Carnegie Mellon University

# Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

Zengliang Zhu, Yuchen Zhang, Jordan Peng

Carnegie Mellon University

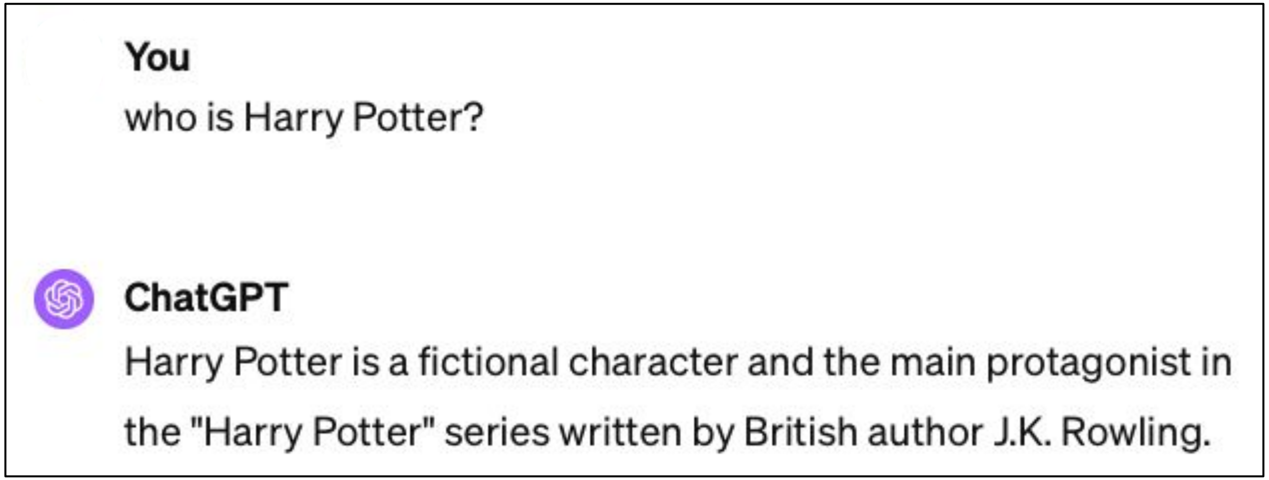
## Outline

- **Motivation**
- **Previous works**: ORQA, REALM, kNN-LM
- **Method**
- **Experiment**
- **Result**

## **Motivation :** Memorization

- **Memorization** makes NLG (Natural Language Generation) systems **better**.

- Especially in QA



**You**
who is Harry Potter?

**ChatGPT**
Harry Potter is a fictional character and the main protagonist in the "Harry Potter" series written by British author J.K. Rowling.

## **Motivation :** Memorization

**How?**

Models gain memory during training.

- If there’s enough original text of **Harry Potter** in the training data, the model can **recite it from memory**.

| **Prompt** | **Llama-7b-chat-hf** |
|---|---|
| Harry Potter’s two best friends are | Ron Weasley and Hermione Granger. In the series... |
| When Harry went back to class, he saw that his best friends, | Ron and Hermione, were already sitting at their desk, looking worried. ”What’s wrong?”... |

Eldan, Ronen, and Mark Russinovich. "Who's Harry Potter? Approximate Unlearning in LLMs." *arXiv preprint arXiv:2310.02238* (2023).

Carnegie Mellon University

## **Motivation :** Memorization

**How?**

Models gain memory during training.

- If there’s enough original text of **Harry Potter** in the training data, the model can **recite it from memory**.

**“Parametric Memory” : inside of learned parameter**

## **Motivation :** Memorization

**How?**

Models gain memory during training.

- If there’s enough original text of **Harry Potter** in the training data, the model can **recite it from memory**.

**“Parametric Memory” : inside of learned parameter**

- **Downside:**
    - Updating memory means updating **training data** and **retraining.**

"Who is the US President up to 2024?"

Carnegie Mellon University

## **Motivation :** Memorization

**How?**

Models gain memory during training.

- If there's enough original text of **Harry Potter** in the training data, the model can **recite it from memory**.

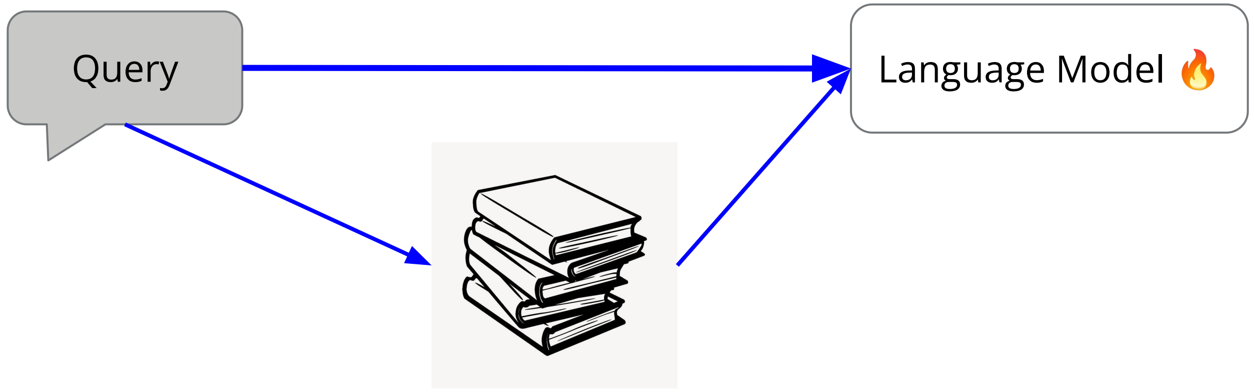
**"Parametric Memory" : inside of learned parameter**

- **Downside:**
  - Updating memory means updating **training data** and **retraining.**
  - **Black box**: Can't tell if the output is based on memorization or hallucination

## **Motivation :** Non-Parametric Memory

**How?**

Retrieve external memory on the fly



Carnegie Mellon University

## **Motivation :** Non-Parametric Memory

**How?**

Retrieve external memory on the fly

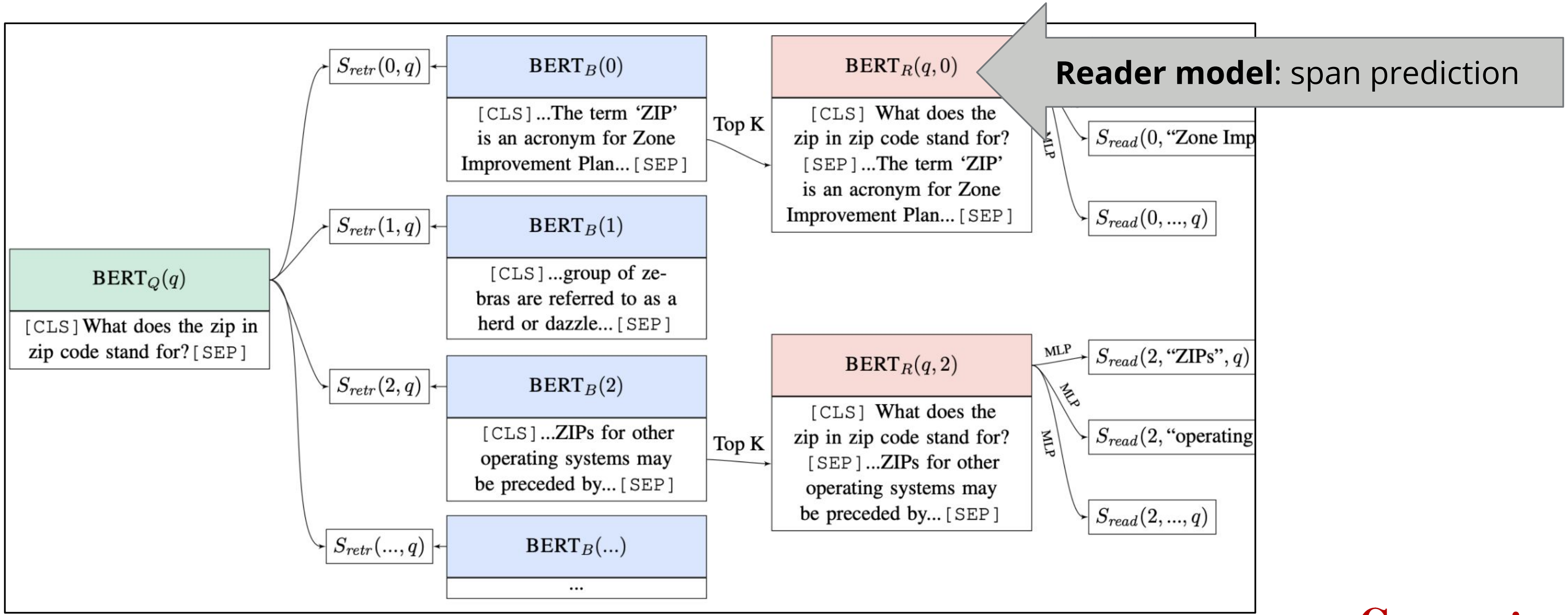
**Previous works:**

Retrieve without LM head: ***ORQA***

Retrieval-augmentation during pre-training: ***REALM***

Retrieval-augmentation after pre-training: ***kNN-LM***, ***RAG*** (this work)

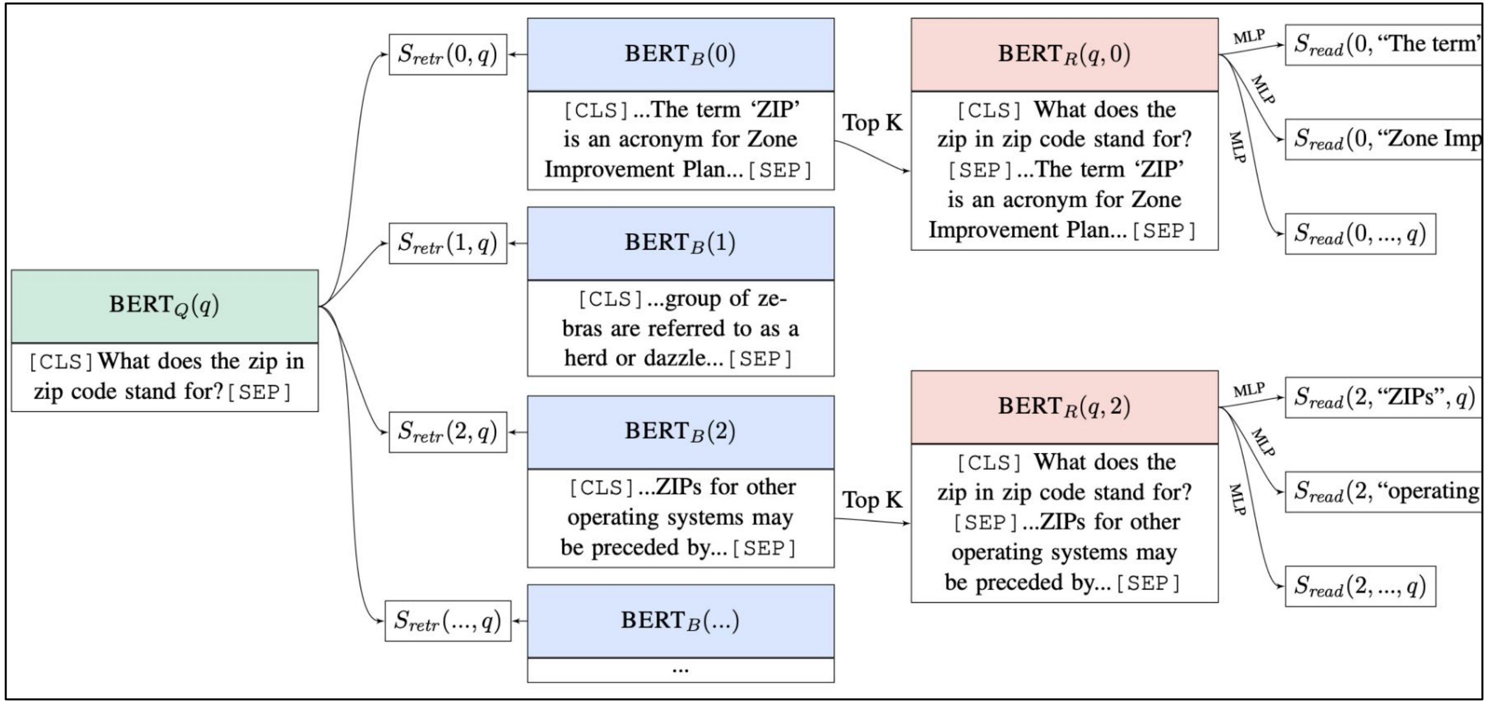
# **Retrieve without LM head :** ORQA[1]


Reader model: span prediction
BERT$_Q(q)$
[CLS] What does the zip in zip code stand for? [SEP]
$S_{retr}(0,q)$
$S_{retr}(1,q)$
$S_{retr}(2,q)$
$S_{retr}(\dots,q)$
BERT$_B(0)$
[CLS] ...The term 'ZIP' is an acronym for Zone Improvement Plan... [SEP]
BERT$_B(1)$
[CLS] ...group of zebras are referred to as a herd or dazzle... [SEP]
BERT$_B(2)$
[CLS] ...ZIPs for other operating systems may be preceded by... [SEP]
BERT$_B(\dots)$
...
Top K
Top K
BERT$_R(q,0)$
[CLS] What does the zip in zip code stand for? [SEP] ...The term 'ZIP' is an acronym for Zone Improvement Plan... [SEP]
BERT$_R(q,2)$
[CLS] What does the zip in zip code stand for? [SEP] ...ZIPs for other operating systems may be preceded by... [SEP]
MLP
$S_{read}(0,$ "Zone Imp
$S_{read}(0,\dots,q)$
$S_{read}(2,$ "ZIPs"$,q)$
$S_{read}(2,$ "operating
$S_{read}(2,\dots,q)$

[1] Lee, Kenton, Ming-Wei Chang, and Kristina Toutanova. "Latent retrieval for weakly supervised open domain question answering." arXiv preprint arXiv:1906.00300 (2019).

Carnegie Mellon University

# Retrieve without LM head : ORQA[1]


$\text{BERT}_Q(q)$
[CLS] What does the zip in zip code stand for? [SEP]
$S_{retr}(0, q)$
$S_{retr}(1, q)$
$S_{retr}(2, q)$
$S_{retr}(..., q)$
$\text{BERT}_B(0)$
[CLS] ...The term 'ZIP' is an acronym for Zone Improvement Plan... [SEP]
$\text{BERT}_B(1)$
[CLS] ...group of zebras are referred to as a herd or dazzle... [SEP]
$\text{BERT}_B(2)$
[CLS] ...ZIPs for other operating systems may be preceded by... [SEP]
$\text{BERT}_B(...)$
...
Top K
Top K
$\text{BERT}_R(q, 0)$
[CLS] What does the zip in zip code stand for? [SEP] ...The term 'ZIP' is an acronym for Zone Improvement Plan... [SEP]
$\text{BERT}_R(q, 2)$
[CLS] What does the zip in zip code stand for? [SEP] ...ZIPs for other operating systems may be preceded by... [SEP]
MLP
MLP
MLP
MLP
MLP
MLP
$S_{read}(0, \text{"The term"}$
$S_{read}(0, \text{"Zone Imp}$
$S_{read}(0, ..., q)$
$S_{read}(2, \text{"ZIPs"}, q)$
$S_{read}(2, \text{"operating}$
$S_{read}(2, ..., q)$

**When to retrieve:**
Per question

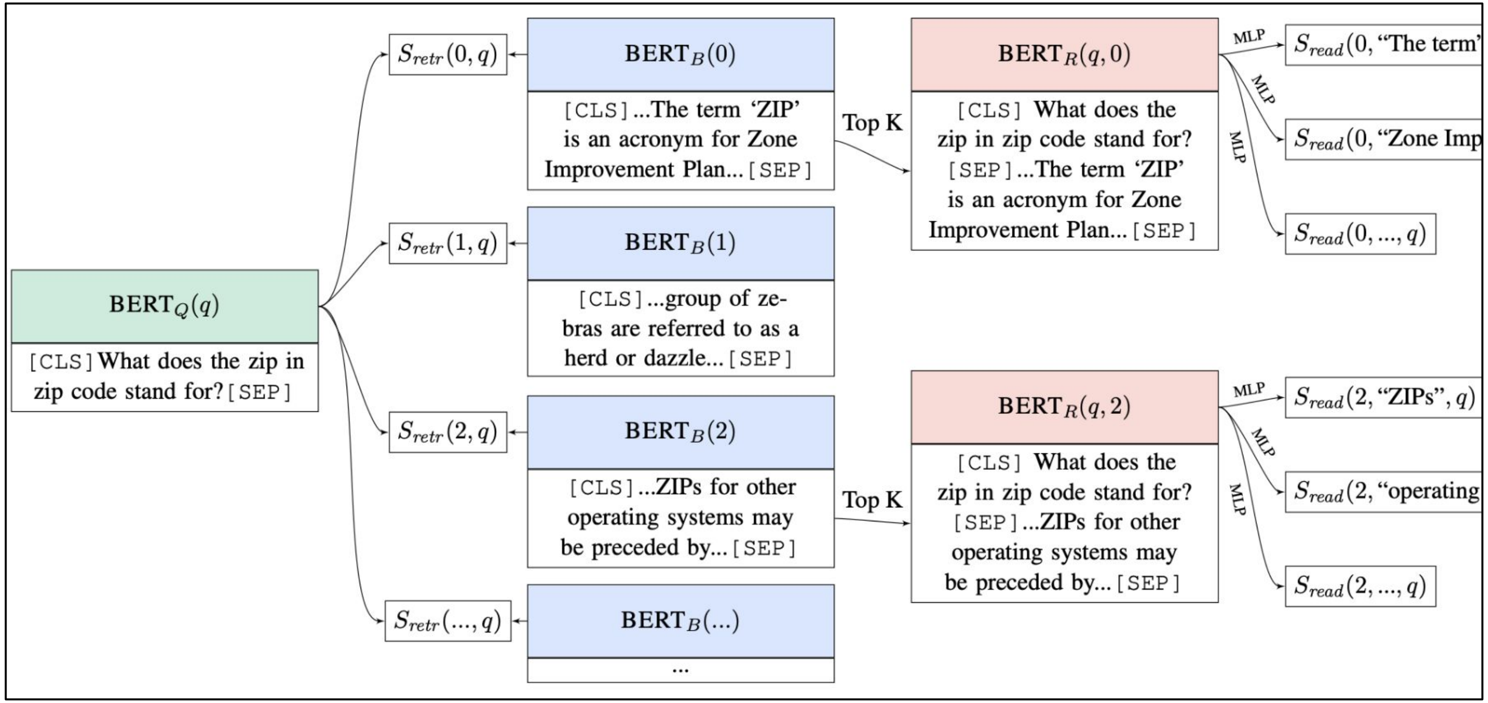
**What to retrieve:**

chunk of 288 tokens

[1] Lee, Kenton, Ming-Wei Chang, and Kristina Toutanova. "Latent retrieval for weakly supervised open domain question answering." arXiv preprint arXiv:1906.00300 (2019).

Carnegie Mellon University

## Retrieve without LM head : ORQA

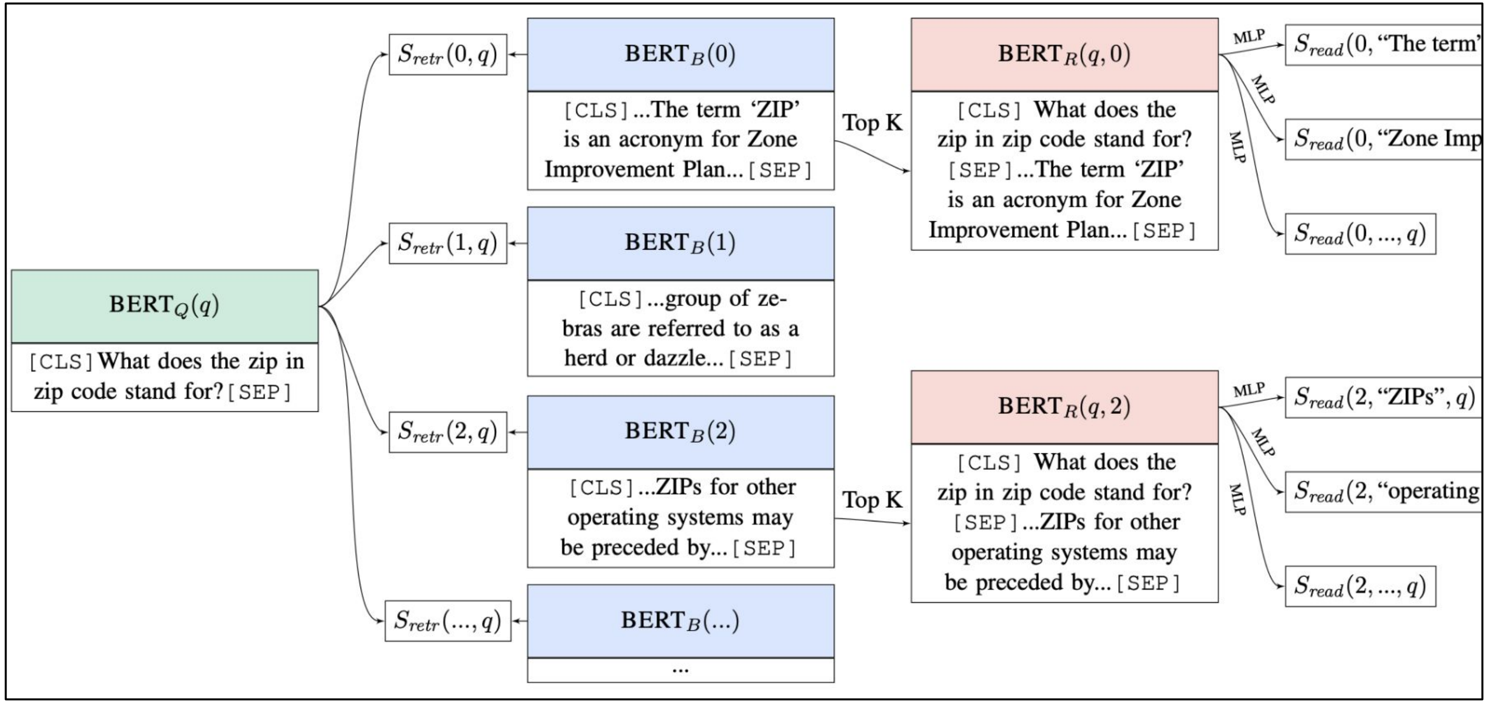

$S_{retr}(0, q)$
$\mathrm{BERT}_B(0)$
[CLS]...The term 'ZIP' is an acronym for Zone Improvement Plan...[SEP]
$S_{retr}(1, q)$
$\mathrm{BERT}_B(1)$
[CLS]...group of zebras are referred to as a herd or dazzle...[SEP]
$\mathrm{BERT}_Q(q)$
[CLS] What does the zip in zip code stand for? [SEP]
$S_{retr}(2, q)$
$\mathrm{BERT}_B(2)$
[CLS]...ZIPs for other operating systems may be preceded by...[SEP]
$S_{retr}(..., q)$
$\mathrm{BERT}_B(...)$
...
Top K
$\mathrm{BERT}_R(q, 0)$
[CLS] What does the zip in zip code stand for? [SEP]...The term 'ZIP' is an acronym for Zone Improvement Plan...[SEP]
Top K
$\mathrm{BERT}_R(q, 2)$
[CLS] What does the zip in zip code stand for? [SEP]...ZIPs for other operating systems may be preceded by...[SEP]
MLP
$S_{read}(0, \text{"The term"})$
$S_{read}(0, \text{"Zone Imp})$
$S_{read}(0, ..., q)$
$S_{read}(2, \text{"ZIPs"}, q)$
$S_{read}(2, \text{"operating})$
$S_{read}(2, ..., q)$

**When to retrieve:**
**What to retrieve:**

**How to retrieve:**
Calculate the product score of question and chunk embedding.

Rank the chunks, retrieve top-k.

## **Retrieve without LM head :** ORQA

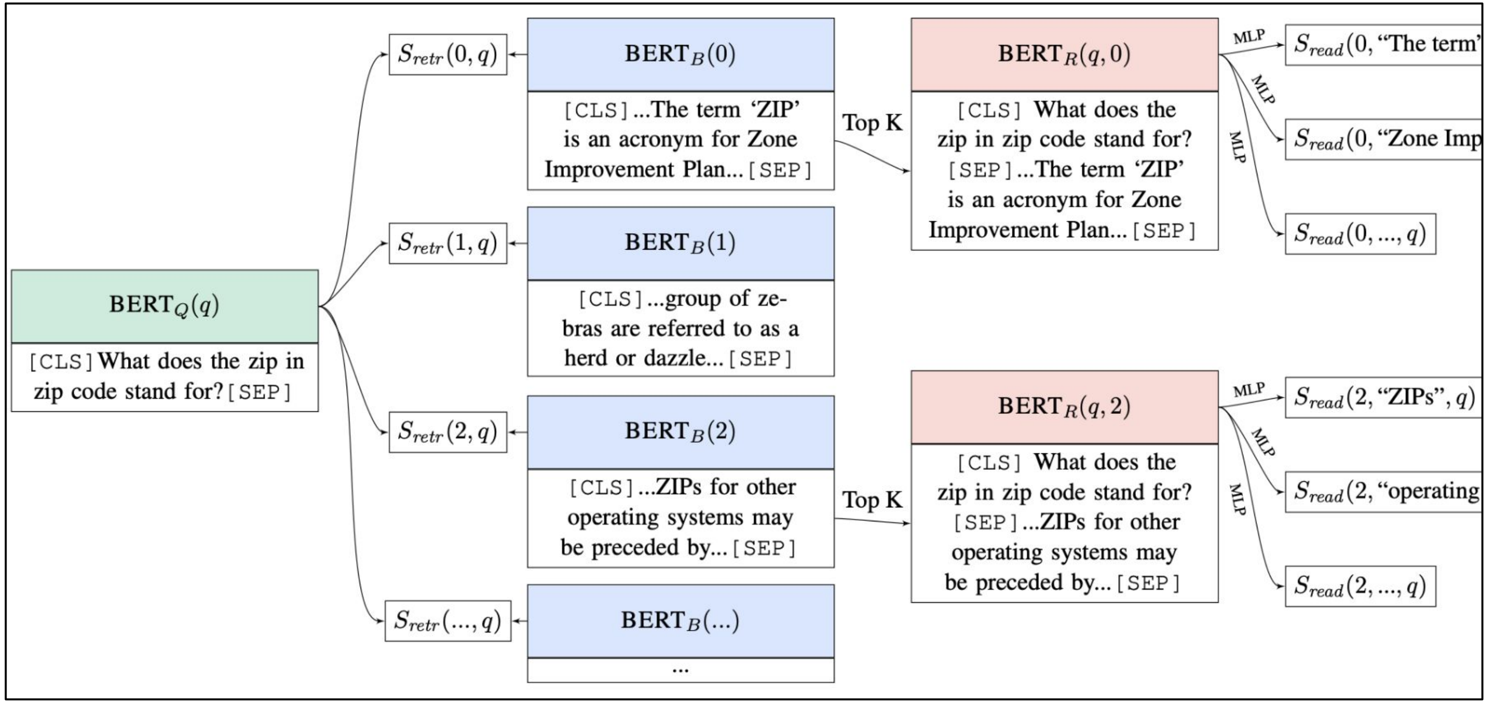


$\text{BERT}_Q(q)$
[CLS] What does the zip in zip code stand for? [SEP]
$S_{retr}(0, q)$
$S_{retr}(1, q)$
$S_{retr}(2, q)$
$S_{retr}(\ldots, q)$
$\text{BERT}_B(0)$
[CLS] ...The term 'ZIP' is an acronym for Zone Improvement Plan... [SEP]
$\text{BERT}_B(1)$
[CLS] ...group of zebras are referred to as a herd or dazzle... [SEP]
$\text{BERT}_B(2)$
[CLS] ...ZIPs for other operating systems may be preceded by... [SEP]
$\text{BERT}_B(\ldots)$
...
Top K
Top K
$\text{BERT}_R(q, 0)$
[CLS] What does the zip in zip code stand for? [SEP] ...The term 'ZIP' is an acronym for Zone Improvement Plan... [SEP]
$\text{BERT}_R(q, 2)$
[CLS] What does the zip in zip code stand for? [SEP] ...ZIPs for other operating systems may be preceded by... [SEP]
MLP
MLP
MLP
MLP
MLP
MLP
$S_{read}(0, \text{"The term"}$
$S_{read}(0, \text{"Zone Imp}$
$S_{read}(0, \ldots, q)$
$S_{read}(2, \text{"ZIPs"}, q)$
$S_{read}(2, \text{"operating}$
$S_{read}(2, \ldots, q)$

**When to retrieve:**
**What to retrieve:**
**How to retrieve:**

**After retrieving:**
Train a **BERT** span-select model to predict the start and end indexes, indicating the substring that act as the answer.

# **Retrieve without LM head: :** ORQA


$\text{BERT}_Q(q)$
[CLS] What does the zip in zip code stand for? [SEP]
$S_{retr}(0, q)$
$S_{retr}(1, q)$
$S_{retr}(2, q)$
$S_{retr}(..., q)$
$\text{BERT}_B(0)$
[CLS] ...The term 'ZIP' is an acronym for Zone Improvement Plan... [SEP]
$\text{BERT}_B(1)$
[CLS] ...group of zebras are referred to as a herd or dazzle... [SEP]
$\text{BERT}_B(2)$
[CLS] ...ZIPs for other operating systems may be preceded by... [SEP]
$\text{BERT}_B(...)$
...
Top K
Top K
$\text{BERT}_R(q, 0)$
[CLS] What does the zip in zip code stand for? [SEP] ...The term 'ZIP' is an acronym for Zone Improvement Plan... [SEP]
$\text{BERT}_R(q, 2)$
[CLS] What does the zip in zip code stand for? [SEP] ...ZIPs for other operating systems may be preceded by... [SEP]
MLP
MLP
MLP
MLP
MLP
MLP
$S_{read}(0, \text{"The term"}$
$S_{read}(0, \text{"Zone Imp}$
$S_{read}(0, ..., q)$
$S_{read}(2, \text{"ZIPs"}, q)$
$S_{read}(2, \text{"operating}$
$S_{read}(2, ..., q)$

**When to retrieve:**
**What to retrieve:**
**How to retrieve:**
**After retrieving:**

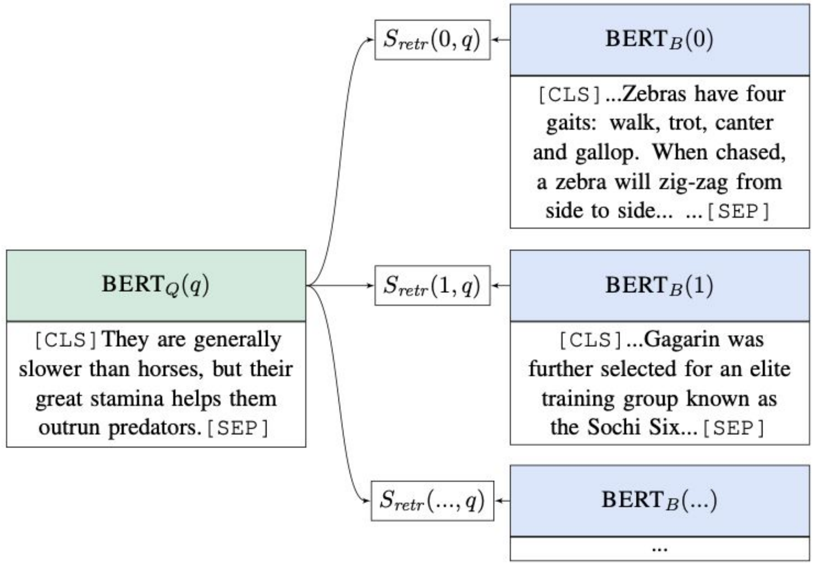
**How to train:**
Pretrain block (context) encoder

Then train end-to-end using ground-truth label.

## **Retrieve without LM head: :** ORQA

- Pre-training task: Inverse Cloze Task

❏ Step 1: Remove a sentence from a snippet
❏ Step 2: Given the **original** snippet and **wrong** snippet, predict the snippet that is the true context



$S_{retr}(0, q)$
$\text{BERT}_B(0)$
[CLS] ...Zebras have four gaits: walk, trot, canter and gallop. When chased, a zebra will zig-zag from side to side... ...[SEP]
$\text{BERT}_Q(q)$
[CLS] They are generally slower than horses, but their great stamina helps them outrun predators. [SEP]
$S_{retr}(1, q)$
$\text{BERT}_B(1)$
[CLS] ...Gagarin was further selected for an elite training group known as the Sochi Six... [SEP]
$S_{retr}(..., q)$
$\text{BERT}_B(...)$
...

Carnegie Mellon University

## Retrieve without LM head: : ORQA

- Fine-tuning goal: Maximize marginal probability

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

Carnegie Mellon University

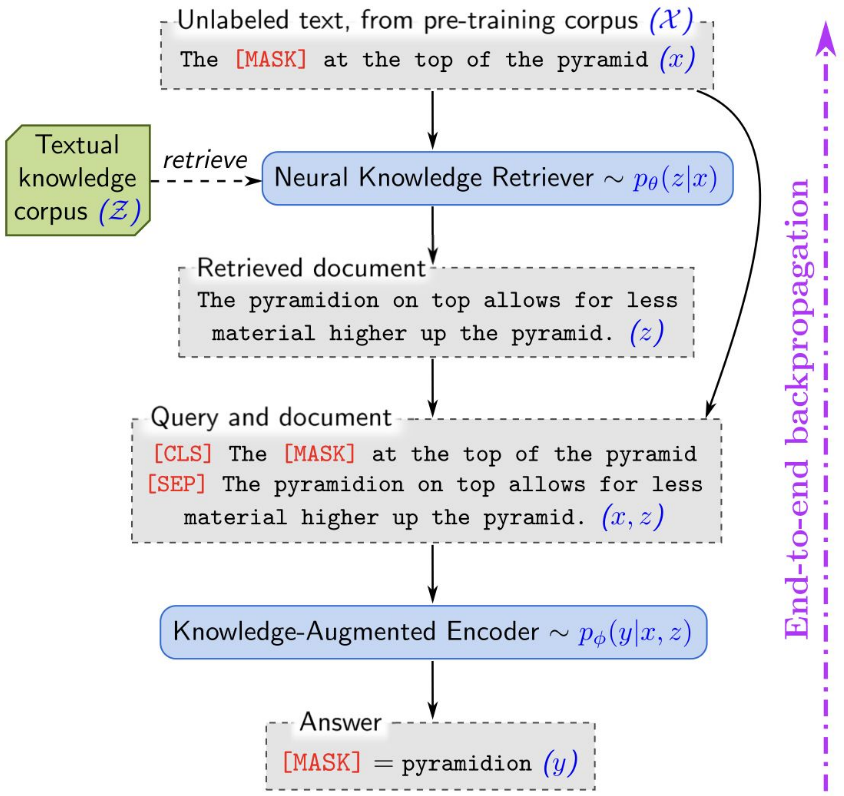
## **Retrieve without LM head: :** ORQA

- Fine-tuning goal: Maximize marginal probability
  - *Z:
    - In theory: The entire corpus
    - In practice: top-k

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

Carnegie Mellon University

# **Retrieval-augmentation during pre-training :** REALM [2]


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

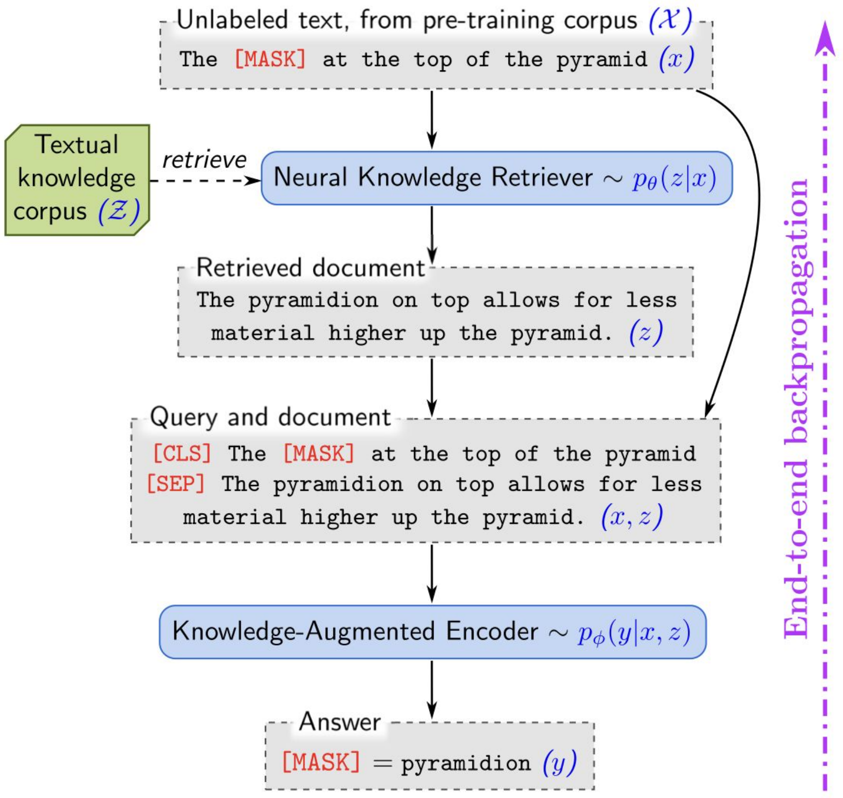
**When to retrieve:**
Per question
**What to retrieve:**
chunk of 288 tokens
**How to retrieve:**
Calculate the product score of question and chunk embedding.

retrieve top-k using approximated Matrix Inner Product Search (MIPS) in sublinear time.

**After retrieving:**
Use a BERT span-select model to predict the start and end indexes, indicating the substring that act as the answer.

[2] Guu, Kelvin, et al. "Retrieval augmented language model pre-training." *International conference on machine learning*. PMLR, 2020.

## **Retrieval-augmentation during pre-training :** REALM [2]

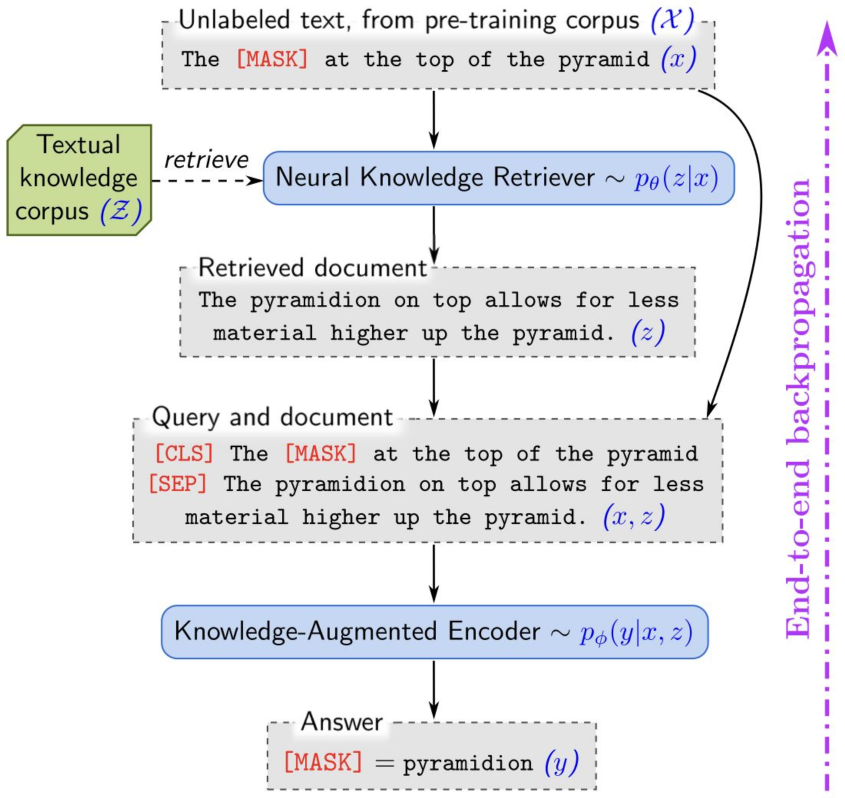

Unlabeled text, from pre-training corpus $(\mathcal{X})$
The [MASK] at the top of the pyramid $(x)$
Textual knowledge corpus $(\mathcal{Z})$
retrieve
Neural Knowledge Retriever $\sim p_\theta(z|x)$
Retrieved document
The pyramidion on top allows for less material higher up the pyramid. $(z)$
Query and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

**When to retrieve:**
**What to retrieve:**
**How to retrieve:**
**After retrieving:**

**How to train:**
Pretraining:
Task – masked token prediction
Train retriever and reader together, using masked token prediction.

Fine-tuning:
Task – span selection
Drop the final projection layer and add a MLP just like ORQA.

## **Retrieval-augmentation during pre-training :** REALM [2]


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

**When to retrieve:**
**What to retrieve:**
**How to retrieve:**
**After retrieving:**

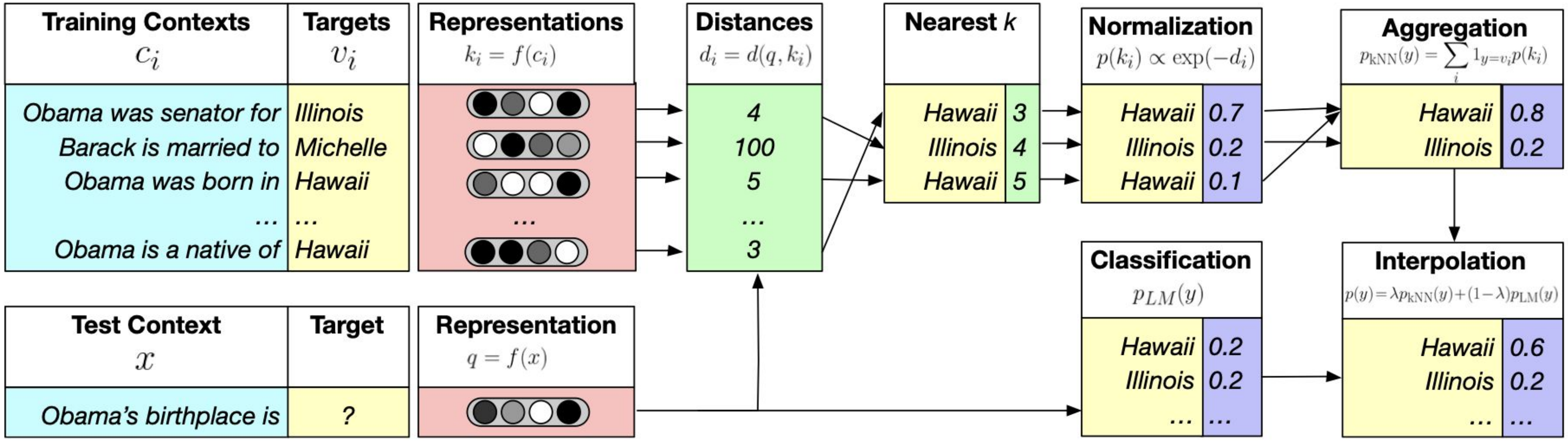
**How to train:**
Pretraining:
Task – masked token prediction
Train retriever and reader together, using masked token prediction.

Fine-tuning:
Task – span selection
Drop the final projection layer and add a MLP just like ORQA.

Finetune end-to-end

Carnegie Mellon University

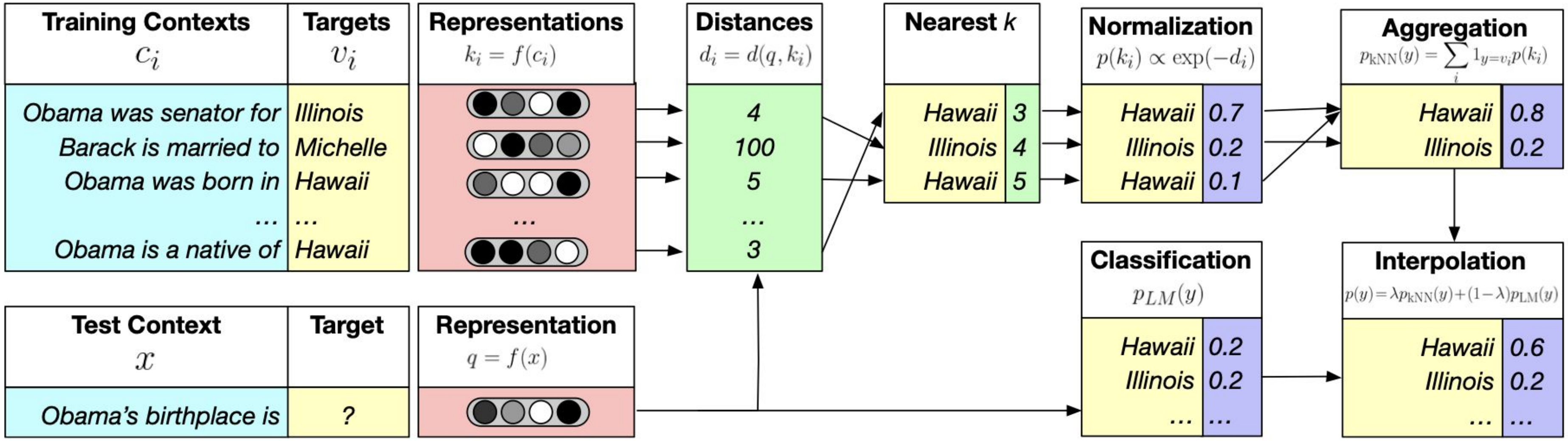
## **Retrieval-augmentation after pre-training :** kNN-LM[3]



**When to retrieve:**
Per output token (autoregressive)

[3] Khandelwal, Urvashi, et al. "Generalization through memorization: Nearest neighbor language models." *arXiv preprint arXiv:1911.00172* (2019).

## **Retrieval-augmentation after pre-training :** kNN-LM[3]

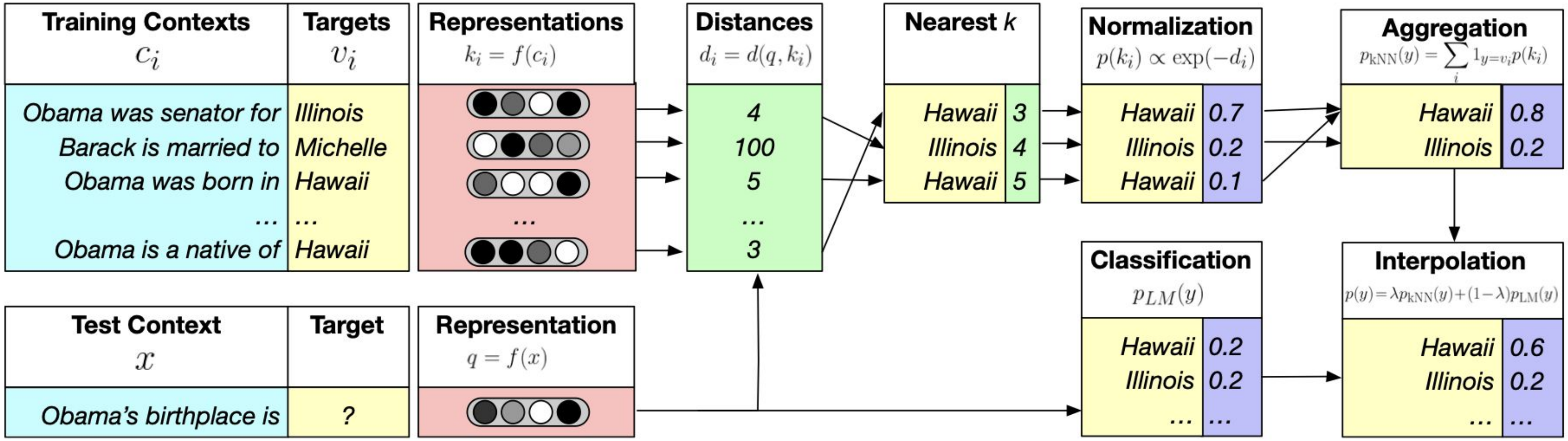


Training Contexts $c_i$
Targets $v_i$
Representations $k_i = f(c_i)$
Distances $d_i = d(q, k_i)$
Nearest $k$
Normalization $p(k_i) \propto \exp(-d_i)$
Aggregation $p_{kNN}(y) = \sum_i 1_{y=v_i} p(k_i)$
Obama was senator for | Illinois | 4
Barack is married to | Michelle | 100
Obama was born in | Hawaii | 5
... | ... | ...
Obama is a native of | Hawaii | 3
Hawaii 3 | Hawaii 0.7
Illinois 4 | Illinois 0.2
Hawaii 5 | Hawaii 0.1
Hawaii 0.8
Illinois 0.2
Test Context $x$
Target
Representation $q = f(x)$
Obama's birthplace is | ?
Classification $p_{LM}(y)$
Hawaii 0.2
Illinois 0.2
... ...
Interpolation $p(y) = \lambda p_{kNN}(y) + (1-\lambda) p_{LM}(y)$
Hawaii 0.6
Illinois 0.2
... ...

**What to retrieve:**

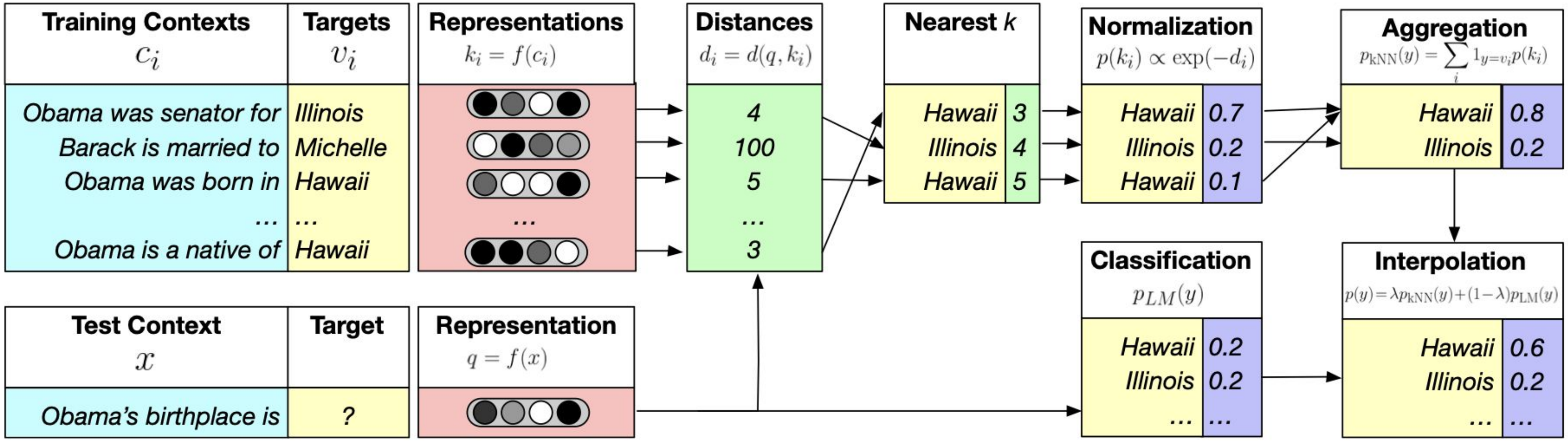
• (Context, Target Word) pairs: ($ci$, $vi$ )

# **Retrieval-augmentation after pre-training :** kNN-LM[3]


Training Contexts $c_i$
Targets $v_i$
Representations $k_i = f(c_i)$
Distances $d_i = d(q, k_i)$
Nearest $k$
Normalization $p(k_i) \propto \exp(-d_i)$
Aggregation $p_{\text{kNN}}(y) = \sum_i \mathbb{1}_{y=v_i} p(k_i)$
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
Hawaii 3 | Hawaii 0.7
Illinois 4 | Illinois 0.2
Hawaii 5 | Hawaii 0.1
Hawaii 0.8
Illinois 0.2
Test Context $x$
Target
Representation $q = f(x)$
Obama's birthplace is | ?
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
... ...

**How to retrieve:**
LM generates embedding for training contexts.
For each step, LM generates embedding for test context.
Use KNN to search nearest k training contexts.

## **Retrieval-augmentation after pre-training :** kNN-LM[3]

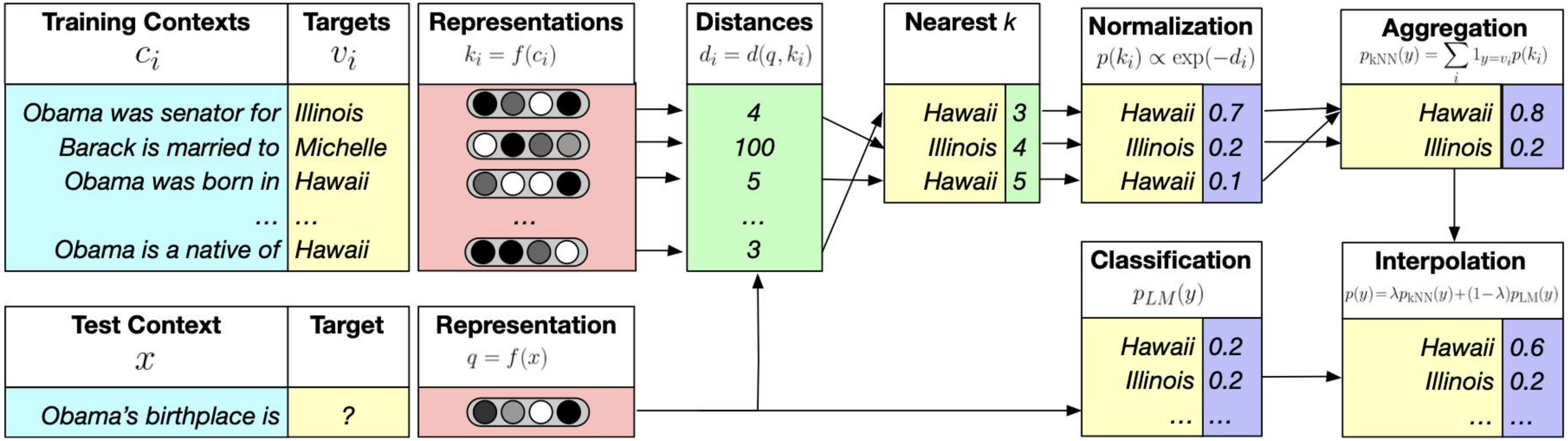


**After retrieving:**

Normalize and Interplot

$p_y = \lambda p_{\text{knn}}\, y + (1 - \lambda) p_{\text{lm}}(y)$

## **Retrieval-augmentation after pre-training :** kNN-LM[3]



Training Contexts $c_i$
Targets $v_i$
Representations $k_i = f(c_i)$
Distances $d_i = d(q, k_i)$
Nearest $k$
Normalization $p(k_i) \propto \exp(-d_i)$
Aggregation $p_{\text{kNN}}(y) = \sum_i 1_{y=v_i} p(k_i)$
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
Hawaii 3 | Hawaii 0.7
Illinois 4 | Illinois 0.2
Hawaii 5 | Hawaii 0.1
Hawaii 0.8
Illinois 0.2
Test Context $x$
Target
Representation $q = f(x)$
Obama's birthplace is | ?
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
... ...

**How to train:**
Standard pre-training.
Plug in new retrieval corpus without additional training.

## Improvement over prev. works

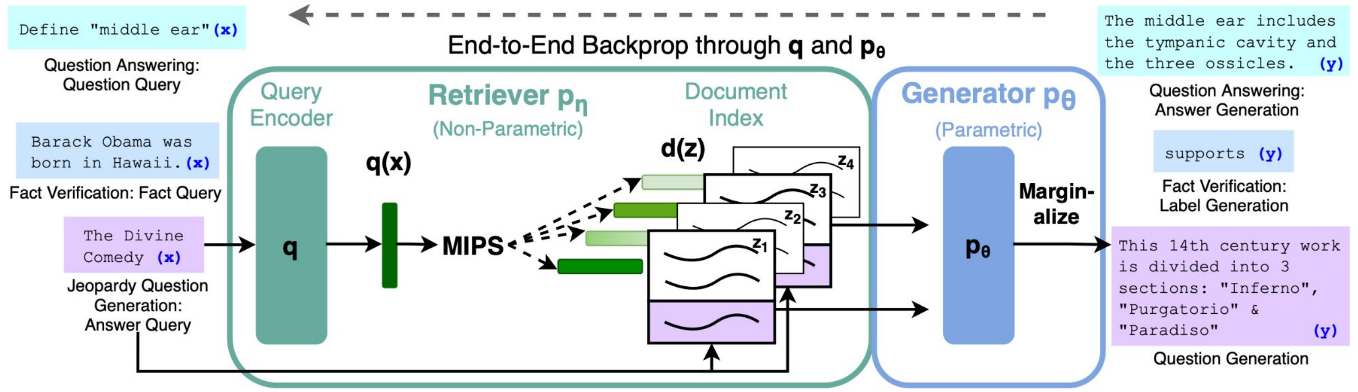
| | ORQA | REALM | kNN-LM | **RAG** (this work) |
|---|---|---|---|---|
| Additional training needed | Yes | Yes | No | **No** |
| Autoregressive? | No | No | Yes | **Yes** |
| Task | Open domain QA | Open domain QA | Open domain QA | Open domain QA, **Abstractive QA, Question Generation** |
| When to retrieve? | Per question | Per question | Per token | **Per question /per token** |

Carnegie Mellon University

## Methods

## Method Overview

1. Encode our query and all documents into features.
2. Match the top-k documents most similar to the query feature
3. Provide the top-k documents to the generator and marginalize


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 pθ
Query Encoder
Retriever pη
(Non-Parametric)
Document Index
q
q(x)
MIPS
d(z)
z4
z3
z2
z1
Generator pθ
(Parametric)
pθ
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

## RAG Models

**RAG-Sequence Model:** use the same retrieved document(s) to generate the complete sequence.

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

**RAG-Token Model:** allows different document(s) to be retrieved for generating each token.

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

Carnegie Mellon University

## RAG Models

**RAG-Sequence Model:** use the same retrieved document(s) to generate the complete sequence.

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

**RAG-Token Model:** allows different document(s) to be retrieved for generating each token.

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

Carnegie Mellon University

## RAG Models

**RAG-Sequence Model:** use the same retrieved document(s) to generate the complete sequence.

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

Approx.

**Marginalization on sequence**

**RAG-Token Model:** allows different document(s) to be retrieved for generating each token.

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

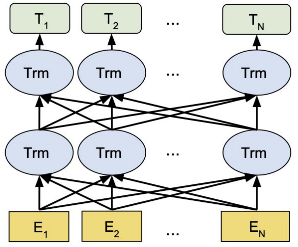
Approx.

**Marginalization on token**

## Architecture: Retriever

Query and Context source are encoded by separate BERT document encoders.

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



The encoded document index d(z) are referred as **non-parametric memory**.

Need to extract the top-k results: Maximum Inner Product Search problem.

- The author used FAISS(Facebook AI similarity search) combined with HNSW (Hierarchical Navigable Small World Approximation) to approximately solve the problem in sub-linear time.

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

Carnegie Mellon University

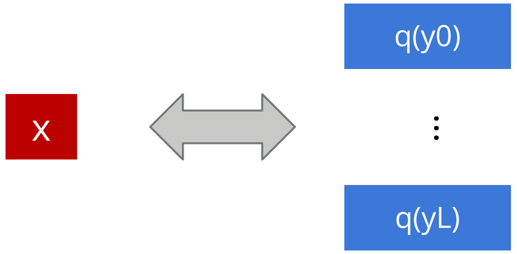
## MIPS Problem - FAISS

Quantization into coarse, fine level. Use coarse quantization result to limit the scope of computing similarity.

$$y \approx q(y) = q_1(y) + q_2(y - q_1(y))$$

**Asymmetric Distance Computation:** compute distance between **x** and all quantized **y**.

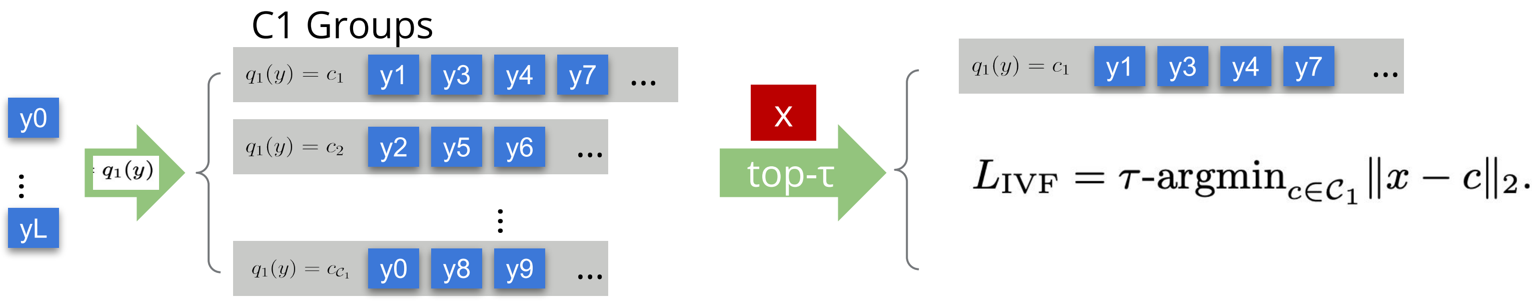
$$L_{\text{ADC}} = k\text{-argmin}_{i=0:\ell} \|x - q(y_i)\|_2.$$



https://arxiv.org/pdf/1702.08734.pdf

## MIPS Problem - FAISS

IVFADC: reduce search range by grouping with q1(.)


C1 Groups
y0
yL
$q_1(y)$
$q_1(y) = c_1$ y1 y3 y4 y7 …
$q_1(y) = c_2$ y2 y5 y6 …
$q_1(y) = c_{\mathcal{C}_1}$ y0 y8 y9 …
x
top-τ
$q_1(y) = c_1$ y1 y3 y4 y7 …

$$L_{\text{IVF}} = \tau\text{-argmin}_{c \in \mathcal{C}_1} \|x - c\|_2.$$

Then proceed with the ADC computation. FIASS Indexes part of the compute on distance between x to quantized values for acceleration.

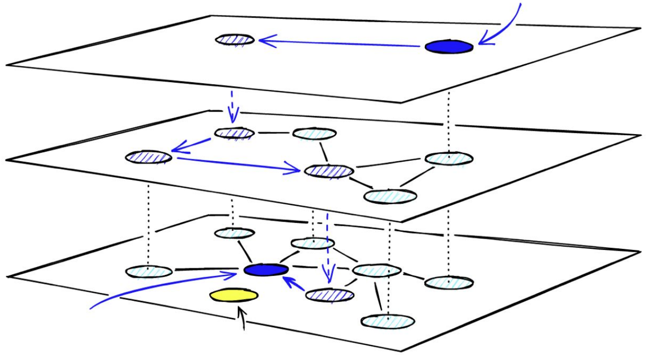
$$L_{\text{IVFADC}} = \underset{i=0:\ell \text{ s.t. } q_1(y_i) \in L_{\text{IVF}}}{k\text{-argmin}} \|x - q(y_i)\|_2.$$

https://arxiv.org/pdf/1702.08734.pdf

## MIPS Problem - Hierarchical Navigable Small World Approximation

**Search:** A pre-constructed hierarchical graph that greedily searches for the nearest neighbor to the query and proceed to the next layer until the last.

**Construction:** After inserting the nodes to layers, edges are connected to a fixed number of nearest neighbors within each layer.

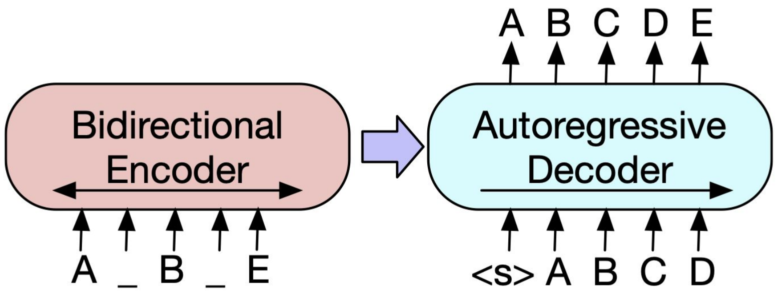


https://www.pinecone.io/learn/series/faiss/hnsw/https://www.pinecone.io/learn/series/faiss/hnsw/

## Architecture: Generator

Used BART-large as the generator architecture

400M parameters



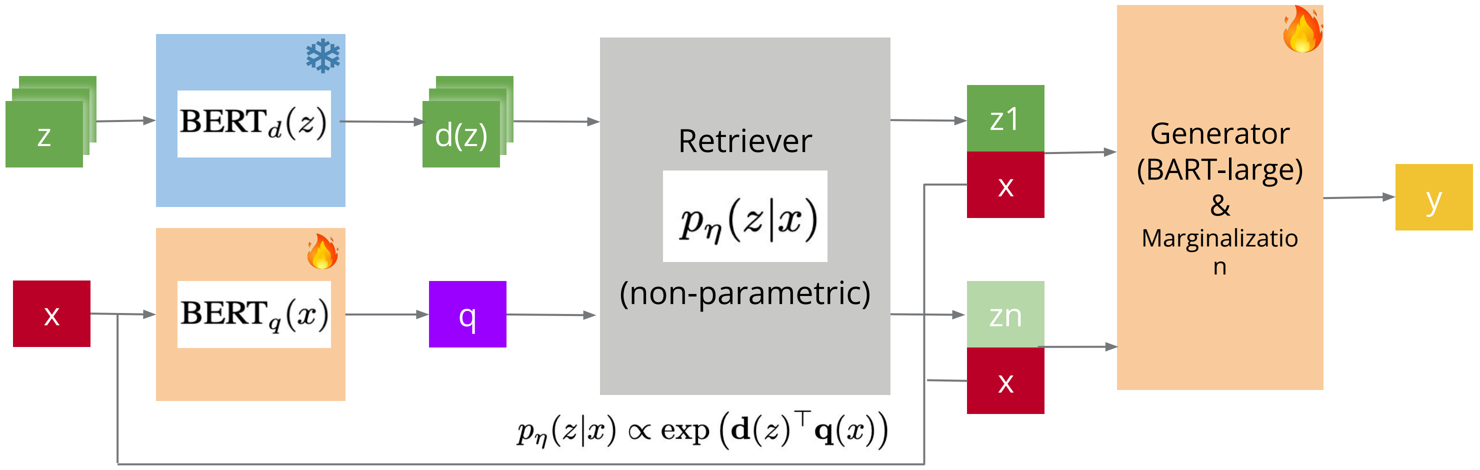
Input **x** and retrieved documents **z** are concatenated to input to BART.

BART parameters θ referred as **parametric memory**.

## Training

Jointly trained the retriever and generator by minimizing the negative marginal log-likelihood of each target on input-output pairs (xi, yi)

$$\sum_j -\log p(y_j|x_j)$$


z
$\mathrm{BERT}_d(z)$
d(z)
x
$\mathrm{BERT}_q(x)$
q
Retriever
$p_\eta(z|x)$
(non-parametric)
z1
x
zn
x
Generator (BART-large) & Marginalizatio n
y
$p_\eta(z|x) \propto \exp\left(\mathbf{d}(z)^\top \mathbf{q}(x)\right)$

## Decoding

**RAG-Token Model:**

Can be treated as normal auto-regressive language model, decoded by a beam decoder with transition probability:

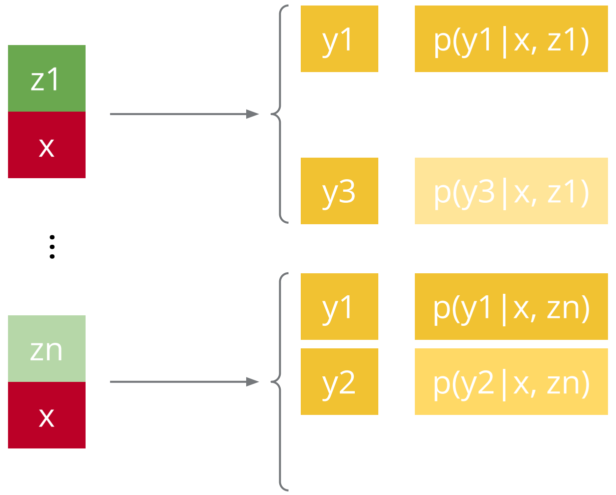
$$p'_\theta(y_i|x, y_{1:i-1}) = \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z_i|x) p_\theta(y_i|x, z_i, y_{1:i-1})$$

**RAG-Sequence Model:**

The conditional p(y|x) cannot break down to per-token transition probabilities. Therefore cannot use beam search directly.

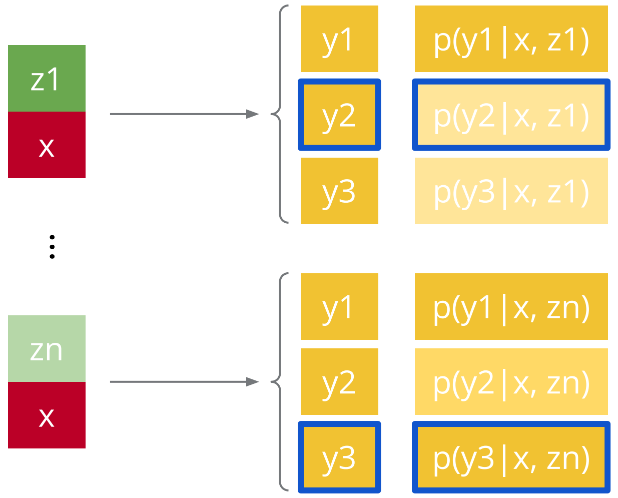
## Thorough Decoding - RAG Sequence Model

Run beam search for each retrieved document.



## Thorough Decoding - RAG Sequence Model

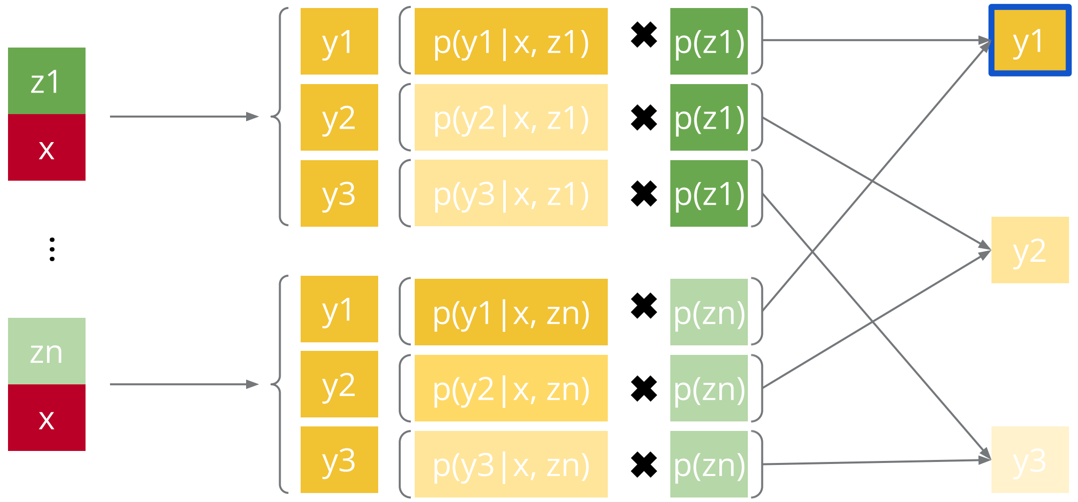
Some hypothesis may not appear in all beam search. Run additional forward for them:



Carnegie Mellon University

# Thorough Decoding - RAG Sequence Model

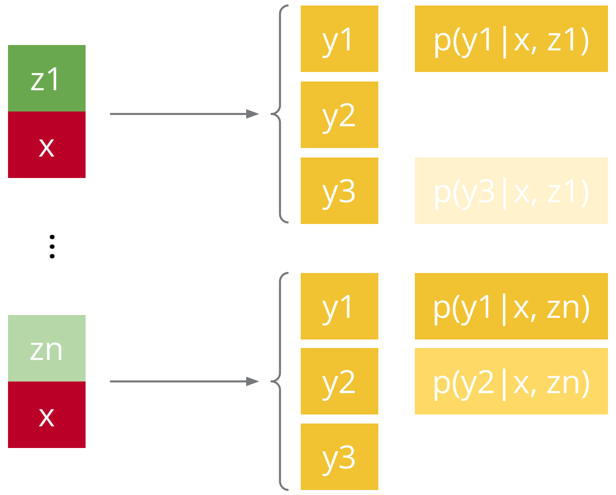
Marginalize hypothesis with weight on the document.



# Fast Decoding - RAG Sequence Model

When the number of hypothesis is large, is it costly to forward all distinct hypotheses.

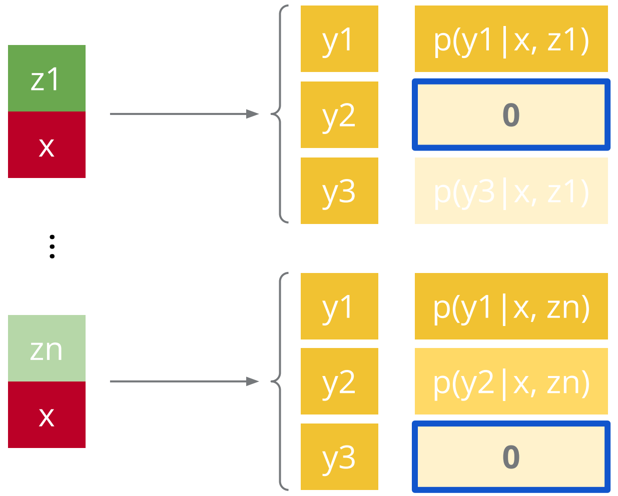
Approximate p(y|x,z)=0 when y is not generated by document z during beam search



# Fast Decoding - RAG Sequence Model

When the number of hypothesis is large, is it costly to forward all distinct hypotheses.

Approximate p(y|x,z)=0 when y is not generated by document z during beam search



## Experiments & Results

**Main Experiments**

Open-domain Question Answering

Jeopardy Question Generation

Abstractive Question Answering

Fact Verification

**Additional Results**

Generation Diversity

Retrieval Ablations

Index hot-swapping

Effect of Retrieving more documents

Carnegie Mellon University

## Open-domain Question Answering

Overview: Open-domain QA involves building systems that can answer questions on any topic without specific domain restrictions. These systems often use large-scale datasets and access a broad range of information sources to find answers.

Example:

- Question: "Who wrote 'Pride and Prejudice'?"
- Answer: "Jane Austen."

Why better performance:

Both REALM and DPR use a Reader model to extract answers from the retrieved documents.

What if there's no correct answer in the documents?

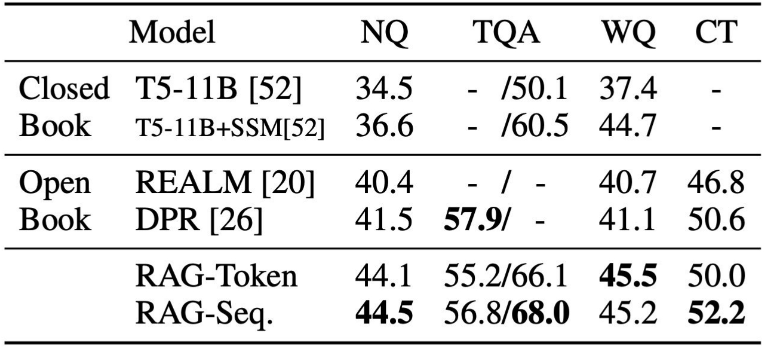
RAG use a BART model as a generator to generate answers with the help from related documents.
RAG can generate correct answers when the related documents only have clues but not the answer verbatim, or do not have correct answer at all.

### open-domain QA datasets:

NQ: Natural Questions
TQA: TriviaQA
WQ: WebQuestions
CT: CuratedTrec

### Settings:

Closed-book: parametric-only
Open-book: retrieval-based



| | 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** |

## Open-domain Question Answering

RAG combines the generation flexibility of the "closed-book" (parametric only) approaches and the performance of "open-book" retrieval-based approaches.

REALM or T5+SSM:
"Salient span masking": Mask entities instead of random masking. A BERT-based tagger is used to identify named entities, which is time-consuming.

DPR:
DPR uses a BERT-based "crossencoder" to re-rank documents, along with an extractive reader.

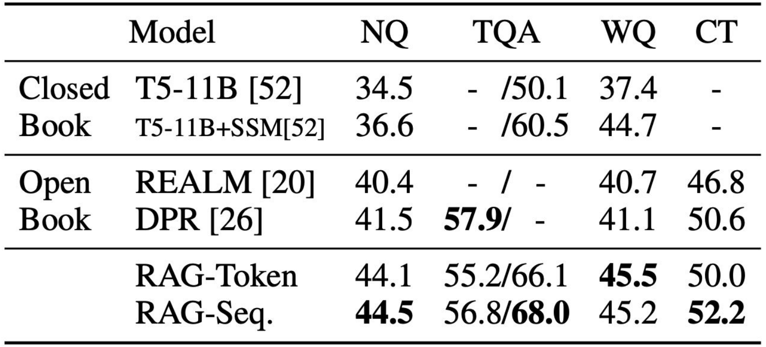
RAG is more efficient.

### open-domain QA datasets:

NQ: Natural Questions
TQA: TriviaQA
WQ: WebQuestions
CT: CuratedTrec

### Settings:

Closed-book: parametric-only
Open-book: retrieval-based



| | 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** |

## Jeopardy Question Generation (Jeopardy)

Overview: This involves generating questions suitable for the game show Jeopardy, where the answers are given first, and the challenge is to form the corresponding question. It tests both the creativity and the informativeness of the generated question.

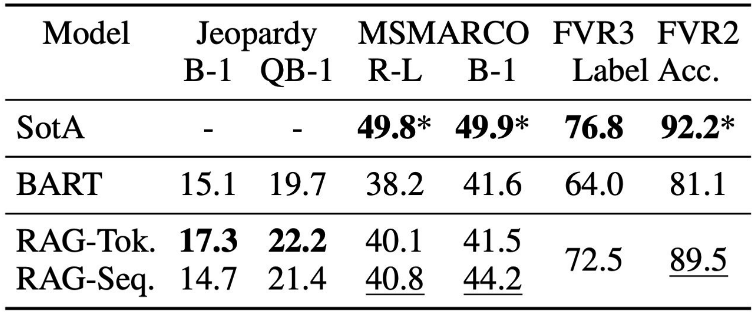
Example:

- Given: "This 19th-century British novelist wrote 'Oliver Twist' and 'A Christmas Carol.'"
- Question: "Who is Charles Dickens?"

New task using splits from SearchQA, with 100K train, 14K dev, and 27K test examples

SotA are different model systems which produce best results in each tasks and use gold context/evidence

B-1: BLEU-1
QB-1: Q-BLEU-1
R-L: Rouge-L



| 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 | | |

## Jeopardy Question Generation (Jeopardy)

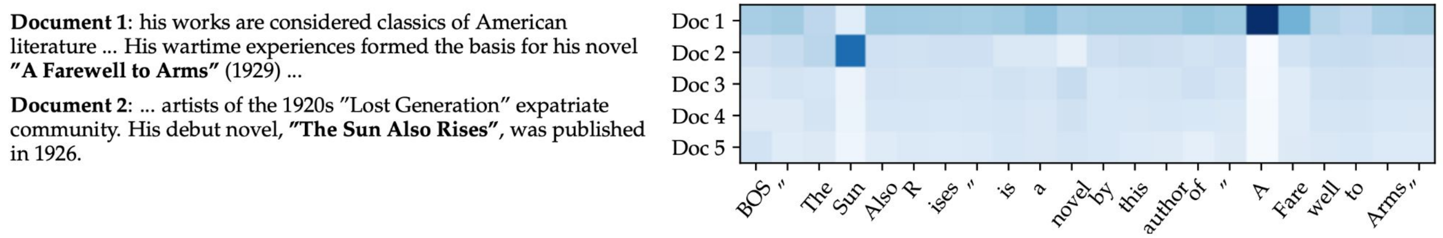
Why RAG-Token perform best?

Different posterior when generating different tokens

RAG-Token can generate responses that combine content from several documents

Input: Hemingway

Output: “The Sun Also Rises” is a novel by this author of “A Farewell to Arms”



**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".

## Jeopardy Question Generation (Jeopardy)

This observation suggests that the generator can complete the titles without depending on specific documents.

BART model:
Given partial result: “The Sun”
Full Generation: "The Sun Also Rises" is a novel by this author of "The Sun Also Rises"
→ the title "The Sun Also Rises" is stored in BART’s parameters.

Given partial result: "The Sun Also Rises" is a novel by this author of "A
Full Generation: "The Sun Also Rises" is a novel by this author of "A Farewell to Arms".
→ the title "A Farewell to Arms" and the authorship are stored in BART’s parameters.

RAG can generate the correct question
→specific knowledge is stored in the parametric memory, and the non-parametric component helps to guide the generation.

**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.

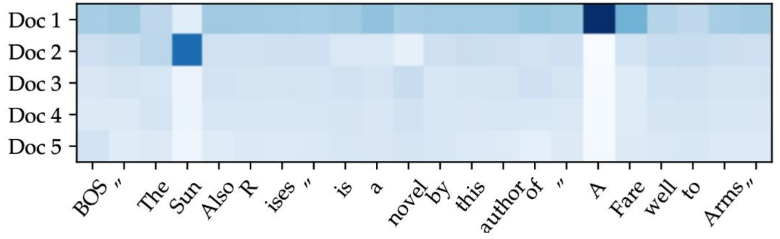


Figure 2: RAG-Token document posterior $p(z_i|x, y_i, y_{-i})$ for each generated token for input “Hemingway" for Jeopardy generation with 5 retrieved documents. The posterior for document 1 is high when generating “A Farewell to Arms" and for document 2 when generating “The Sun Also Rises".

## Jeopardy Question Generation (Jeopardy)

BART is more factual than RAG in only 7.1% of cases
RAG is more factual in 42.7% of cases

→ The effectiveness of RAG on the task over a state-of-the-art generation model

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% |

Evaluators also find RAG generations to be more specific by a large margin.

| | | | |
|---|---|---|---|
| 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" |

'?' indicates factually incorrect responses, * indicates partially correct responses.

## Abstractive Question Answering (MSMARCO)

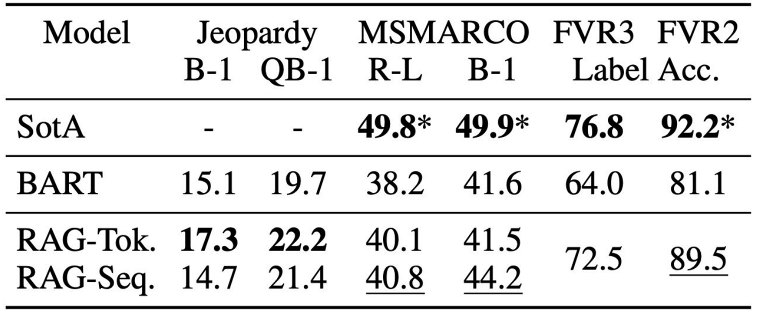
Overview: Unlike the typical QA that retrieves parts of text containing the answer, abstractive QA systems generate a concise, coherent response that may not directly quote the source materials. This requires deep understanding and reformulation of the information.

Example:

- Question: "Why is the sky blue?"
- Answer: "The sky appears blue because of the scattering of sunlight by the Earth's atmosphere, which is more effective at shorter wavelengths, such as blue."

SotA are different model systems which produce best results in each tasks and use gold context/evidence

B-1: BLEU-1
QB-1: Q-BLEU-1
R-L: Rouge-L



| 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 | | |

## Abstractive Question Answering (MSMARCO)

MSMARCO:The task consists of questions, ten gold passages retrieved from a search engine for each question, and a full sentence answer annotated from the retrieved passages.

RAG do not use the supplied passages, only the questions and answers, to treat MSMARCO as an open-domain abstractive QA task.

RAG approaches state-of-the-art model performance.

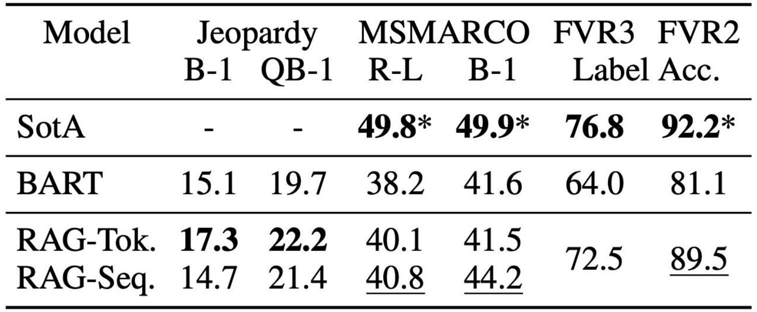
1. SotA models access gold passages with specific information required to generate the reference answer.

2. many questions are unanswerable without the gold passages.

3. not all questions are answerable from Wikipedia alone.

SotA are different model systems which produce best results in each tasks and use gold context/evidence

B-1: BLEU-1
QB-1: Q-BLEU-1
R-L: Rouge-L



| 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 | | |

## Fact Verification (FVR3 & FVR2)

Overview: Fact Verification systems check the truthfulness of a statement by cross-referencing it with credible sources. It's crucial in combating misinformation, especially on social media and other online platforms.

Example:

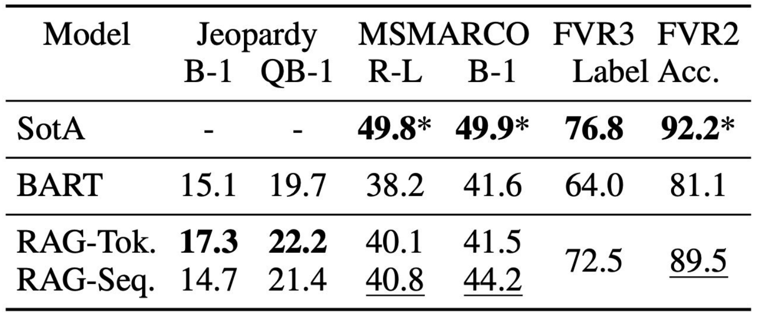
- Statement: "The capital of France is Paris."
- Classified as "True"

FEVER requires classifying whether a natural language claim is supported or refuted by Wikipedia, or whether there is not enough information to decide.

RAG map FEVER class labels (supports, refutes, or not enough info) to single output tokens and directly train with claim-class pairs.
RAG do not use supervision on retrieved evidence.

SotA are different model systems which produce best results in each tasks and use gold context/evidence

B-1: BLEU-1
QB-1: Q-BLEU-1
R-L: Rouge-L



| 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 | | |

## Fact Verification (FVR3 & FVR2)

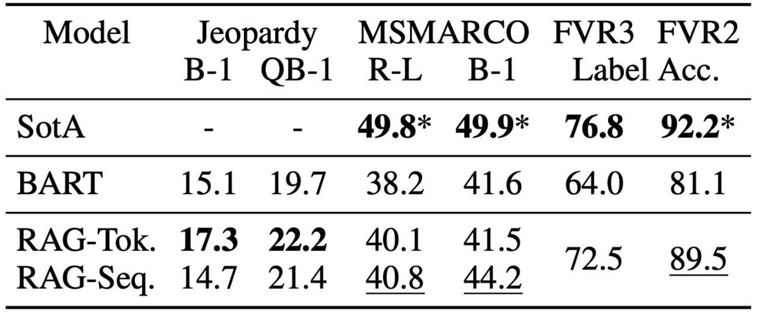
3-way classification (supports, refutes, or not enough info):
RAG scores are within 4.3% of state-of-the-art models.
2-way classification (supports, refutes):
RAG scores are within 2.7% of state-of-the-art models.
(RAG do not require supervision. RAG retrieve its own evidence.)

overlap in article titles between the top k documents retrieved by RAG and gold evidence annotations:
Top retrieved document is from a gold article in 71% of cases.
A gold article is present in the top 10 retrieved articles in 90% of cases.

This indicates the retriever can effectively retrieve the correct documents.

SotA are different model systems which produce best results in each tasks and use gold context/evidence

B-1: BLEU-1
QB-1: Q-BLEU-1
R-L: Rouge-L

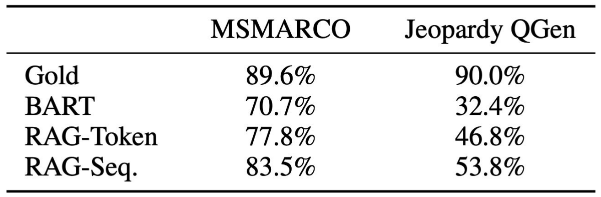


| Model | Jeopardy | | MSMARCO | | FVR3 | FVR2 |
|---|---|---|---|---|---|---|
| | B-1 | QB-1 | R-L | B-1 | Label Acc. | |
| SotA | - | - | **49.8*** | **49.9*** | **76.8** | **92.2*** |
| BART | 15.1 | 19.7 | 38.2 | 41.6 | 64.0 | 81.1 |
| RAG-Tok. | **17.3** | **22.2** | 40.1 | 41.5 | 72.5 | <u>89.5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

## Additional Results

### 1. Generation Diversity

Table 5: Ratio of distinct to total tri-grams for generation tasks.



| | MSMARCO | Jeopardy QGen |
|---|---|---|
| Gold | 89.6% | 90.0% |
| BART | 70.7% | 32.4% |
| RAG-Token | 77.8% | 46.8% |
| RAG-Seq. | 83.5% | 53.8% |

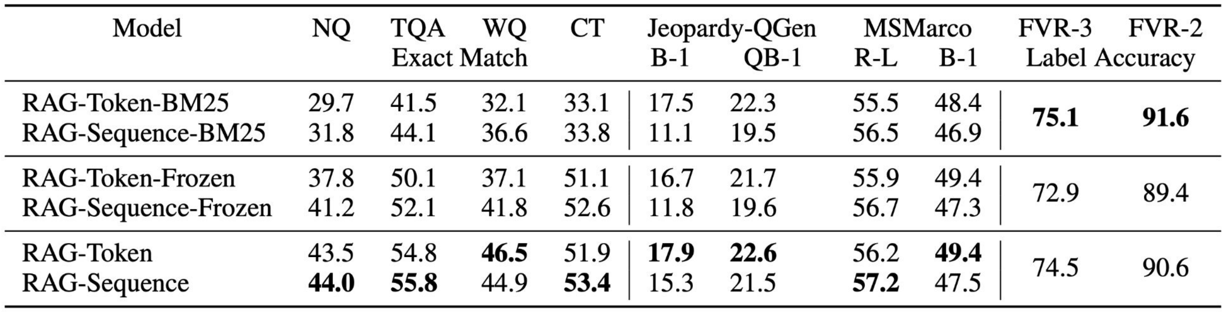
Investigate generation diversity by calculating the ratio of distinct ngrams to total ngrams generated by different models.

RAG-Sequence's generations are more diverse than RAG-Token's, and both are significantly more diverse than BART without needing any diversity-promoting decoding.

## Additional Results

### 2. Retrieval Ablations

Table 6: Ablations on the dev set. As FEVER is a classification task, both RAG models are equivalent.



| Model | NQ | TQA | WQ | CT | Jeopardy-QGen | | MSMarco | | FVR-3 | FVR-2 |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Exact Match | | | B-1 | QB-1 | R-L | B-1 | Label Accuracy | |
| RAG-Token-BM25 | 29.7 | 41.5 | 32.1 | 33.1 | 17.5 | 22.3 | 55.5 | 48.4 | **75.1** | **91.6** |
| RAG-Sequence-BM25 | 31.8 | 44.1 | 36.6 | 33.8 | 11.1 | 19.5 | 56.5 | 46.9 | | |
| RAG-Token-Frozen | 37.8 | 50.1 | 37.1 | 51.1 | 16.7 | 21.7 | 55.9 | 49.4 | 72.9 | 89.4 |
| RAG-Sequence-Frozen | 41.2 | 52.1 | 41.8 | 52.6 | 11.8 | 19.6 | 56.7 | 47.3 | | |
| RAG-Token | 43.5 | 54.8 | **46.5** | 51.9 | **17.9** | **22.6** | 56.2 | **49.4** | 74.5 | 90.6 |
| RAG-Sequence | **44.0** | **55.8** | 44.9 | **53.4** | 15.3 | 21.5 | **57.2** | 47.5 | | |

BM25, short for "Best Matching 25," is a ranking function used by search engines to estimate the relevance of documents to a given search query.

It is part of a family of retrieval functions under the probabilistic retrieval model, and it's widely used due to its effectiveness and simplicity.

To assess the effectiveness of the retrieval mechanism, we run ablations where we freeze the retriever during training

RAG-BM25: replace RAG's retriever with a fixed BM25 system
RAG-Frozen: Freeze the retriever during training

For FEVER, BM25 performs best, perhaps since FEVER claims are heavily entity-centric and thus well-suited for word overlap-based retrieval. Differentiable retrieval improves results on all other tasks, especially for Open-Domain QA, where it is crucial.

## Additional Results

### 3. Index hot-swapping

RAG: knowledge can be easily updated at test time.
Parametric-only models like T5 or BART: need further training to update their behavior as the world changes

Experiment:
build an index using the DrQA Wikipedia dump from December 2016 and December 2018
use a template “Who is {position}?” (e.g. “Who is the President of Peru?”) to query our NQ RAG model with each index

Results:
70% correctly using the 2016 index for 2016 world leaders and 68% using the 2018 index for 2018 world leaders
12% with the 2018 index and 2016 leaders, 4% with the 2016 index and 2018 leaders

Conclusion:
This shows we can update RAG’s world knowledge by simply replacing its non-parametric memory.

Carnegie Mellon University

## Additional Results

**4. Effect of Retrieving more documents**

Observation:
Models are trained with either 5 or 10 retrieved latent documents, and RAG do not observe significant differences in performance between them

Experiment:
Adjust the number of retrieved documents at test time, which can affect performance and runtime

Results:
1. Left Figure: Retrieving more documents at test time monotonically improves Open-domain QA results for RAG-Sequence, but the performance peaks for RAG-Token at 10 retrieved documents.
2. Center Figure: Retrieving more docume leads to a increasing recall for all models.
3. Right Figure: Retrieving more documents leads to higher Rouge-L for RAG-Token, but the effect is less pronounced for RAG-Sequence.

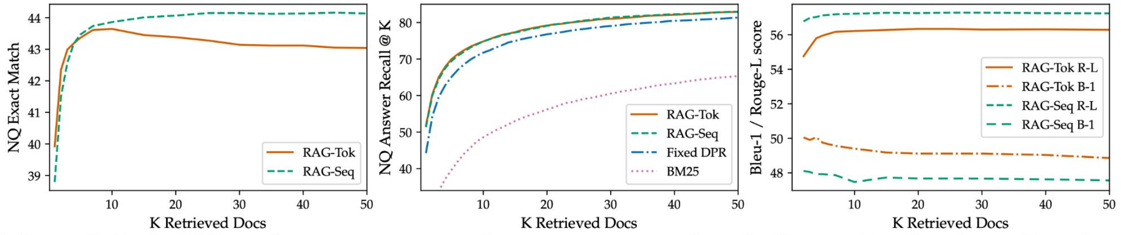
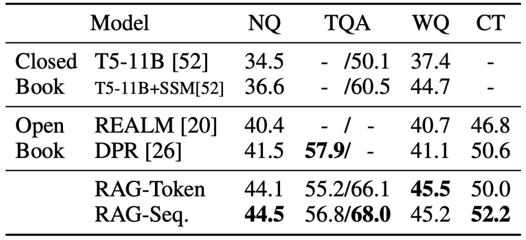


Figure 3: Left: NQ performance as more documents are retrieved. Center: Retrieval recall performance in NQ. Right: MS-MARCO Bleu-1 and Rouge-L as more documents are retrieved.

Carnegie Mellon University

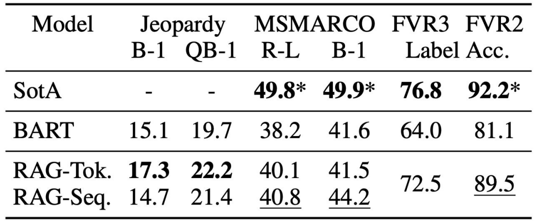
## Results Overview

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.



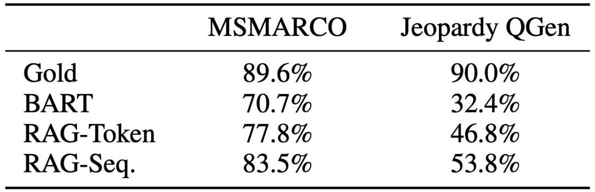
| | Model | NQ | TQA | WQ | CT |
|---|---|---|---|---|---|
| Closed Book | T5-11B [52] | 34.5 | - /50.1 | 37.4 | - |
| | T5-11B+SSM[52] | 36.6 | - /60.5 | 44.7 | - |
| Open Book | REALM [20] | 40.4 | - / - | 40.7 | 46.8 |
| | DPR [26] | 41.5 | **57.9**/ - | 41.1 | 50.6 |
| | RAG-Token | 44.1 | 55.2/66.1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



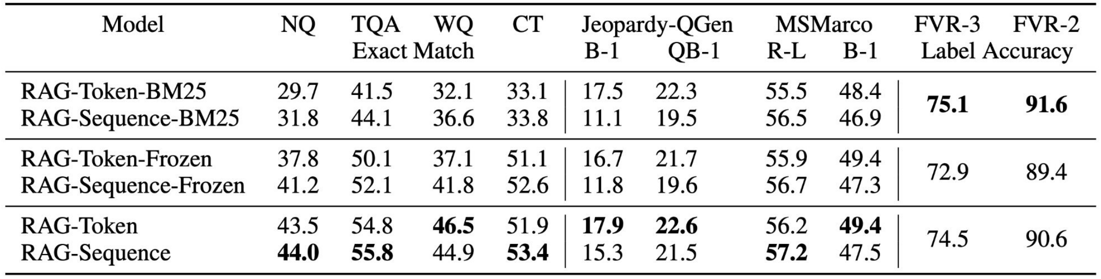
| Model | Jeopardy | | MSMARCO | | FVR3 | FVR2 |
|---|---|---|---|---|---|---|
| | B-1 | QB-1 | R-L | B-1 | Label Acc. | |
| SotA | - | - | **49.8*** | **49.9*** | **76.8** | **92.2*** |
| BART | 15.1 | 19.7 | 38.2 | 41.6 | 64.0 | 81.1 |
| RAG-Tok. | **17.3** | **22.2** | 40.1 | 41.5 | 72.5 | 89.5 |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

Table 5: Ratio of distinct to total tri-grams for generation tasks.



| | MSMARCO | Jeopardy QGen |
|---|---|---|
| Gold | 89.6% | 90.0% |
| BART | 70.7% | 32.4% |
| RAG-Token | 77.8% | 46.8% |
| RAG-Seq. | 83.5% | 53.8% |

Table 6: Ablations on the dev set. As FEVER is a classification task, both RAG models are equivalent.



| Model | NQ | TQA | WQ | CT | Jeopardy-QGen | | MSMarco | | FVR-3 | FVR-2 |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Exact Match | | | B-1 | QB-1 | R-L | B-1 | Label Accuracy | |
| RAG-Token-BM25 | 29.7 | 41.5 | 32.1 | 33.1 | 17.5 | 22.3 | 55.5 | 48.4 | **75.1** | **91.6** |
| RAG-Sequence-BM25 | 31.8 | 44.1 | 36.6 | 33.8 | 11.1 | 19.5 | 56.5 | 46.9 | | |
| RAG-Token-Frozen | 37.8 | 50.1 | 37.1 | 51.1 | 16.7 | 21.7 | 55.9 | 49.4 | 72.9 | 89.4 |
| RAG-Sequence-Frozen | 41.2 | 52.1 | 41.8 | 52.6 | 11.8 | 19.6 | 56.7 | 47.3 | | |
| RAG-Token | 43.5 | 54.8 | **46.5** | 51.9 | **17.9** | **22.6** | 56.2 | **49.4** | 74.5 | 90.6 |
| RAG-Sequence | **44.0** | **55.8** | 44.9 | **53.4** | 15.3 | 21.5 | **57.2** | 47.5 | | |

Carnegie Mellon University