Retrieval-based Language Models

Tianle Cai and Beiqi Zou

Nov 9th, 2022

Outline

1). Motivation: why retrieval-based LMs?

2). Related Work: existing retrieval-based LMs

3). Method: RETRO (Borgeaud et al., 2022)

4). Results

Prior work: GPT-2 & GPT-3

- GPT-3 is massive!
- 175B parameters (~117x GPT-2)

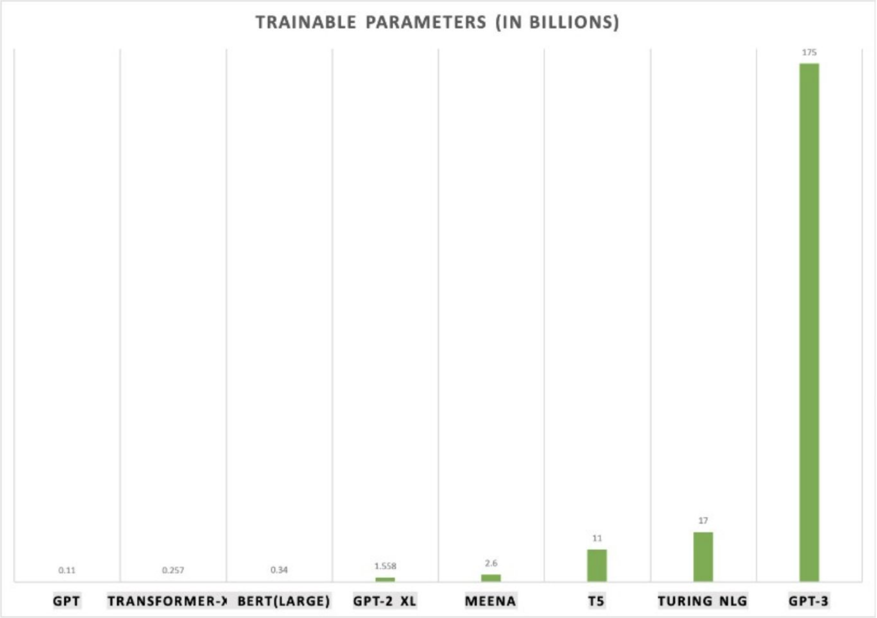


Figure from blog post

Motivation

- It seems scaling larger and larger models is the main way of improving the performance…

Motivation

- It seems scaling larger and larger models is the main way of improving the performance…

But with a tremendous increase in training energy cost!

Motivation

- It seems scaling larger and larger models is the main way of improving the performance…

But with a tremendous increase in training energy cost!

1). Additional computations at training and inference time

2). Increased memorization of the training data

Motivation

- It seems scaling larger and larger models is the main way of improving the performance…

But with a tremendous increase in training energy cost!

1). Additional computations at training and inference time

2). Increased memorization of the training data

Can we separate language information from world knowledge information?

Retrieval-based Language Models

- Knowledge is encoded explicitly
- The model **learns to search** for relevant passages, then **use the retrieved information** for crafting knowledgeable response

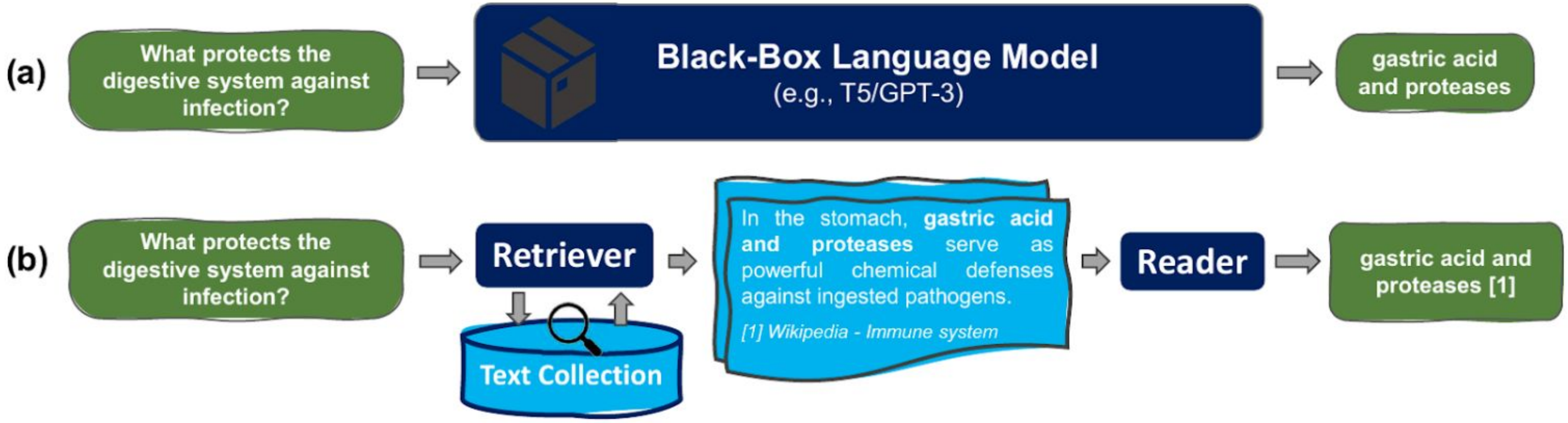


Figure from *[stanford blog post]*

Why is retrieval important?

- Tackling inefficiency
  - Retrieval-based models can be much **smaller and faster**

Why is retrieval important?

- Tackling inefficiency
  - Retrieval-based models can be much **smaller and faster**

- Tackling static knowledge
  - The retrieval knowledge store can be **efficiently updated or expanded** by modifying the text corpus

Figure from Danqi's talk

Why is retrieval important?

- Tackling inefficiency
  - Retrieval-based models can be much **smaller and faster**

- Tackling static knowledge
  - The retrieval knowledge store can be **efficiently updated or expanded** by modifying the text corpus

- Tackling Opaqueness
  - We are able to inspect the sources the model retrieved, which is more **transparent**

Existing methods

	# Retrieval tokens		Granularity		Retriever training	Retrieval integration
Continuous Cache	$\mathcal{O}(10^3)$		Token		Frozen (LSTM)	Add to probs
*k*NN-LM	$\mathcal{O}(10^9)$	Type 1	Token		Frozen (Transformer)	Add to probs
SPALM	$\mathcal{O}(10^9)$		Token		Frozen (Transformer)	Gated logits
DPR	$\mathcal{O}(10^9)$		Prompt		Contrastive proxy	Extractive QA
REALM	$\mathcal{O}(10^9)$		Prompt		End-to-End	Prepend to prompt
RAG	$\mathcal{O}(10^9)$	Type 2	Prompt	?	Fine-tuned DPR	Cross-attention
FID	$\mathcal{O}(10^9)$		Prompt		Frozen DPR	Cross-attention
EMDR[2]	$\mathcal{O}(10^9)$		Prompt		End-to-End (EM)	Cross-attention
RETRO (ours)	$\mathcal{O}(10^{12})$		**Chunk**		**Frozen (BERT)**	**Chunked cross-attention**

Type 1: *Token*-level Retrieval (mainly) for *LM* – augmenting prediction of next token

Type 2: *Passage*-level Retrieval (mainly) for ***QA*** – retrieving passages relevant to the question

Type 1: *Token*-level retrieval for *LM*

- Augment LM model with *k*NN-based model.
- Target is the next token.

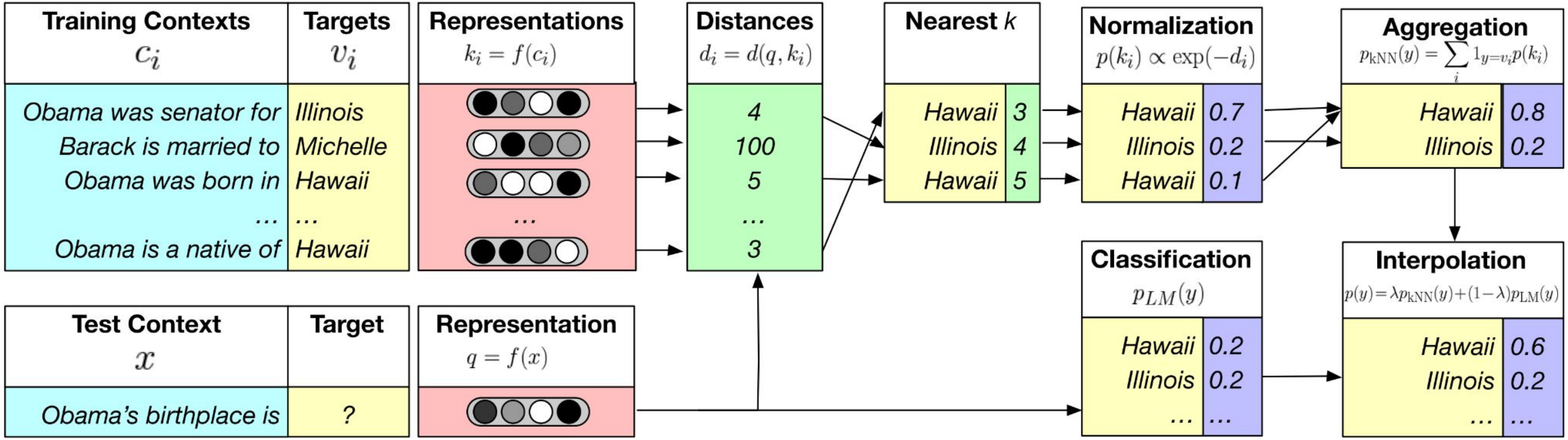


*Figure from k*NN-LM *paper (Khandelwal et al. 2019)*

Type 1: *Token*-level retrieval for *LM*

$$p_{\mathrm{kNN}}(y|x) \propto \sum_{(k_i, v_i) \in \mathcal{N}} \mathbb{1}_{y=v_i} \exp(-d(k_i, f(x)))$$

$$p(y|x) = \lambda \, p_{\mathrm{kNN}}(y|x) + (1-\lambda) \, p_{\mathrm{LM}}(y|x)$$

- No interaction between context encoder (for retrieval) and LM during training.
- What's the relationship between lambda and the size of database?

Type 1: *Token*-level retrieval for *LM*

> *No interaction between Context encoder and LM during training*.

How to train them together?

- SPALM: Adding an extra gating network to post-process the retrieved data.
- **TRIME: Training with in-batch memories.**
    - Incorporating retrieval into the training objective:

$$P(w \mid c) \propto \exp(E_w^\top f_\theta(c)) + \sum_{(c_j, x_j) \in \mathcal{M}_{\text{train}} : x_j = w} \exp(\text{sim}(g_\theta(c), g_\theta(c_j))).$$

Type 1: *Token*-level retrieval for *LM*

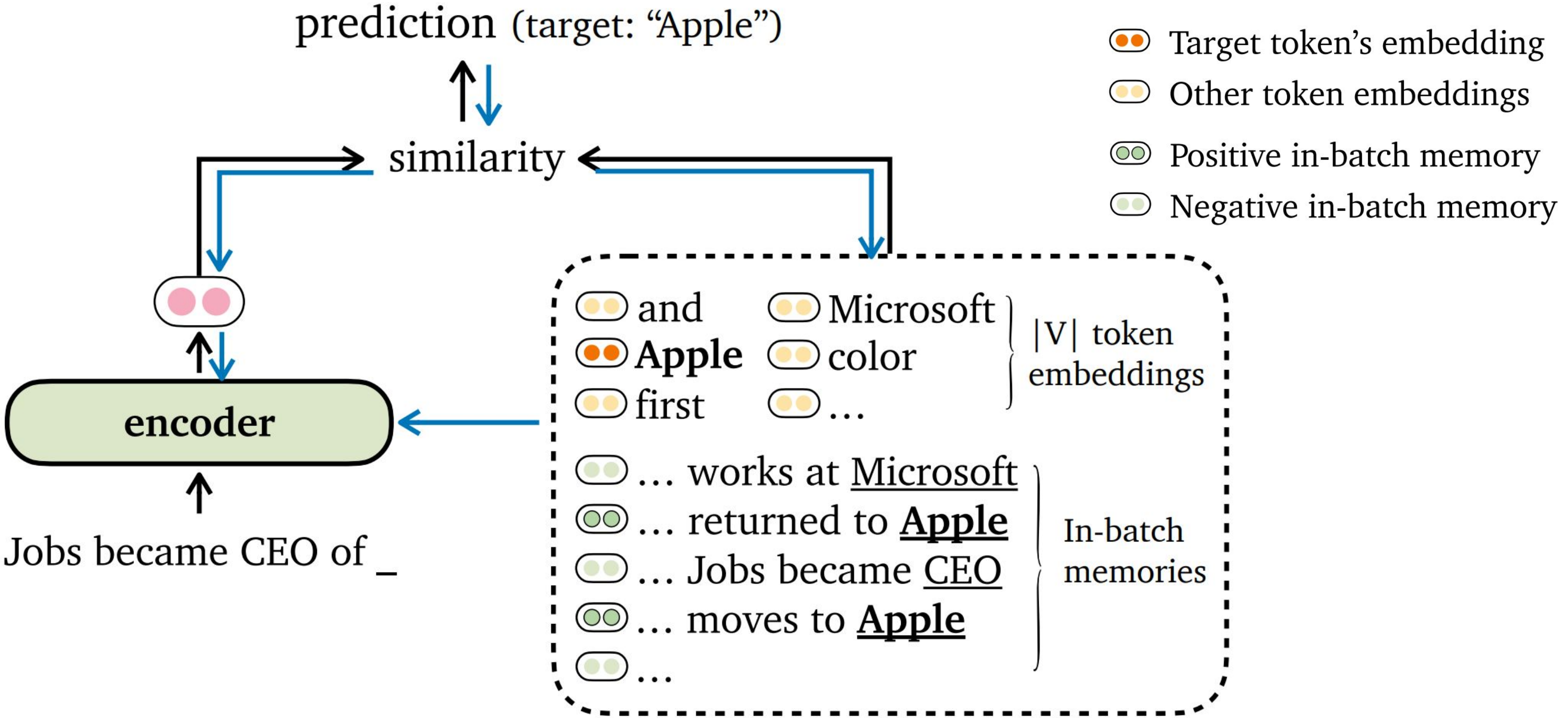


Figure from TRIME paper (Zhong et al. 2022)

Existing methods

	# Retrieval tokens		Granularity	Retriever training	Retrieval integration
Continuous Cache	$\mathcal{O}(10^3)$		Token	Frozen (LSTM)	Add to probs
*k*NN-LM	$\mathcal{O}(10^9)$	Type 1	Token	Frozen (Transformer)	Add to probs
SPALM	$\mathcal{O}(10^9)$		Token	Frozen (Transformer)	Gated logits
DPR	$\mathcal{O}(10^9)$		Prompt	Contrastive proxy	Extractive QA
REALM	$\mathcal{O}(10^9)$		Prompt	End-to-End	Prepend to prompt
RAG	$\mathcal{O}(10^9)$	Type 2	Prompt	Fine-tuned DPR	Cross-attention
FID	$\mathcal{O}(10^9)$		Prompt	Frozen DPR	Cross-attention
EMDR[2]	$\mathcal{O}(10^9)$		Prompt	End-to-End (EM)	Cross-attention
RETRO (ours)	$\mathcal{O}(\mathbf{10^{12}})$		**Chunk**	**Frozen (BERT)**	**Chunked cross-attention**

Type 1: *Token*-level Retrieval (mainly) for *LM* – augmenting prediction of next token

Type 2: *Passage*-level Retrieval (mainly) for *QA* – retrieving passages relevant to the question

Type 2: *Passage*-level Retrieval for *QA*

- Contrastively train the retriever.
- Can be plugged into a QA system for retrieving context.

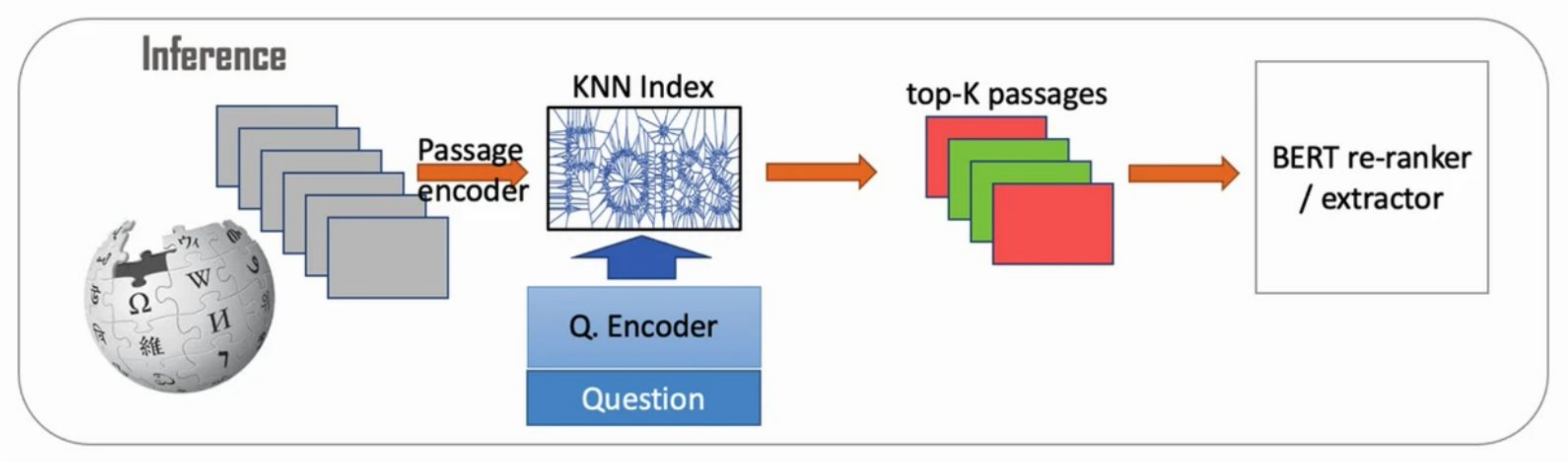


Figure from the talk of DRP paper (Karpukhin et al. 2020)

Type 2: *Passage*-level Retrieval for *QA*

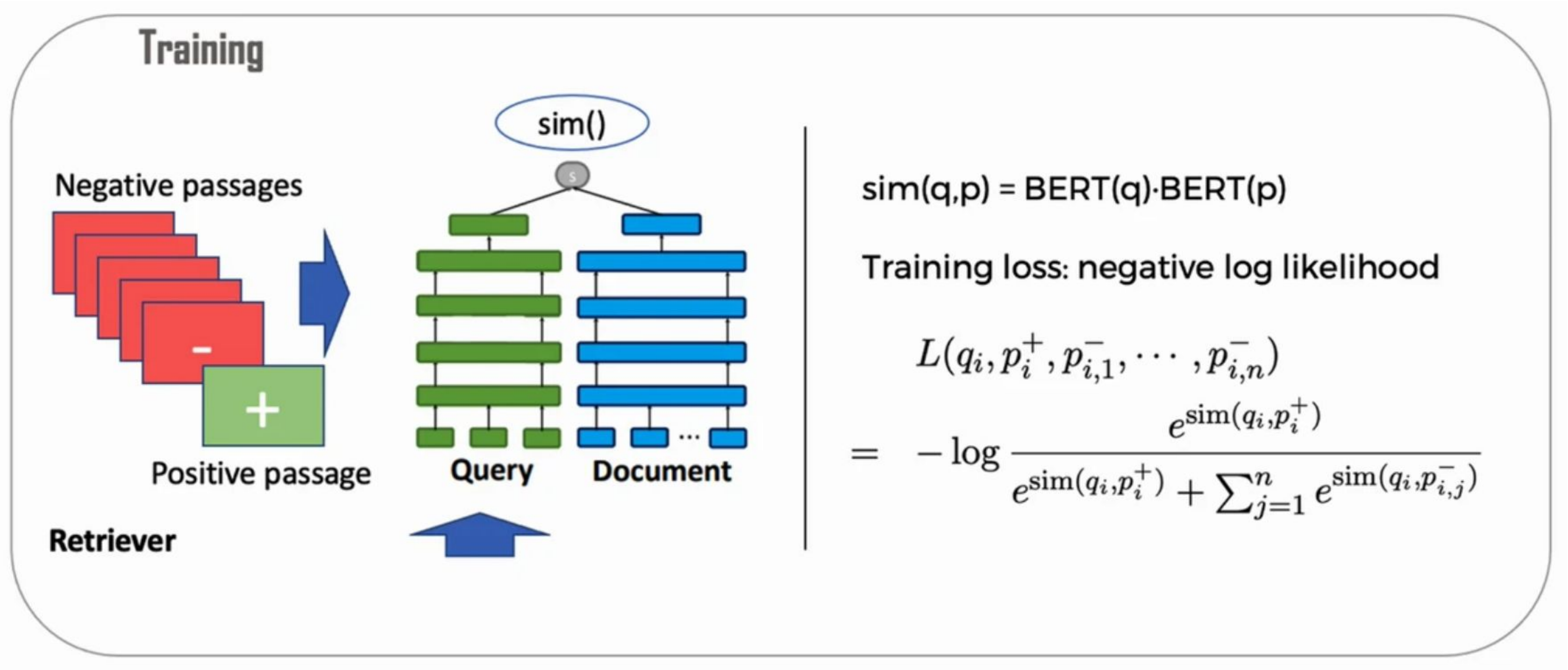


Figure from the talk of DRP paper (Karpukhin et al. 2020)

Existing methods

	# Retrieval tokens	Granularity	Retriever training	Retrieval integration
Continuous Cache	$\mathcal{O}(10^3)$	Token	Frozen (LSTM)	Add to probs
*k*NN-LM	$\mathcal{O}(10^9)$	Token	Frozen (Transformer)	Add to probs
SPALM	$\mathcal{O}(10^9)$	Token	Frozen (Transformer)	Gated logits
DPR	$\mathcal{O}(10^9)$	Prompt	Contrastive proxy	Extractive QA
REALM	$\mathcal{O}(10^9)$	Prompt	End-to-End	Prepend to prompt
RAG	$\mathcal{O}(10^9)$	Prompt	Fine-tuned DPR	Cross-attention
FID	$\mathcal{O}(10^9)$	Prompt	Frozen DPR	Cross-attention
EMDR^2	$\mathcal{O}(10^9)$	Prompt	End-to-End (EM)	Cross-attention
RETRO (ours)	$\mathcal{O}(\mathbf{10^{12}})$	**Chunk**	**Frozen (BERT)**	**Chunked cross-attention**

"Limited" scale:

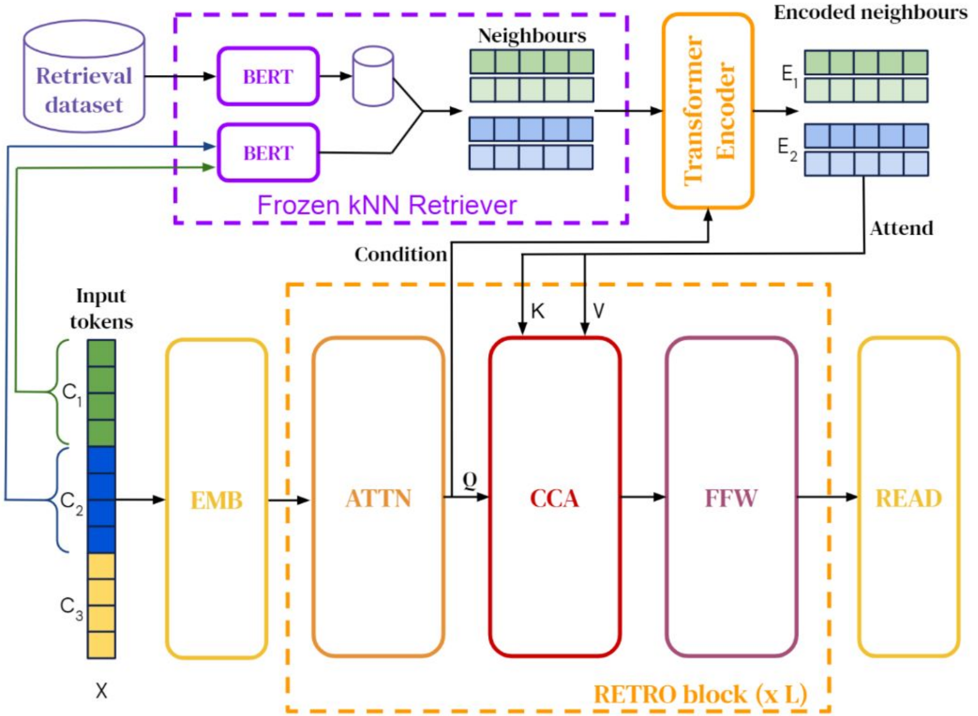
- Datasets are up to billions of tokens.
- Models are ~100M parameters.

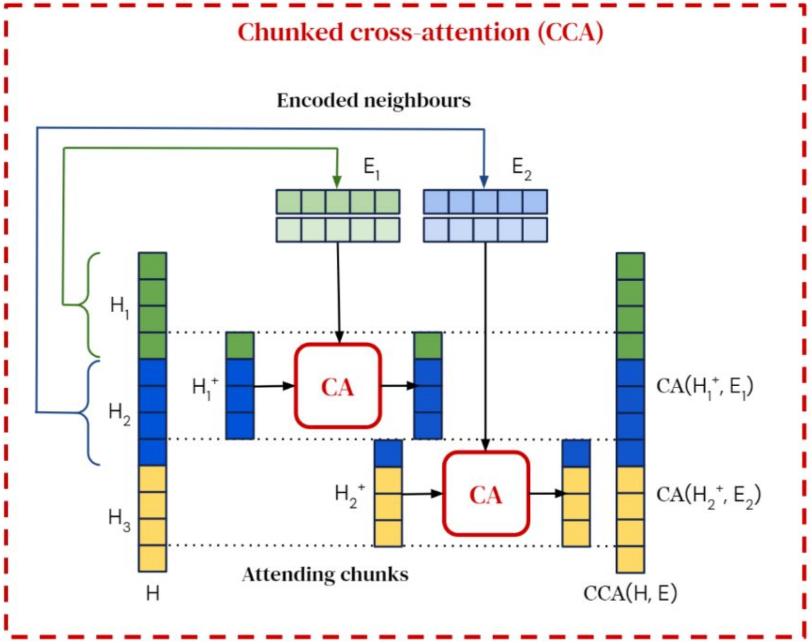
Improving language models by retrieving from trillions of tokens

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

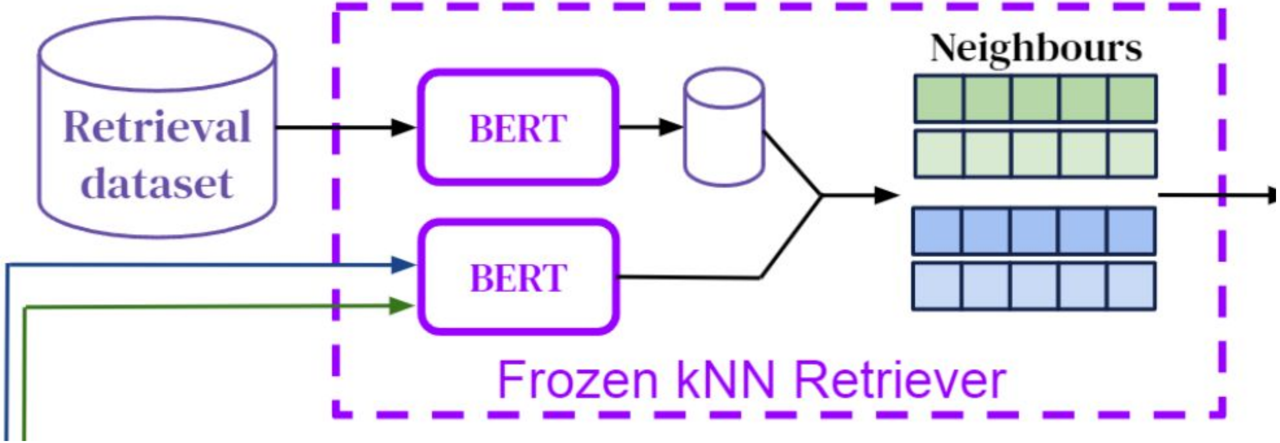
This paper: RETRO *architecture* (for *LM*)





Component 1: Frozen BERT encoder for retrieval

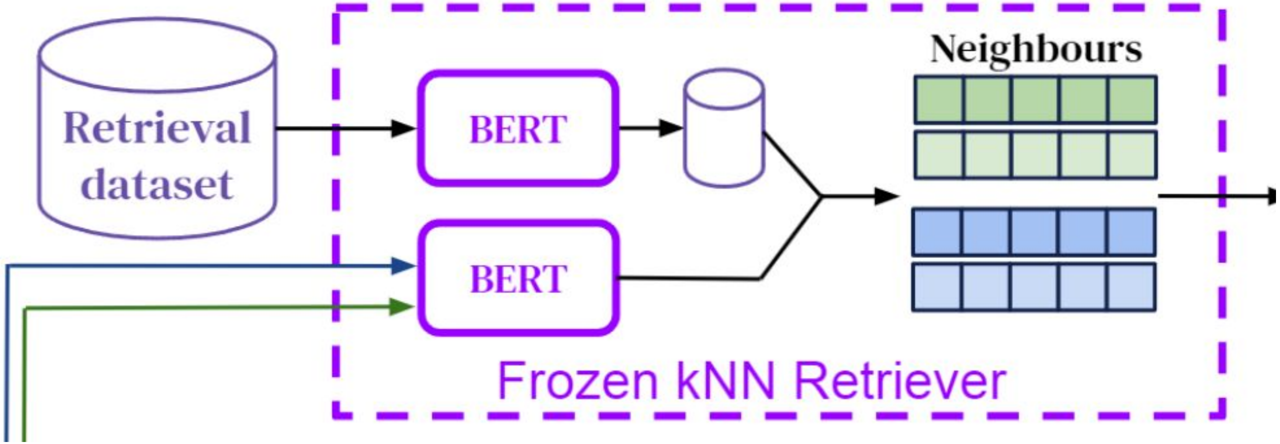
Why frozen the encoders given that training them is helpful (as shown in previous works like DPR)?



Component 1: Frozen BERT encoder for retrieval

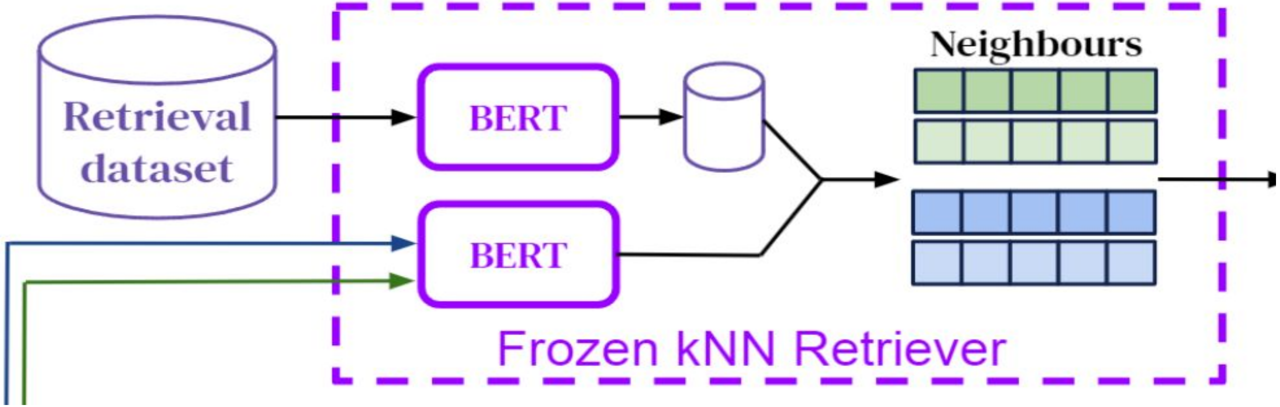
Why frozen the encoders given that training them is helpful (as shown in previous works like DPR)?

> *"avoid having to periodically re-compute embeddings over the entire database during training"*



Component 1: Frozen BERT encoder for retrieval

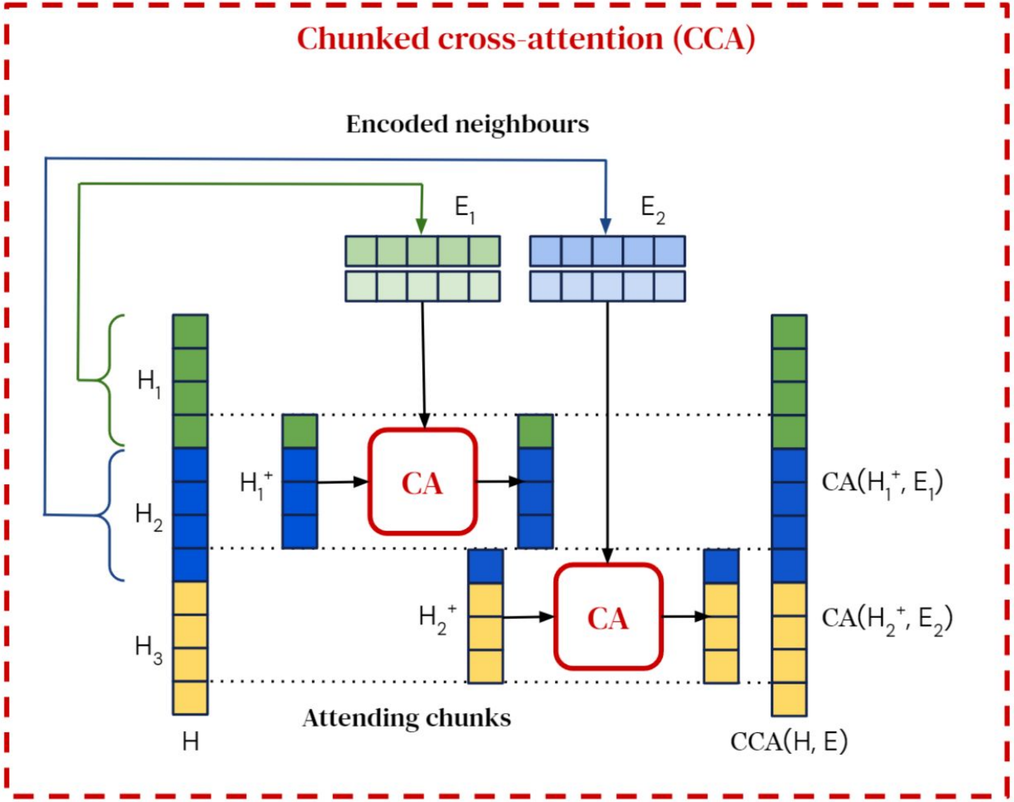
- Format of the retrieval neighbors: [N, F] where N is used as key and F is the continuation of N.
- Metric: d(C, N) = ||BERT(C) - BERT(N)||.
- RET(C) = ([N^1, F^1], ..., [N^k, F^k]).



Component 2: Chunked cross-attention (CCA)

Background

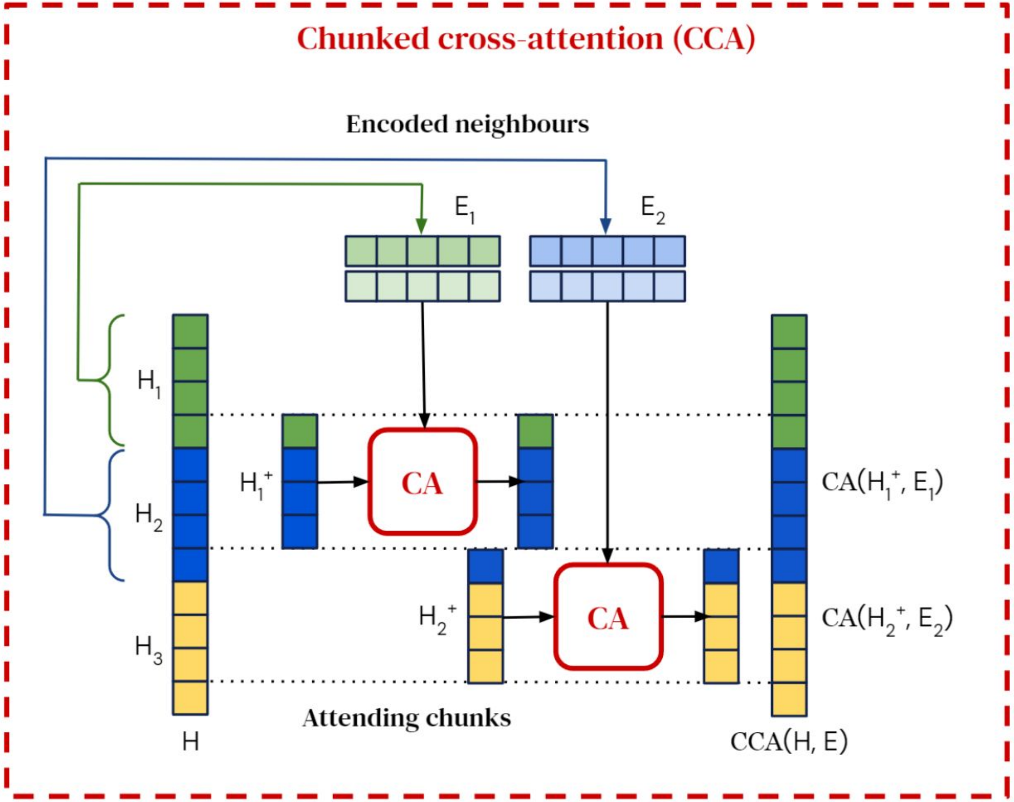
- Input chunks: Divide input of length 2048 into chunks of length 64.
- N, F in the retrieval database are also of length 64.



Component 2: Chunked cross-attention (CCA)

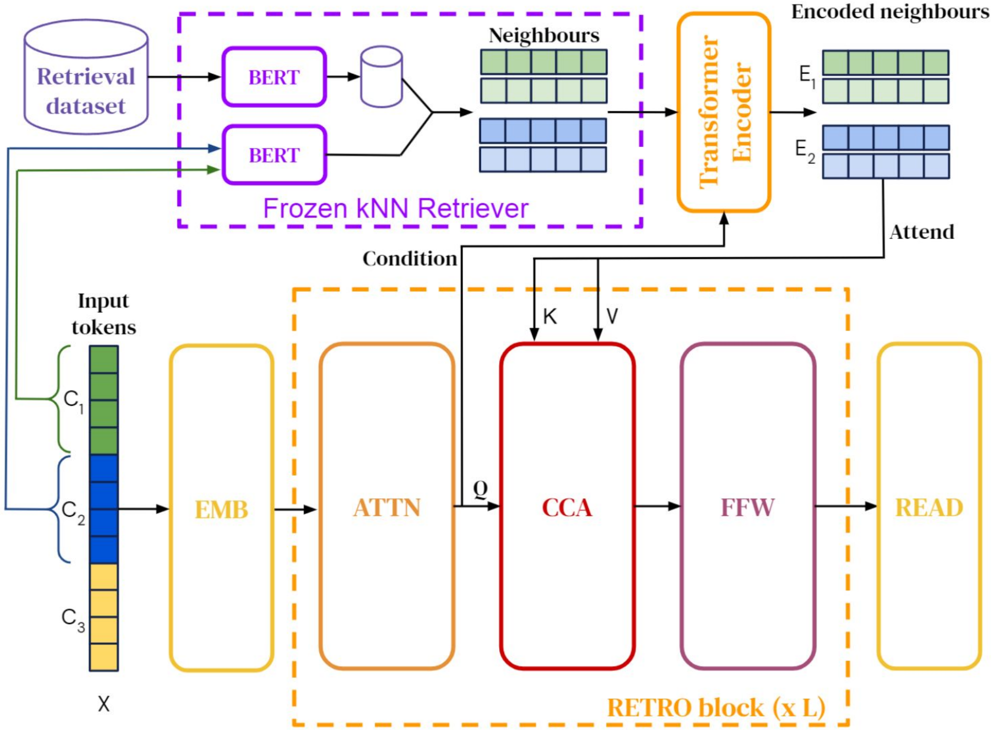
How to maintain causality?

- Chunk-wise autoregressive
- Adding (encoded) neighbor of chunk *i* to the last token of chunk *i* and chunk *i+1*.
- Intuition: Ideally if neighbor is exactly same as chunk, its continuation will be the next chunk.



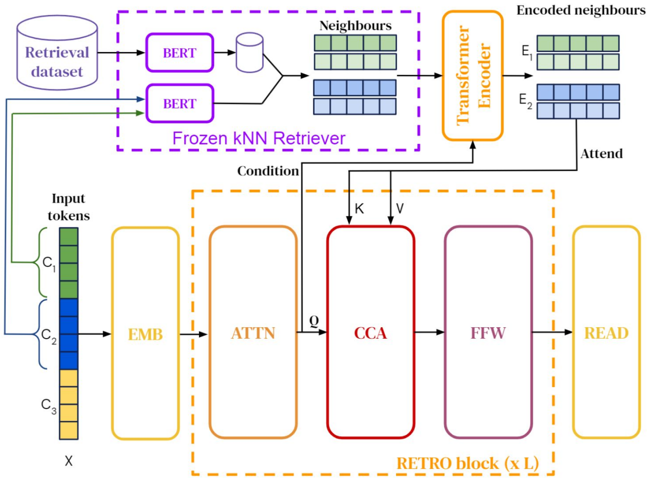
Miscellaneous

- Encoder for post-processing neighbors: A small (19M params) BERT encoder for *conditioning* neighbors on query.
- In the implementation, the retrieval models contain one RETRO-block every 3 blocks, starting from layer 6. (Why?)



Q1. Describe how the text is stored in RETRO's database (keys and values) and how they are encoded and integrated into the language model.

- Format of the retrieval neighbors:
    - [N, F] where N is used as key and F is the continuation of N.
- Chunked cross-attention.



Experiments Outline

1). Models and Datasets

2). Scaling on Models and Data

3). RETRO-fitting

4). RETRO on Question Answering

5). Evaluations on leakage filtering

Models

1). Baseline Transformer

- Replace LayerNorm with RMSNorm
- Relative position encodings

2). RETRO [Off]

- Without retrieval data

3). RETRO [On]

Models

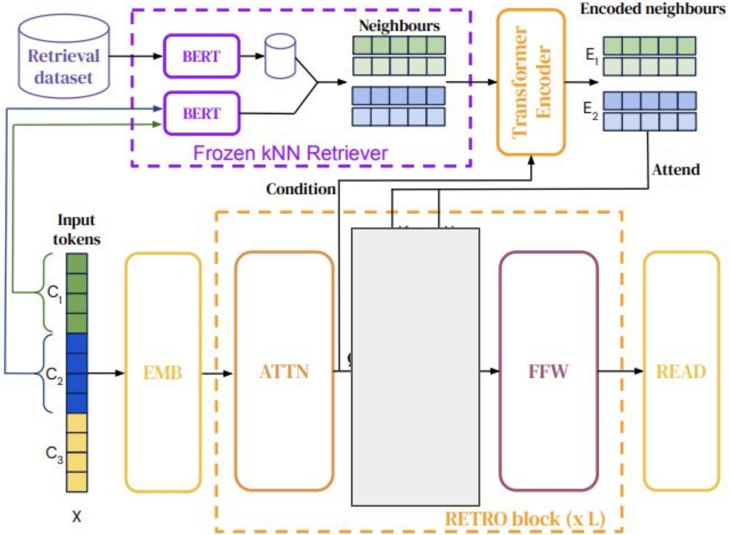
1). Baseline Transformer

- Replace LayerNorm with RMSNorm
- Relative position encodings

2). RETRO [Off]

- Without retrieval data

3). RETRO [On]



Models

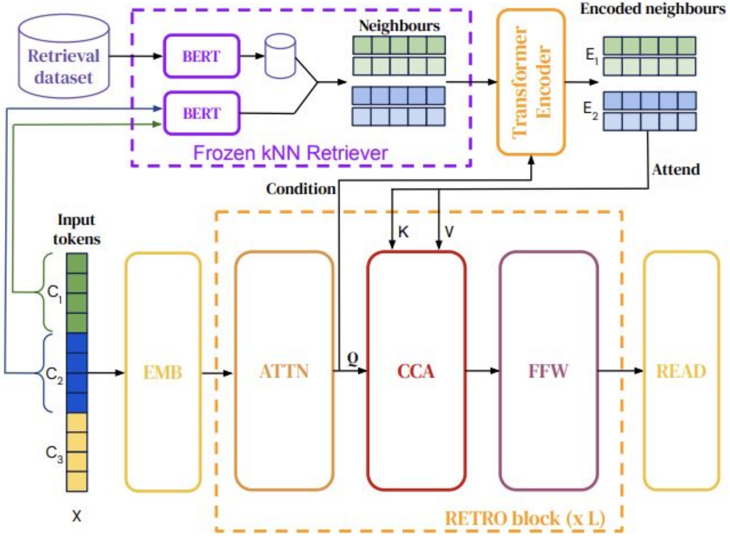
1). Baseline Transformer

- Replace LayerNorm with RMSNorm
- Relative position encodings

2). RETRO [Off]

- Without retrieval data

3). RETRO [On]



Models

1). Baseline Transformer

2). RETRO [Off]

3). RETRO [On]

Baseline parameters	RETRO	d	d_{ffw}	# heads	Head size	# layers
132M	172M (+30%)	896	3,584	16	64	12
368M	425M (+15%)	1,536	6,144	12	128	12
1,309M	1,451M (+11%)	2,048	8,192	16	128	24
6,982M	7,532M (+8%)	4,096	16,384	32	128	32

Models

1). Baseline Transformer

2). RETRO [Off]

3). RETRO [On]

Baseline parameters	RETRO		d	d_{ffw}	# heads	Head size	# layers
132M	172M	(+30%)	896	3,584	16	64	12
368M	425M	(+15%)	1,536	6,144	12	128	12
1,309M	1,451M	(+11%)	2,048	8,192	16	128	24
6,982M	7,532M	(+8%)	4,096	16,384	32	128	32

Less percentage increase for larger models

Datasets

- Multilingual version of MassiveText (Rae et al., 2021) for both training and retrieval data

Source	Token count (M)	Documents (M)	Multilingual	Sampling frequency
Web	977,563	1,208	Yes	55%
Books	3,423,740	20	No	25%
News	236,918	398	No	10%
Wikipedia	13,288	23	Yes	5%
GitHub	374,952	143	No	5%

Datasets

- C4 (Raffel et al., 2020)
- The Pile (Gao et al., 2020)
- Curation Corpus (Curation, 2020)
- A set of manually selected Wikipedia articles

Bits-per-byte (bpb)

- WikiText-103 (Merity et al., 2017)

Perplexity

- Lambada (Paperno et al., 2016)

Accuracy

Example Data from LAMBADA

- Designed to evaluate the capabilities of computational models for text understanding by means of a word prediction task
- Models must be able to keep track of information in the broader discourse
- Measured in *accuracy*

Context: “Why?” “I would have thought you’d find him rather dry,” she said. “I don’t know about that,” said <u>Gabriel</u>. “He was a great craftsman,” said Heather. “That he was,” said Flannery.
Target sentence: “And Polish, to boot,” said _____.
Target word: Gabriel

Figure from LAMBADA (Paperno et al., 2016)

Evaluation Metric: Bits-per-bytes

$$\forall\, \alpha \in [0,1], \quad C_\alpha \triangleq \{C \in \mathcal{C}, r(C) \leqslant \alpha\}, \quad \mathrm{bpb}(\alpha) \triangleq \frac{\sum_{C \in C_\alpha} \ell(C)}{\sum_{C \in C_\alpha} N(C)}$$

Evaluation Metric: Bits-per-bytes

$$\forall\, \alpha \in [0, 1], \quad C_\alpha \triangleq \{C \in \mathcal{C}, \boxed{r(C)} \leqslant \alpha\}, \quad \mathrm{bpb}(\alpha) \triangleq \frac{\sum_{C \in C_\alpha} \ell(C)}{\sum_{C \in C_\alpha} N(C)}$$

Evaluation Metric: Bits-per-bytes

1). Split the evaluation sequences into chunks of length m <= 64

2). For each evaluation chunk C, retrieve 10 closest neighbours in the training data

3). Compute the longest token substring common to both the evaluation chunk and its neighbours

$$r(C) = \frac{\boxed{s}}{\boxed{m}}$$

Evaluation Metric: Bits-per-bytes

1). Split the evaluation sequences into chunks of length m <= 64

2). For each evaluation chunk C, retrieve 10 closest neighbours in the training data

3). Compute the longest token substring common to both the evaluation chunk and its neighbours

$$r(C) = \frac{s}{m}$$

- Ranges from 0 (chunk never seen) to 1 (chunk entirely seen)
- Indicates how much overlap there is between the evaluation chunk and training data

Evaluation Metric: Bits-per-bytes

4). Obtain the log-likelihood of each chunk C, and the number of bytes it encodes

<u>*Filtered*</u> *bits-per-bytes (bpb)* as follows:

$$\forall\, \alpha \in [0,1], \quad \mathcal{C}_\alpha \triangleq \{C \in \mathcal{C}, r(C) \leqslant \alpha\}, \quad \mathrm{bpb}(\alpha) \triangleq \frac{\sum_{C \in \mathcal{C}_\alpha} \boxed{\ell(C)}}{\sum_{C \in \mathcal{C}_\alpha} \boxed{N(C)}}$$

Evaluation Metric: Bits-per-bytes

4). Obtain the log-likelihood of each chunk C, and the number of bytes it encodes

<u>*Filtered*</u> *bits-per-bytes (bpb)* as follows:

$$\forall\, \alpha \in [0,1], \quad \mathcal{C}_\alpha \triangleq \{C \in \mathcal{C}, r(C) \leqslant \alpha\}, \quad \mathrm{bpb}(\alpha) \triangleq \frac{\sum_{C \in \mathcal{C}_\alpha} \boxed{\ell(C)}}{\sum_{C \in \mathcal{C}_\alpha} \boxed{N(C)}}$$

Shows bpb on the set of chunks that overlap less than α% with the training chunks

Evaluation Metric: Bits-per-bytes

4). Obtain the log-likelihood of each chunk C, and the number of bytes it encodes

Filtered *bits-per-bytes (bpb)* as follows:

$$\forall\, \alpha \in [0,1], \quad \mathcal{C}_\alpha \triangleq \{C \in \mathcal{C}, r(C) \leqslant \alpha\}, \quad \mathrm{bpb}(\alpha) \triangleq \frac{\sum_{C \in \mathcal{C}_\alpha} \boxed{\ell(C)}}{\sum_{C \in \mathcal{C}_\alpha} \boxed{N(C)}}$$

Shows bpb on the set of chunks that overlap less than α% with the training chunks

Full evaluation *bits-per-bytes (bpb)* performance is recovered by bpb(1)

C4, The Pile, Curation, Manually selected Wiki articles

Evaluation Metric: Bits-per-bytes

4). Obtain the log-likelihood of each chunk C, and the number of bytes it encodes

Filtered *bits-per-bytes (bpb)* as follows:

$$\forall\, \alpha \in [0,1], \quad \mathcal{C}_\alpha \triangleq \{C \in \mathcal{C}, r(C) \leqslant \alpha\}, \quad \mathrm{bpb}(\alpha) \triangleq \frac{\sum_{C \in \mathcal{C}_\alpha} \boxed{\ell(C)}}{\sum_{C \in \mathcal{C}_\alpha} \boxed{N(C)}}$$

Shows bpb on the set of chunks that overlap less than α% with the training chunks

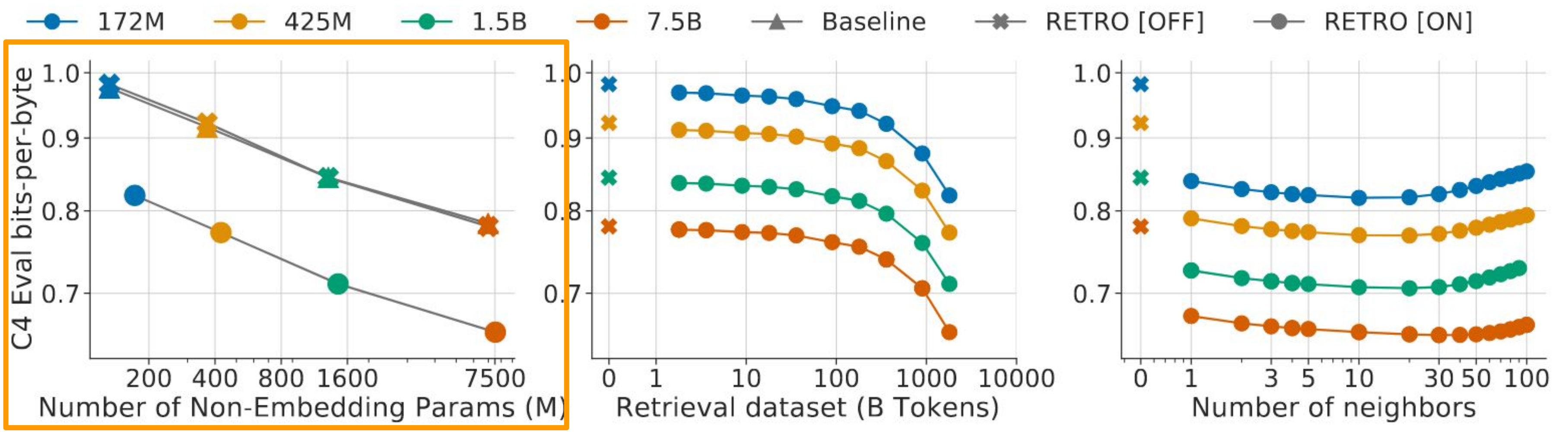
Full evaluation *bits-per-bytes (bpb)* performance is recovered by bpb(1)

Tokenizer agnostic

The lower, the better

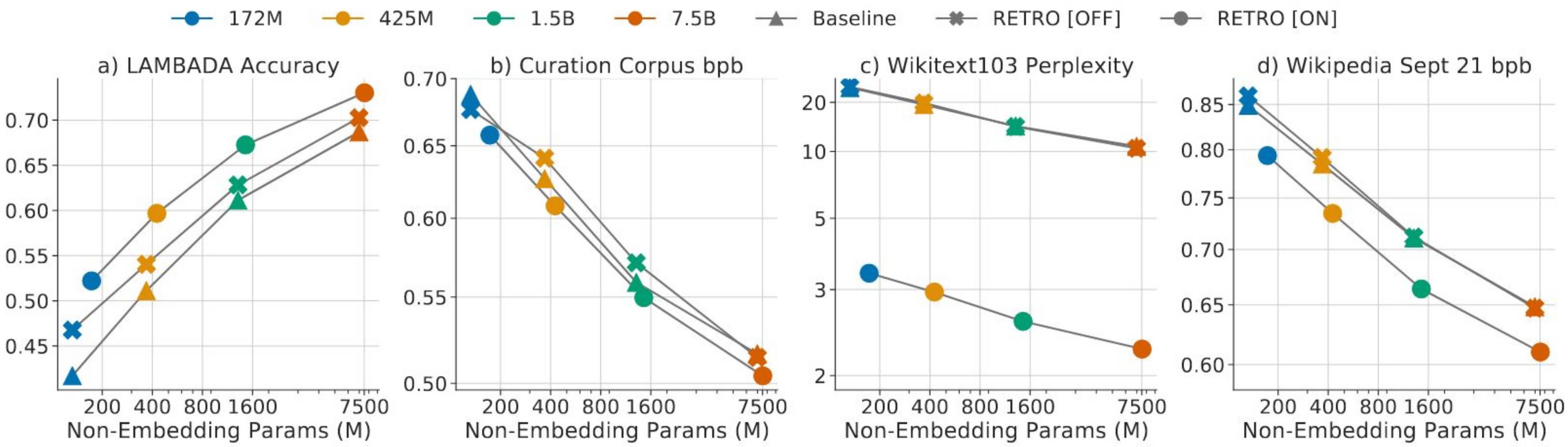
Model Scaling

- On all datasets, RETRO outperforms the baseline at all model sizes
- Improvements do not diminish as we scale the models



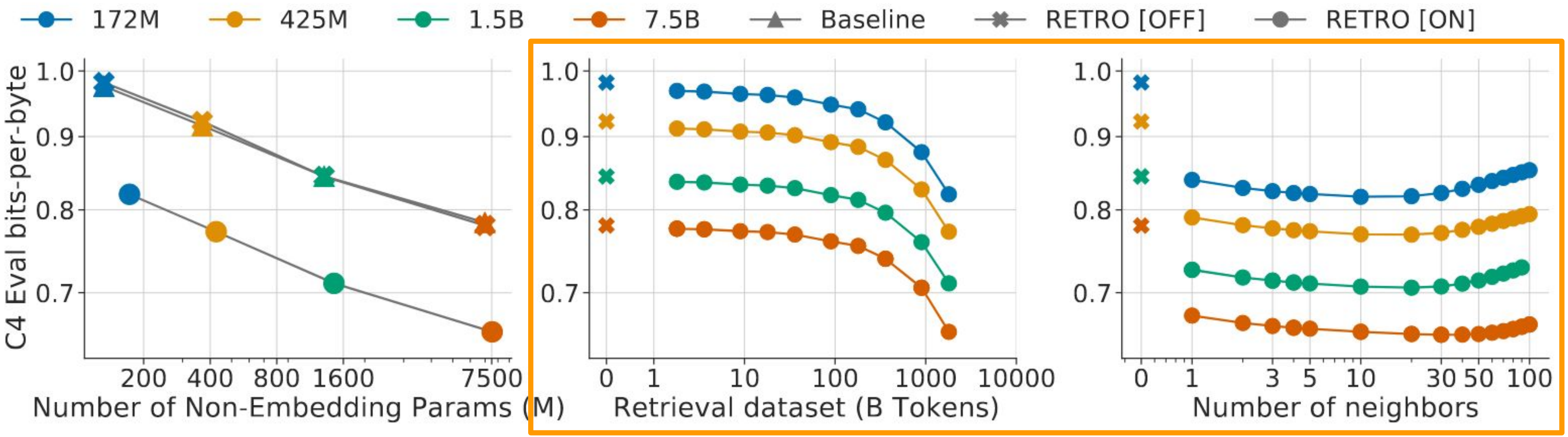
Model Scaling

- On all datasets, RETRO outperforms the baseline at all model sizes
- Improvements do not diminish as we scale the models



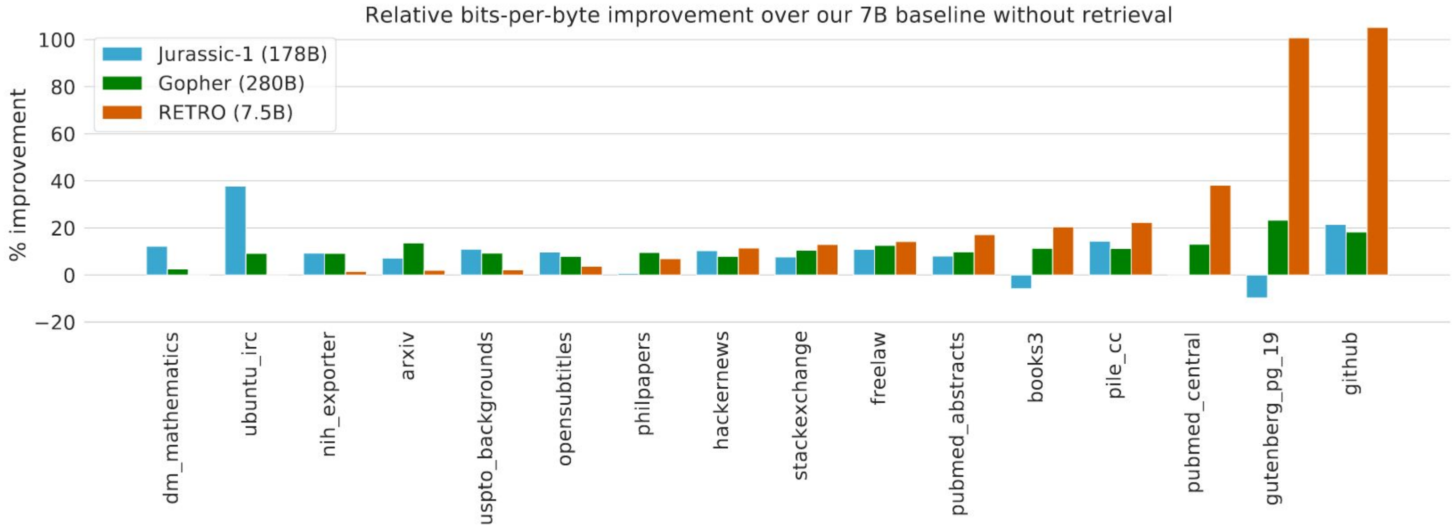
Data Scaling

- Scaling the retrieval database at evaluation improves performance



Relative bpb improvement on the Pile

- RETRO outperforms baseline on almost all datasets except *dm_mathematics* and *ubuntu_irc*



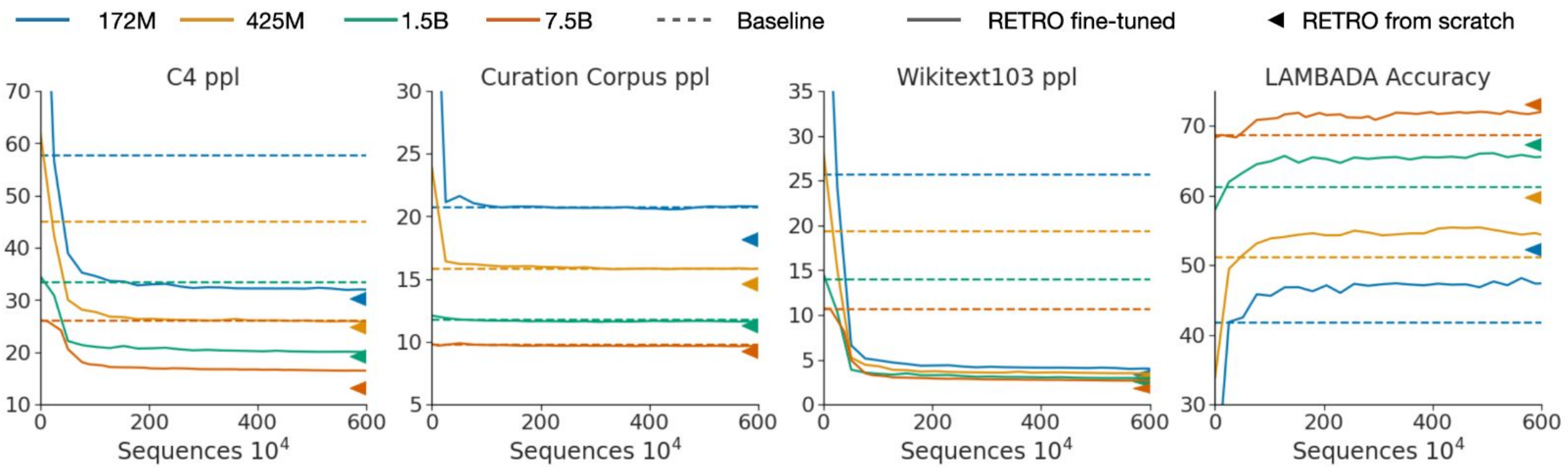
[Jurassic-1](#) and [Gopher](#) outperform GPT-3!

RETRO-fitting

- Extend baseline models into RETRO models
- Freeze the pre-trained weights
- Only train chunked cross-attention and neighbour encoder parameters

RETRO-fitting

- RETRO-fitting Models quickly surpasses the performance of baseline models
- Close to RETRO models trained from scratch



Performance on QA

- Fine-tune on the Natural Questions dataset
- Measures exact string match accuracy

The Green Mile (film)	what was the prison called in the green mile	In 1935, Paul supervises officers Brutus Howell, Dean Stanton, Harry Terwilliger, and Percy Wetmore at Cold Mountain Penitentiary. Paul is suffering from a severe bladder infection and receives John Coffey, a physically imposing but mentally challenged black man, into his custody. John had been sentenced to death after being convicted of raping and murdering two white girls. One of the other inmates is a Native-American named Arlen Bitterbuck, who is charged with murder and is the first to be executed. Percy demonstrates a severe sadistic streak, but, as the nephew of Louisiana's First Lady, he is beyond reproach. He is particularly abusive with inmate Eduard Delacroix; he breaks Del's fingers with his baton, steps on a pet mouse named Mr. Jingles, which Del had adopted, repeatedly calls him by a gay slur, and ultimately sabotages his execution by failing to soak the sponge used to conduct electricity to Del's head; Del dies screaming in pain.	Cold Mountain Penitentiary

Format in: “question: {question} \n answer: {answer}”

Figure from [Natural Questions]

Performance on QA

- Fine-tune on the Natural Questions dataset
- Measures exact string match accuracy

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 on QA

- RETRO 7.5B (DPR retrieval)
  - Has access to the question as well as the **top 20 DPR Wiki passages** and their titles via CCA

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 on QA

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

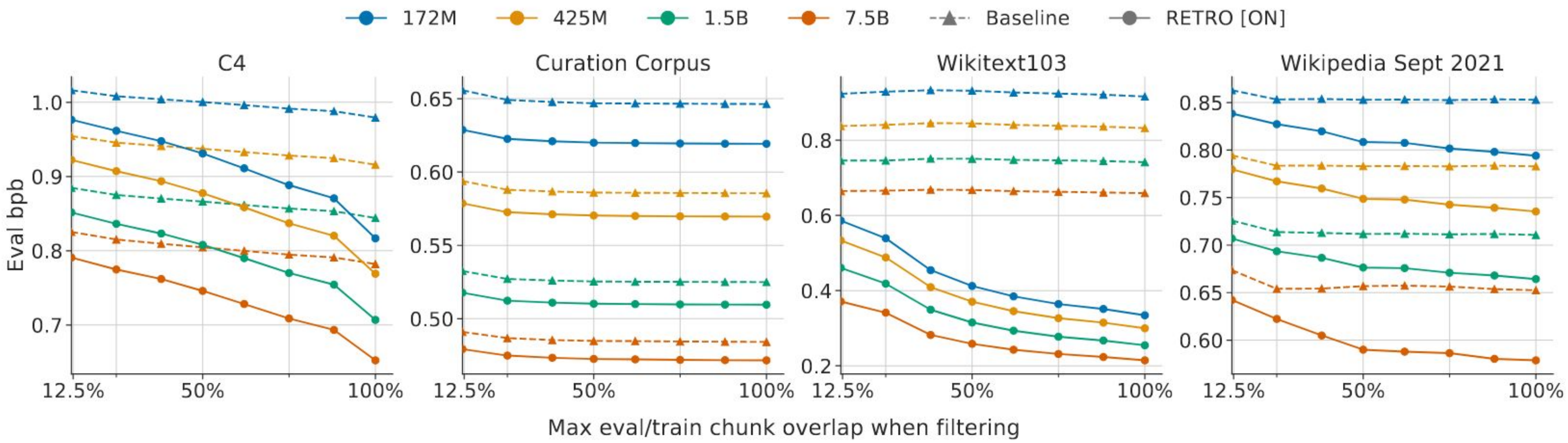
Performance on QA

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 wrt. Dataset Leakage

- Filtered bpb as eval loss

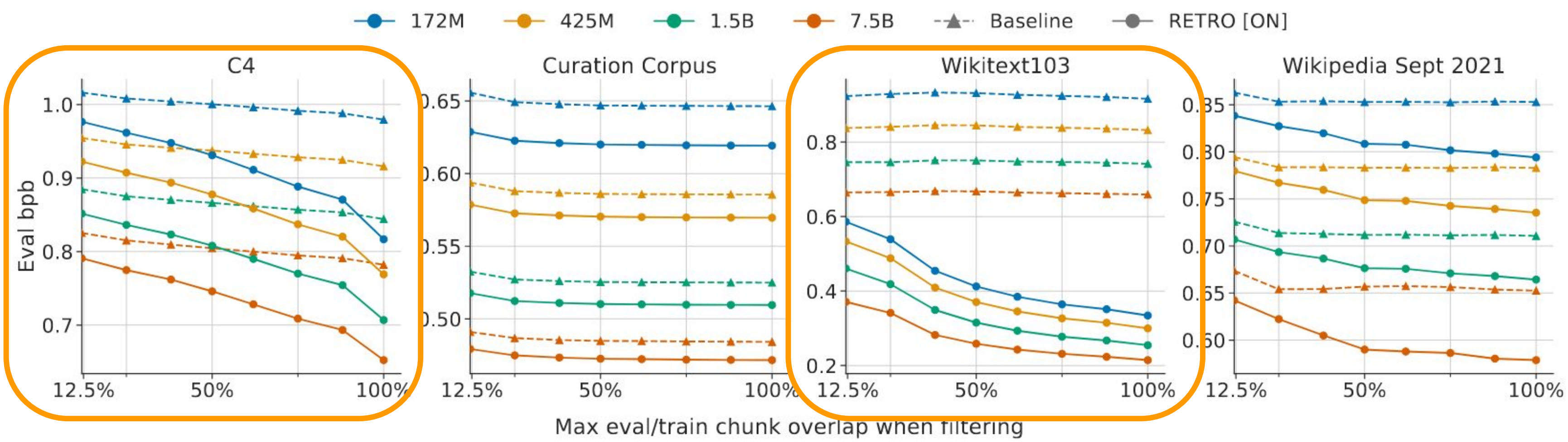
$$\mathrm{bpb}(\alpha) \triangleq \frac{\sum_{C \in C_\alpha} \ell(C)}{\sum_{C \in C_\alpha} N(C)}$$



Performance wrt. Dataset Leakage

- Filtered bpb as eval loss

$$\text{bpb}(\alpha) \triangleq \frac{\sum_{C \in C_\alpha} \ell(C)}{\sum_{C \in C_\alpha} N(C)}$$

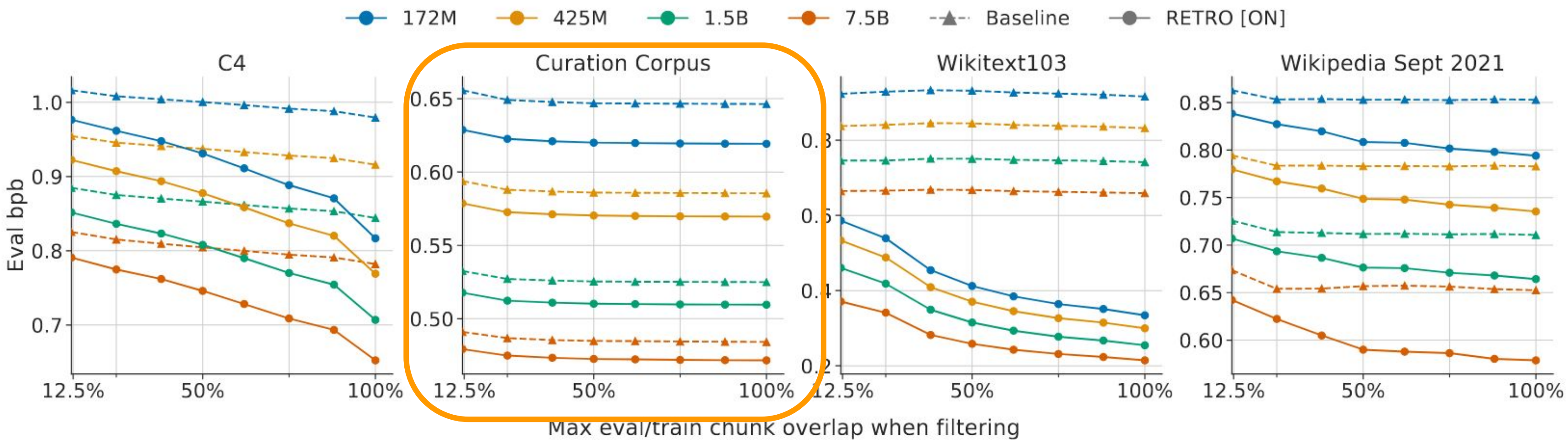


More details in table 19

Performance wrt. Dataset Leakage

- Filtered bpb as eval loss

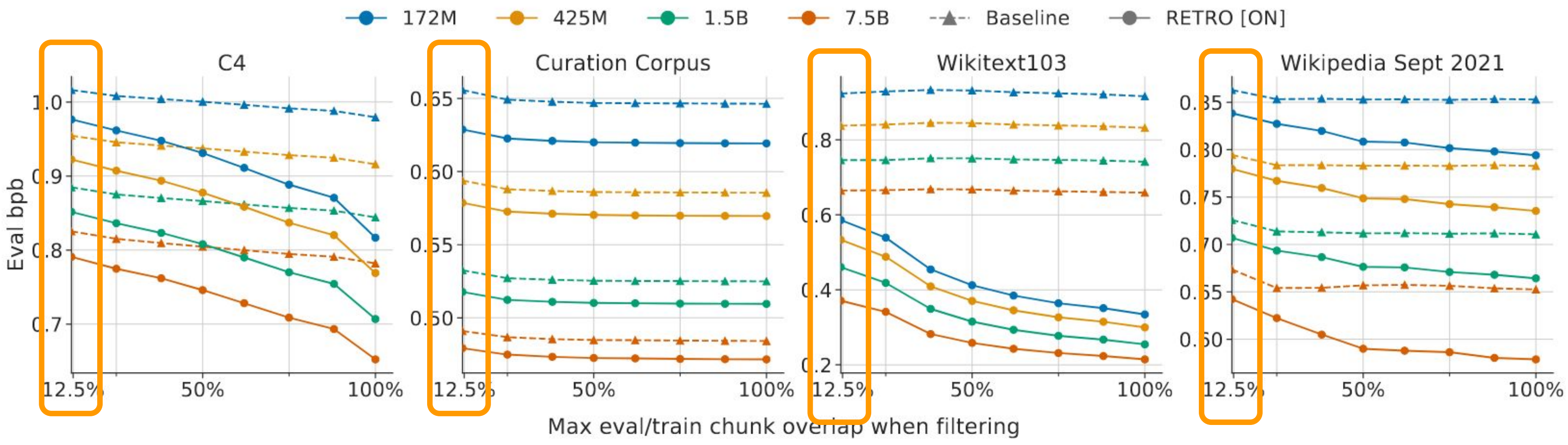
$$\mathrm{bpb}(\alpha) \triangleq \frac{\sum_{C \in C_\alpha} \ell(C)}{\sum_{C \in C_\alpha} N(C)}$$



Performance wrt. Dataset Leakage

- Filtered bpb as eval loss

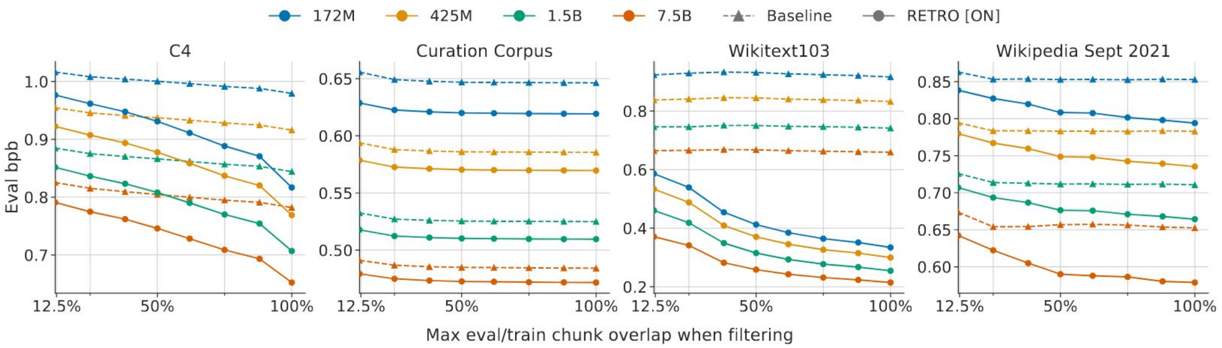
$$\text{bpb}(\alpha) \triangleq \frac{\sum_{C \in C_\alpha} \ell(C)}{\sum_{C \in C_\alpha} N(C)}$$



More details in table 19 and Fig. 12

Q2. Describe how RETRO defines dataset leakage. Do retrieval-based models like RETRO actually exploit evaluation dataset leakage or not?

- Filtered bits-per-bytes.
- Yes, retrieval-based models like RETRO do exploit evaluation dataset leakage, as indicated in the figure below.



Sampling Results: “Beavers are interesting animals”

Prompt and sample of RETRO[OFF]	Prompt and sample of RETRO[ON] colored by LCP with RET(C_u−1) LCP = 0, 1, 2, 3,4,⩾ 5	$[N_u^1, F_u^1]$ colored by LCP with C_{u+1} LCP = 0, 1, 2, 3,4,⩾ 5	$[N_u^2, F_u^2]$ colored by LCP with C_{u+1} LCP = 0, 1, 2, 3,4,⩾ 5
Beavers are interesting animals that live near rivers. They build	Beavers are interesting animals that live near rivers. They build	.Beavers build their lodges in pon ds they have created in wooded areas .Like many things in nature, there is a connection between creatures i n the wild.Beaver ponds cause tree s to drown, but the dead trees attra ct the great blue heron, which often return year after year. Over time, a beaver pond can attract more than 50 nests in a colony, called a rooke ry.An example of this can be found in the large pond off Bradford Road at Carter Fields near the Boxford l ine.Chris Leahy, an expert with th e Massachusetts Audubon Society who wrote	naw them into smaller sections and d rag them into the water.Engineers of the Pond Beavers are interesting animals because they change the hab itat in which they live. Beavers do this by blocking up streams to creat e ponds. Then they build their homes , called lodges, in these ponds. Bea vers’ bodies make them well-suited f or underwater building Special muscl es close off their noses, ears, and throats to keep the water out. Beave rs’ broad tails act like rudders for steering. Their two very large, ora nge front teeth are used to gnaw dow n trees. They begin building their d am
dams to create ponds. Frogs are am phibians, so they can live in both l and and water. They have great camou flage to hide from predators. The G olden Retriever, or Golden, is a ver y big dog. They are very strong, and have a thick coat to help them live in	their houses called beaver dams in the riverbeds. They also live on lan d.Beavers use their strong teeth an d strong jaws to cut down trees and branches to build their homes. They also use their teeth and jaws to che w up food. Beavers use their big, fl at tails to swim. They use	, then they mean that you are very b usy. Beavers swim easily in streams, picking up rocks and sticks to buil d their dams. They gnaw at trees wit h their big front teeth to cut them down. Then they use parts of the tre es to build their houses.Beavers ar e clever builders. They know exactly what they need to build their beave r dams. They use mud from the stream to make their dams stay together. T hey use their tails to pat down the mud.Beavers put a snug room at the top of their dams for their babies. They store their food underwater. Be avers eat the bark from the	ar-like tail, and two protruding tee th that are strong enough to gnaw do wn trees. The beaver uses trees, bra nches, and mud to build dams across rivers and streams. These dams creat e a deep pool of water in which the beaver builds its home. A beaver hom e is called a lodge. A baby beaver o r “kit” remains in the family lodge until the age of two.Beaver fur, kn own as pelt, was once highly popular as a trim for hats and coats. How m ight the popularity of beaver fur co ntributed to the colonization of New York?wwwWw

Thanks for listening!

Q3: Do you think that retrieval-based LMs can work similarly as standard dense LLMs in terms of downstream applications (e.g., prompting, fine-tuning)?

What are key challenges of scaling up retrieval-based LMs?