Retrieval-Augmented Language Generation (RAG)

Spring 2024

CS6501: Natural Language Processing

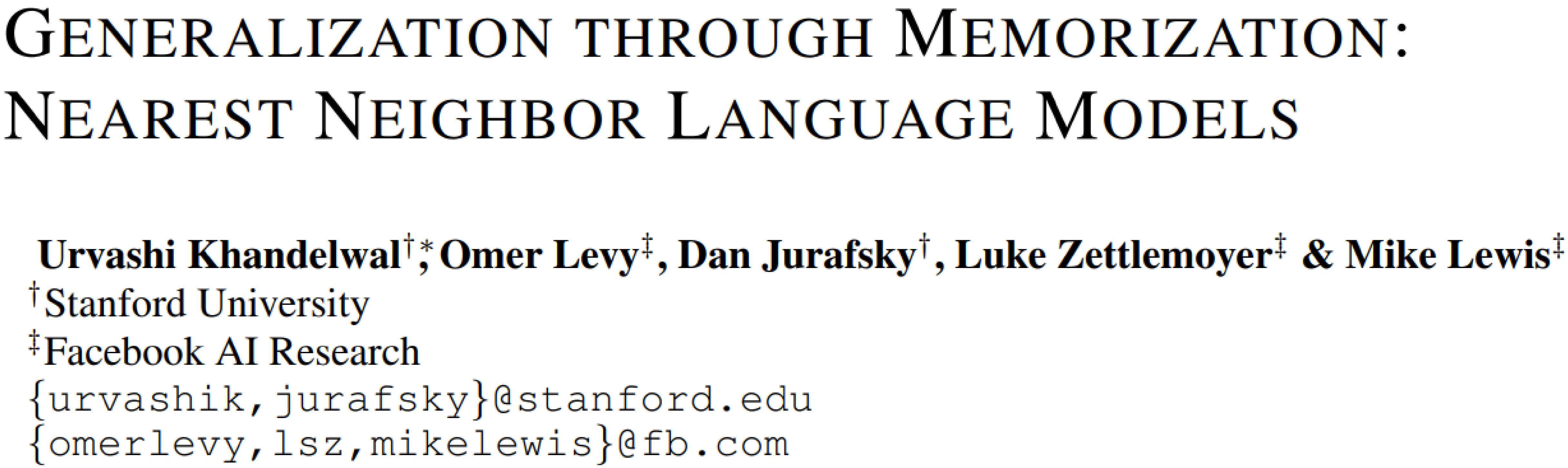
JiHo Lee
Department of Computer Science
University of Virginia
Charlottesville, VA
jiholee@virginia.edu

Jacob Christopher
Department of Computer Science
University of Virginia
Charlottesville, VA
csk4sr@virginia.edu

Yanson Khuu
Department of Computer Science
University of Virginia
Charlottesville, VA
yansonkhuu@virginia.edu

Agenda

- Generalization through memorization: nearest neighbor language models
- Retrieval-augmented generation for knowledge-intensive NLP Tasks
- Dense passage retrieval for open-domain question answering
- Improving language models by retrieving from trillions of tokens



GENERALIZATION THROUGH MEMORIZATION: NEAREST NEIGHBOR LANGUAGE MODELS

Urvashi Khandelwal[†,*] **Omer Levy**[‡], **Dan Jurafsky**[†], **Luke Zettlemoyer**[‡] **& Mike Lewis**[‡]

[†]Stanford University

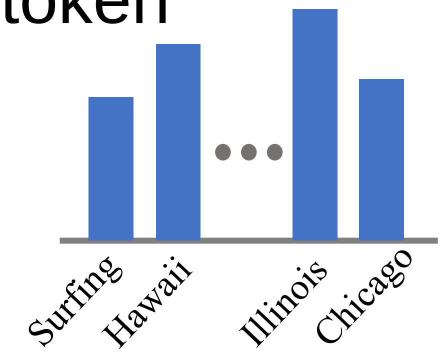
[‡]Facebook AI Research

{urvashik,jurafsky}@stanford.edu

{omerlevy,lsz,mikelewis}@fb.com

Neural Autoregressive Language Models

Given prior text, estimate the probability for the target token



$C_t = (w_1, \ldots, w_{t-1}) \longrightarrow$

Obama was born in

$f(c_t)$

Neural Language Model

$\longrightarrow P(w_t | c_t)$

Illinois (0.5)

Chicago (0.25)

Hawaii (0.1)

Congress (0.02)

Surfing (0.000009)

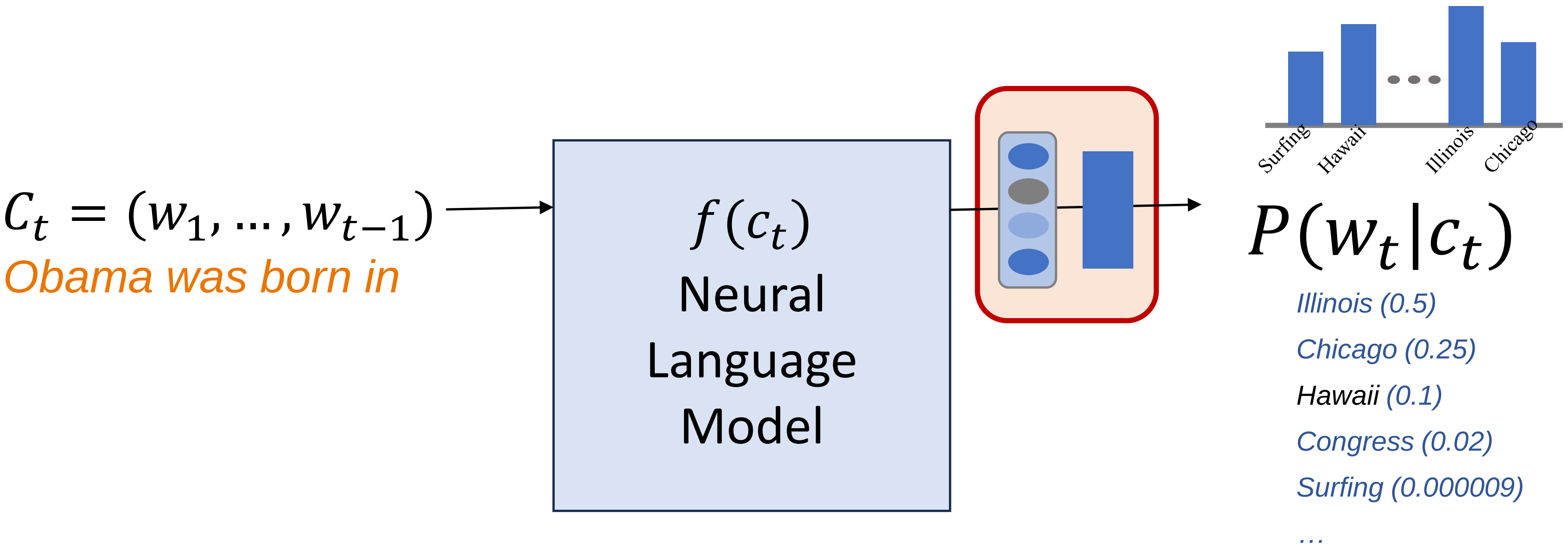
…

Language Models

- Lots of text is very easily available, so we train models on large amounts of data.

- But improving LM performance or scaling to larger datasets, by training bigger and bigger models with billions of parameters, requires massive amounts of GPU compute.

Language Models



$c_t = (w_1, \ldots, w_{t-1})$

Obama was born in

$f(c_t)$

Neural Language Model

$P(w_t | c_t)$

Illinois (0.5)

Chicago (0.25)

Hawaii (0.1)

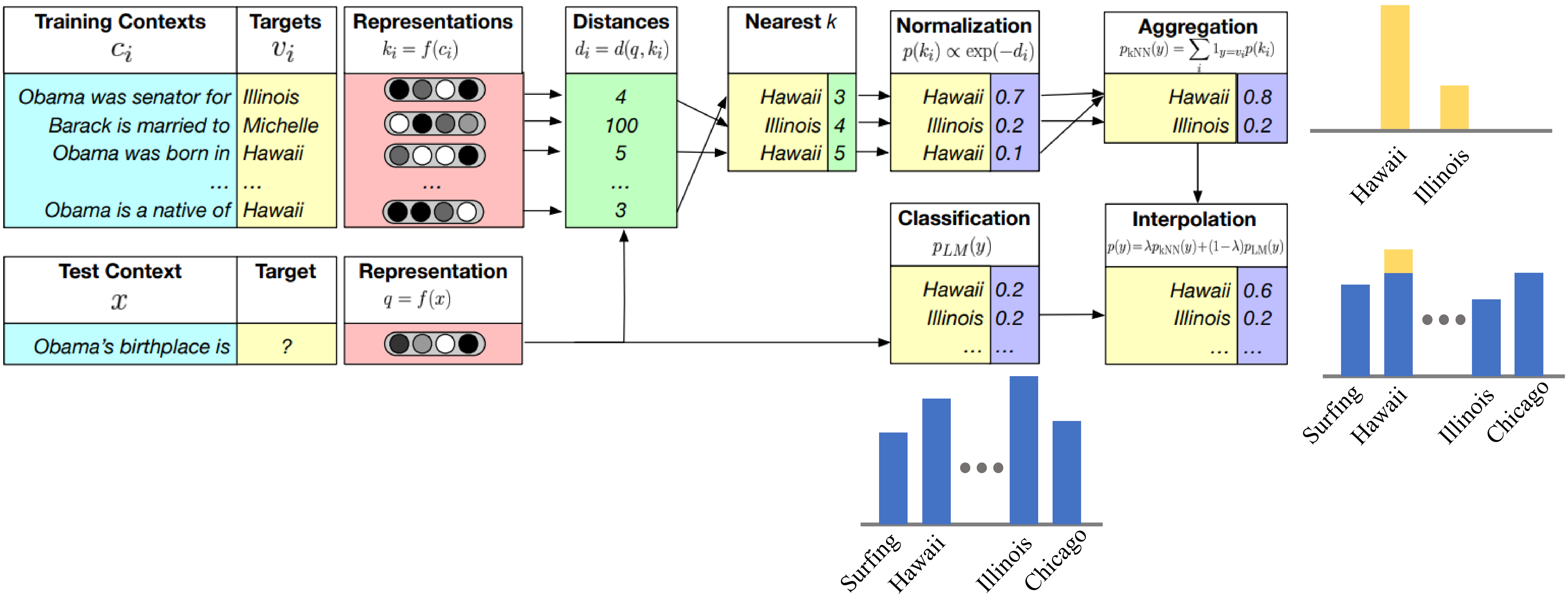
Congress (0.02)

Surfing (0.000009)

…

- Encoding + Prediction
- kNN-LM
  - The representation learning problem may be easier than the prediction problem.
  - Pre-trained LM with retrieval mechanism

Nearest Neighbor Language Models (KNN-LM)



Training Contexts c_i	Targets v_i	Representations $k_i = f(c_i)$	Distances $d_i = d(q, k_i)$
Obama was senator for	Illinois		4
Barack is married to	Michelle		100
Obama was born in	Hawaii		5
...	...	...	...
Obama is a native of	Hawaii		3

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

Nearest k	
Hawaii	3
Illinois	4
Hawaii	5

Normalization $p(k_i) \propto \exp(-d_i)$	
Hawaii	0.7
Illinois	0.2
Hawaii	0.1

Aggregation $p_{\text{kNN}}(y) = \sum_i \mathbb{1}_{y=v_i} p(k_i)$	
Hawaii	0.8
Illinois	0.2

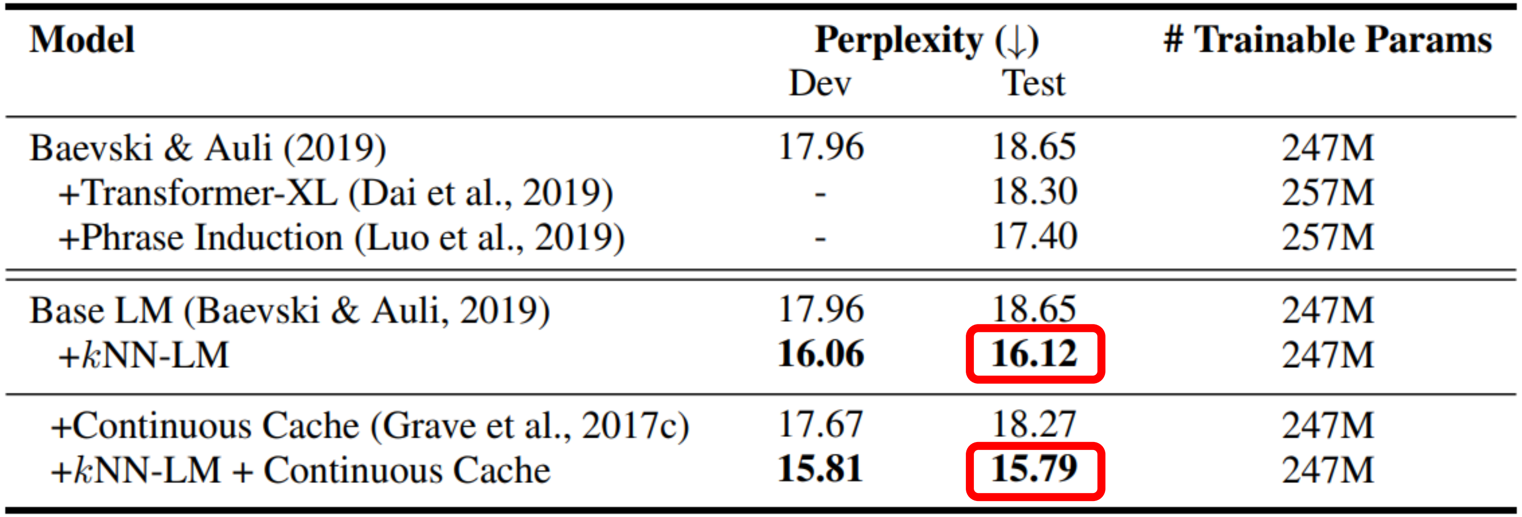
Classification $p_{LM}(y)$	
Hawaii	0.2
Illinois	0.2
...	...

Interpolation $p(y) = \lambda p_{\text{kNN}}(y) + (1-\lambda) p_{\text{LM}}(y)$	
Hawaii	0.6
Illinois	0.2
...	...

Key Results

- Explicitly memorizing the training data helps generalization.
- LMs can scale to larger text collections without the added cost of training.
- A single LM can adapt to multiple domains without any in-domain training.

Experiments



Model	Perplexity (↓)		# Trainable Params
	Dev	Test	
Baevski & Auli (2019)	17.96	18.65	247M
+Transformer-XL (Dai et al., 2019)	-	18.30	257M
+Phrase Induction (Luo et al., 2019)	-	17.40	257M
Base LM (Baevski & Auli, 2019)	17.96	18.65	247M
+*k*NN-LM	**16.06**	**16.12**	247M
+Continuous Cache (Grave et al., 2017c)	17.67	18.27	247M
+*k*NN-LM + Continuous Cache	**15.81**	**15.79**	247M

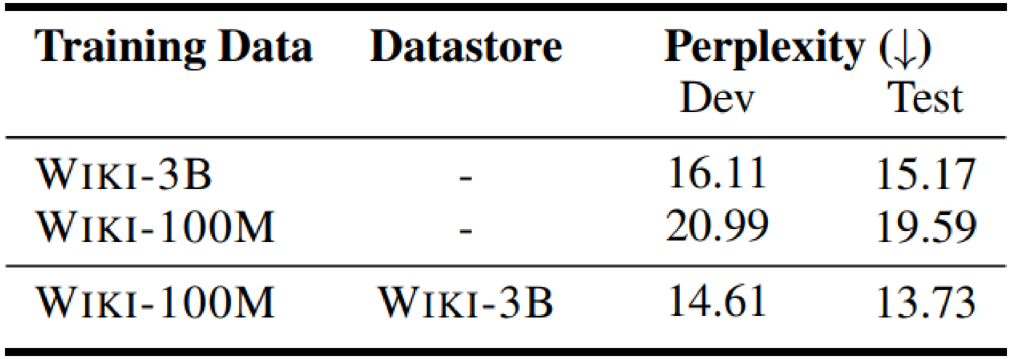
- kNN-LM is compatible with any model that produces fixed size context representations.
- kNN-LM improves perplexity on WIKITEXT-103 from 18.65 to 16.12.
- Improvements from the continuous cache are additive with the kNN-LM.

Key Results

- Explicitly memorizing the training data helps generalization.
- LMs can scale to larger text collections without the added cost of training.
- A single LM can adapt to multiple domains without any in-domain training.

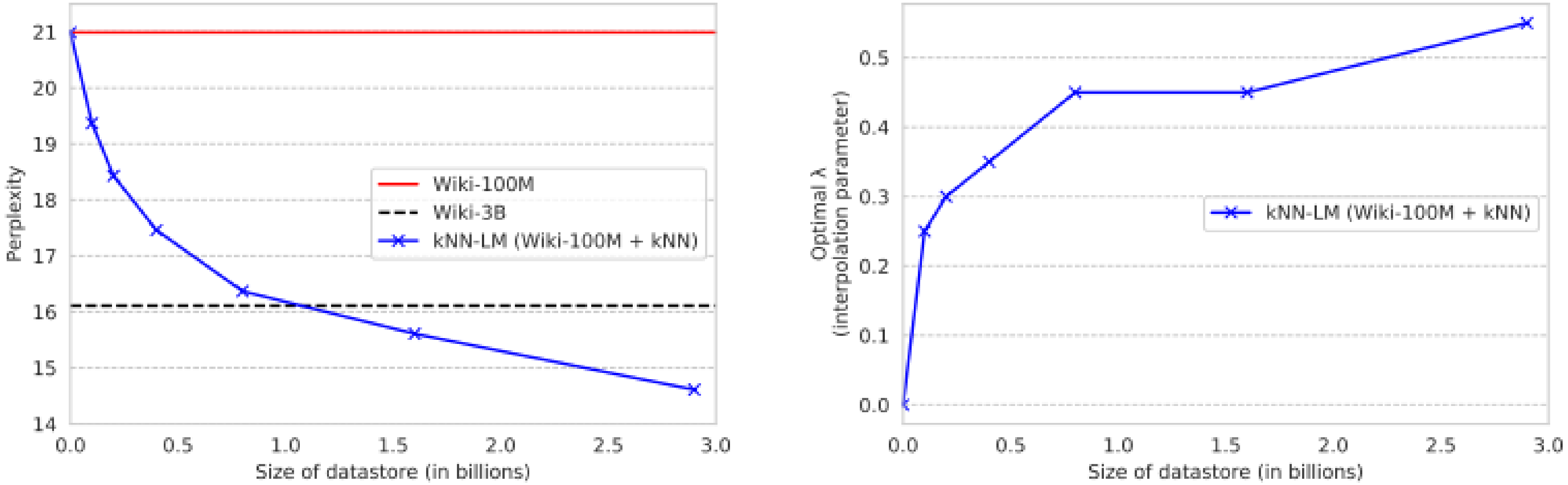
Scaling up from Wiki-100M to Wiki-3B

- Retrieving nearest neighbors from the corpus outperforms training on it.



Training Data	Datastore	Perplexity (↓)	
		Dev	Test
WIKI-3B	-	16.11	15.17
WIKI-100M	-	20.99	19.59
WIKI-100M	WIKI-3B	14.61	13.73

Datastore Size and Interpolation



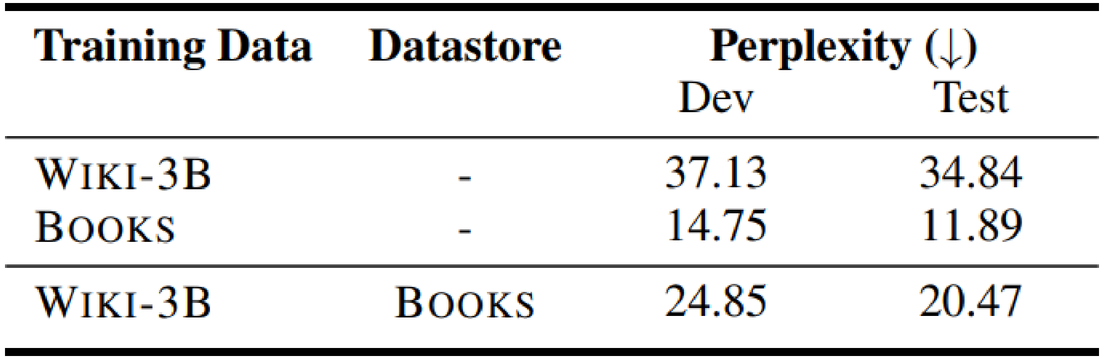
- As the datastore expands, the vocab distribution produced by the retriever becomes significantly effective.

Key Results

- Explicitly memorizing the training data helps generalization.
- LMs can scale to larger text collections without the added cost of training.
- A single LM can adapt to multiple domains without any in-domain training.

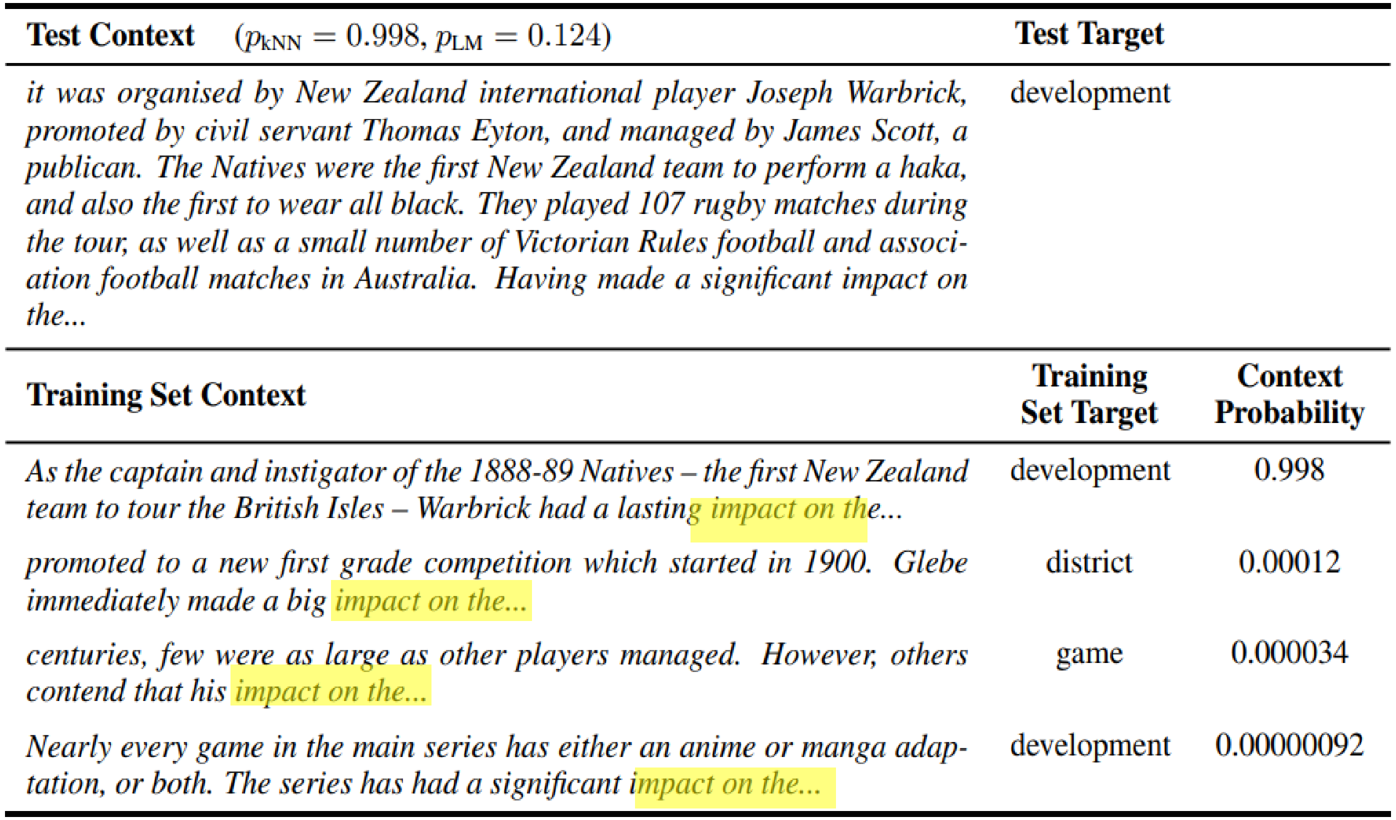
Domain Adaptation from Wiki to Books

- A single LM can be useful in multiple domains by simply adding a domain-specific datastore.



Training Data	Datastore	Perplexity (↓)	
		Dev	Test
WIKI-3B	-	37.13	34.84
BOOKS	-	14.75	11.89
WIKI-3B	BOOKS	24.85	20.47

Qualitative Analysis



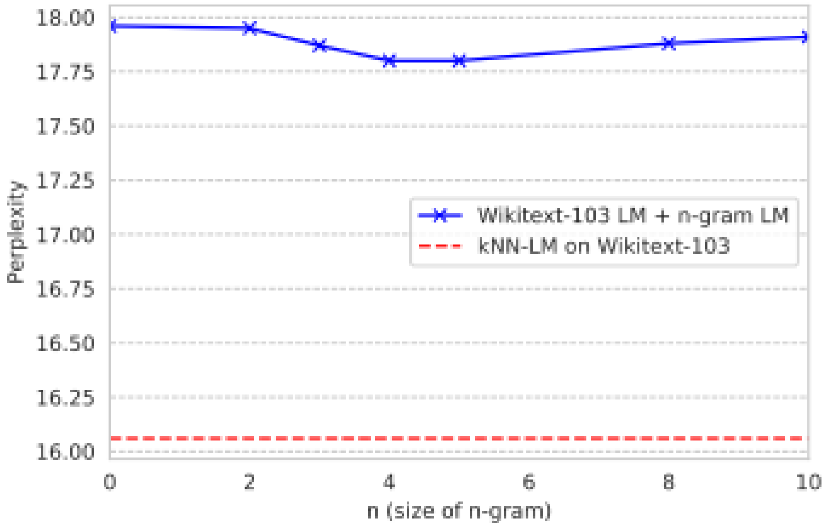
Test Context $(p_{\mathrm{kNN}} = 0.998, p_{\mathrm{LM}} = 0.124)$	**Test Target**
it was organised by New Zealand international player Joseph Warbrick, promoted by civil servant Thomas Eyton, and managed by James Scott, a publican. The Natives were the first New Zealand team to perform a haka, and also the first to wear all black. They played 107 rugby matches during the tour, as well as a small number of Victorian Rules football and association football matches in Australia. Having made a significant impact on the...	development

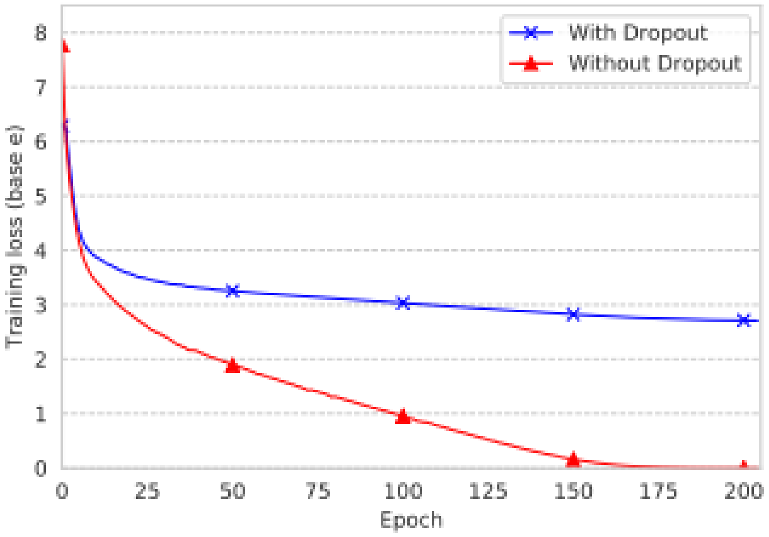
Training Set Context	**Training Set Target**	**Context Probability**
As the captain and instigator of the 1888-89 Natives – the first New Zealand team to tour the British Isles – Warbrick had a lasting impact on the...	development	0.998
promoted to a new first grade competition which started in 1900. Glebe immediately made a big impact on the...	district	0.00012
centuries, few were as large as other players managed. However, others contend that his impact on the...	game	0.000034
Nearly every game in the main series has either an anime or manga adaptation, or both. The series has had a significant impact on the...	development	0.00000092

- *k*NN model has much higher confidence in the correct target than the LM.

- The nearest neighbor search is highly confident of specific and very relevant context.

Memorization vs Retrieval





- LM can implicitly memorize the training data with less effective generalization.
- *k*NN-LM memorizes the training data while retaining an effective generalization for the validation perplexity.

Conclusion

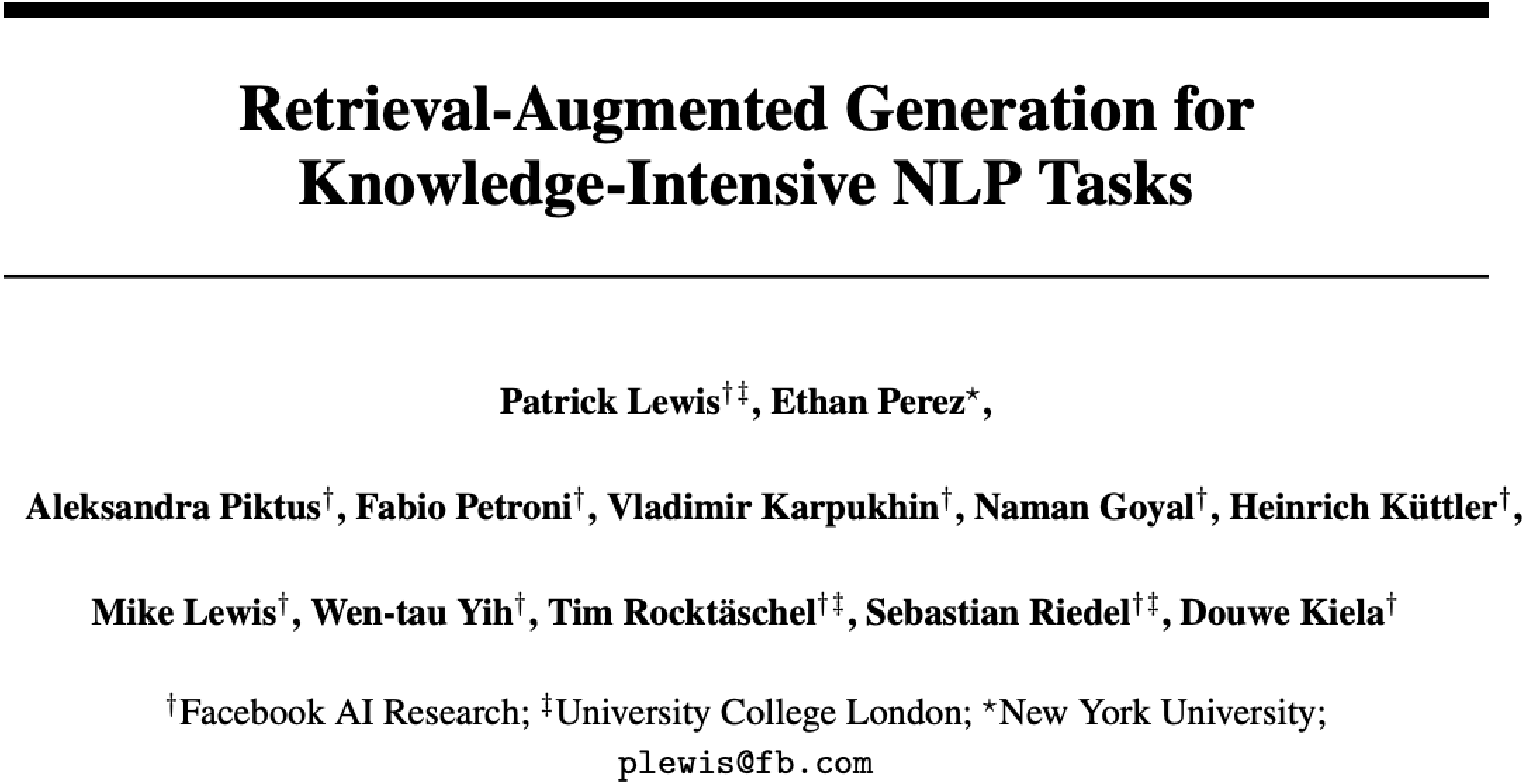
- Summary & Pros
  - Significantly outperform standard language models by directly querying training examples at test time.
  - The approach can be applied to any neural language model.

- Cons
  - Additional cost for constructing datastore.
  - Memory and computational costs to retrieve targets from datastore.
  - Perplexity is the only evaluation metric for the comparison.

Future work

- Learning similarity functions between contexts may be an easier problem than predicting the next word from some given context.

- Future work should explore explicitly training similarity functions, and reducing the size of the datastore.



Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

Patrick Lewis[†‡], Ethan Perez[⋆],

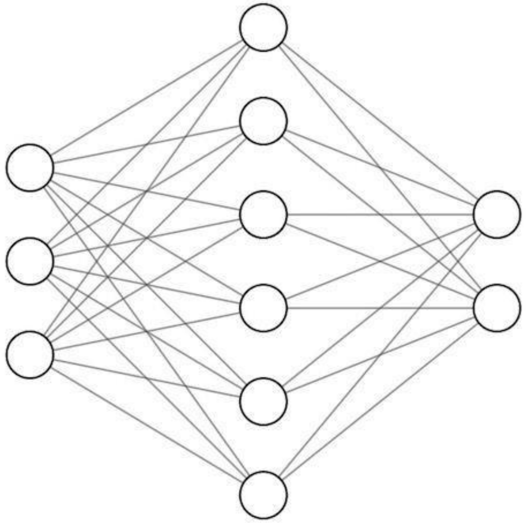
Aleksandra Piktus[†], Fabio Petroni[†], Vladimir Karpukhin[†], Naman Goyal[†], Heinrich Küttler[†],

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

[†]Facebook AI Research; [‡]University College London; [⋆]New York University;
`plewis@fb.com`

Parametric vs Non-Parametric Memory

- As shown in the previous paper, pre-trained have some capability in exercising recall of factual information contained in the trained parameters for ***general knowledge tasks***
- These are less effective for ***knowedge-intensive tasks***, those falling outside the domain of facts widely known
- Incorporating both parametric memory and non-parametric memory into pre-training (avoid additional training)



Approach Overview

- Retriever Model: $p_\eta(z|x)$
  - A model with parameters η retrieving over passages z using query x

- Generator Model: $p_\theta(y_i|x, z, y_{1:i-1})$
  - A model with parameters θ that generates token i based on context of the query, previously generated tokens, and retrieved passage

- End-to-End Training
  - These models are trained in an end-to-end manner (collectively)

Approach Overview

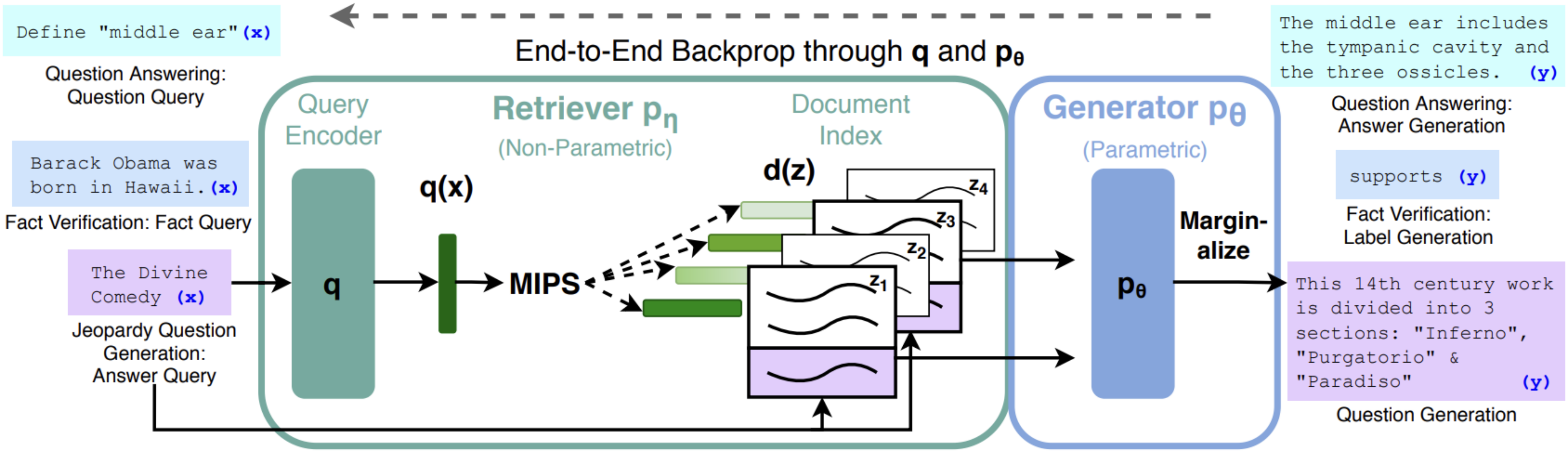

Define "middle ear" (x)
Question Answering: Question Query
Barack Obama was born in Hawaii. (x)
Fact Verification: Fact Query
The Divine Comedy (x)
Jeopardy Question Generation: Answer Query
End-to-End Backprop through q and p_θ
Query Encoder
q
q(x)
Retriever p_η
(Non-Parametric)
MIPS
Document Index
d(z)
z_4
z_3
z_2
z_1
Generator p_θ
(Parametric)
p_θ
Margin-alize
The middle ear includes the tympanic cavity and the three ossicles. (y)
Question Answering: Answer Generation
supports (y)
Fact Verification: Label Generation
This 14th century work is divided into 3 sections: "Inferno", "Purgatorio" & "Paradiso" (y)
Question Generation

Figure 1: Overview of our approach. We combine a pre-trained retriever (*Query Encoder* + *Document Index*) with a pre-trained seq2seq model (*Generator*) and fine-tune end-to-end. For query x, we use Maximum Inner Product Search (MIPS) to find the top-K documents z_i. For final prediction y, we treat z as a latent variable and marginalize over seq2seq predictions given different documents.

RAG-Sequence vs RAG-Token

- RAG-Sequence Model
  - Single retrieved document *z* for complete sequence

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

- RAG-Token Model
  - Retirever selects a series of documents and selects the "best" document to use as context for each output token

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

Dense Passage Retrieval

- The retriever model utilizes a bi-encoder architecture, such that

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

  BERT encodes the documents and the query to build the document index, compiling the non-parametric memory.

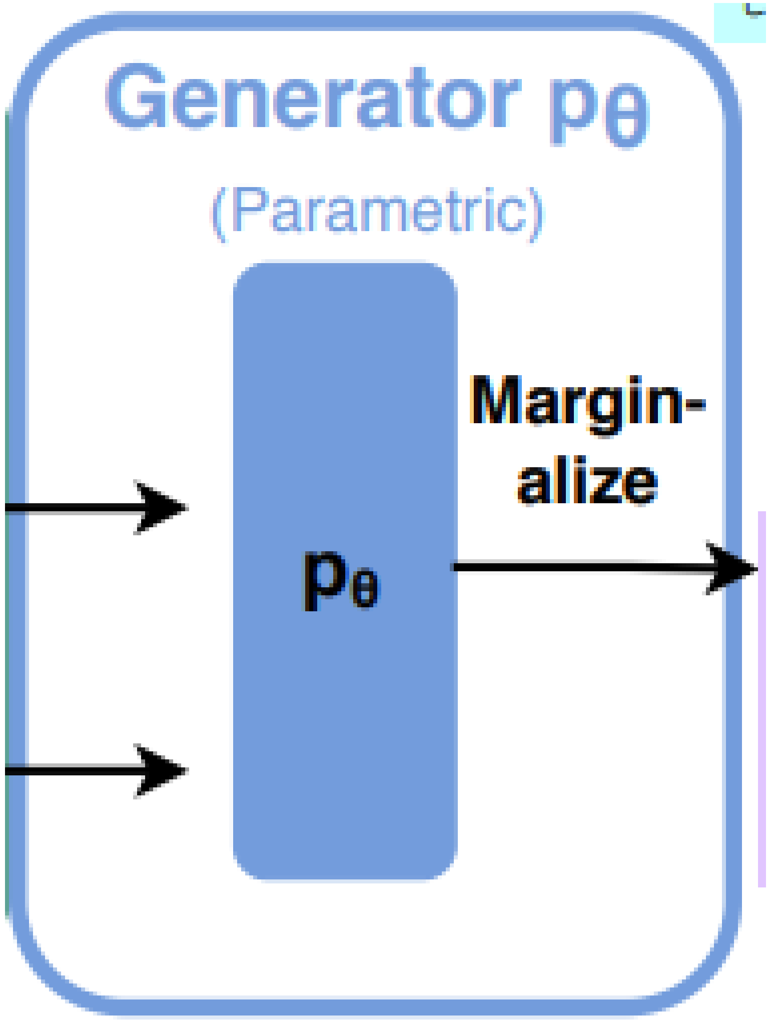
- This allows us to train the encoders to use a Maximum Inner Product Search (MIPS) to retrieve the most useful set of documents for the query.

Generator

- BART-large is used for this component

- Context becomes a concatenation of the documents *z*, the query *x*, and the previous token predictions $y_{1:i-1}$

- The model parameters *θ* act as the parameteric memory



Decoding

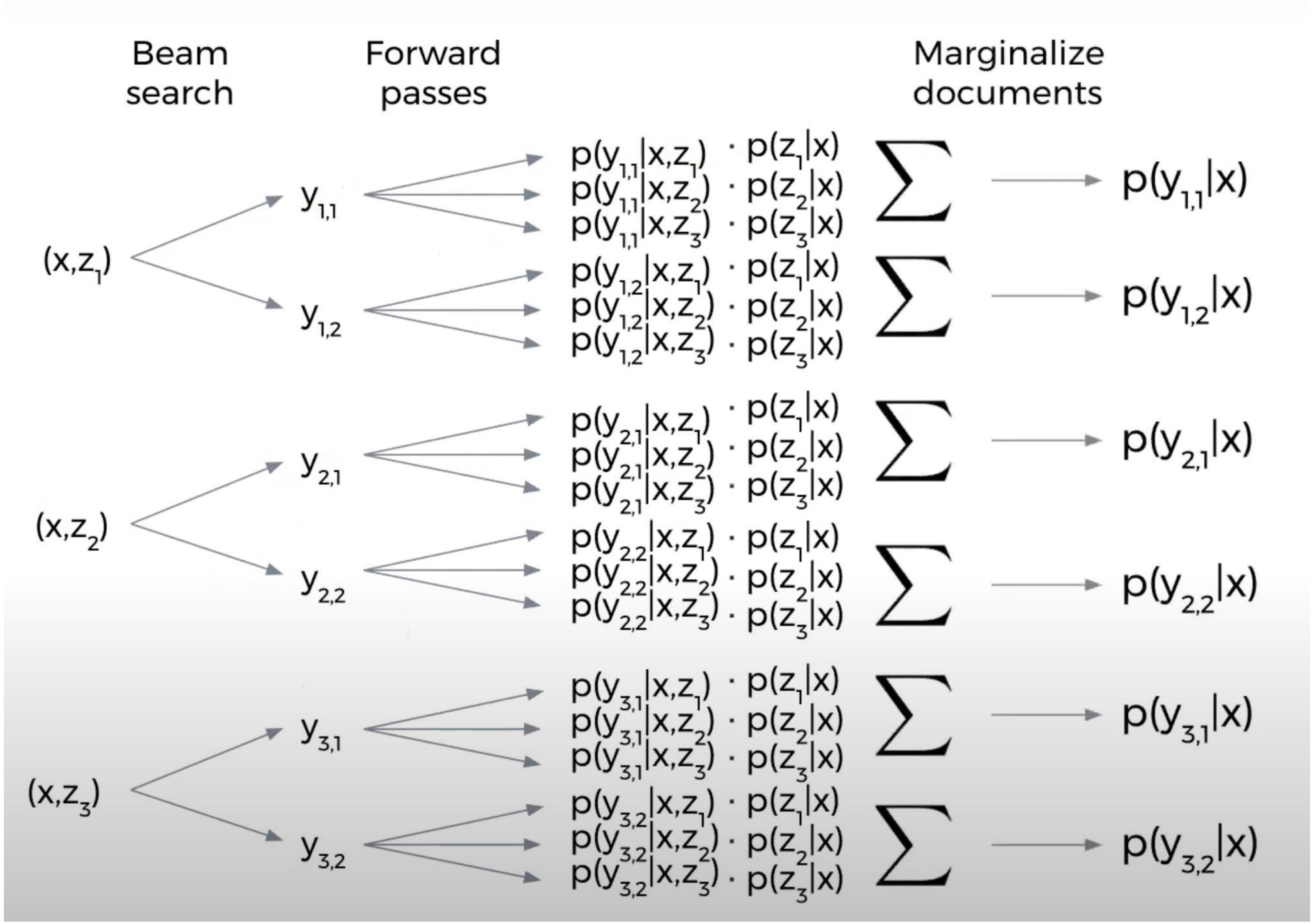
- **RAG-Token:**
  Standard beam decoder

- **RAG-Sequence:**
  Since there are multiple contexts being considered (multiple documents), beam search is applied to each, providing an independent score

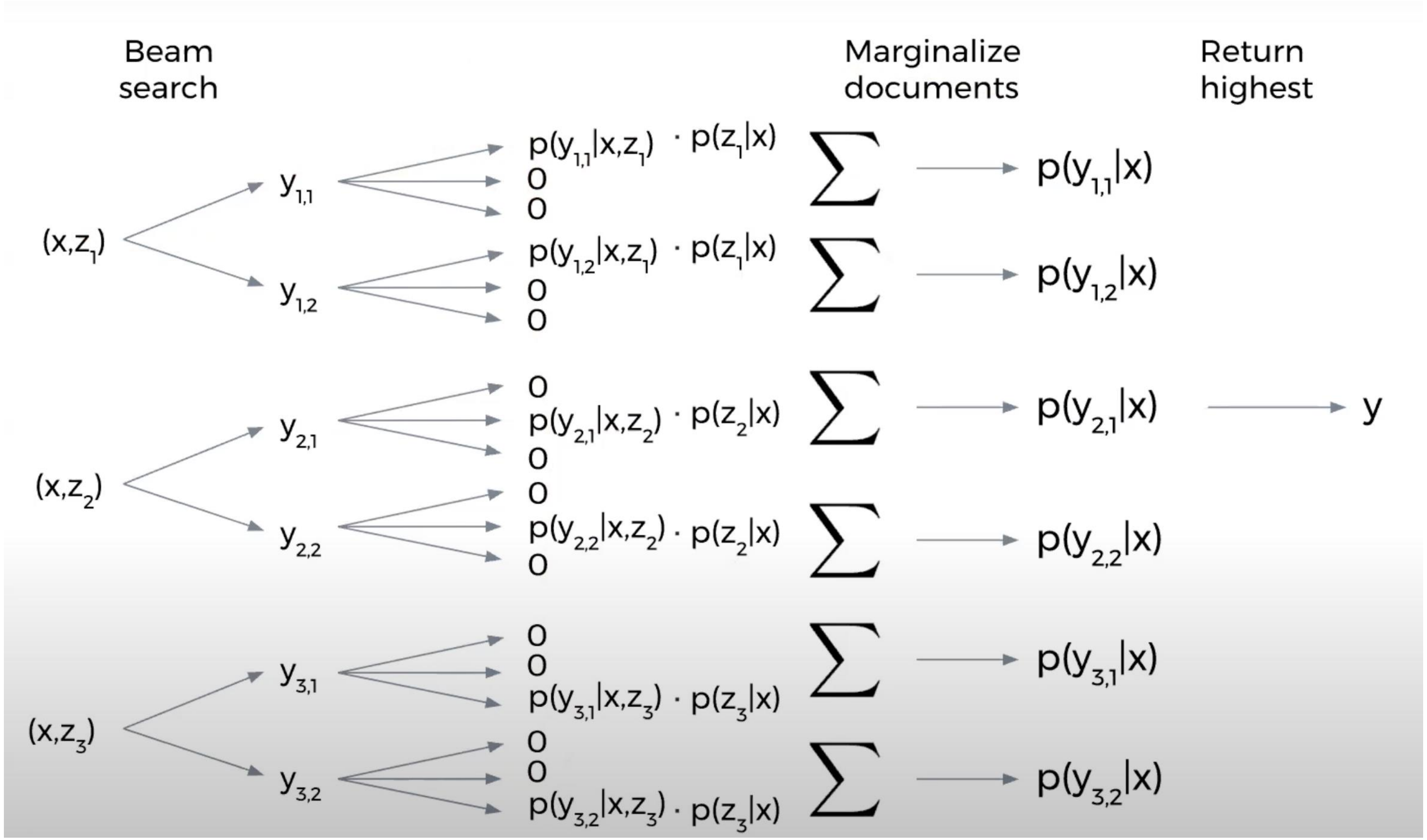
Decoding

- **RAG-Sequence:**
  - *Thorough Decoding:*
    - Each context beam has associated outputs y (hypothesis) which is scored and $y \in Y$
    - The probability for y is evaluated using an additional forward pass for documents which do not elicit y
    - The generator probability of this forward pass is then used to create a collective probability across beams
  - *Fast Decoding:*
    - As the former can be quite computationally expensive, the probability of y for contexts not containing it in the beam can be approximated as zero
    - Eliminates need for additional forward passes

Decoding

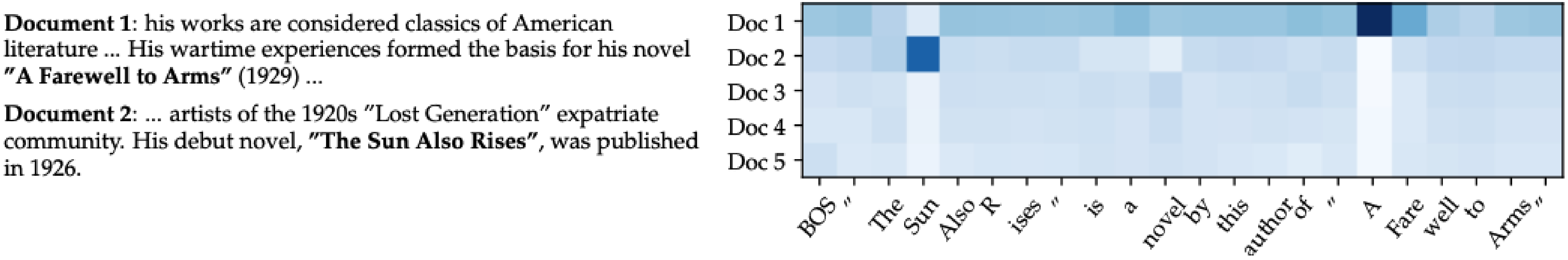

Beam search
Forward passes
Marginalize documents
(x,z_1)
$y_{1,1}$
$y_{1,2}$
$p(y_{1,1}|x,z_1) \cdot p(z_1|x)$
$p(y_{1,1}|x,z_2) \cdot p(z_2|x)$
$p(y_{1,1}|x,z_3) \cdot p(z_3|x)$
$\sum$
$p(y_{1,1}|x)$
$p(y_{1,2}|x,z_1) \cdot p(z_1|x)$
$p(y_{1,2}|x,z_2) \cdot p(z_2|x)$
$p(y_{1,2}|x,z_3) \cdot p(z_3|x)$
$\sum$
$p(y_{1,2}|x)$
(x,z_2)
$y_{2,1}$
$y_{2,2}$
$p(y_{2,1}|x,z_1) \cdot p(z_1|x)$
$p(y_{2,1}|x,z_2) \cdot p(z_2|x)$
$p(y_{2,1}|x,z_3) \cdot p(z_3|x)$
$\sum$
$p(y_{2,1}|x)$
$p(y_{2,2}|x,z_1) \cdot p(z_1|x)$
$p(y_{2,2}|x,z_2) \cdot p(z_2|x)$
$p(y_{2,2}|x,z_3) \cdot p(z_3|x)$
$\sum$
$p(y_{2,2}|x)$
(x,z_3)
$y_{3,1}$
$y_{3,2}$
$p(y_{3,1}|x,z_1) \cdot p(z_1|x)$
$p(y_{3,1}|x,z_2) \cdot p(z_2|x)$
$p(y_{3,1}|x,z_3) \cdot p(z_3|x)$
$\sum$
$p(y_{3,1}|x)$
$p(y_{3,2}|x,z_1) \cdot p(z_1|x)$
$p(y_{3,2}|x,z_2) \cdot p(z_2|x)$
$p(y_{3,2}|x,z_3) \cdot p(z_3|x)$
$\sum$
$p(y_{3,2}|x)$

Decoding


Beam search
Marginalize documents
Return highest
(x,z_1)
$y_{1,1}$
$y_{1,2}$
$p(y_{1,1}|x,z_1) \cdot p(z_1|x)$
0
0
$p(y_{1,2}|x,z_1) \cdot p(z_1|x)$
0
0
$\sum$
$p(y_{1,1}|x)$
$p(y_{1,2}|x)$
(x,z_2)
$y_{2,1}$
$y_{2,2}$
0
$p(y_{2,1}|x,z_2) \cdot p(z_2|x)$
0
0
$p(y_{2,2}|x,z_2) \cdot p(z_2|x)$
0
$p(y_{2,1}|x)$
y
$p(y_{2,2}|x)$
(x,z_3)
$y_{3,1}$
$y_{3,2}$
0
0
$p(y_{3,1}|x,z_3) \cdot p(z_3|x)$
0
0
$p(y_{3,2}|x,z_3) \cdot p(z_3|x)$
$p(y_{3,1}|x)$
$p(y_{3,2}|x)$

Decoding

The paper visualizes an example of this approach:



Document 1: his works are considered classics of American literature ... His wartime experiences formed the basis for his novel **"A Farewell to Arms"** (1929) ...

Document 2: ... artists of the 1920s "Lost Generation" expatriate community. His debut novel, **"The Sun Also Rises"**, was published in 1926.

Figure 2: RAG-Token document posterior $p(z_i|x, y_i, y_{-i})$ for each generated token for input "Hemingway" for Jeopardy generation with 5 retrieved documents. The posterior for document 1 is high when generating "A Farewell to Arms" and for document 2 when generating "The Sun Also Rises".

Example Test Cases

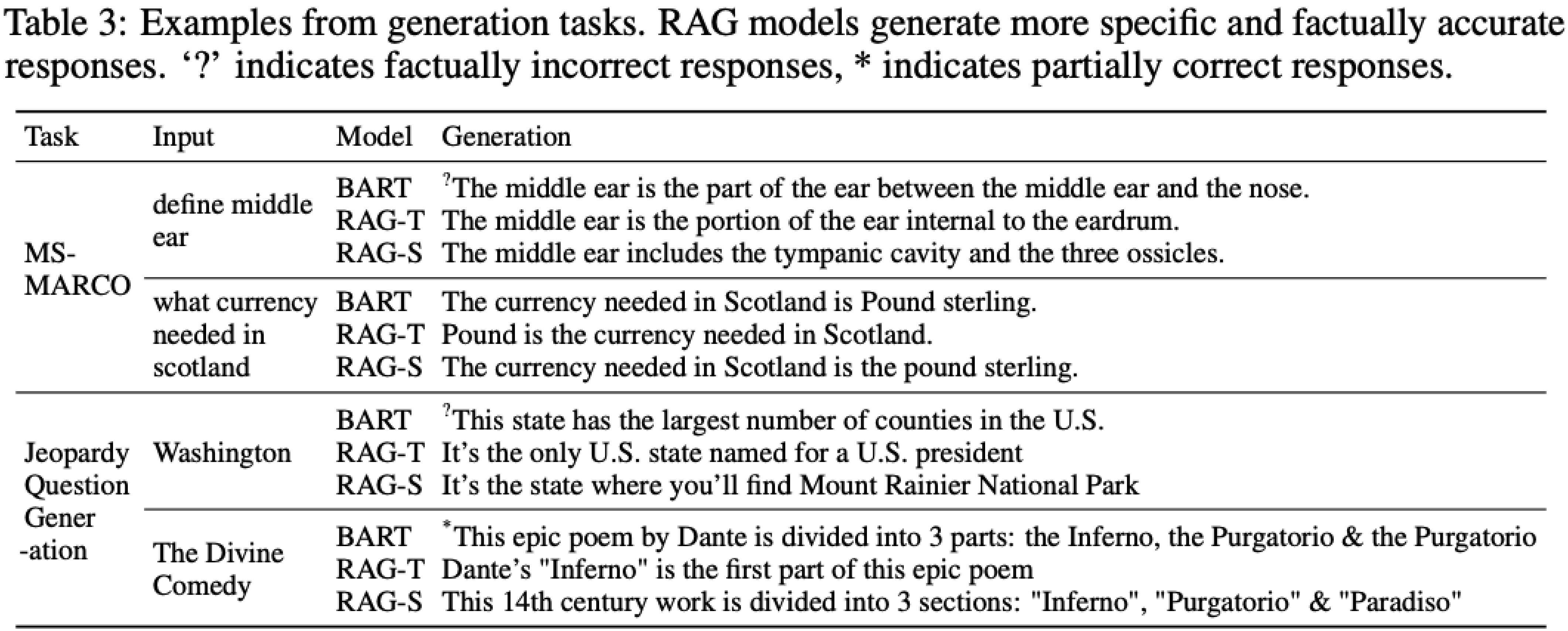


Table 3: Examples from generation tasks. RAG models generate more specific and factually accurate responses. ‘?’ indicates factually incorrect responses, * indicates partially correct responses.

Task	Input	Model	Generation
MS-MARCO	define middle ear	BART	$^{?}$The middle ear is the part of the ear between the middle ear and the nose.
		RAG-T	The middle ear is the portion of the ear internal to the eardrum.
		RAG-S	The middle ear includes the tympanic cavity and the three ossicles.
	what currency needed in scotland	BART	The currency needed in Scotland is Pound sterling.
		RAG-T	Pound is the currency needed in Scotland.
		RAG-S	The currency needed in Scotland is the pound sterling.
Jeopardy Question Gener -ation	Washington	BART	$^{?}$This state has the largest number of counties in the U.S.
		RAG-T	It’s the only U.S. state named for a U.S. president
		RAG-S	It’s the state where you’ll find Mount Rainier National Park
	The Divine Comedy	BART	*This epic poem by Dante is divided into 3 parts: the Inferno, the Purgatorio & the Purgatorio
		RAG-T	Dante’s "Inferno" is the first part of this epic poem
		RAG-S	This 14th century work is divided into 3 sections: "Inferno", "Purgatorio" & "Paradiso"

Results

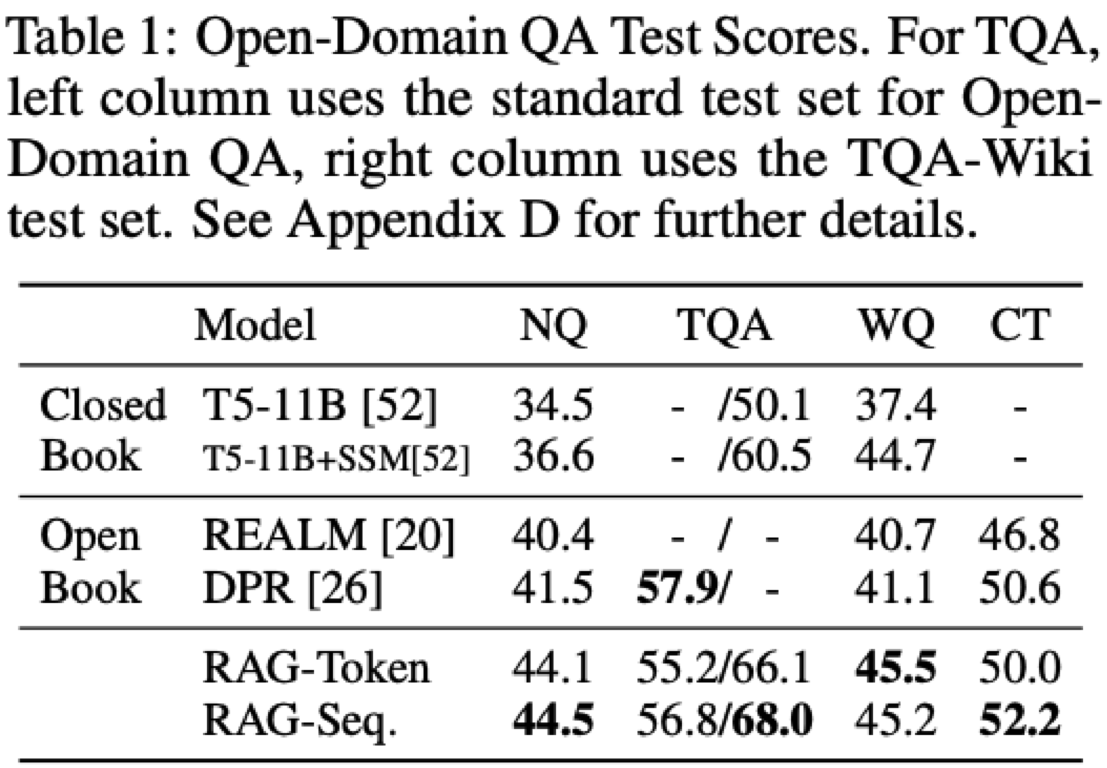


Table 1: Open-Domain QA Test Scores. For TQA, left column uses the standard test set for Open-Domain QA, right column uses the TQA-Wiki test set. See Appendix D for further details.

	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**

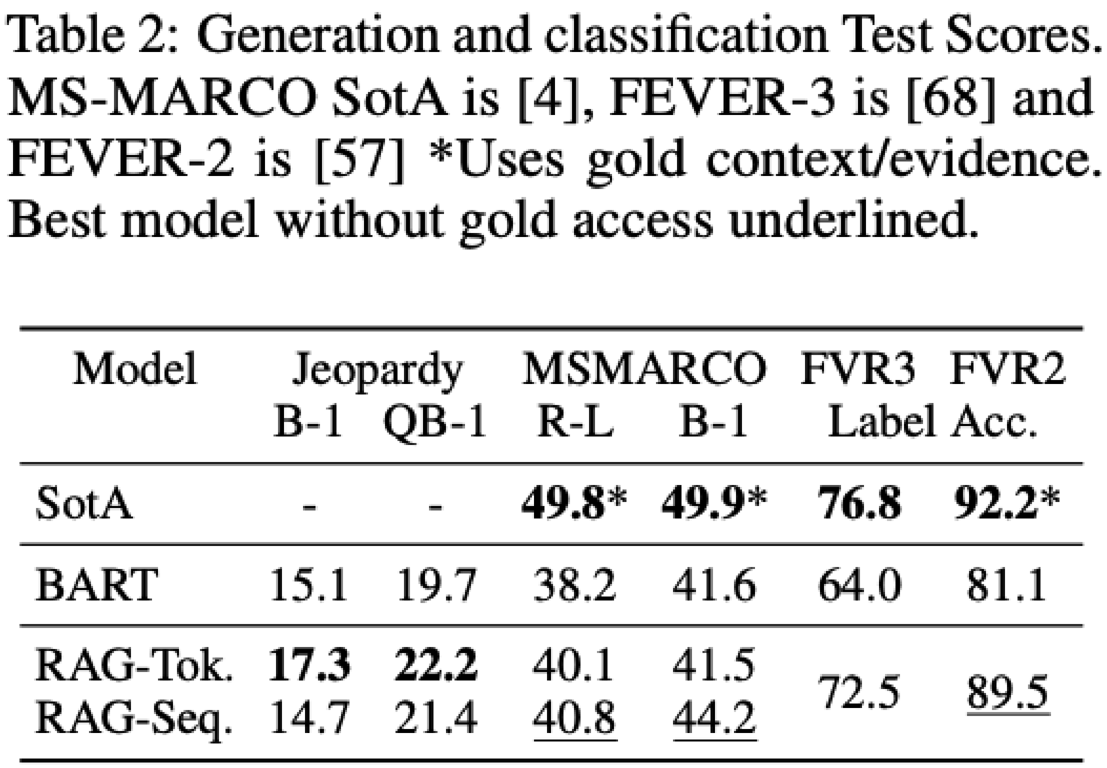


Table 2: Generation and classification Test Scores. MS-MARCO SotA is [4], FEVER-3 is [68] and FEVER-2 is [57] *Uses gold context/evidence. Best model without gold access underlined.

Model	Jeopardy		MSMARCO		FVR3	FVR2
	B-1	QB-1	R-L	B-1	Label Acc.	
SotA	-	-	**49.8***	**49.9***	**76.8**	**92.2***
BART	15.1	19.7	38.2	41.6	64.0	81.1
RAG-Tok.	**17.3**	**22.2**	40.1	41.5	72.5	89.5
RAG-Seq.	14.7	21.4	40.8	44.2		

Results

They evaluate based on human preference:

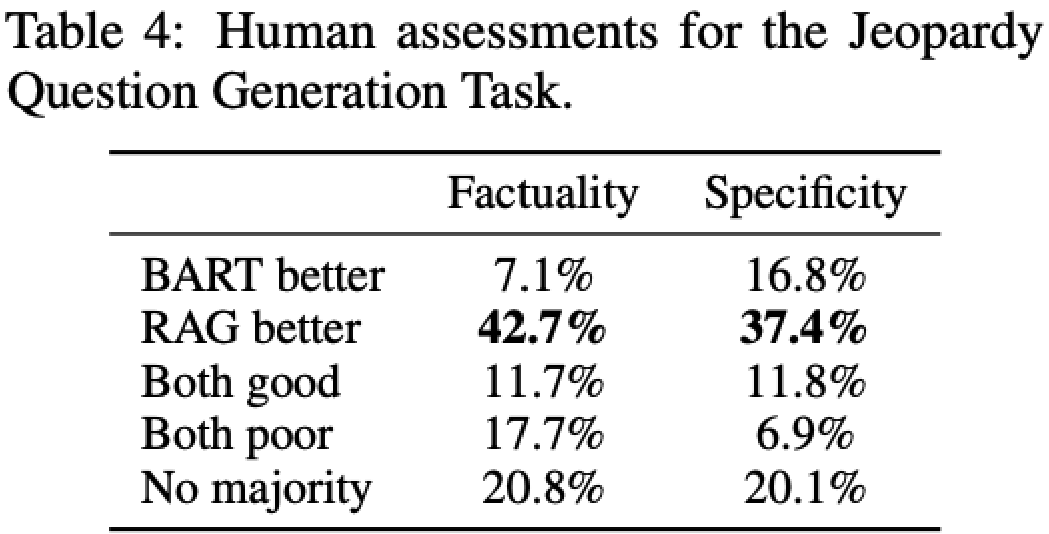


Table 4: Human assessments for the Jeopardy Question Generation Task.

	Factuality	Specificity
BART better	7.1%	16.8%
RAG better	**42.7%**	**37.4%**
Both good	11.7%	11.8%
Both poor	17.7%	6.9%
No majority	20.8%	20.1%

Results

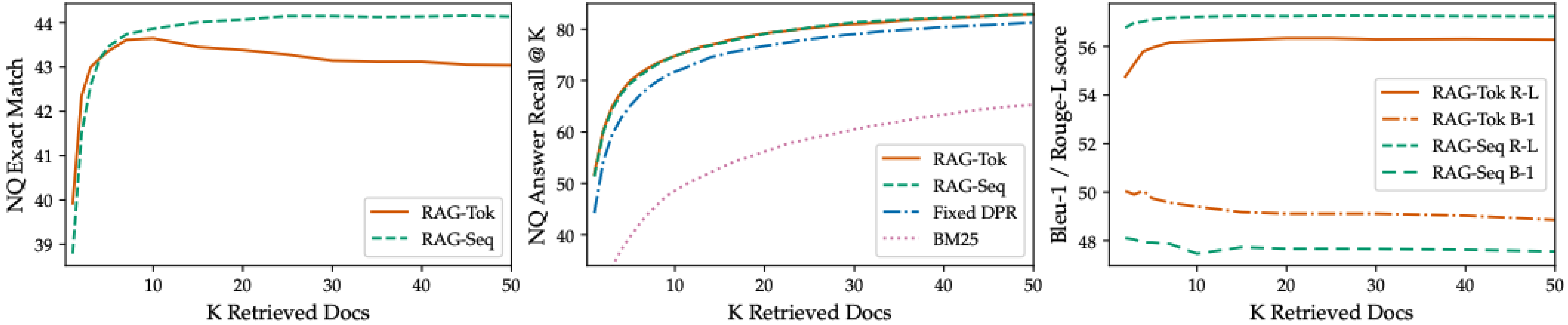


Figure 3: Left: NQ performance as more documents are retrieved. Center: Retrieval recall performance in NQ. Right: MS-MARCO Bleu-1 and Rouge-L as more documents are retrieved.

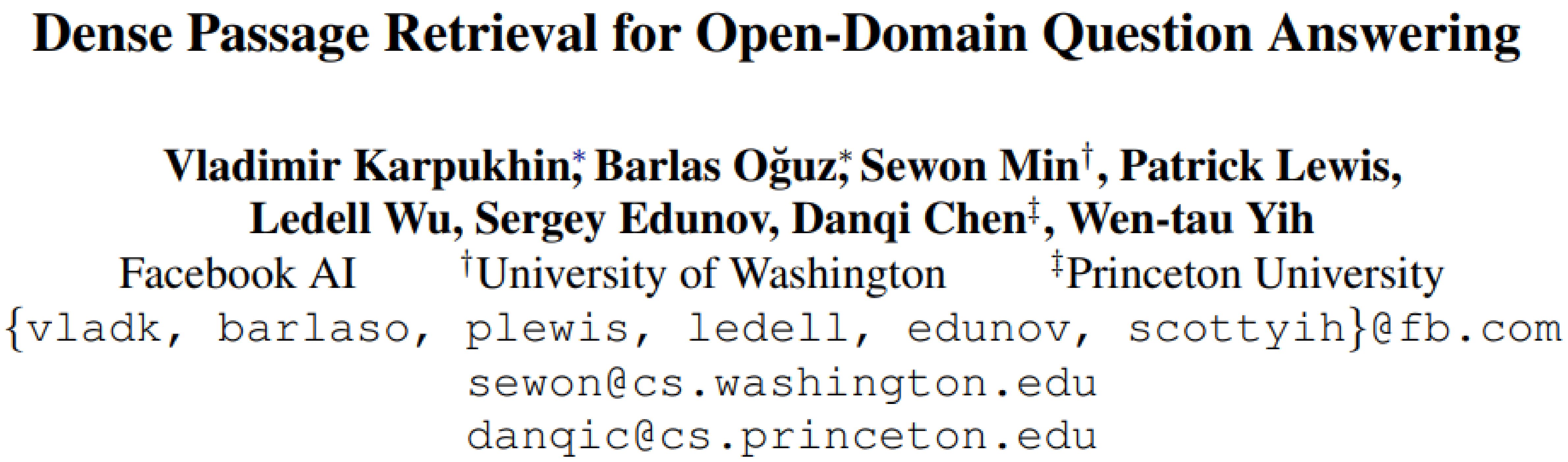
Limitations

- End-to-End Training:
  - Increased training cost
  - Cannot natively generalize to out-of-distribution problem sets
- Retrieval Collapse:
  - If the retriever does not provide any useful documents (when initialized), gradients become unstable and training fails
- Document sets may have factually incorrect information, so ans wers still should be assessed critically as method is adopted

Conclusion

- RAG methods can improve accuarcy by expanding to non-parametric memory

- Retrieval methods require no additional training to extend to various document sets



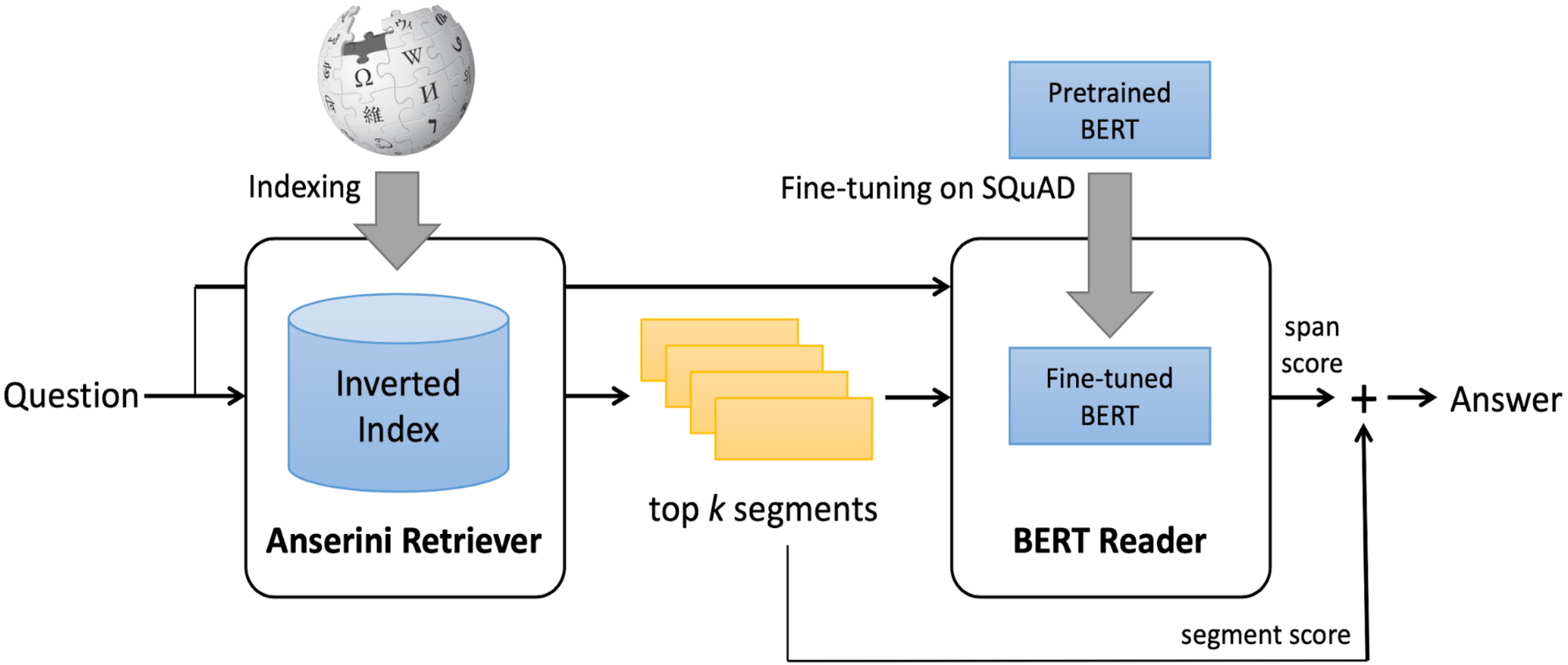
Dense Passage Retrieval for Open-Domain Question Answering

**Vladimir Karpukhin[*], Barlas Oğuz[*], Sewon Min[†], Patrick Lewis,
Ledell Wu, Sergey Edunov, Danqi Chen[‡], Wen-tau Yih**

Facebook AI [†]University of Washington [‡]Princeton University

{vladk, barlaso, plewis, ledell, edunov, scottyih}@fb.com
sewon@cs.washington.edu
danqic@cs.princeton.edu

What is Open-Domain Question Answering?



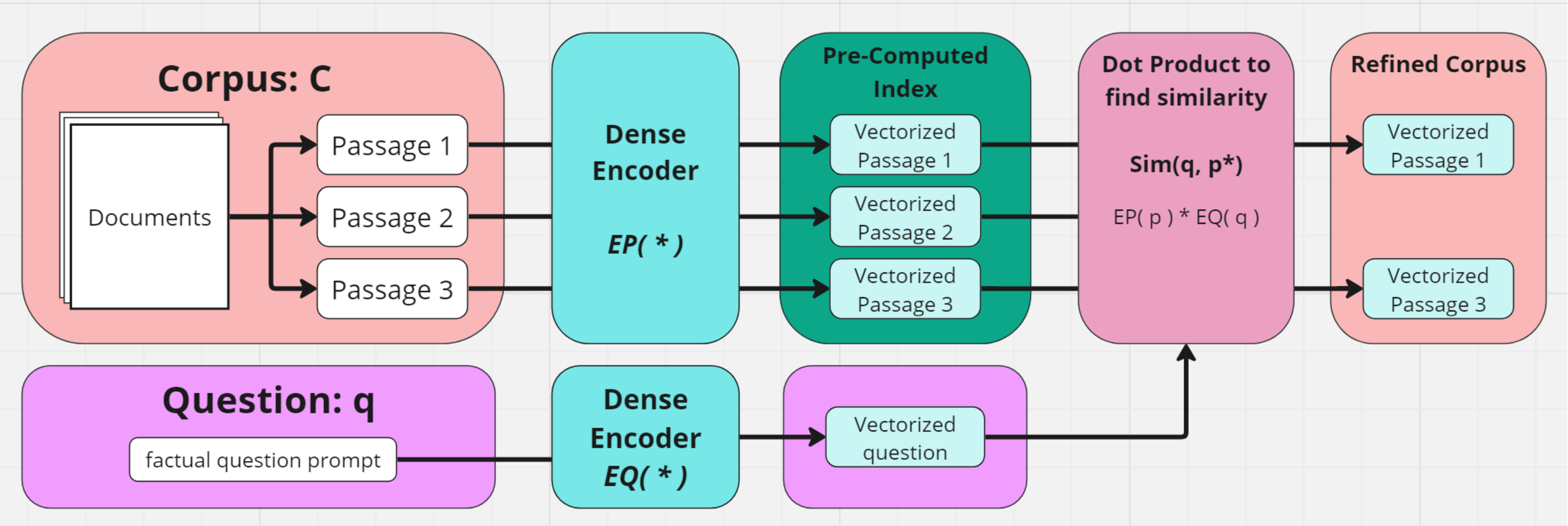
Overview

- **Goal:** We focus our research in this work on improving the retrieval component in open-domain QA.

- **Current state:** Utilization of "TF-IDF" or "BM25" as a means of comparing passages to questions

- **General strategy:** Encode passages and question prompts as Dense Vectors to search for answers. This method is labled Dense Passage Retrieval.

Dense Passage Retrieval

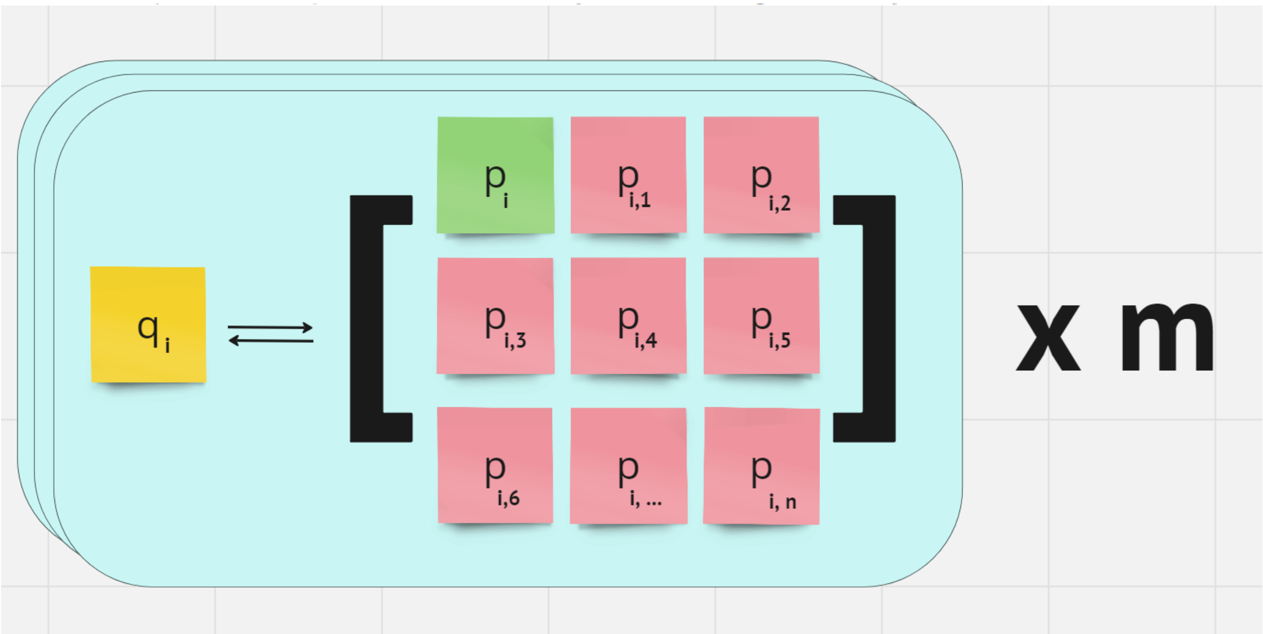


Training

- Train Encoders with question-passage pairing

- 1 Positive result
- *n* Irrelevant results

$$\mathcal{D} = \{\langle q_i, p_i^+, p_{i,1}^-, \cdots, p_{i,n}^- \rangle\}_{i=1}^m$$



- Loss function

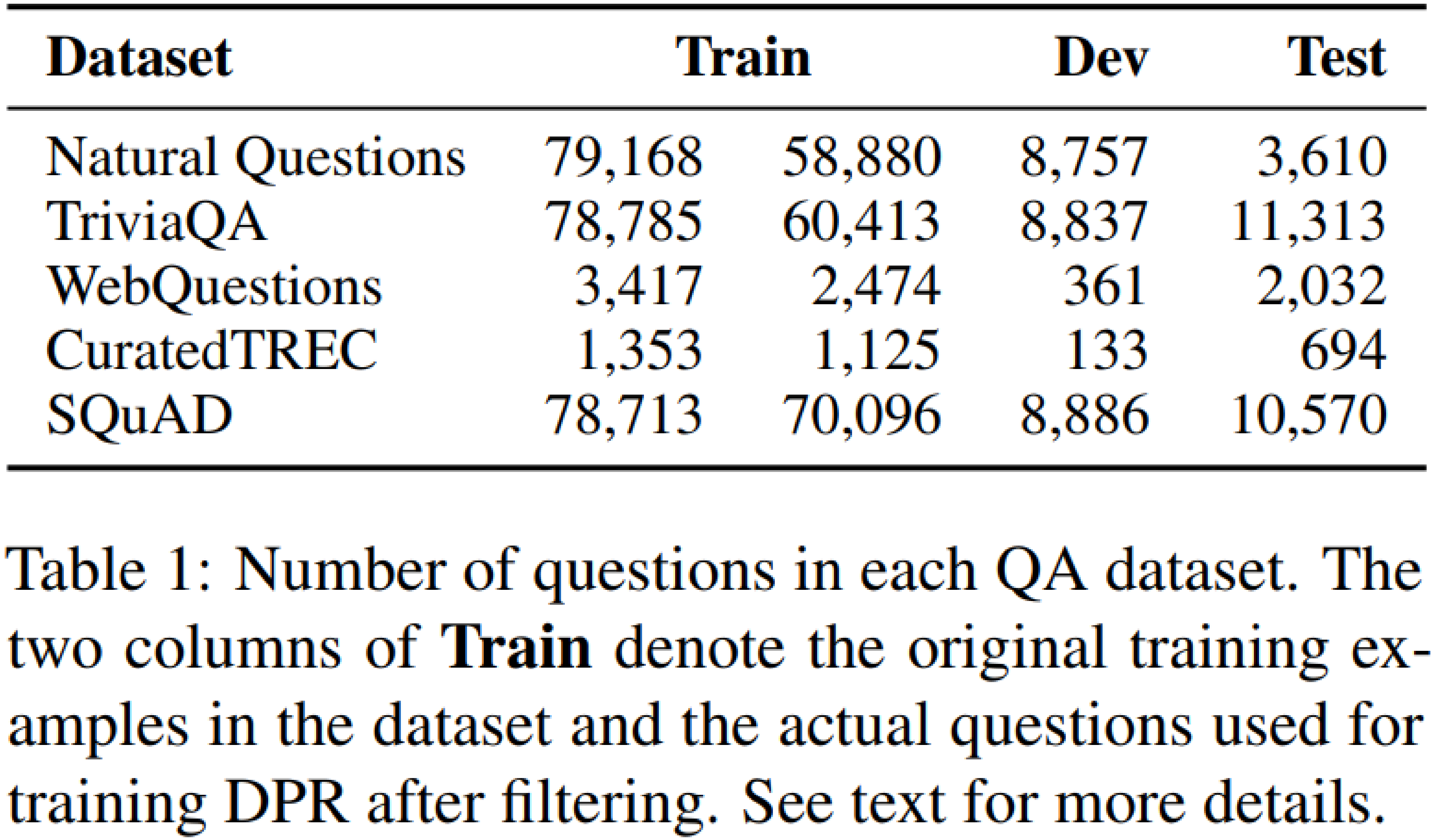
$$L(q_i, p_i^+, p_{i,1}^-, \cdots, p_{i,n}^-)$$

$$= -\log \frac{e^{\mathrm{sim}(q_i, p_i^+)}}{e^{\mathrm{sim}(q_i, p_i^+)} + \sum_{j=1}^n e^{\mathrm{sim}(q_i, p_{i,j}^-)}}$$

Training

- 3 Different Negative Types
  - Random: any random passage from the corpus;
  - BM25: top passages returned by BM25 which don’t contain the answer but match most question tokens;
  - Gold: positive passages paired with other questions which appear in th e training set.

Training Data

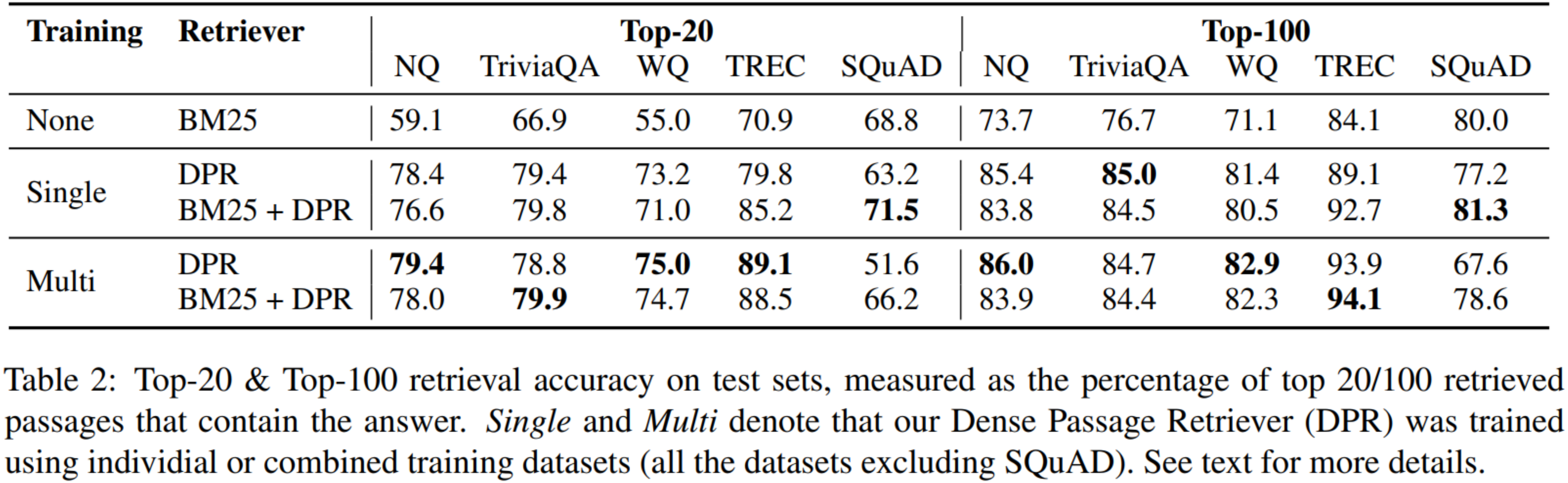


Dataset	Train		Dev	Test
Natural Questions	79,168	58,880	8,757	3,610
TriviaQA	78,785	60,413	8,837	11,313
WebQuestions	3,417	2,474	361	2,032
CuratedTREC	1,353	1,125	133	694
SQuAD	78,713	70,096	8,886	10,570

Table 1: Number of questions in each QA dataset. The two columns of **Train** denote the original training examples in the dataset and the actual questions used for training DPR after filtering. See text for more details.

Experiment

Results of Retrievers in different situations



Training	Retriever	Top-20					Top-100				
		NQ	TriviaQA	WQ	TREC	SQuAD	NQ	TriviaQA	WQ	TREC	SQuAD
None	BM25	59.1	66.9	55.0	70.9	68.8	73.7	76.7	71.1	84.1	80.0
Single	DPR	78.4	79.4	73.2	79.8	63.2	85.4	**85.0**	81.4	89.1	77.2
	BM25 + DPR	76.6	79.8	71.0	85.2	**71.5**	83.8	84.5	80.5	92.7	**81.3**
Multi	DPR	**79.4**	78.8	**75.0**	**89.1**	51.6	**86.0**	84.7	**82.9**	93.9	67.6
	BM25 + DPR	78.0	**79.9**	74.7	88.5	66.2	83.9	84.4	82.3	**94.1**	78.6

Table 2: Top-20 & Top-100 retrieval accuracy on test sets, measured as the percentage of top 20/100 retrieved passages that contain the answer. *Single* and *Multi* denote that our Dense Passage Retriever (DPR) was trained using individial or combined training datasets (all the datasets excluding SQuAD). See text for more details.

Experiment

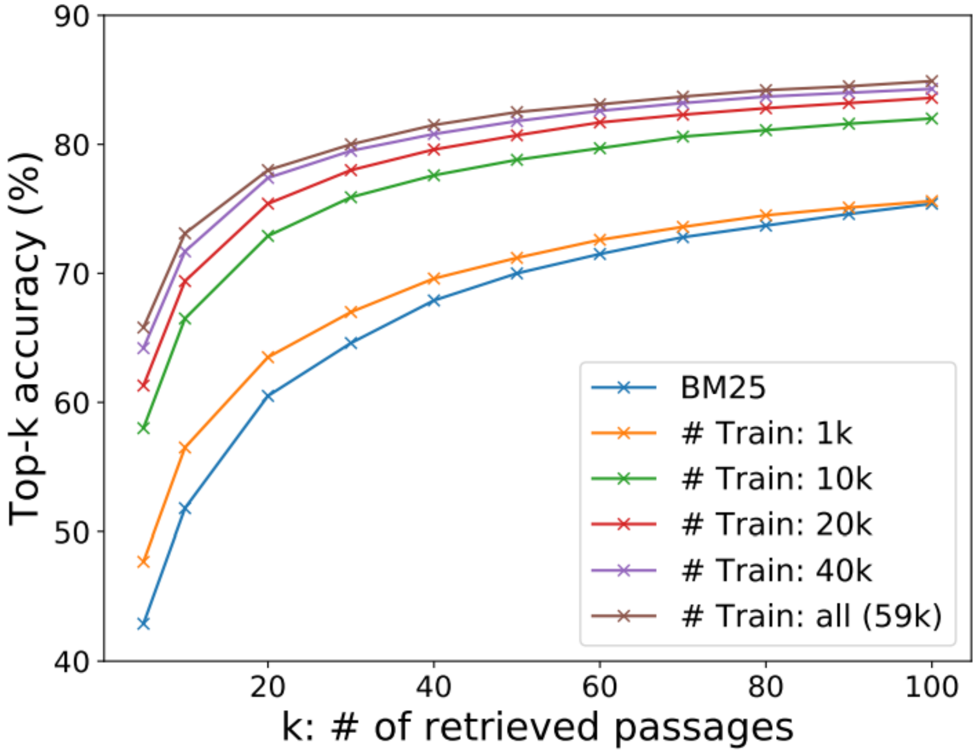
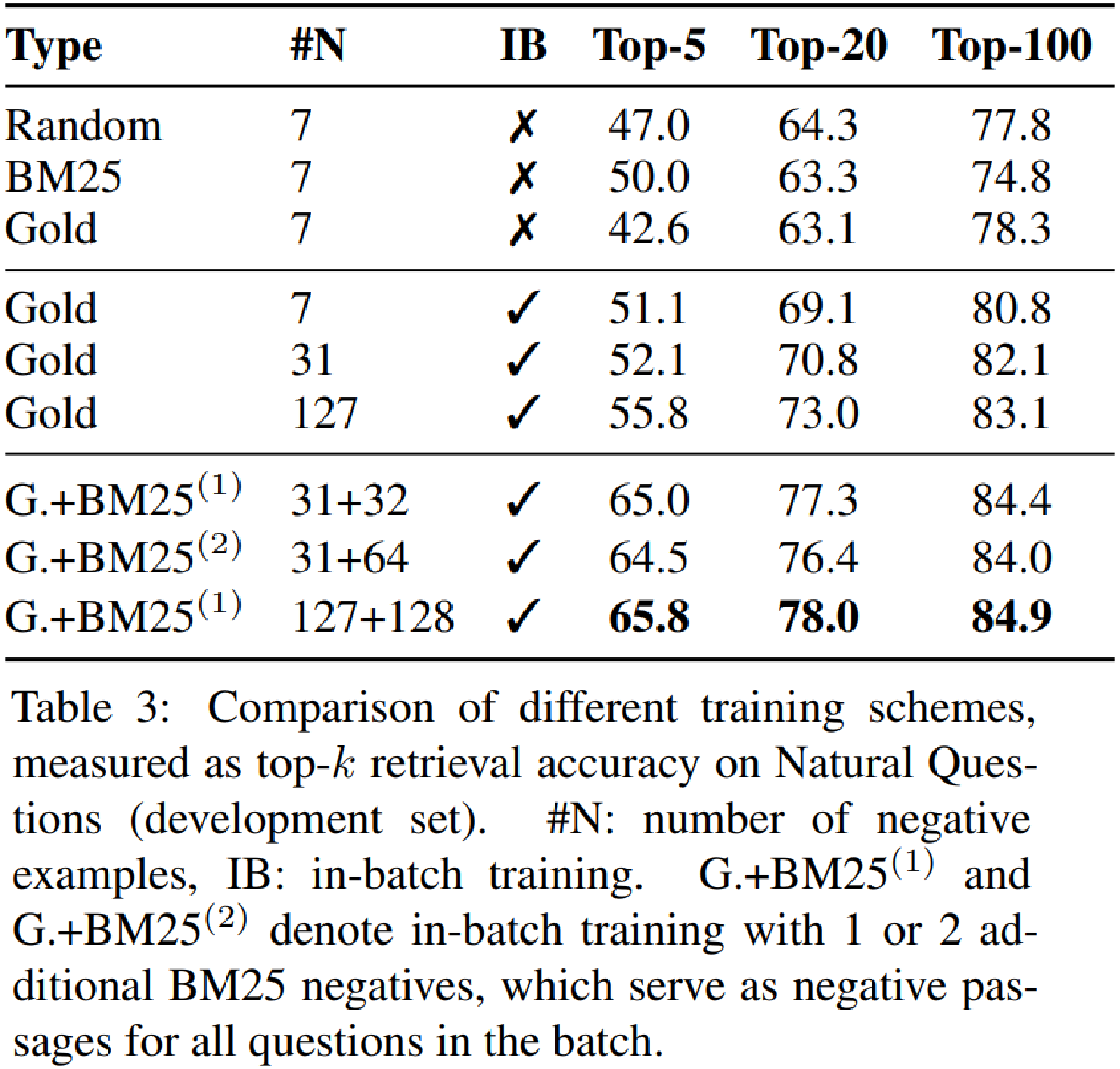


Figure 1: Retriever top-k accuracy with different numbers of training examples used in our dense passage retriever vs BM25. The results are measured on the development set of Natural Questions. Our DPR trained using 1,000 examples already outperforms BM25.

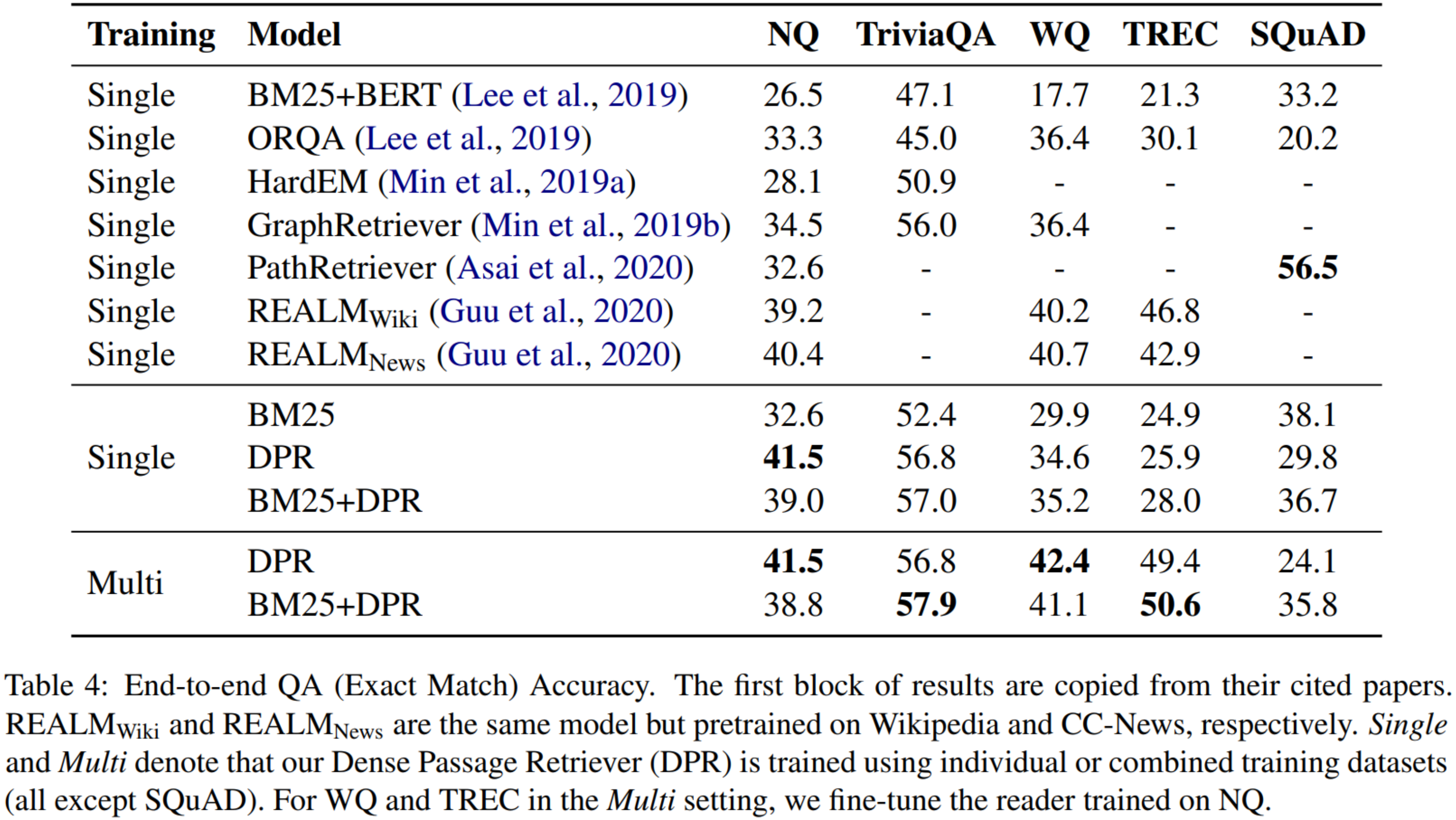
Experiment



Type	#N	IB	Top-5	Top-20	Top-100
Random	7	✗	47.0	64.3	77.8
BM25	7	✗	50.0	63.3	74.8
Gold	7	✗	42.6	63.1	78.3
Gold	7	✓	51.1	69.1	80.8
Gold	31	✓	52.1	70.8	82.1
Gold	127	✓	55.8	73.0	83.1
G.+BM25$^{(1)}$	31+32	✓	65.0	77.3	84.4
G.+BM25$^{(2)}$	31+64	✓	64.5	76.4	84.0
G.+BM25$^{(1)}$	127+128	✓	**65.8**	**78.0**	**84.9**

Table 3: Comparison of different training schemes, measured as top-k retrieval accuracy on Natural Questions (development set). #N: number of negative examples, IB: in-batch training. G.+BM25$^{(1)}$ and G.+BM25$^{(2)}$ denote in-batch training with 1 or 2 additional BM25 negatives, which serve as negative passages for all questions in the batch.

Results



Training	Model	NQ	TriviaQA	WQ	TREC	SQuAD
Single	BM25+BERT (Lee et al., 2019)	26.5	47.1	17.7	21.3	33.2
Single	ORQA (Lee et al., 2019)	33.3	45.0	36.4	30.1	20.2
Single	HardEM (Min et al., 2019a)	28.1	50.9	-	-	-
Single	GraphRetriever (Min et al., 2019b)	34.5	56.0	36.4	-	-
Single	PathRetriever (Asai et al., 2020)	32.6	-	-	-	**56.5**
Single	REALM$_{\text{Wiki}}$ (Guu et al., 2020)	39.2	-	40.2	46.8	-
Single	REALM$_{\text{News}}$ (Guu et al., 2020)	40.4	-	40.7	42.9	-
Single	BM25	32.6	52.4	29.9	24.9	38.1
	DPR	**41.5**	56.8	34.6	25.9	29.8
	BM25+DPR	39.0	57.0	35.2	28.0	36.7
Multi	DPR	**41.5**	56.8	**42.4**	49.4	24.1
	BM25+DPR	38.8	**57.9**	41.1	**50.6**	35.8

Table 4: End-to-end QA (Exact Match) Accuracy. The first block of results are copied from their cited papers. REALM$_{\text{Wiki}}$ and REALM$_{\text{News}}$ are the same model but pretrained on Wikipedia and CC-News, respectively. *Single* and *Multi* denote that our Dense Passage Retriever (DPR) is trained using individual or combined training datasets (all except SQuAD). For WQ and TREC in the *Multi* setting, we fine-tune the reader trained on NQ.

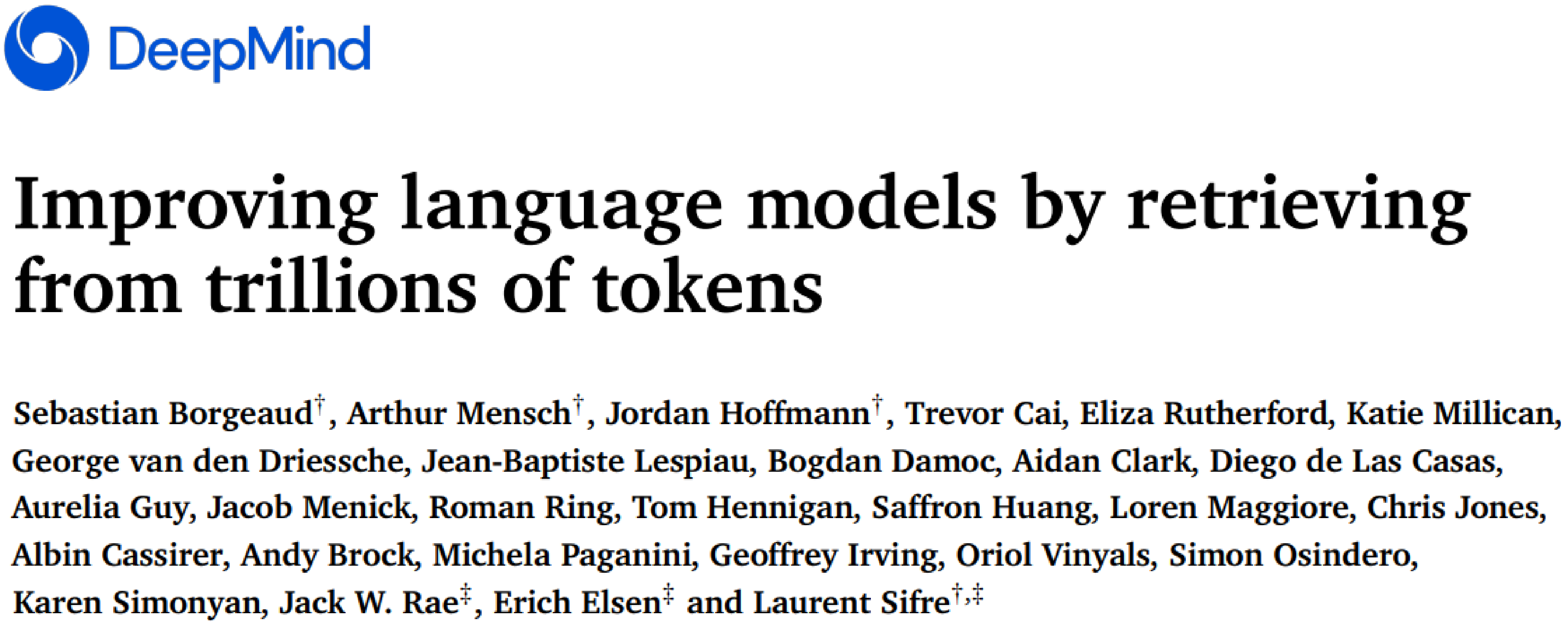
Conclusion

- Dense retrieval can outperform traditional sparse retrieval components in open-domain question answering.
- Training a dense retriever successfully is no longer a pipe dream.
- These developments have set a new standard for multiple open-domain question answering benchmarks

Limitations

- While impressive it is not ready to completely replace current methods like BM25
  - DPR alone with randomly selected negative passages or golden passages have limited effectiveness.
  - Needs the careful touch of BM25 to reach peak performance

- DPR training/computing is much higher in cost than that of BM25 when it comes to creating a pre-computed index



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

Semi-parametric approach

- Large performance improvements on Language Model (LM)
  - Increasing the amount of data, training compute, or model parameters

- Increasing the number of parameters
  - Increased memorization of the training data
  - Additional computations at training and inference time

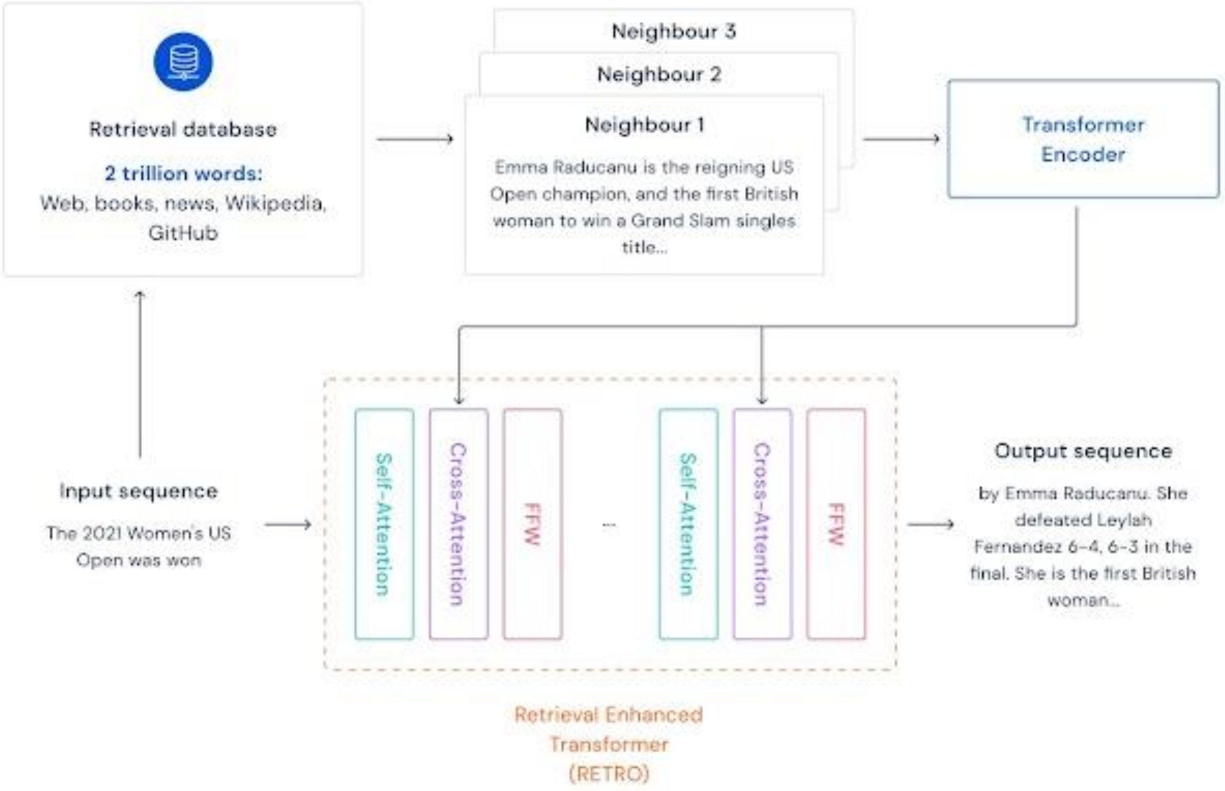
  → Decoupling this approach: “Retrieval from **a large text database** as a complementary path”

- Limitation of previous works
  - Performed on the limited amount of knowledge base

  → “Our work is the first to show the benefit of **scaling the retrieval database** to **trillions** of tokens”

RETRO

- Retrieval-enhanced transformer (RETRO): High-level overview

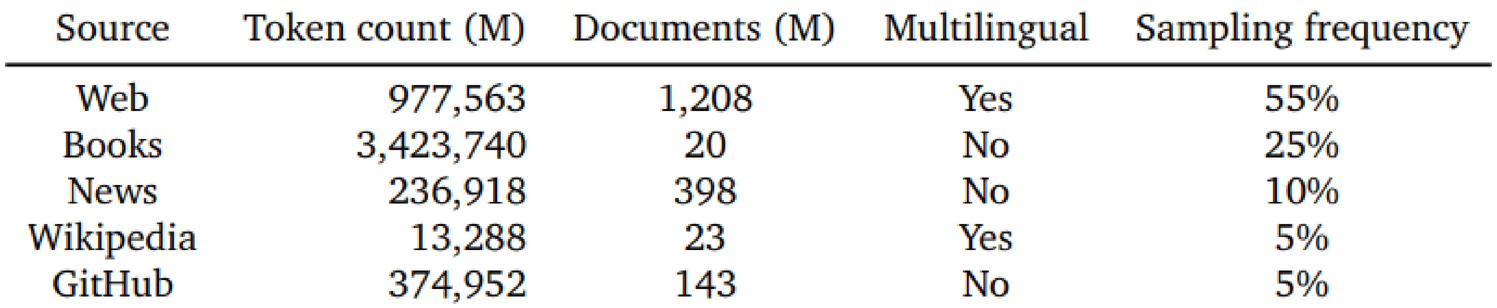

Retrieval database
2 trillion words:
Web, books, news, Wikipedia, GitHub
Neighbour 3
Neighbour 2
Neighbour 1
Emma Raducanu is the reigning US Open champion, and the first British woman to win a Grand Slam singles title...
Transformer Encoder
Input sequence
The 2021 Women's US Open was won
Self-Attention
Cross-Attention
FFW
...
Self-Attention
Cross-Attention
FFW
Output sequence
by Emma Raducanu. She defeated Leylah Fernandez 6-4, 6-3 in the final. She is the first British woman...
Retrieval Enhanced Transformer (RETRO)

https://deepmind.google/discover/blog/improving-language-models-by-retrieving-from-trillions-of-tokens/

Dataset

- *Massive Text*
  - Multiple sources, multiple languages
  - SentencePiece tokenization (128,000 vocab tokens)

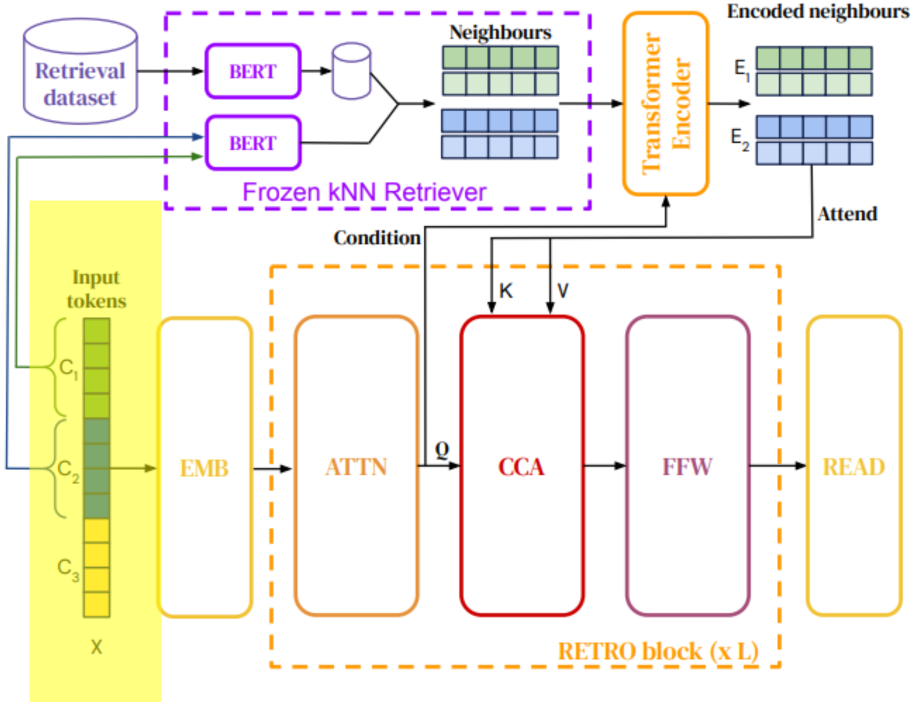
- Retrieval database
  - For training: 600B tokens
  - For evaluation: 1.75T tokens



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%

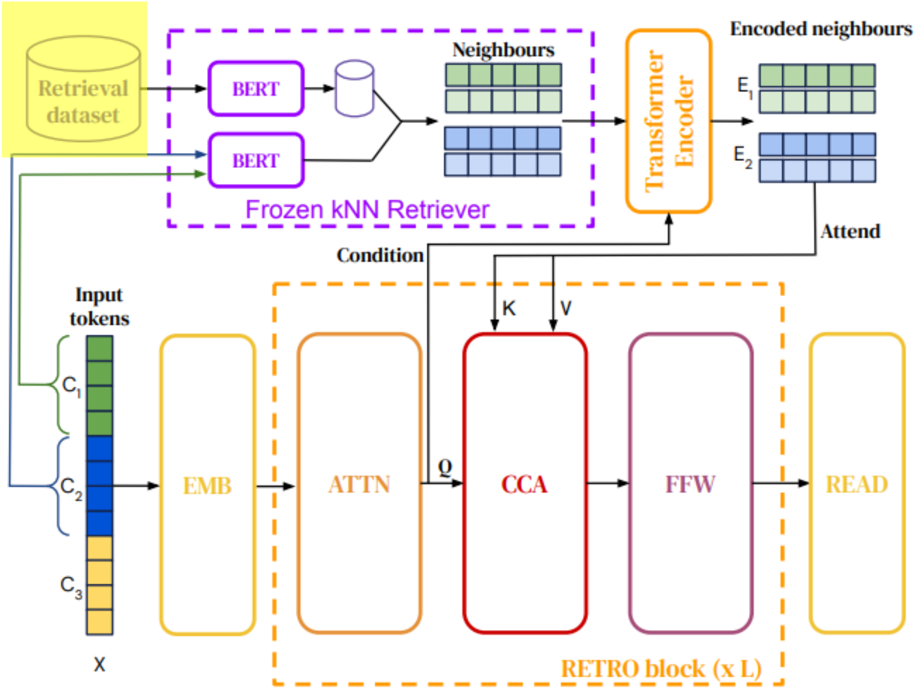
Retrieval-enhanced autoregressive token models

- Token granularity for efficient retrieval
- Split each n-token-long example $X = (x_1, \dots, x_n)$ into a sequence of l chunks $(C_1, \dots, C_l)$



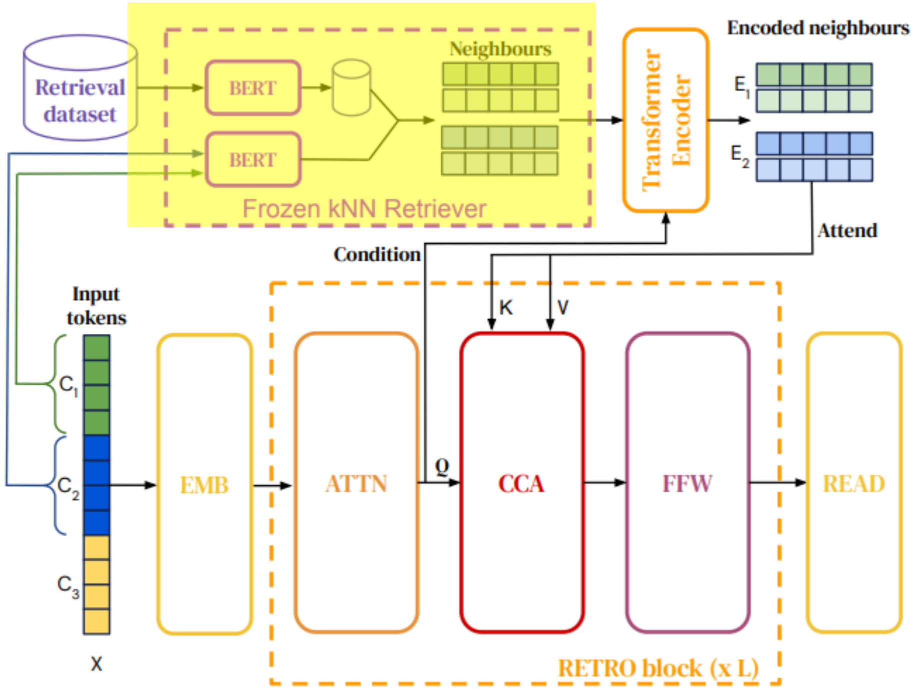
Constructing key-value database

- KEY: frozen BERT embeddings
- VALUE: raw chunks of text tokens



Nearest neighbor retrieval

- Retrieval as a way to augment input examples
- K-nearest neighbor based on L2 distance



RETRO Block

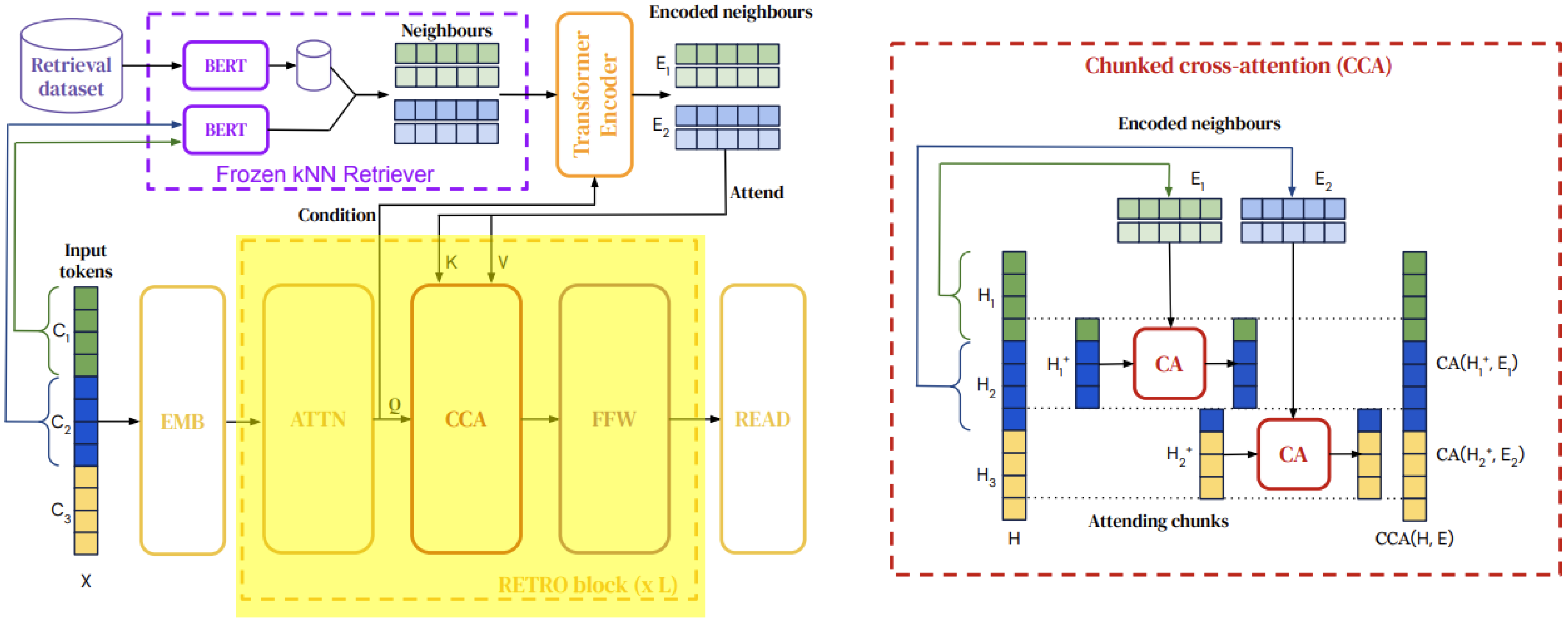
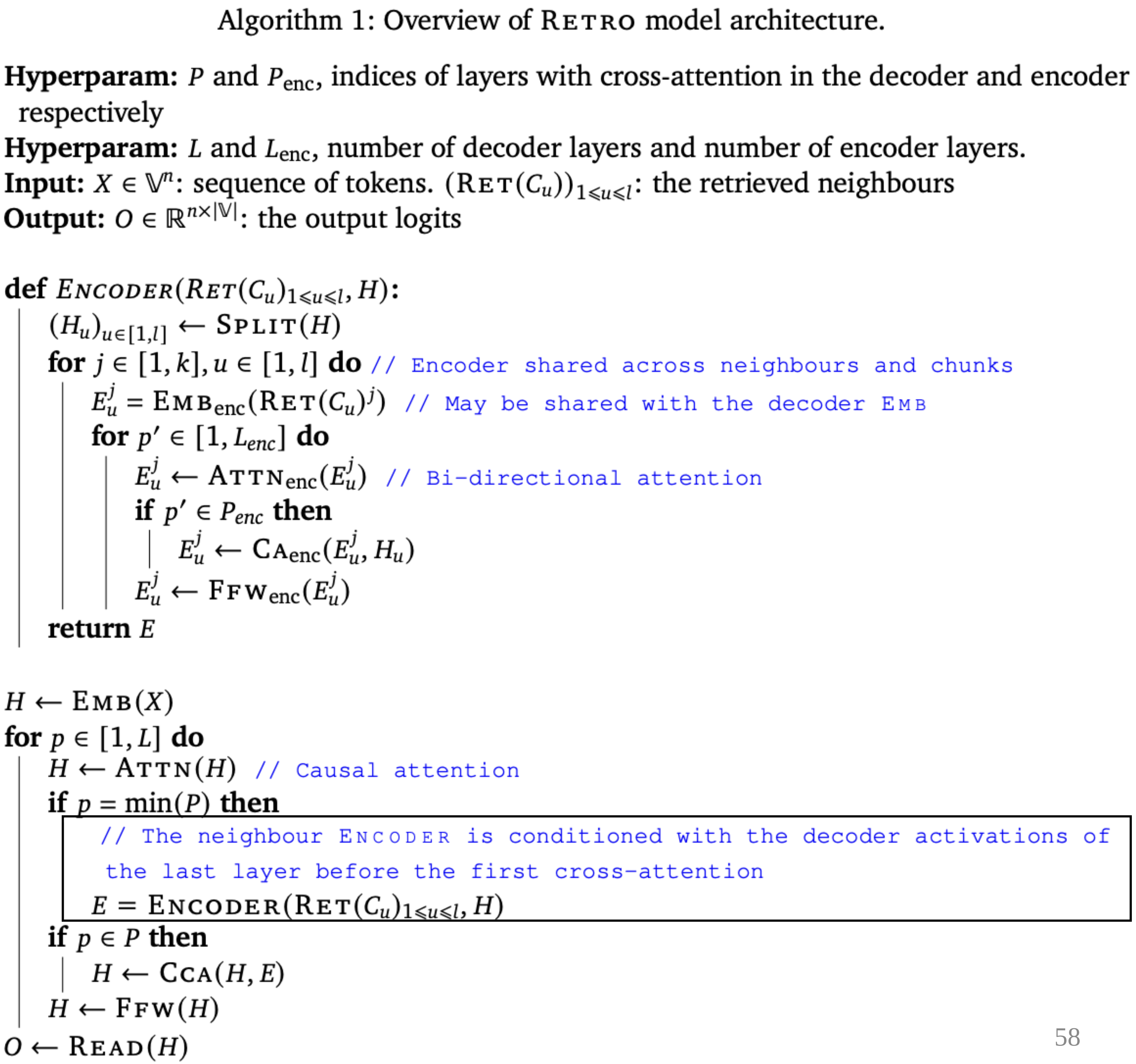


Figure 2 | **RETRO architecture.** *Left:* simplified version where a sequence of length $n = 12$ is split into $l = 3$ chunks of size $m = 4$. For each chunk, we retrieve $k = 2$ neighbours of $r = 5$ tokens each. The retrieval pathway is shown on top. *Right:* Details of the interactions in the CCA operator. Causality is maintained as neighbours of the first chunk only affect the last token of the first chunk and tokens from the second chunk.

RETRO Block

- k retrieval neighbors input to bi-directional encoder
- Conditioned on H_u which is the intermediate activation of chunk C_u
- This yields the set of encoded neighbors E, indexed by u



Algorithm 1: Overview of RETRO model architecture.

Hyperparam: P and P_{enc}, indices of layers with cross-attention in the decoder and encoder respectively
Hyperparam: L and L_{enc}, number of decoder layers and number of encoder layers.
Input: $X \in \mathbb{V}^n$: sequence of tokens. $(\textsc{Ret}(C_u))_{1 \leqslant u \leqslant l}$: the retrieved neighbours
Output: $O \in \mathbb{R}^{n \times |\mathbb{V}|}$: the output logits

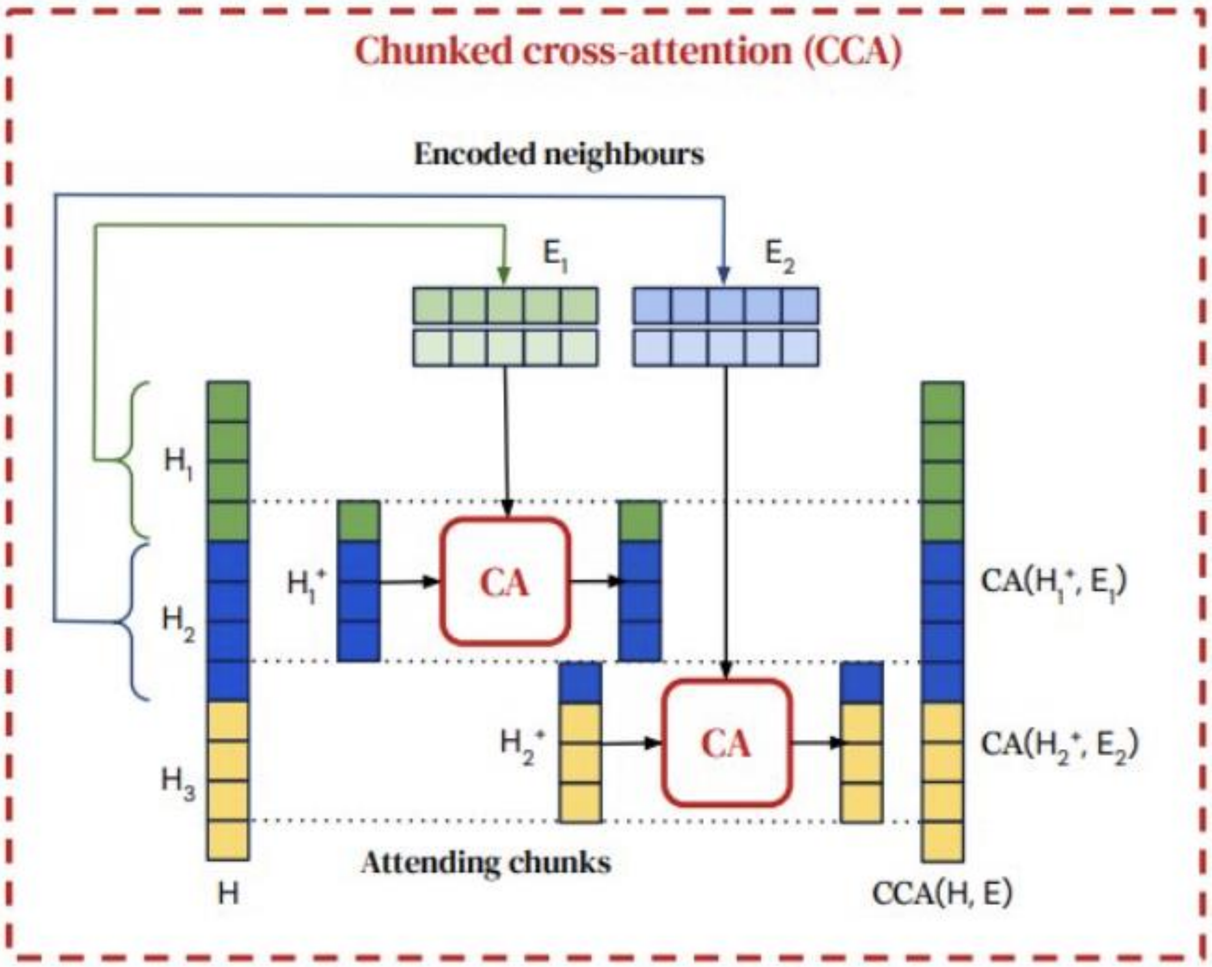
```
def $\textsc{Encoder}(\textsc{Ret}(C_u)_{1\leqslant u\leqslant l}, H)$:
    $(H_u)_{u\in[1,l]} \leftarrow \textsc{Split}(H)$
    for $j \in [1,k], u \in [1,l]$ do // Encoder shared across neighbours and chunks
        $E_u^j = \textsc{Emb}_{\text{enc}}(\textsc{Ret}(C_u)^j)$ // May be shared with the decoder EMB
        for $p' \in [1, L_{enc}]$ do
            $E_u^j \leftarrow \textsc{Attn}_{\text{enc}}(E_u^j)$ // Bi-directional attention
            if $p' \in P_{enc}$ then
                $E_u^j \leftarrow \textsc{Ca}_{\text{enc}}(E_u^j, H_u)$
            $E_u^j \leftarrow \textsc{Ffw}_{\text{enc}}(E_u^j)$
    return $E$

$H \leftarrow \textsc{Emb}(X)$
for $p \in [1, L]$ do
    $H \leftarrow \textsc{Attn}(H)$ // Causal attention
    if $p = \min(P)$ then
        // The neighbour ENCODER is conditioned with the decoder activations of
        the last layer before the first cross-attention
        $E = \textsc{Encoder}(\textsc{Ret}(C_u)_{1\leqslant u\leqslant l}, H)$
    if $p \in P$ then
        $H \leftarrow \textsc{Cca}(H, E)$
    $H \leftarrow \textsc{Ffw}(H)$
$O \leftarrow \textsc{Read}(H)$
```

Chunked Cross Attention

- Break down the intermediate activations to compute the attending chunks H_u^+

- Compute attention between H_u^+ and embedded neighbors E_u

- This CCA mechanism allows for C_u to attend to the neighbors of preceding chunk C_{u-1} but the activations of each chunk is dependent upon all previous chunks



RETRO Block

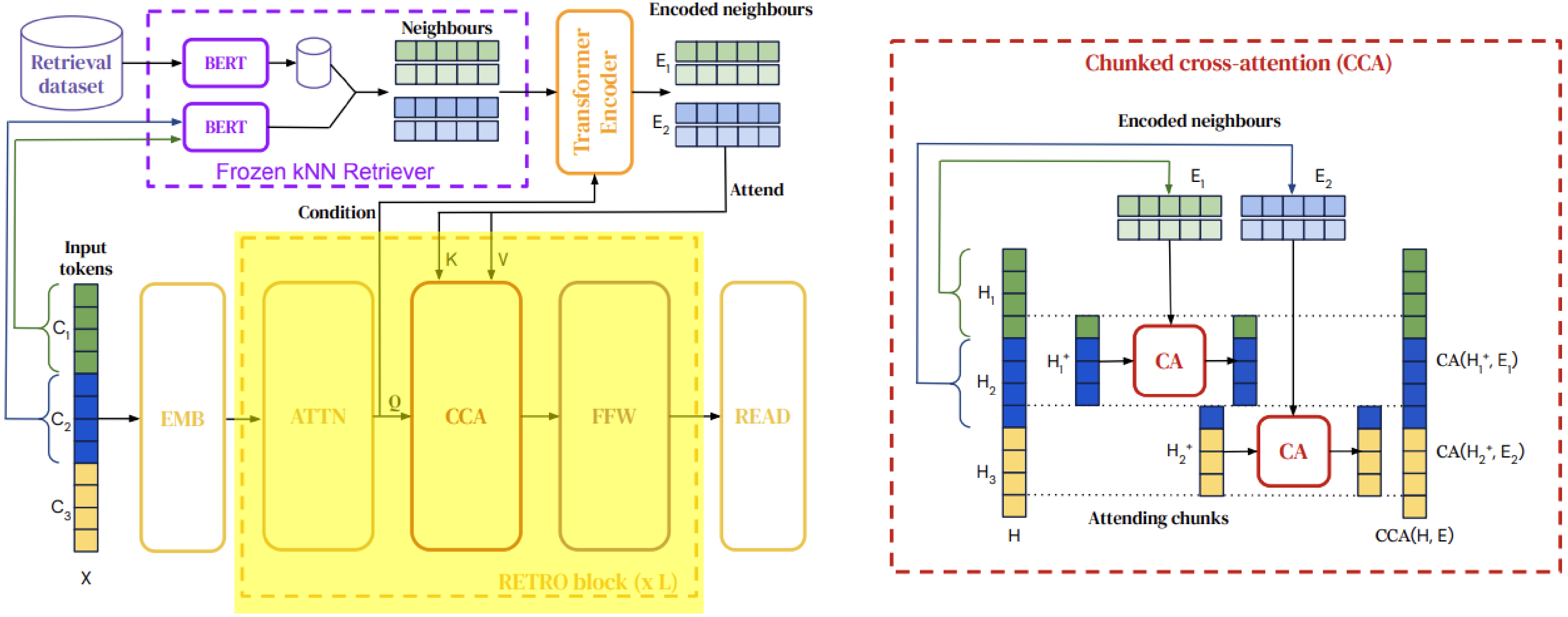


Figure 2 | **Retro architecture.** *Left:* simplified version where a sequence of length $n = 12$ is split into $l = 3$ chunks of size $m = 4$. For each chunk, we retrieve $k = 2$ neighbours of $r = 5$ tokens each. The retrieval pathway is shown on top. *Right:* Details of the interactions in the Cca operator. Causality is maintained as neighbours of the first chunk only affect the last token of the first chunk and tokens from the second chunk.

Evaluating Data Leakage

- Evaluation sequences and training data are treated as chunks
- Closest neighbors *in the training data* are retrieved for each evaluation chunk
- Compare the closest pair between the training data and evalutation sequence
- Frame as log likelihood metric

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

Results

RETRO effectiveness on different corpus and models of different sizes

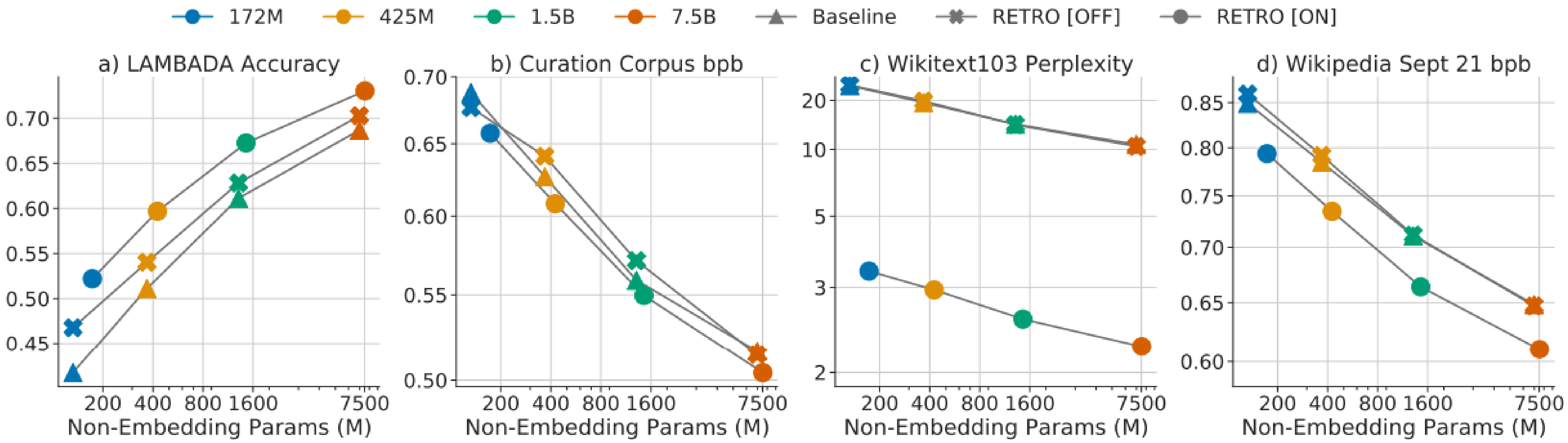


Figure 3 | **Scaling with respect to model size.** (a) LAMBADA top-1 accuracy. (b) Evaluation loss on curation corpus. (c) Perplexity on Wikitext103 valid. (d) Bits-per-byte on selected Wikipedia articles from September 2021.

Results

RETRO outperforms Jurassic for dataflow

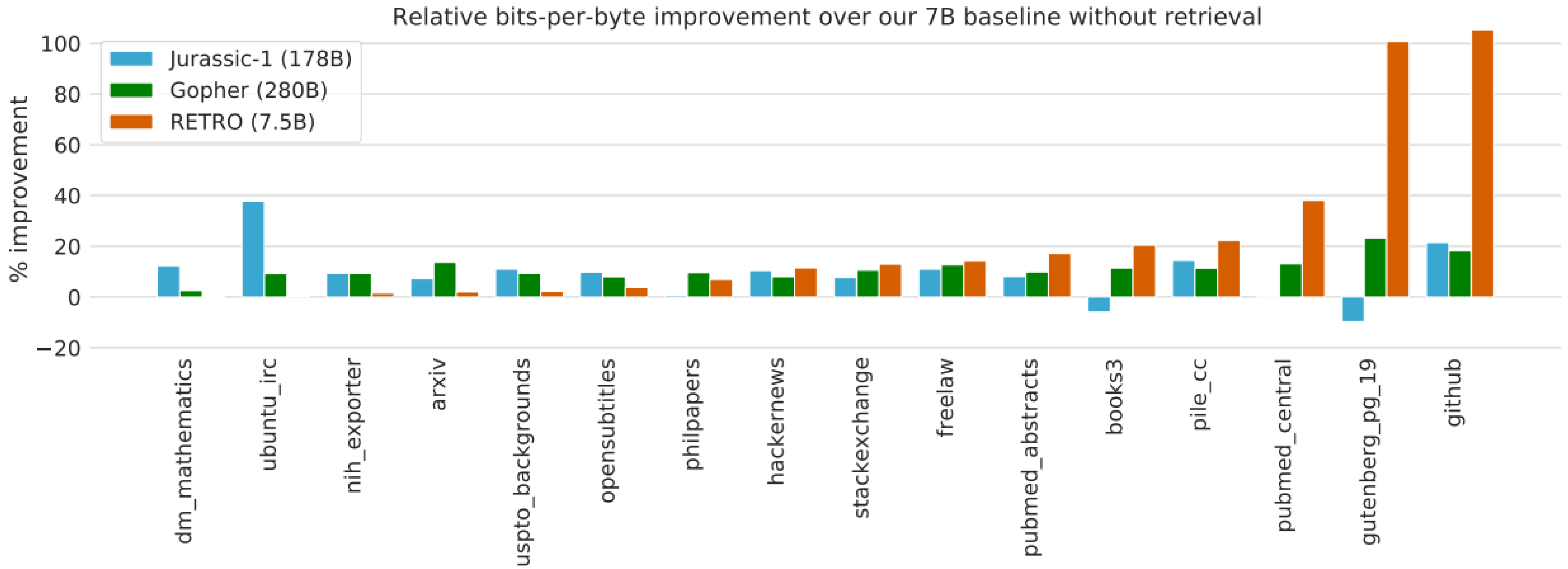


Figure 4 | **The Pile: Comparison of our 7B baseline against Jurassic-1, Gopher, and RETRO.** We observe that the retrieval model outperforms the baseline on all test sets and outperforms Jurassic-1 on a majority of them, despite being over an order of magnitude smaller.

Results

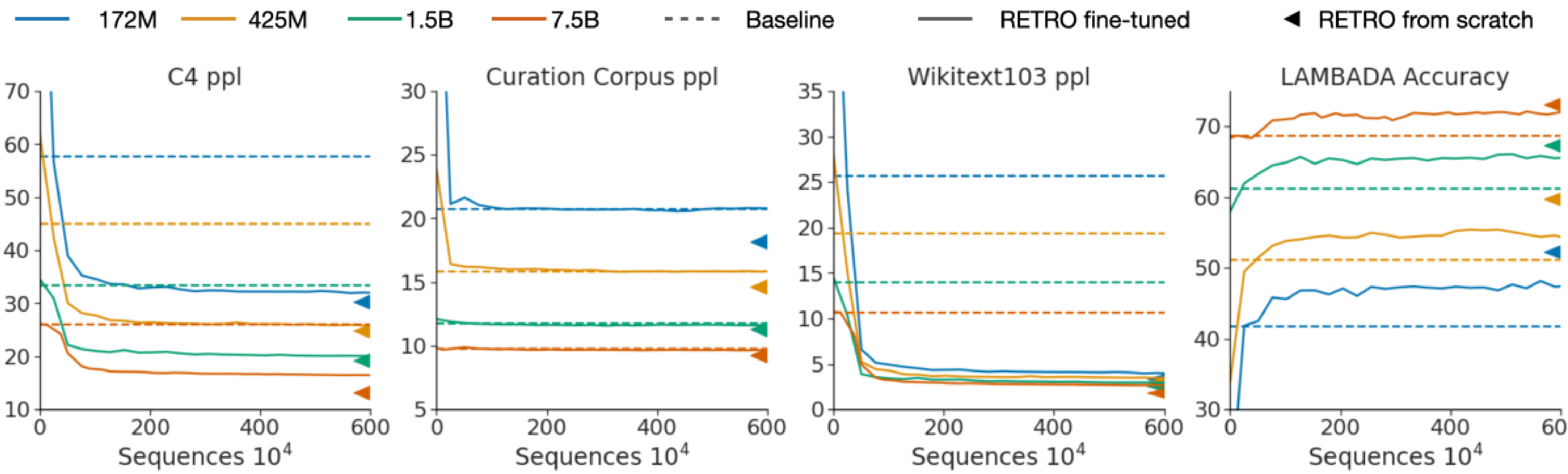
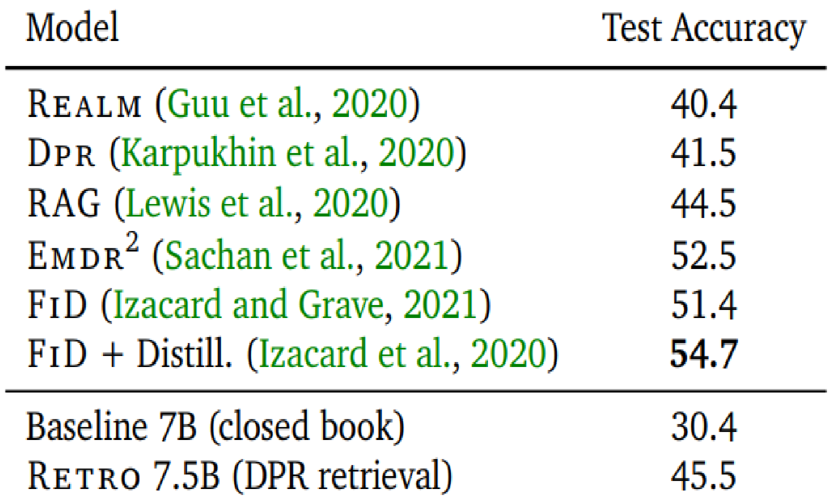


Figure 5 | **Retro-fitting a baseline transformer.** Any transformer can be fine-tuned into a retrieval-enhanced transformer by randomly initializing and training only the chunked cross-attention and retrieval encoder weights. Fine-tuning in this way quickly recovers and surpasses the non-retrieval performance, and almost achieves the same performance as training a retrieval model from scratch (shown by the arrow on the right hand side of each plot). We find good performance Retro-fitting our models training on only 3% the number of tokens seen during pre-training.

Results

RETRO is competitive with other retrieval methods however it underperforms compared to the recent FiD method.



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

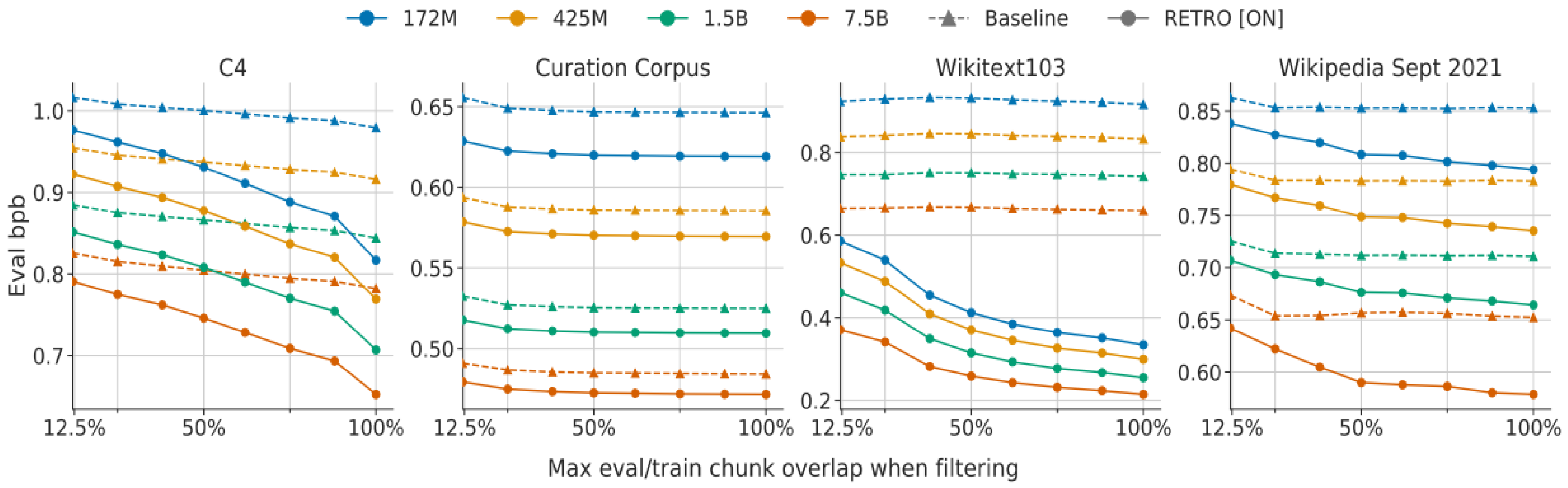


Figure 6 | **Performance vs. longest common retrieval substring.** Evaluation loss as a function of allowed longest common substring between evaluation data chunks and their nearest neighbours. Retrieval still helps when considering chunks with no more than 8 contiguous tokens overlapping with training dataset chunks.

Conclusion

- Retrieval-Enhanced Transformers (RETRO) is an efficient, competitive, and scalable retrieval model for datasets up to 7B
- Previous models can be fine tuned easily to become RETRO
- Limited data leakage, however could still be a factor on larger scale models

Limitations

- Difficult to know how RETRO could apply to other transformer models
  - Study only used one fixed encoder, (BERT), to generate embeddings
    - Costly to retrain if BERT ever updates

- Future work can explore RETRO and its effectiveness under other encoder models

Main Ideas

- RAG approaches allow for language models to generalize to complex domains which may be poorly represented in the training corpus
- Some key ideas based upon in-context learning are integral to the applicability of these techniques
- Future work may focus on finding better methods to index documents and retrieve appropriate text for a given task

Questions?