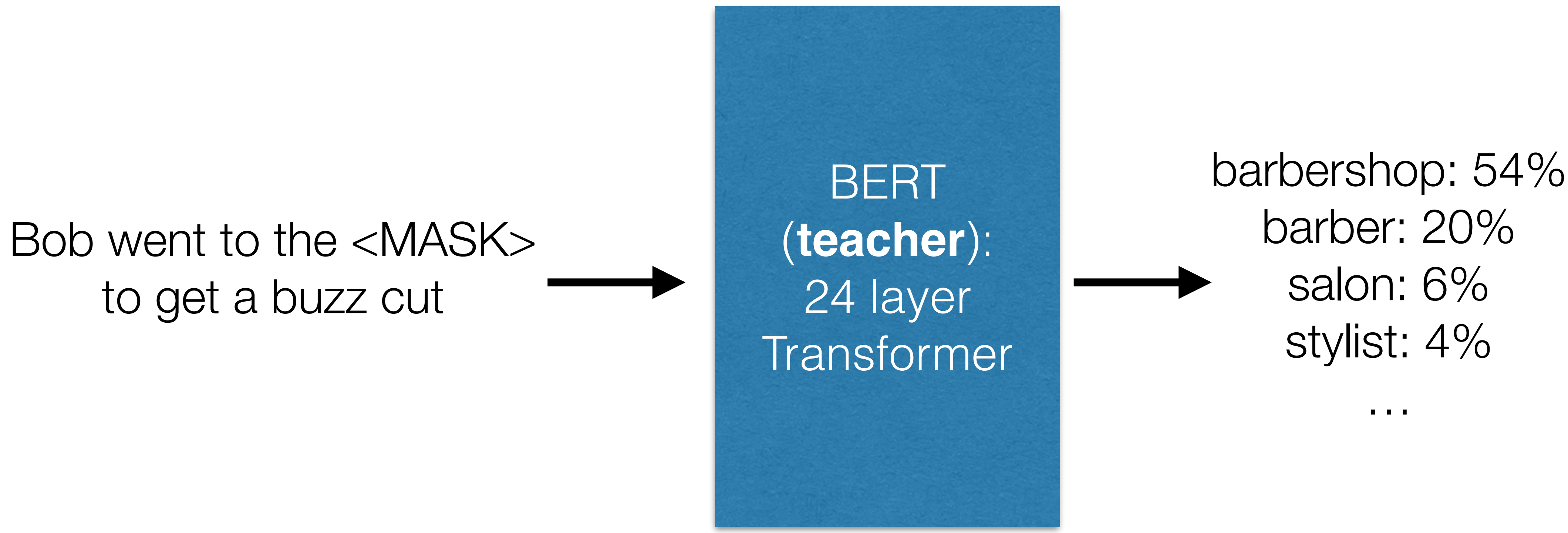
Retrieval-augmented language models

CS 685, Spring 2022

Advanced Natural Language Processing

Mohit Iyyer
College of Information and Computer Sciences
University of Massachusetts Amherst



Bob went to the <MASK> to get a buzz cut
BERT (**teacher**): 24 layer Transformer
barbershop: 54%
barber: 20%
salon: 6%
stylist: 4%
...

World knowledge is *implicitly* encoded in BERT's parameters! (e.g., that barbershops are places to get buzz cuts)

Bob went to the <MASK> to get a buzz cut

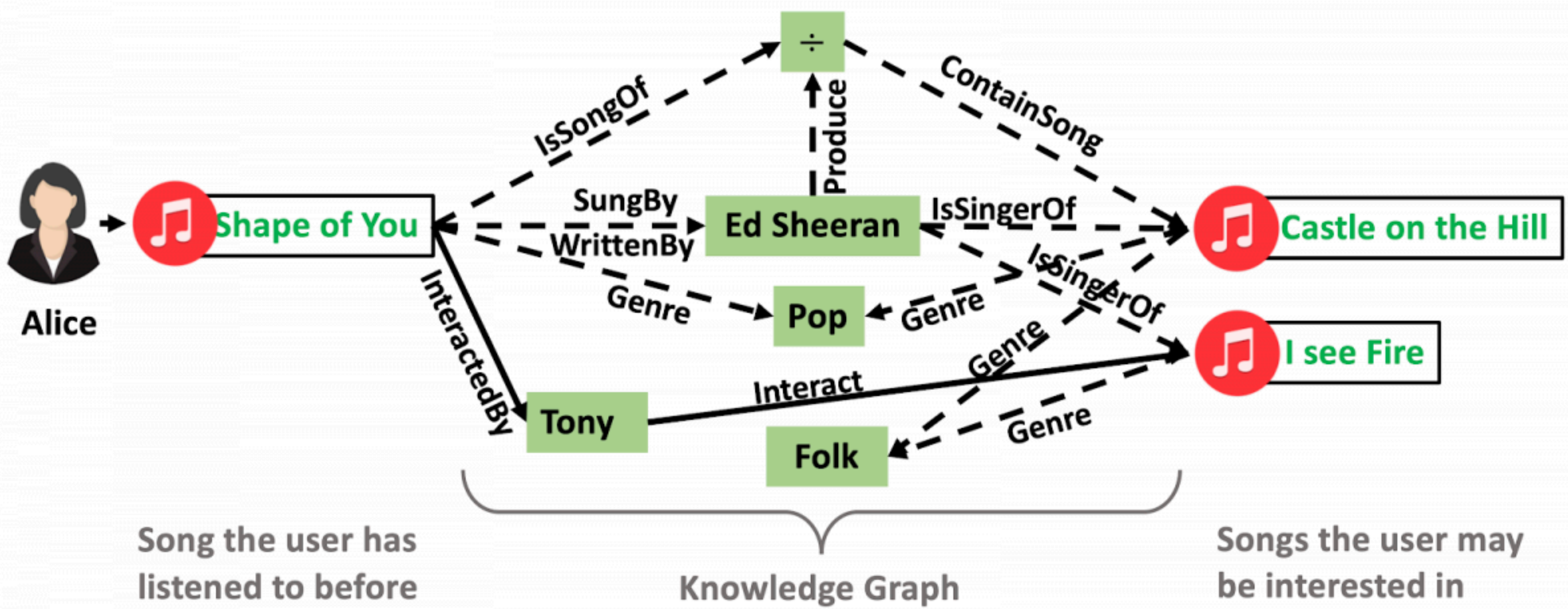
BERT (**teacher**): 24 layer Transformer

barbershop: 54%
barber: 20%
salon: 6%
stylist: 4%
…

In these language models, the learned world knowledge is stored *implicitly* in the parameters of the underlying neural network. This makes it difficult to determine what knowledge is stored in the network and where. Furthermore, storage space is limited by the size of the network—to capture more world knowledge, one must train ever-larger networks, which can be prohibitively slow or expensive.

Guu et al., 2020 (“REALM”)

One option: condition predictions on explicit *knowledge graphs*



Wang et al., 2019

Pros / cons

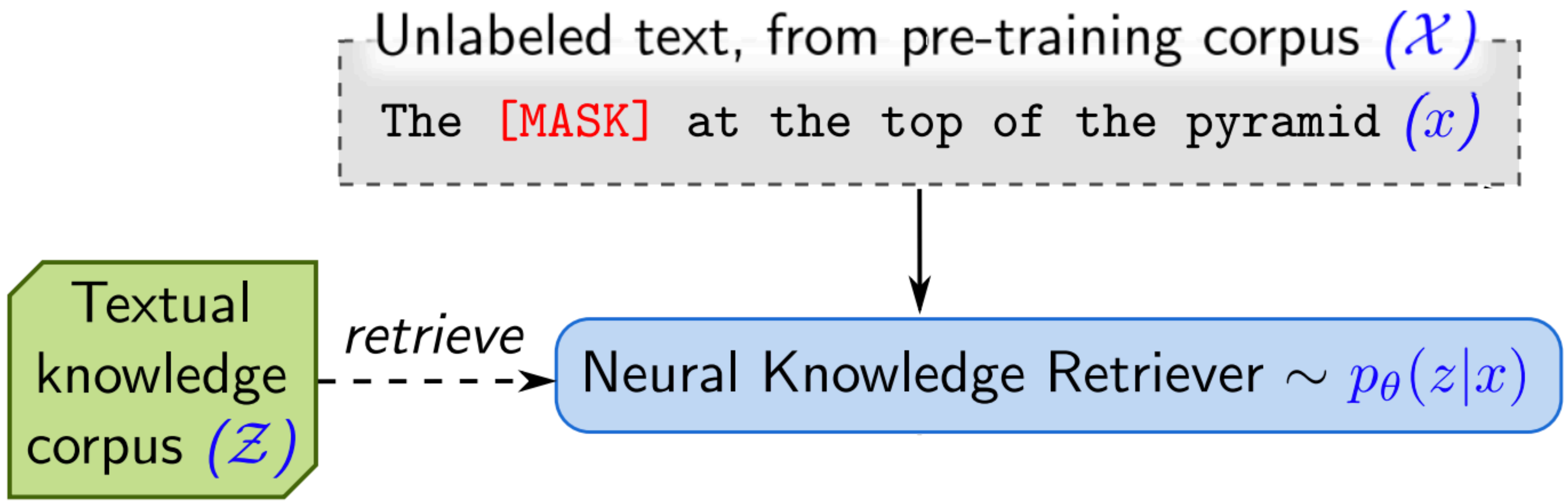
- Explicit graph structure makes KGs easy to navigate
- Knowledge graphs are expensive to produce at scale
- Automatic knowledge graph induction is an open research problem
- Knowledge graphs struggle to encode complex relations between entities

Another source of knowledge: unstructured text!

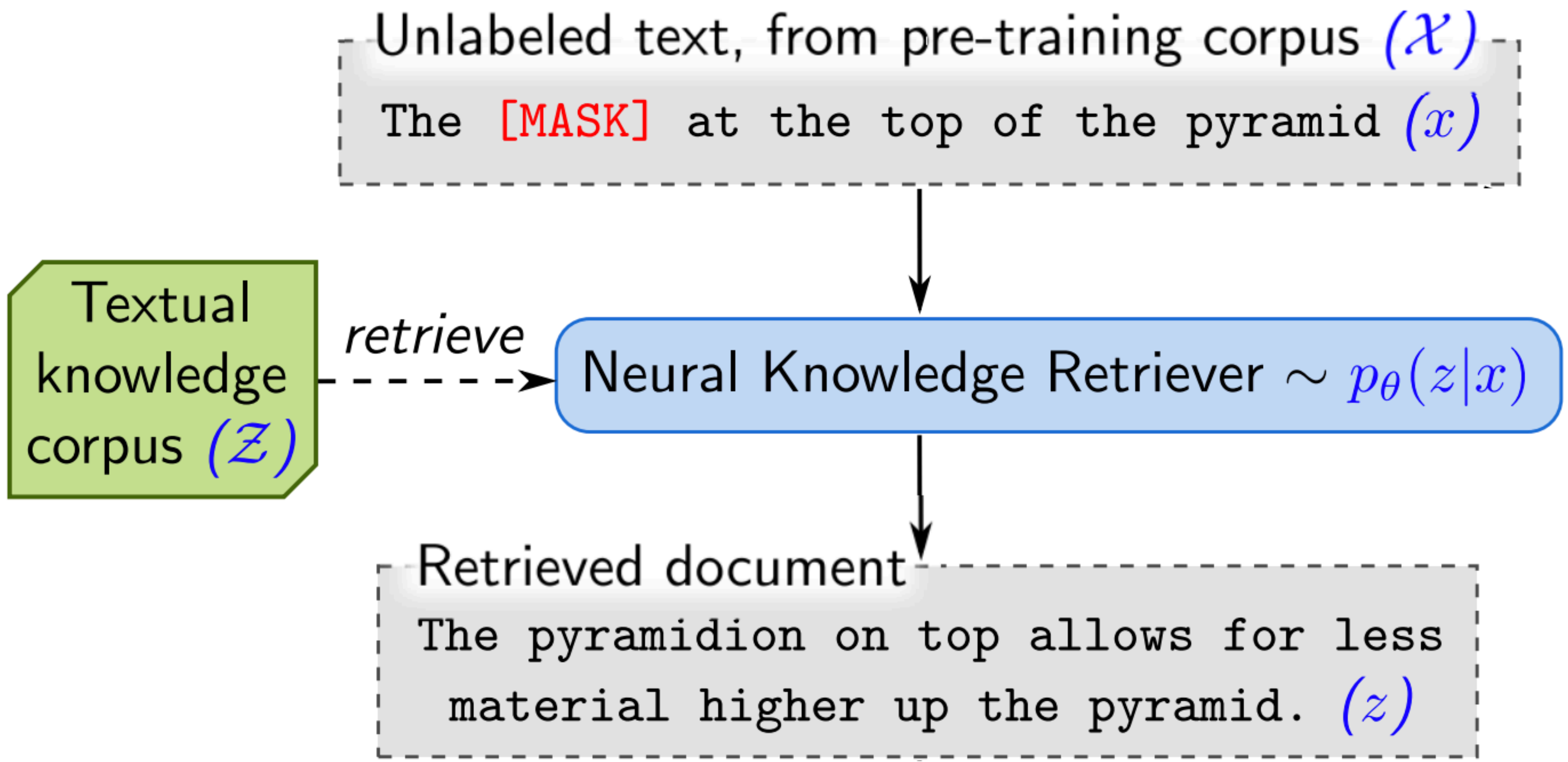
- Readily available at scale, requires no processing
- We have powerful methods of encoding semantics (e.g., BERT)
- However, these methods don't really work with larger units of text (e.g., books)
- Extracting relevant information from unstructured text is more difficult than it is with KGs

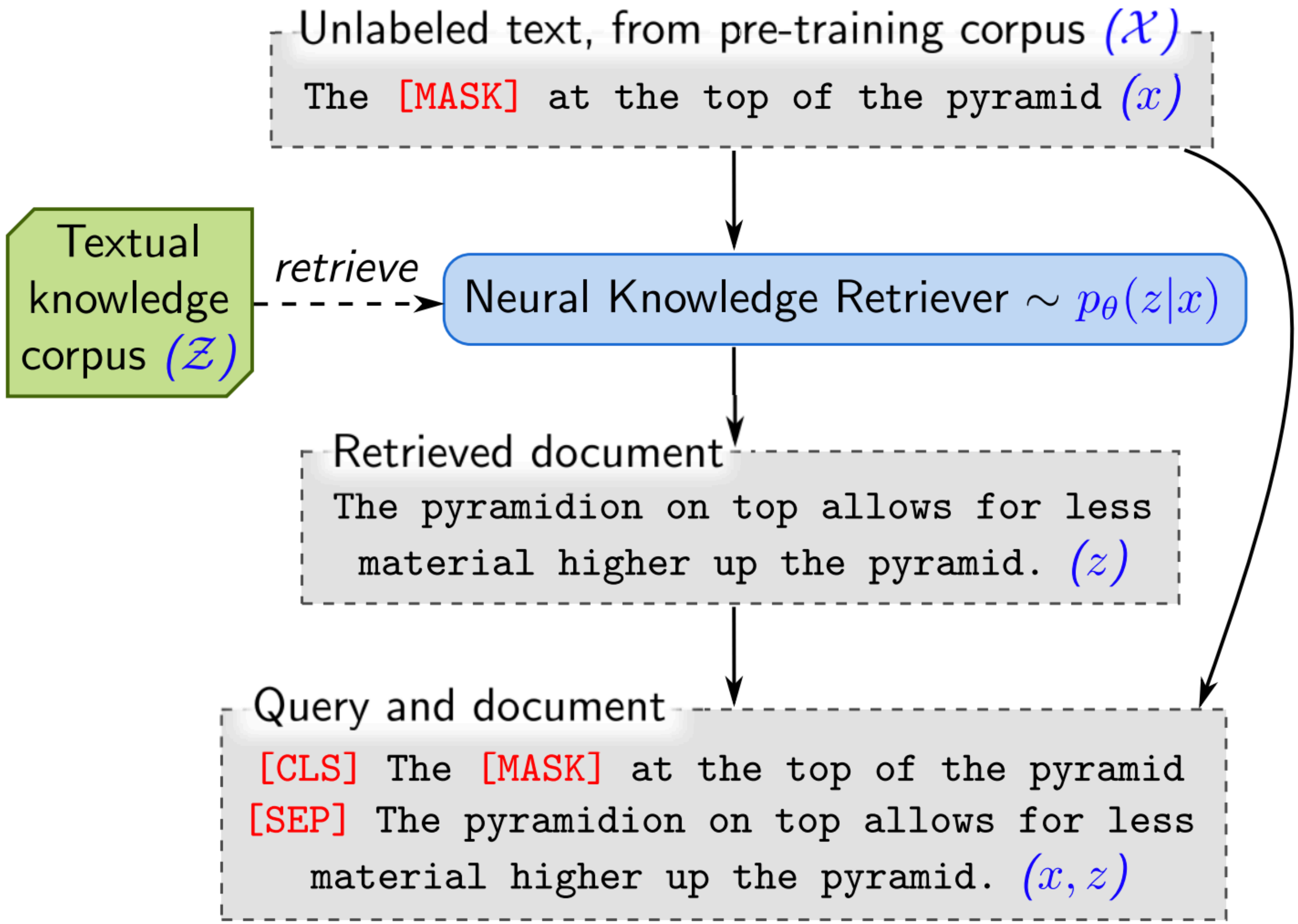
Unlabeled text, from pre-training corpus $(\mathcal{X})$

`The [MASK] at the top of the pyramid` (x)

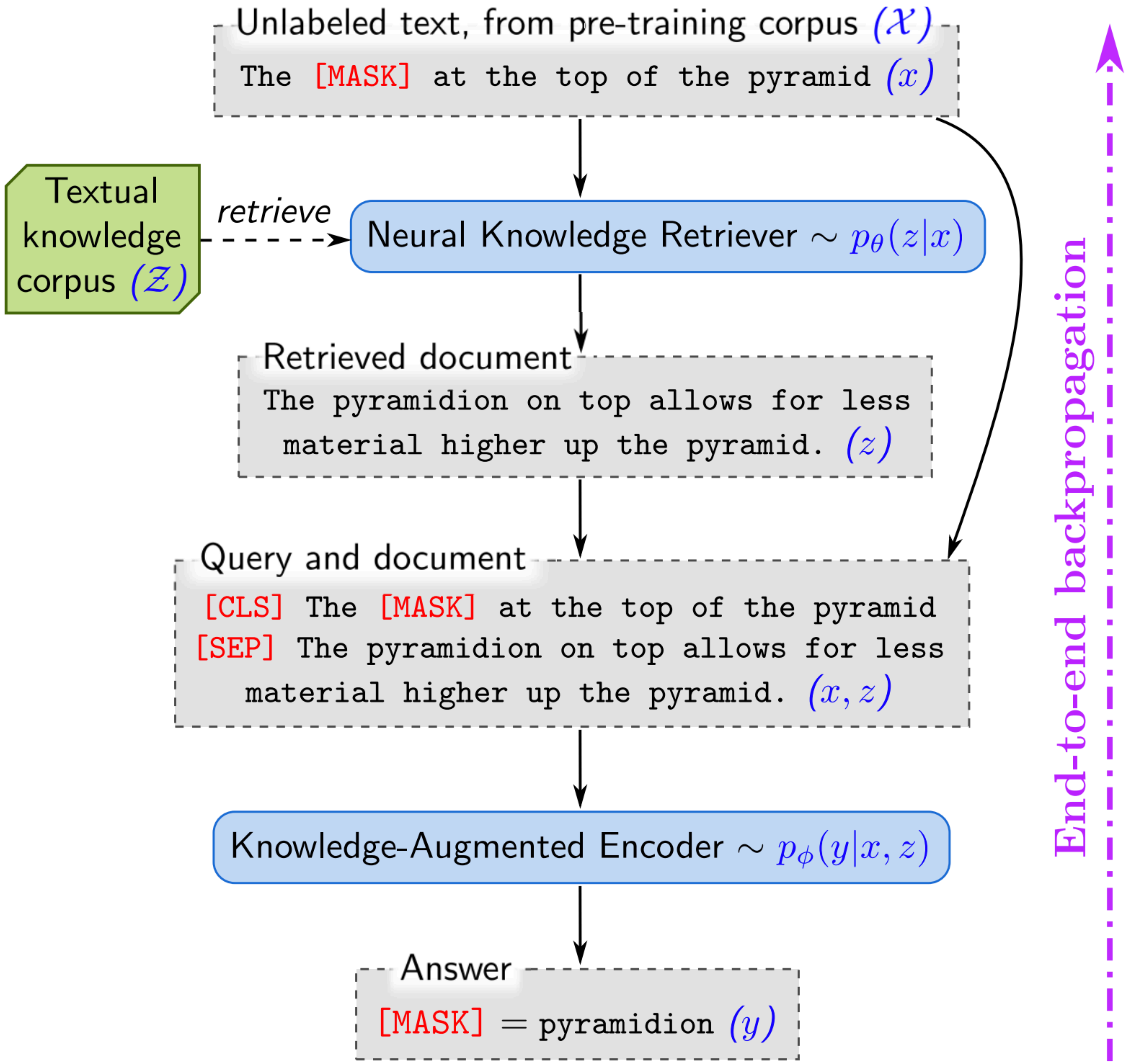


Unlabeled text, from pre-training corpus $(\mathcal{X})$
The [MASK] at the top of the pyramid (x)
Textual knowledge corpus $(\mathcal{Z})$
retrieve
Neural Knowledge Retriever $\sim p_\theta(z|x)$

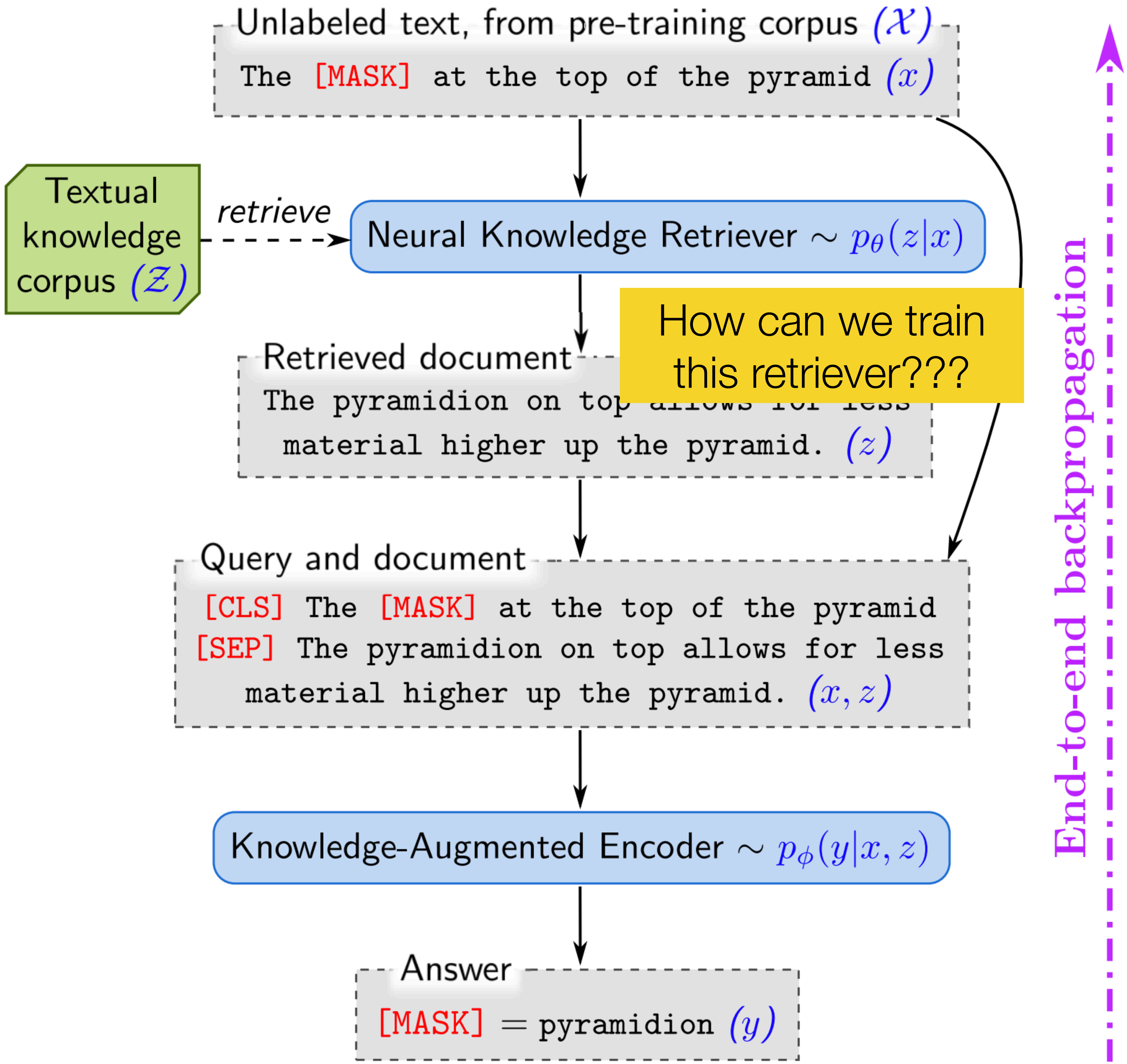

Unlabeled text, from pre-training corpus $(\mathcal{X})$
The [MASK] at the top of the pyramid (x)
Textual knowledge corpus $(\mathcal{Z})$
retrieve
Neural Knowledge Retriever $\sim p_\theta(z|x)$
Retrieved document
The pyramidion on top allows for less material higher up the pyramid. (z)



Unlabeled text, from pre-training corpus $(\mathcal{X})$
The [MASK] at the top of the pyramid (x)
Textual knowledge corpus $(\mathcal{Z})$
retrieve
Neural Knowledge Retriever $\sim p_\theta(z|x)$
Retrieved document
The pyramidion on top allows for less material higher up the pyramid. (z)
Query and document
[CLS] The [MASK] at the top of the pyramid [SEP] The pyramidion on top allows for less material higher up the pyramid. (x, z)



Unlabeled text, from pre-training corpus $(\mathcal{X})$
The [MASK] at the top of the pyramid (x)
Textual knowledge corpus $(\mathcal{Z})$
retrieve
Neural Knowledge Retriever $\sim p_\theta(z|x)$
Retrieved document
The pyramidion on top allows for less material higher up the pyramid. (z)
Query and document
[CLS] The [MASK] at the top of the pyramid [SEP] The pyramidion on top allows for less material higher up the pyramid. (x, z)
Knowledge-Augmented Encoder $\sim p_\phi(y|x, z)$
Answer
[MASK] = pyramidion (y)
End-to-end backpropagation



Unlabeled text, from pre-training corpus $(\mathcal{X})$
The [MASK] at the top of the pyramid (x)
Textual knowledge corpus $(\mathcal{Z})$
retrieve
Neural Knowledge Retriever $\sim p_\theta(z|x)$
How can we train this retriever???
Retrieved document
The pyramidion on top
material higher up the pyramid. (z)
Query and document
[CLS] The [MASK] at the top of the pyramid
[SEP] The pyramidion on top allows for less
material higher up the pyramid. (x, z)
Knowledge-Augmented Encoder $\sim p_\phi(y|x, z)$
Answer
[MASK] = pyramidion (y)
End-to-end backpropagation

REALM decomposes $p(y \mid x)$ into two steps: *retrieve*, then *predict*. Given an input x, we first retrieve possibly helpful documents z from a knowledge corpus $\mathcal{Z}$. We model this as a sample from the distribution $p(z \mid x)$. Then, we condition on both the retrieved z and the original input x to generate the output y—modeled as $p(y \mid z, x)$. To obtain the overall likelihood of generating y, we treat z as a latent variable and marginalize over all possible documents z, yielding

$$p(y \mid x) = \sum_{z \in \mathcal{Z}} p(y \mid z, x)\, p(z \mid x)$$

REALM decomposes $p(y \mid x)$ into two steps: *retrieve*, then *predict*. Given an input x, we first retrieve possibly helpful documents z from a knowledge corpus $\mathcal{Z}$. We model this as a sample from the distribution $p(z \mid x)$. Then, we condition on both the retrieved z and the original input x to generate the output y—modeled as $p(y \mid z, x)$. To obtain the overall likelihood of generating y, we treat z as a latent variable and marginalize over all possible documents z, yielding

Knowledge-augmented encoder

$$p(y \mid x) = \sum_{z \in \mathcal{Z}} p(y \mid z, x)\, p(z \mid x)$$

Neural knowledge retriever

Knowledge Retriever The retriever is defined using a dense inner product model:

$$p(z \mid x) = \frac{\exp f(x, z)}{\sum_{z'} \exp f(x, z')},$$
$$f(x, z) = \texttt{Embed}_{\texttt{input}}(x)^{\top} \texttt{Embed}_{\texttt{doc}}(z),$$

where $\texttt{Embed}_{\texttt{input}}$ and $\texttt{Embed}_{\texttt{doc}}$ are embedding functions that map x and z respectively to d-dimensional vectors. The *relevance score* $f(x, z)$ between x and z is defined as the inner product of the vector embeddings. The retrieval distribution is the softmax over all relevance scores.

Embed function is just BERT!

$$\texttt{join}_{\texttt{BERT}}(x) = \texttt{[CLS]}\, x \,\texttt{[SEP]}$$
$$\texttt{join}_{\texttt{BERT}}(x_1, x_2) = \texttt{[CLS]}\, x_1 \,\texttt{[SEP]}\, x_2 \,\texttt{[SEP]}$$

$$\texttt{Embed}_{\texttt{input}}(x) = \mathbf{W}_{\texttt{input}} \mathrm{BERT}_{\texttt{CLS}}(\texttt{join}_{\texttt{BERT}}(x))$$
$$\texttt{Embed}_{\texttt{doc}}(z) = \mathbf{W}_{\texttt{doc}} \mathrm{BERT}_{\texttt{CLS}}(\texttt{join}_{\texttt{BERT}}(z_{\text{title}}, z_{\text{body}}))$$

Knowledge-Augmented Encoder Given an input x and a retrieved document z, the knowledge-augmented encoder defines $p(y \mid z, x)$. We join x and z into a single sequence that we feed into a Transformer (distinct from the one used in the retriever).

$$p(y \mid z, x) = \prod_{j=1}^{J_x} p(y_j \mid z, x)$$

$$p(y_j \mid z, x) \propto \exp\left(w_j^\top \mathtt{BERT}_{\mathtt{MASK}(j)}(\mathtt{join}_{\mathtt{BERT}}(x, z_{\text{body}}))\right)$$

where $\mathtt{BERT}_{\mathtt{MASK}(j)}$ denotes the Transformer output vector corresponding to the j^{th} masked token, J_x is the total number of [MASK] tokens in x, and w_j is a learned word embedding for token y_j.

Isn't training the retriever extremely expensive?

The key computational challenge is that the marginal probability $p(y \mid x) = \sum_{z \in \mathcal{Z}} p(y \mid x, z)\, p(z \mid x)$ involves a summation over all documents z in the knowledge corpus $\mathcal{Z}$. We approximate this by instead summing over the top k documents with highest probability under $p(z \mid x)$—this is reasonable if most documents have near zero probability.

Imagine if your knowledge corpus was every article in Wikipedia… this would be super expensive without the approximation

Maximum inner product search (MIPS)

- Algorithms that *approximately* find the top-*k* documents
- Scales *sub-linearly* with the number of documents (both time and storage)
    - Shrivastava and Li, 2014 (“Asymmetric LSH…”)
- Requires precomputing the BERT embedding of every document in the knowledge corpus and then building an index over the embeddings

Need to refresh the index!

- We are training the parameters of the retriever, i.e., the BERT architecture that produces $\mathbf{Embed_{doc}(z)}$

- If we precompute all of the embeddings, the search index becomes stale when we update the parameters of the retriever

- REALM solution: asynchronously refresh the index by re-embedding all docs after a few hundred training iterations

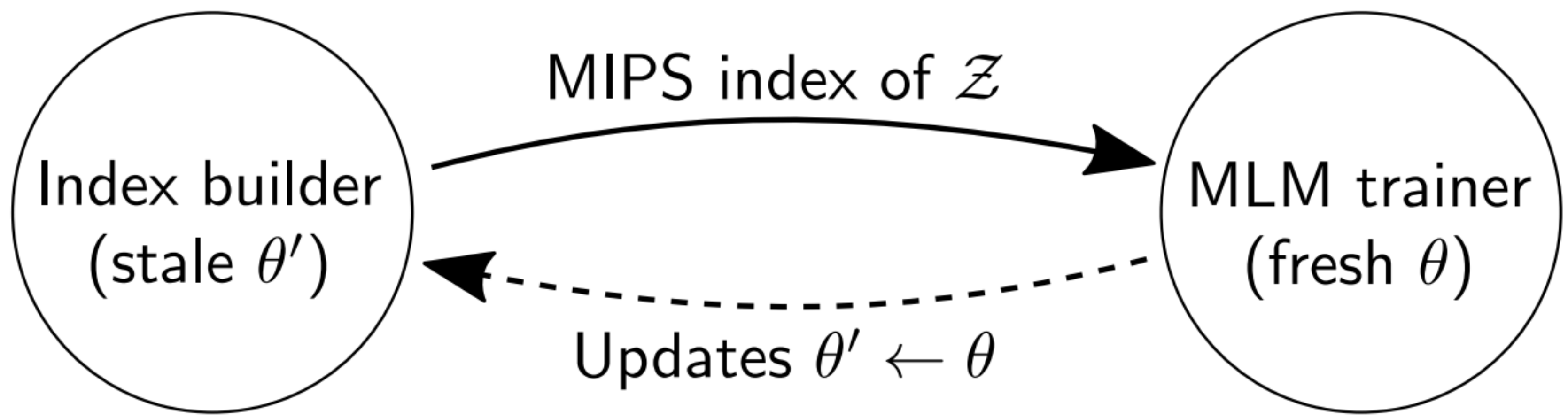


Figure 3. REALM pre-training with asynchronous MIPS refreshes.

Other tricks in REALM

- *Salient span masking*: mask out spans of text corresponding to named entities and dates

- *Null document*: always include an empty document in the top-*k* retrieved docs, allowing the model to rely on its implicit knowledge as well

Evaluation on *open-domain QA*

- Unlike SQuAD-style QA, in open-domain QA we are only given a question, not a supporting document that is guaranteed to contain the answer
- Open-domain QA generally has a large *retrieval* component, since the answer to any given question could occur anywhere in a large collection of documents

Name	Architectures	Pre-training	NQ (79k/4k)	WQ (3k/2k)	CT (1k /1k)	# params
BERT-Baseline (Lee et al., 2019)	Sparse Retr.+Transformer	BERT	26.5	17.7	21.3	110m
T5 (base) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	27.0	29.1	-	223m
T5 (large) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	29.8	32.2	-	738m
T5 (11b) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	34.5	37.4	-	11318m
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
Ours ($\mathcal{X}$ = Wikipedia, $\mathcal{Z}$ = Wikipedia)	Dense Retr.+Transformer	REALM	39.2	40.2	**46.8**	330m
Ours ($\mathcal{X}$ = CC-News, $\mathcal{Z}$ = Wikipedia)	Dense Retr.+Transformer	REALM	**40.4**	**40.7**	42.9	330m

Table 3. An example where REALM utilizes retrieved documents to better predict masked tokens. It assigns much higher probability (0.129) to the correct term, "`Fermat`", compared to BERT. (Note that the blank corresponds to 3 BERT wordpieces.)

	x:	An equilateral triangle is easily constructed using a straightedge and compass, because 3 is a ___ prime.		
(a) BERT	$p(y = \text{“}\texttt{Fermat}\text{”} \mid x)$	$= 1.1 \times 10^{-14}$	(No retrieval.)
(b) REALM	$p(y = \text{“}\texttt{Fermat}\text{”} \mid x, z)$	$= 1.0$	(Conditional probability with document $z =$"257 is … a Fermat prime. Thus a regular polygon with 257 sides is constructible with compass …")
(c) REALM	$p(y = \text{“}\texttt{Fermat}\text{”} \mid x)$	$= 0.129$	(Marginal probability, marginalizing over top 8 retrieved documents.)

Can retrieval-augmented LMs improve other tasks?

Nearest-neighbor machine translation

Test Input x	**Generated tokens** $\hat{y}_{1:i-1}$	**Representation** $q = f(x, \hat{y}_{1:i-1})$	**Target** y_i
J'ai été dans ma propre chambre.	*I have*		*?*

Khandelwal et al., 2020

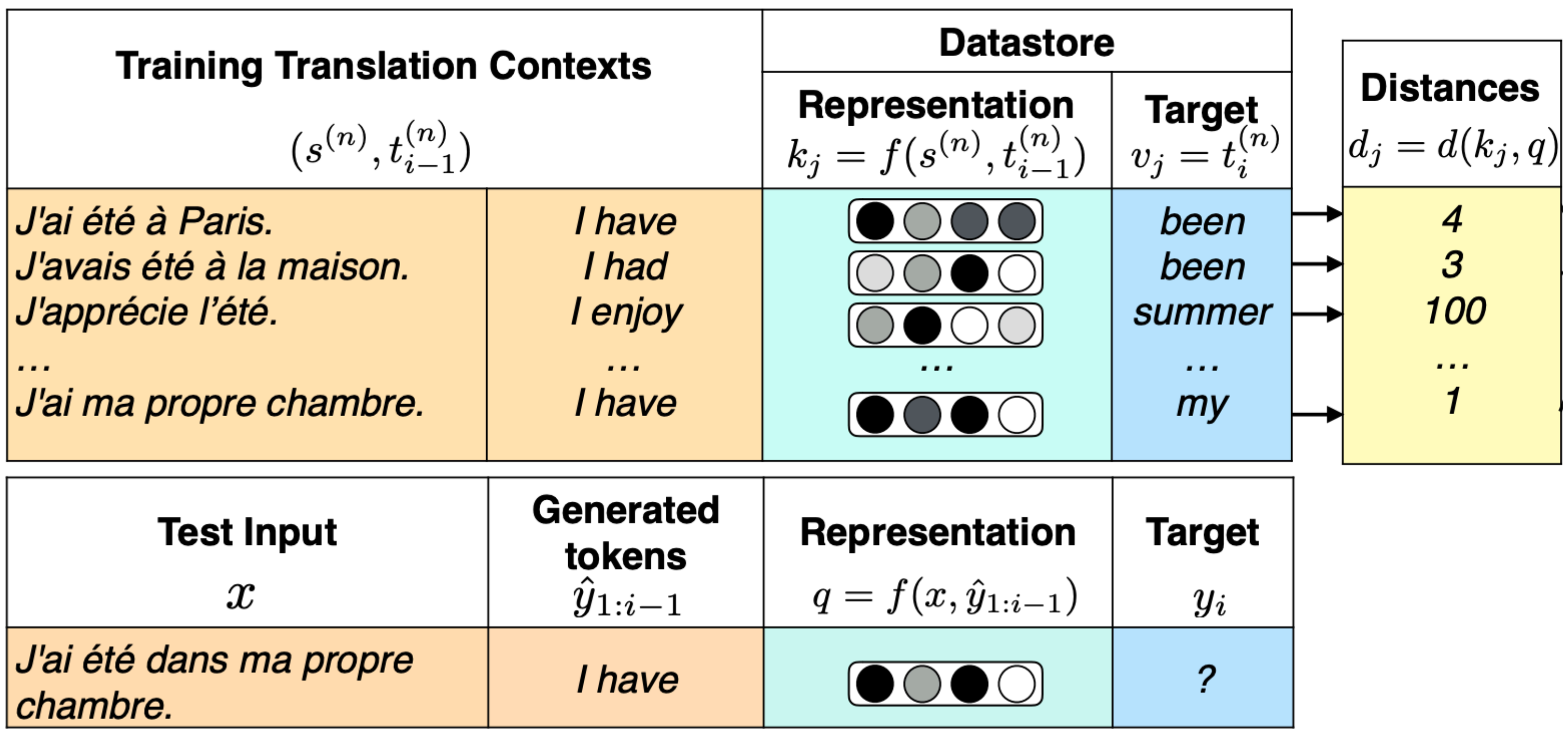
Nearest-neighbor machine translation

Training Translation Contexts $(s^{(n)}, t_{i-1}^{(n)})$		**Datastore**	
		Representation $k_j = f(s^{(n)}, t_{i-1}^{(n)})$	**Target** $v_j = t_i^{(n)}$
J'ai été à Paris.	*I have*		*been*
J'avais été à la maison.	*I had*		*been*
J'apprécie l'été.	*I enjoy*		*summer*
…	…	…	…
J'ai ma propre chambre.	*I have*		*my*

Test Input x	**Generated tokens** $\hat{y}_{1:i-1}$	**Representation** $q = f(x, \hat{y}_{1:i-1})$	**Target** y_i
J'ai été dans ma propre chambre.	*I have*		*?*

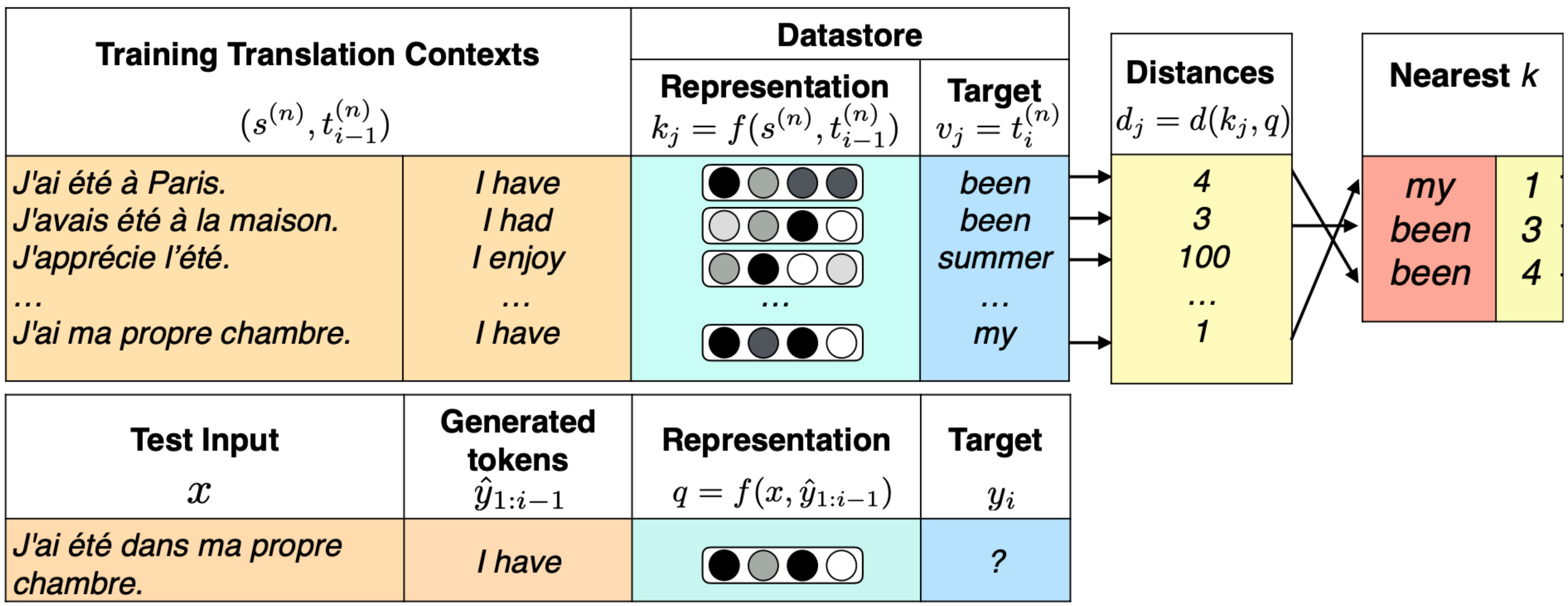
Khandelwal et al., 2020

Nearest-neighbor machine translation



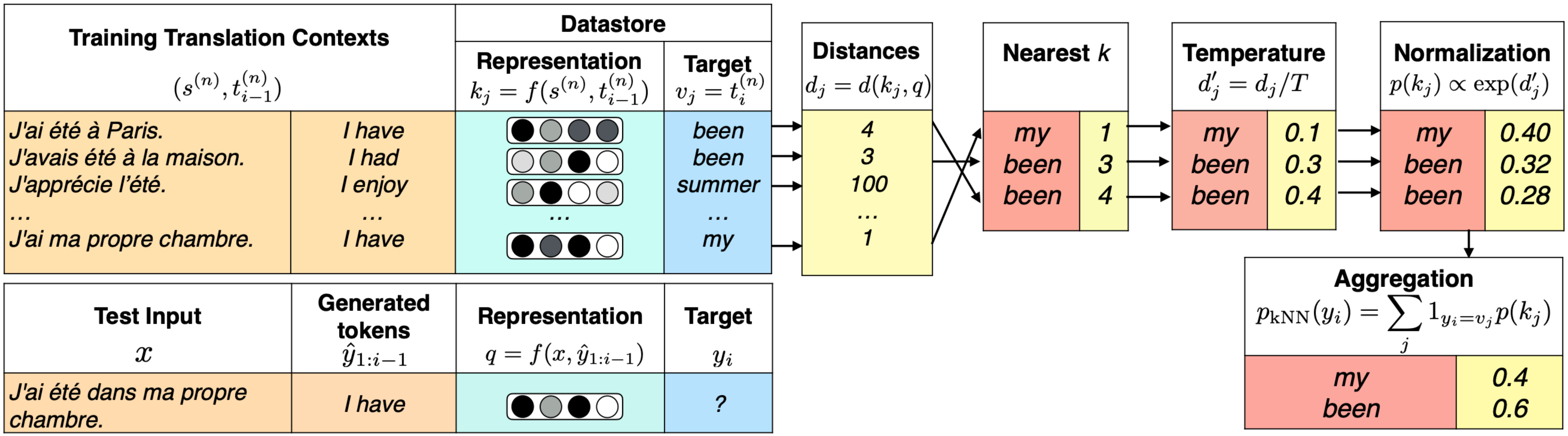
Khandelwal et al., 2020

Nearest-neighbor machine translation



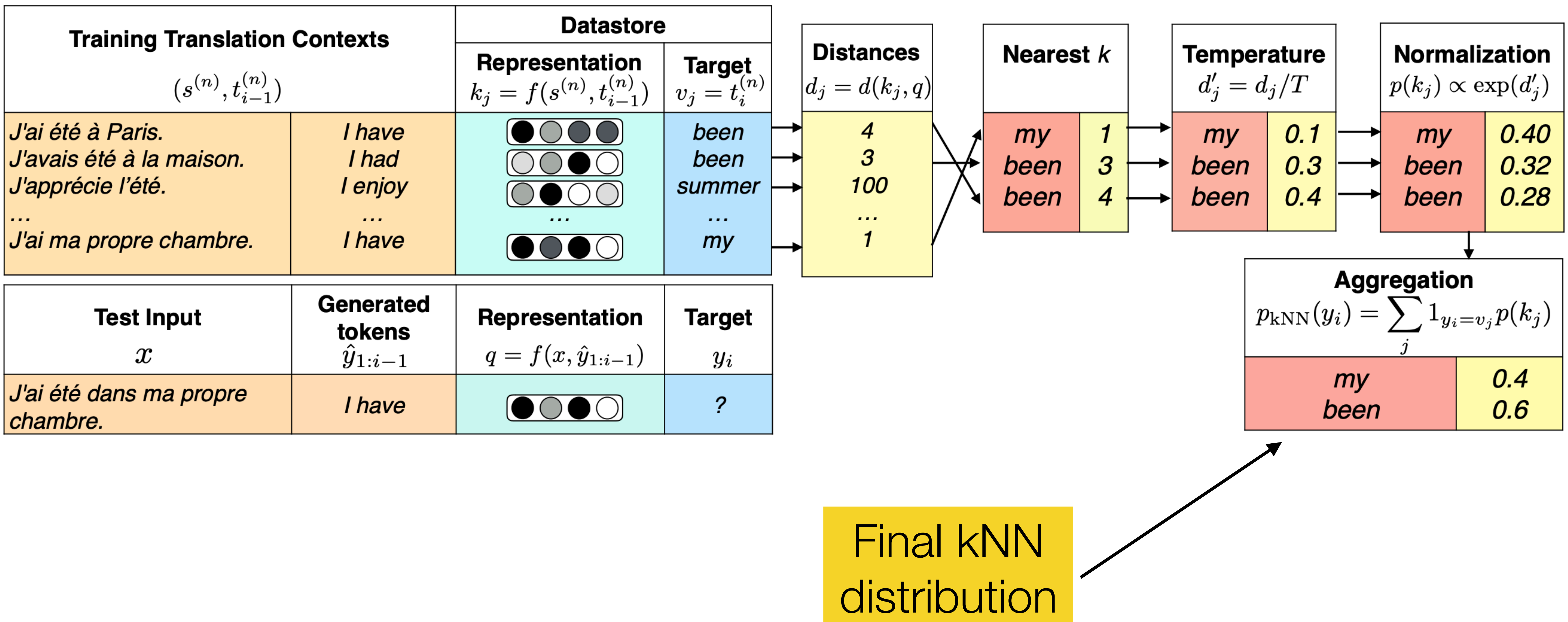
Khandelwal et al., 2020

Nearest-neighbor machine translation



Khandelwal et al., 2020

Nearest-neighbor machine translation



Khandelwal et al., 2020

Interpolate between kNN prediction and decoder's actual prediction

$$p(y_i|x, \hat{y}_{1:i-1}) = \lambda \, p_{\mathrm{kNN}}(y_i|x, \hat{y}_{1:i-1}) + (1-\lambda) \, p_{\mathrm{MT}}(y_i|x, \hat{y}_{1:i-1})$$

Final kNN distribution

Decoder's predicted distribution

Khandelwal et al., 2020

Unlike REALM, this approach doesn’t require any training! It retrieves the kNNs via L2 distance using a fast kNN library (FAISS)

This is quite expensive!

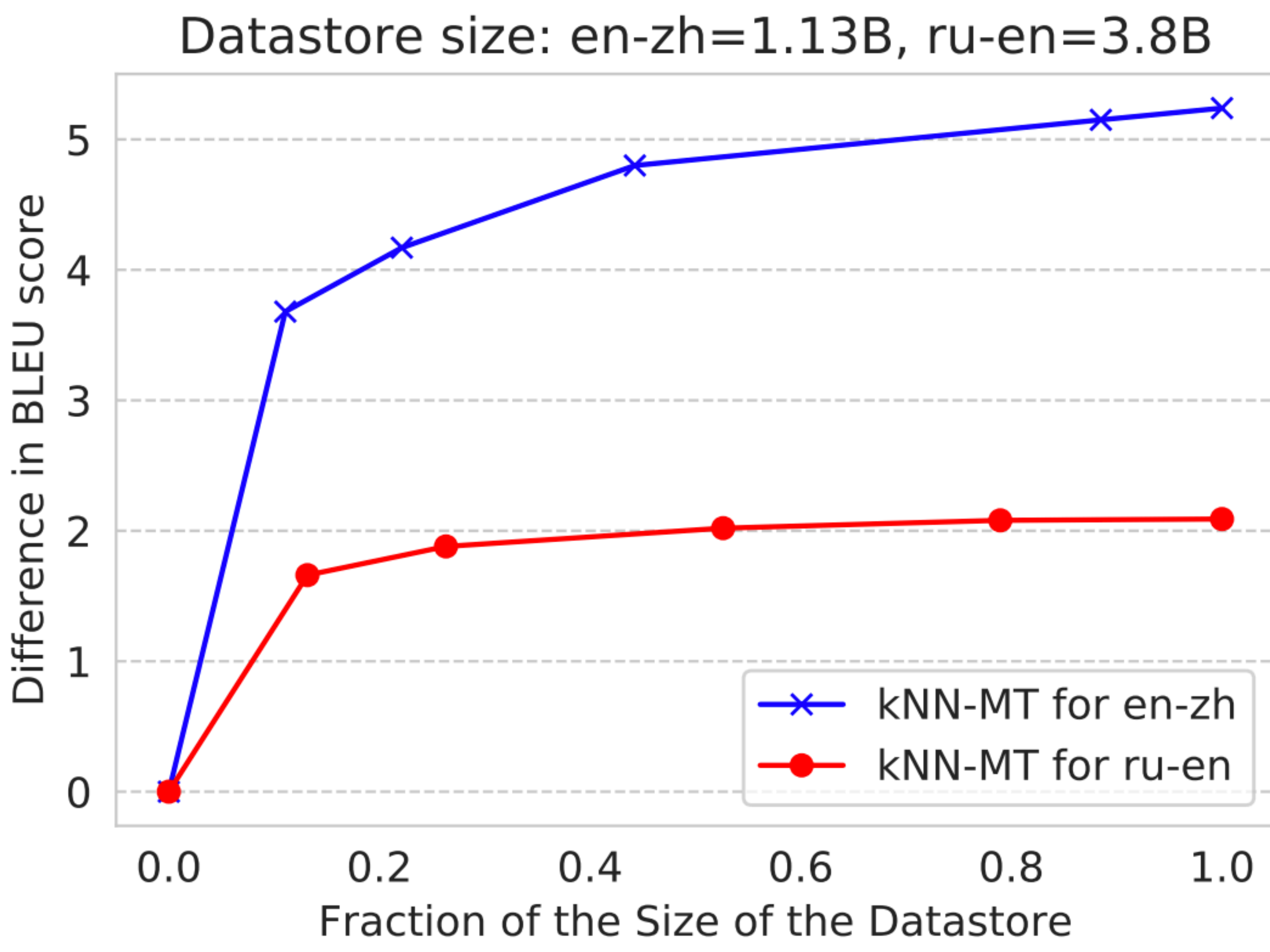
Computational Cost While kNN-MT does not add trainable model parameters, it does add some computational overhead. The primary cost of building the datastore is a single forward pass over all examples in the datastore, which is a fraction of the cost for training on the same examples for one epoch. During inference, retrieving 64 keys from a datastore containing billions of items results in a generation speed that is two orders of magnitude slower than the base MT system.

But also increases translation quality!

	de-en	ru-en	zh-en	ja-en	fi-en	lt-en	de-fr	de-cs	en-cs
Test set sizes	2,000	2,000	2,000	993	1,996	1,000	1,701	1,997	2,000
Base MT	34.45	36.42	24.23	12.79	25.92	29.59	32.75	21.15	22.78
+kNN-MT	**35.74**	**37.83**	**27.51**	13.14	26.55	29.98	**33.68**	21.62	**23.76**
Datastore Size	5.56B	3.80B	1.19B	360M	318M	168M	4.21B	696M	533M

	en-de	en-ru	en-zh	en-ja	en-fi	en-lt	fr-de	cs-de	Avg.
Test set sizes	1,997	1,997	1,997	1,000	1,997	998	1,701	1,997	-
Base MT	36.47	26.28	30.22	21.35	21.37	17.41	26.04	22.78	26.00
+kNN-MT	**39.49**	**27.91**	**33.63**	**23.23**	22.20	18.25	**27.81**	23.55	**27.40**
Datastore Size	6.50B	4.23B	1.13B	433M	375M	204M	3.98B	689M	-

Can make it faster by using a smaller datastore



Hurdles to Progress in Longform QA

Kalpesh Krishna

Aurko Roy

NAACL 2021

Posted by u/halloichbins987 5 months ago 4 2 6 6 4 4 2

34.3k

ELI5 why do humans need to eat many different kind of foods to get their vitamins etc but large animals like cows only need grass to survive?

Biology

1.7k Comments Share Save Hide Report

SORT BY **BEST**

View all comments

View discussions in 1 other community

zapawu 5 months ago 3 & 6 More

In addition to the 'efficiently breaking down grass' thing, and the 'they eat a variety of plants' thing, there's also the fact that species typically evolve the ability to make vitamins that they can't get easily in their diet. For example, humans make vitamin D because there aren't many food sources of it, but we can't make vitamin C, but can find it in food. But other species can make their own vitamin C.

It's a trade off between needing to find a variety of food and not needing the cellular machines to make more stuff.

13.9k Reply Share Report Save

ELI5 dataset crawled from Reddit & filtered by Fan et al., ACL 2019, ~275K QA pairs

LFQA involves *retrieval* as well as generation

Why do humans need to eat many kinds of foods to get their vitamins but cows only need grass to survive?

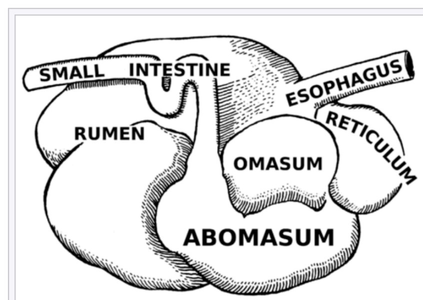
LFQA involves *retrieval* as well as generation

Why do humans need to eat many kinds of foods to get their vitamins but cows only need grass to survive?

Ruminant

From Wikipedia, the free encyclopedia

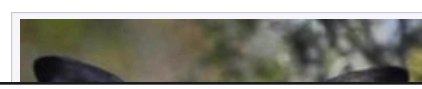
Ruminants are herbivorous mammals of the suborder Ruminantia that are able to acquire nutrients from plant-based food by fermenting it in a specialized stomach prior to digestion, principally through microbial actions. The process, which takes place in the front part of the digestive system and therefore is called foregut fermentation, typically requires the fermented ingesta (known as cud) to be regurgitated and chewed again. The process of rechewing the cud to further break down plant matter and stimulate digestion is called **rumination**.[1][2] The word "ruminant" comes from the Latin *ruminare*, which means "to chew over again".



Stylised illustration of a ruminant digestive system

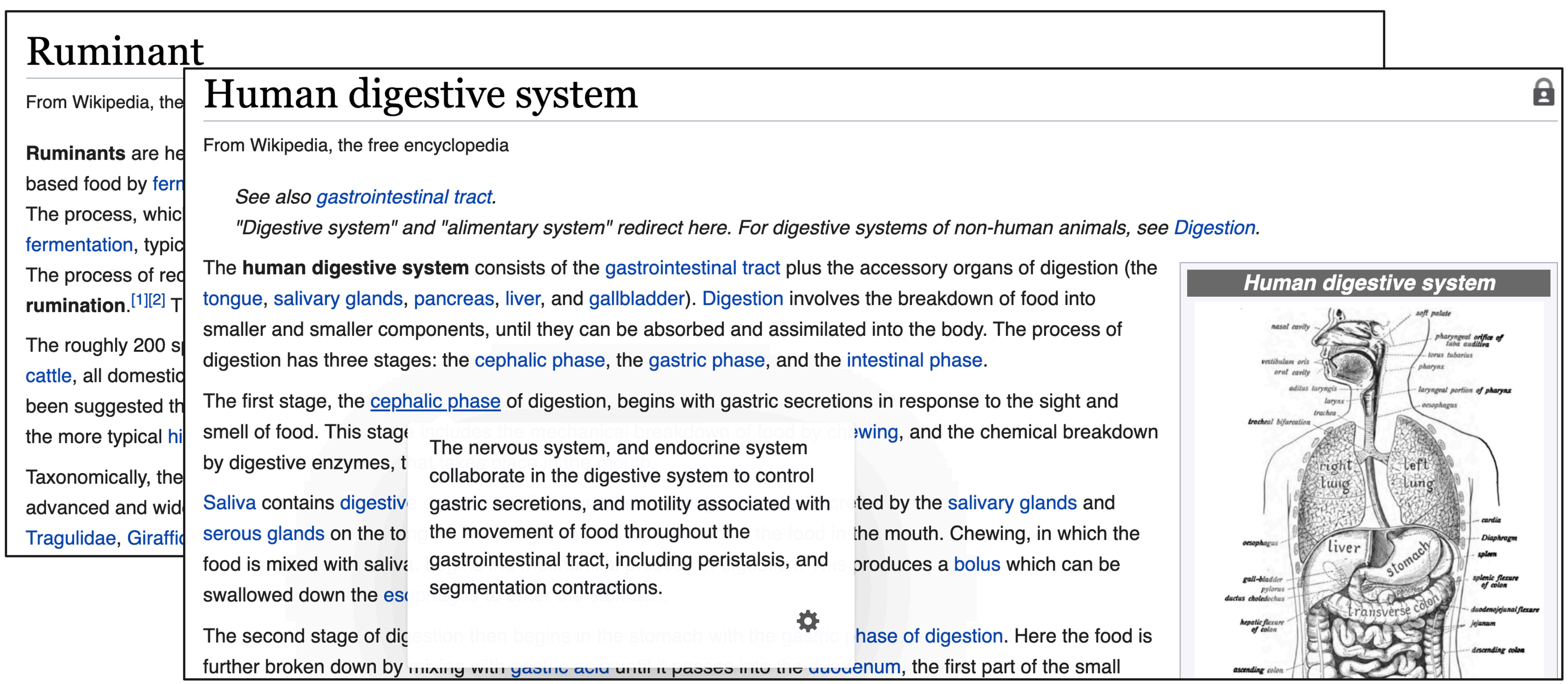
The roughly 200 species of ruminants include both domestic and wild species.[3] Ruminating mammals include cattle, all domesticated and wild bovines, goats, sheep, giraffes, deer, gazelles, and antelopes.[4] It has also been suggested that notoungulates also relied on rumination, as opposed to other atlantogenates that rely on the more typical hindgut fermentation, though this is not entirely certain.[5]

Taxonomically, the suborder Ruminantia is a lineage of herbivorous artiodactyls that includes the most advanced and widespread of the world's ungulates.[6] The suborder Ruminantia includes six different families: Tragulidae, Giraffidae, Antilocapridae, Moschidae, Cervidae, and Bovidae.[3]



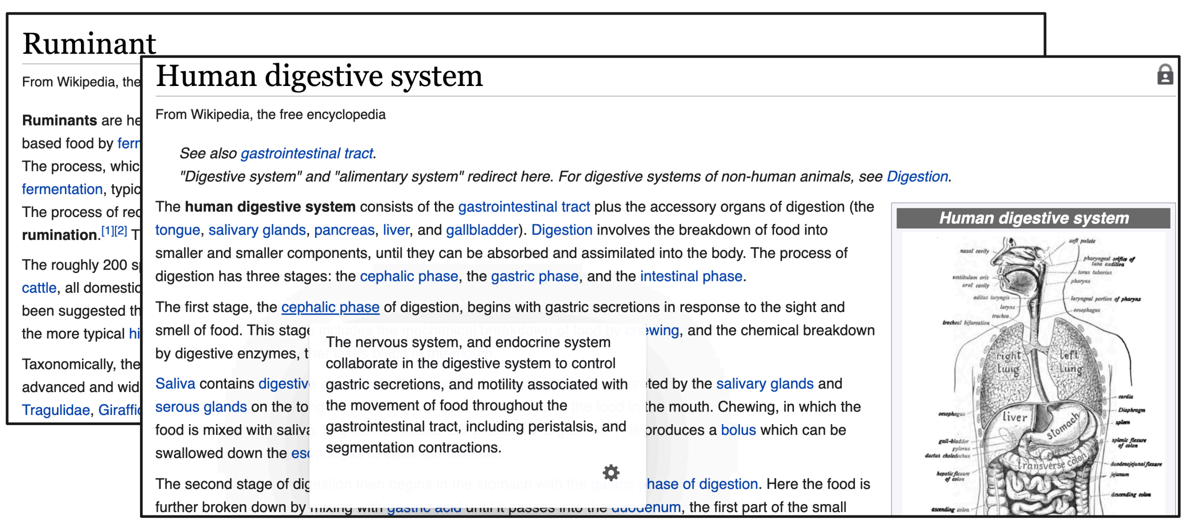
LFQA involves *retrieval* as well as generation

Why do humans need to eat many kinds of foods to get their vitamins but cows only need grass to survive?



LFQA involves *retrieval* as well as generation

Why do humans need to eat many kinds of foods to get their vitamins but cows only need grass to survive?



In addition to the "efficiently breaking down grass" thing and the "they eat a variety of plants" thing, there's also the fact that species typically evolve the ability to make vitamins that they can't easily get in their diet…

Evaluate both retrieval and generation quality

Rank	Participant team	R-Prec	Recall@5	ROUGE-L	F1	KILT-RL	KILT-F1	Last submission at
1	Anonymous Cardinal (Routing Transformer, c-REALM)	10.67	24.56	23.19	22.88	2.36	2.34	4 months ago
10	Host_23415_Team (BART) B	0.00	0.00	20.55	19.23	0.00	0.00	5 months ago
9	kilt_ (T5-base)	0.00	0.00	19.08	16.10	0.00	0.00	5 months ago
5	Metrics Test 2 (Training Set Retrieval (top 1))	0.00	0.00	18.66	21.62	0.00	0.00	3 months ago
4	dzorlu (multi-task small)	0.00	0.00	17.67	16.40	0.00	0.00	8 days ago
2	Host_23415_Team (BART + DPR) B	10.67	26.92	17.41	17.88	1.90	2.01	5 months ago

KILT benchmark (Petroni et al., 2020)

We build a state-of-the-art model for this task

Model	*Retrieval* RPr.	*Retrieval* R@5	*Generation* F1	*Generation* R-L	KRL
T5-base	0.0	0.0	16.1	19.1	0.0
BART	0.0	0.0	19.2	20.6	0.0
RAG	**11.0**	22.9	14.5	14.1	1.7
BART + DPR	10.7	**26.9**	17.9	17.4	1.9
$p = 0.9$					
RT + REALM	6.7	15.5	25.1	21.5	1.4
RT + C-REALM	10.2	24.4	**25.4**	21.5	2.1
$p = 0.6$					
RT + REALM	6.7	15.7	23.1	**23.4**	1.5
RT + C-REALM	10.7	24.6	22.9	23.2	**2.4**

However, it doesn't seem to even use the retrievals!

	R-L	vs predicted retr.		vs random retr.	
		1-g	2-g	1-g	2-g
Predicted	$\boxed{24.42}$	52.3	9.0	38.8	3.9
Random	$\boxed{24.20}$	51.2	8.5	38.5	3.9
Gold Ans	-	54.1	9.1	40.2	3.8

Human preference evaluation shows slight preference for generations grounded in *random* retrieval!

A	B	Prefer A	Prefer B	Tie
For $p = 0.9$				
pred.	random	31% (52)	$\boxed{37\%}$ (63)	32% (54)
pred.	gold ans.	17% (49)	**72%** (203)	11% (31)

several issues with the dataset and evaluation

(a) Many held-out questions are paraphrased in the training set. Best answer to similar train questions gets 27.4 ROUGE-L

Val Q: Can you protect electronics from EMPs/solar flares? If so, how?

Train Q1: How does an EMP ruin electronics? What does it do? How would they be fixed? Can It be protected against? How?

Train Q2: If Earth were hit with a massive EMP, would all of our currently technology be completely unusable permanently?

Train Q3: Whenever a electromagnetic pulse (EMP) is released what does it do to electronics to disable them?

Train Q4: If earth was hit with an EMP, could we ever restore electricity? If not, why?

Train Q5: What are solar flares and why does it impact our electronics?

Train Q6. When an EMP goes off, can the electronics affected be replaced?

(a) Many held-out questions are paraphrased in the training set. Best answer to similar train questions gets 27.4 ROUGE-L

Val Q: Can you protect electronics from EMPs/solar flares? If so, how?

Train Q1: How does an EMP ruin electronics? What does it do? How would they be fixed? <u>Can It be protected against? How?</u>

Train Q2: If Earth were hit with a massive EMP, would all of our currently technology be completely unusable permanently?

Train Q3: Whenever a electromagnetic pulse (EMP) is released what does it do to electronics to disable them?

Train Q4: If earth was hit with an EMP, could we ever restore electricity? If not, why?

Train Q5: What are solar flares and why does it impact our electronics?

Train Q6. When an EMP goes off, can the electronics affected be replaced?

This means the model doesn't need to rely on retrieving external documents, because it already sees the answers to most questions during training. Our analysis shows that **81%** of val questions have an exact paraphrase in the training set.

(a) Many held-out questions are paraphrased in the training set. Best answer to similar train questions gets 27.4 ROUGE-L

(b) Simply retrieving answers to random unrelated training questions yields relatively high ROUGE-L, while actual gold answers underperform generations

Val Q: Can you protect electronics from EMPs/solar flares? If so, how?

Train Q1: How does an EMP ruin electronics? What does it do? How would they be fixed? <u>Can It be protected against? How?</u>

Train Q2: If Earth were hit with a massive EMP, would all of our currently technology be completely unusable permanently?

Train Q3: Whenever a electromagnetic pulse (EMP) is released what does it do to electronics to disable them?

Train Q4: If earth was hit with an EMP, could we ever restore electricity? If not, why?

Train Q5: What are solar flares and why does it impact our electronics?

Train Q6. When an EMP goes off, can the electronics affected be replaced?

Random Train Ans, 19.4 ROUGE-L

The fast lane/slow lane is a bit of a misnomer. It gives the impression that new, faster lanes are being built. In reality, normal speed will be...

Gold Answer, 18.6 ROUGE-L

I'll start with the grounding question, because that's the easiest to answer: Doesn't help a bit. All that matters is that the metal container is conductive and doesn't have gaps...completely seal your Faraday cage. Consider soldering the lid on to that paint can... look at little baggie it comes in. Sealed mylar. That protected that chip from air travel at 35,000 feet, land travel through rural, urban, and suburban areas, and all the electromagnetic radiation that the trip entails... No lead shielding. No safes....

(a) Many held-out questions are paraphrased in the training set. Best answer to similar train questions gets 27.4 ROUGE-L

(b) Simply retrieving answers to random unrelated training questions yields relatively high ROUGE-L, while actual gold answers underperform generations

Val Q: Can you protect electronics from EMPs/solar flares? If so, how?

Train Q1: How does an EMP ruin electronics? What does it do? How would they be fixed? Can It be protected against? How?

Train Q2: If Earth were hit with a massive EMP, would all of our currently technology be completely unusable permanently?

Train Q3: Whenever a electromagnetic pulse (EMP) is released what does it do to electronics to disable them?

Train Q4: If earth was hit with an EMP, could we ever restore electricity? If not, why?

Train Q5: What are solar flares and why does it impact our electronics?

Train Q6. When an EMP goes off, can the electronics affected be replaced?

Random Train Ans, 19.4 ROUGE-L

The fast lane/slow lane is a bit of a misnomer. It gives the impression that new, faster lanes are being built. In reality, normal speed will be...

Gold Answer, 18.6 ROUGE-L

I'll start with the grounding question, because that's the easiest to answer: Doesn't help a bit. All that matters is that the metal container is conductive and doesn't have gaps...completely seal your Faraday cage. Consider soldering the lid on to that paint can... look at little baggie it comes in. Sealed mylar. That protected that chip from air travel at 35,000 feet, land travel through rural, urban, and suburban areas, and all the electromagnetic radiation that the trip entails... No lead shielding. No safes....

ROUGE-L is clearly not a good metric for LFQA, as *gold* answers frequently underperform randomly selected unrelated answers

(a) Many held-out questions are paraphrased in the training set. Best answer to similar train questions gets 27.4 ROUGE-L

(b) Simply retrieving answers to random unrelated training questions yields relatively high ROUGE-L, while actual gold answers underperform generations

(c) Conditioning answer generation on *random* documents instead of relevant ones **does not** measurably impact its factual correctness. Longer outputs get higher ROUGE-L

(d) Annotators find it difficult to judge long answers (with repetition) & correctness of technical content

Val Q: Can you protect electronics from EMPs/solar flares? If so, how?

Train Q1: How does an EMP ruin electronics? What does it do? How would they be fixed? <u>Can It be protected against? How?</u>

Train Q2: If Earth were hit with a massive EMP, would all of our currently technology be completely unusable permanently?

Train Q3: Whenever a electromagnetic pulse (EMP) is released what does it do to electronics to disable them?

Train Q4: If earth was hit with an EMP, could we ever restore electricity? If not, why?

Train Q5: What are solar flares and why does it impact our electronics?

Train Q6. When an EMP goes off, can the electronics affected be replaced?

Random Train Ans, 19.4 ROUGE-L

The fast lane/slow lane is a bit of a misnomer. It gives the impression that new, faster lanes are being built. In reality, normal speed will be...

Gold Answer, 18.6 ROUGE-L

I'll start with the grounding question, because that's the easiest to answer: Doesn't help a bit. All that matters is that the metal container is conductive and doesn't have gaps...completely seal your Faraday cage. Consider soldering the lid on to that paint can... look at little baggie it comes in. Sealed mylar. That protected that chip from air travel at 35,000 feet, land travel through rural, urban, and suburban areas, and all the electromagnetic radiation that the trip entails... No lead shielding. No safes....

Generation using *predicted* retrievals, 19.0 ROUGE-L

Yes, you can shield them. But it's a slow process... Also, the equipment that's powered by them is a lot more expensive than you'd think, so it's hard to make sure that you're not just shielding them from your remote control.

Generation using *random* retrievals, 24.8 ROUGE-L

Yes, you absolutely can, in fact you can build a Faraday cage around your electronics, and protect them from solar flares... This is what is done with the Faraday cage around your electronics, which is the problem. The reason it is expensive is because it requires a huge amount of power and is expensive to replace... designed to shield your electronics from solar flares, you will have to pay for the protection. This is because you have to buy a piece of equipment that is designed to shield your electronics from solar flares, and that is expensive. ... This is also expensive, but not as expensive as the protection you need to shield your electronics from solar flares... designed to be as cheap as possible...

Lower and upper bounds demonstrate the issues

Scheme	Validation F1	Validation R-L	Test F1	Test R-L
random train answer (↓)	17.8	$\boxed{16.2}$	17.1	15.5
copy input (↓)	16.6	20.0	14.8	16.9
RAG (2020c)	17.2	16.1	14.5	14.1
BART + DPR (2020)	18.8	18.5	17.9	17.4
longest top-1 train answer	25.2	20.7	21.6	18.7
longest top-7 train answer	26.9	21.1	22.0	18.5
RT + c-REALM (ours)	25.6	24.4	22.9	23.2
best top-1 train answer (↑)	25.9	22.4	-	-
best top-7 train answer (↑)	31.5	28.5	-	-
longest gold answer (↑)	26.7	21.2	-	-
best gold answer (↑)	29.5	$\boxed{26.2}$	-	-

Upper (↑) and lower (↓) bounds

We have a long way to go…

Q: Why are almost all boats white?
A: Boats are generally painted white for obvious reasons, as they are inexpensive and easy to clean. Ships are also often painted to provide a visible visual contrast to the sea. Boats also often have high visibility ports to allow for easy international trade. In addition, many boats have an air-breathing function that makes them easier to spot and stay afloat, which is a good thing if you're a pirate or trying to get into some country's waters. Lastly, many ships are built in the last few decades, so they are not as exotic looking as they were thirty years ago. And lastly, many ships are made out of materials that are easy to work with, and so many of them are painted white as a practical matter. There are many, many more reasons for why ships are white, but those are the ones I know of. I'm sure there are many other reasons, but those are the ones I know of. Edit: I forgot to mention that the other answers about visibility and easy to see ports are also true. Ships are painted to appear to be larger, and more expensive. That makes them appear more impressive, and so they are more likely to be bought by people who want to show off their wealth and status.