracy based on a given set of documents. Indeed, adapting LLMs to the specialized domains (e.g., recent news, enterprise private documents, or program resources constructed after the training cutoff) is essential to many emerging applications (Vu et al., 2023; Lazaridou et al., 2022) and is the focus of this work.

This paper studies the following question – *How to adapt pre-trained LLMs for Retrieval Augmented Generation (RAG) in specialized domains?*

Important Resources

- LangChain ; LlamaIndex – *overall frameworks*
- Lucene – *BM25 sparse retriever*
- ANNOY, FAISS, CromaDB - *dense embeddings and retrievers*
- Comprehensive RAG (CRAG) Benchmark – *KDD Cup 2024*

Content credits

- Graham Neubig's lecture - https://phontron.com/class/anlp2024/assets/slides/anlp-10-rag.pdf

- ACL 2023 Tutorial - https://acl2023-retrieval-lm.github.io/

Slide source: