End-to-End Training of Multi-Document Reader and Retriever for Open-Domain QA

- Devendra Singh Sachan

PhD student @ Mila and McGill University

Devendra Sachan

Siva Reddy

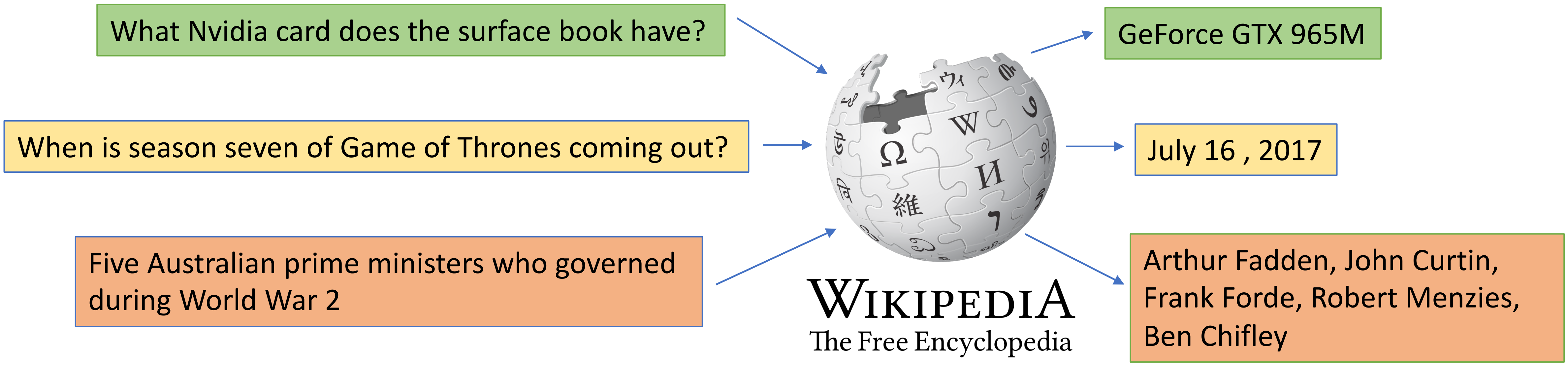
Will Hamilton

Chris Dyer

Dani Yogatama

Problem Setup: Open-Domain QA

- **Input**: Question (q) and evidence documents (D) such as Wikipedia (millions of documents)
- **Output**: Answer (a)



What Nvidia card does the surface book have?

When is season seven of Game of Thrones coming out?

Five Australian prime ministers who governed during World War 2

GeForce GTX 965M

July 16 , 2017

Arthur Fadden, John Curtin, Frank Forde, Robert Menzies, Ben Chifley

Chen et al., 2017. Reading Wikipedia to Answer Open-domain Questions

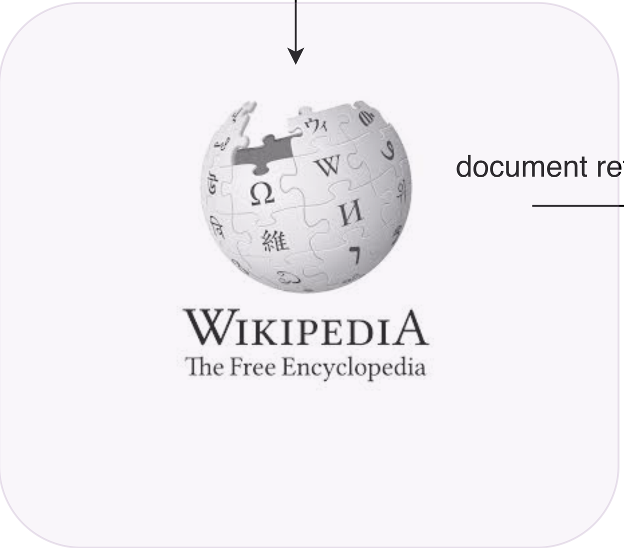
Background: Open-Domain QA

Two-stage approach

Stage 1. Document retriever from evidence

Stage 2. Answer extraction

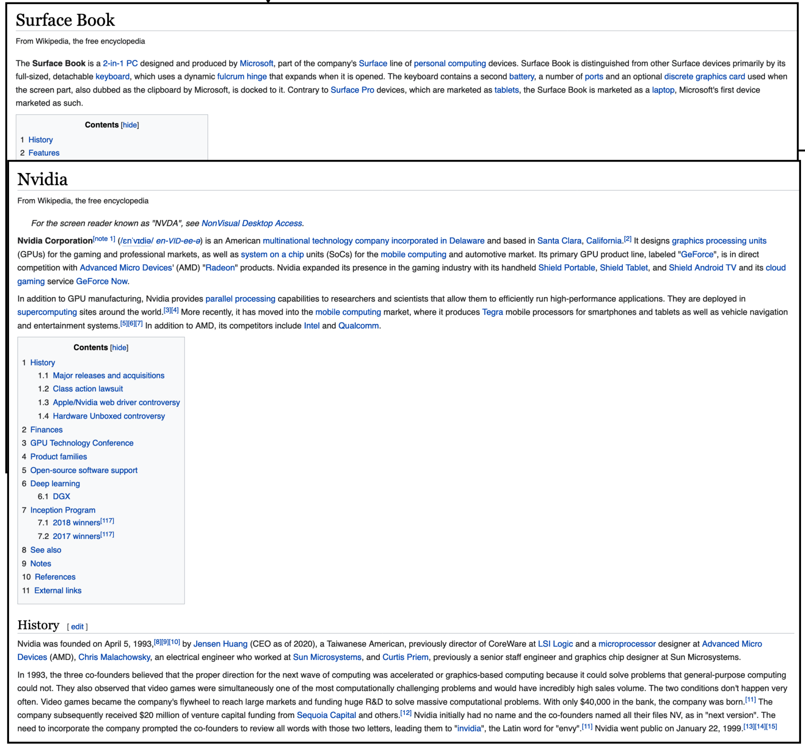
What Nvidia card does the surface book have?



Information Retrieval

Ex: TFIDF, BM-25

document retriever



Surface Book

From Wikipedia, the free encyclopedia

The **Surface Book** is a 2-in-1 PC designed and produced by Microsoft, part of the company's Surface line of personal computing devices. Surface Book is distinguished from other Surface devices primarily by its full-sized, detachable keyboard, which uses a dynamic fulcrum hinge that expands when it is opened. The keyboard contains a second battery, a number of ports and an optional discrete graphics card used when the screen part, also dubbed as the clipboard by Microsoft, is docked to it. Contrary to Surface Pro devices, which are marketed as tablets, the Surface Book is marketed as a laptop, Microsoft's first device marketed as such.

Contents [hide]

1 History
2 Features

Nvidia

From Wikipedia, the free encyclopedia

For the screen reader known as "NVDA", see NonVisual Desktop Access.

Nvidia Corporation[note 1] (/ɛnˈvɪdiə/ *en-VID-ee-ə*) is an American multinational technology company incorporated in Delaware and based in Santa Clara, California.[2] It designs graphics processing units (GPUs) for the gaming and professional markets, as well as system on a chip units (SoCs) for the mobile computing and automotive market. Its primary GPU product line, labeled "GeForce", is in direct competition with Advanced Micro Devices' (AMD) "Radeon" products. Nvidia expanded its presence in the gaming industry with its handheld Shield Portable, Shield Tablet, and Shield Android TV and its cloud gaming service GeForce Now.

In addition to GPU manufacturing, Nvidia provides parallel processing capabilities to researchers and scientists that allow them to efficiently run high-performance applications. They are deployed in supercomputing sites around the world.[3][4] More recently, it has moved into the mobile computing market, where it produces Tegra mobile processors for smartphones and tablets as well as vehicle navigation and entertainment systems.[5][6][7] In addition to AMD, its competitors include Intel and Qualcomm.

Contents [hide]

1 History
  1.1 Major releases and acquisitions
  1.2 Class action lawsuit
  1.3 Apple/Nvidia web driver controversy
  1.4 Hardware Unboxed controversy
2 Finances
3 GPU Technology Conference
4 Product families
5 Open-source software support
6 Deep learning
  6.1 DGX
7 Inception Program
  7.1 2018 winners[117]
  7.2 2017 winners[117]
8 See also
9 Notes
10 References
11 External links

History [edit]

Nvidia was founded on April 5, 1993,[8][9][10] by Jensen Huang (CEO as of 2020), a Taiwanese American, previously director of CoreWare at LSI Logic and a microprocessor designer at Advanced Micro Devices (AMD), Chris Malachowsky, an electrical engineer who worked at Sun Microsystems, and Curtis Priem, previously a senior staff engineer and graphics chip designer at Sun Microsystems.

In 1993, the three co-founders believed that the proper direction for the next wave of computing was accelerated or graphics-based computing because it could solve problems that general-purpose computing could not. They also observed that video games were simultaneously one of the most computationally challenging problems and would have incredibly high sales volume. The two conditions don't happen very often. Video games became the company's flywheel to reach large markets and funding huge R&D to solve massive computational problems. With only $40,000 in the bank, the company was born.[11] The company subsequently received $20 million of venture capital funding from Sequoia Capital and others.[12] Nvidia initially had no name and the co-founders named all their files NV, as in "next version". The need to incorporate the company prompted the co-founders to review all words with those two letters, leading them to "invidia", the Latin word for "envy".[11] Nvidia went public on January 22, 1999.[13][14][15]

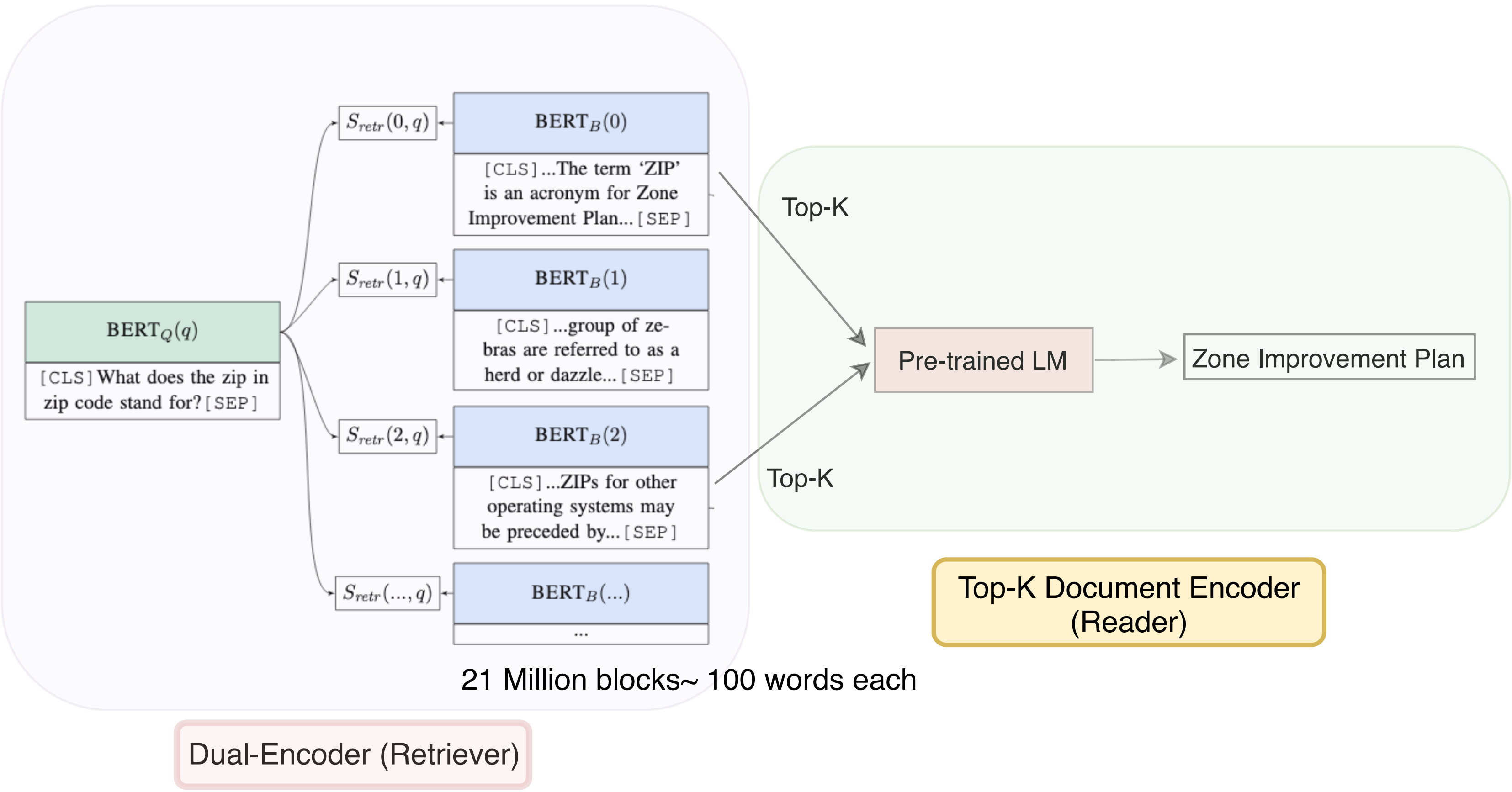
Answer Extraction

GeForce GTX 965 M

Background: Neural Models for Open-Domain QA

Stage 1: Trainable Information Retrieval

Stage 2: Trainable Answer Extraction



$\text{BERT}_Q(q)$

[CLS] What does the zip in zip code stand for? [SEP]

$S_{retr}(0, q)$

$\text{BERT}_B(0)$

[CLS] ...The term 'ZIP' is an acronym for Zone Improvement Plan... [SEP]

$S_{retr}(1, q)$

$\text{BERT}_B(1)$

[CLS] ...group of zebras are referred to as a herd or dazzle... [SEP]

$S_{retr}(2, q)$

$\text{BERT}_B(2)$

[CLS] ...ZIPs for other operating systems may be preceded by... [SEP]

$S_{retr}(..., q)$

$\text{BERT}_B(...)$

...

Top-K

Top-K

Pre-trained LM

Zone Improvement Plan

Top-K Document Encoder (Reader)

21 Million blocks~ 100 words each

Dual-Encoder (Retriever)

Image from "Lee et al., 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering"

EMDR^2: End-to-End Training of Multi-Document Reader and Retriever

Modeling Components

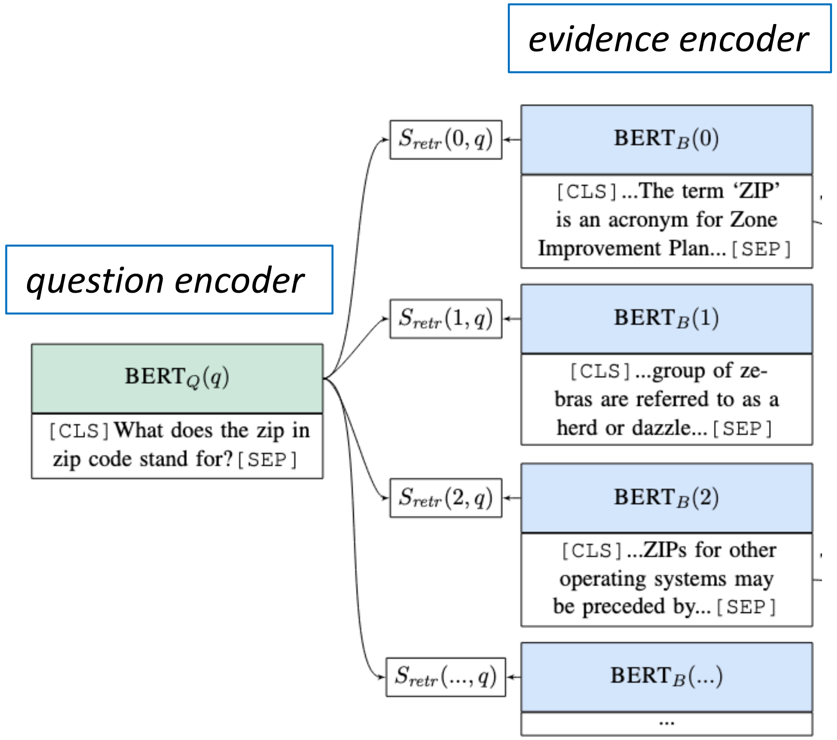
- **Retriever**: Dual Encoder
- **Reader / Answer Extractor**: Fusion-in-Decoder (FiD)

EMDR^2: End-to-End Training of Multi-Document Reader and Retriever

Modeling Components

- **Retriever**: **Dual Encoder**
- **Reader / Answer Extractor**: Fusion-in-Decoder (FiD)

Dual Encoder Retriever



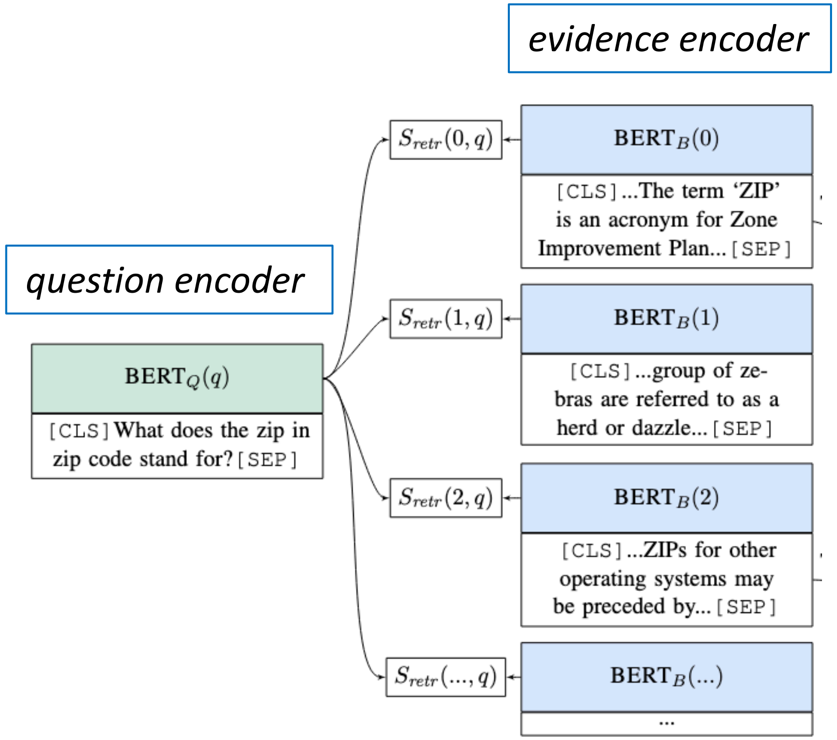
evidence encoder

question encoder

$\text{BERT}_Q(q)$
[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]
$\text{BERT}_B(...)$
...

Image from "Lee et al., 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering"

Dual Encoder Retriever



evidence encoder

question encoder

[CLS] What does the zip in zip code stand for? [SEP]

[CLS] ...The term ‘ZIP’ is an acronym for Zone Improvement Plan... [SEP]

[CLS] ...group of zebras are referred to as a herd or dazzle... [SEP]

[CLS] ...ZIPs for other operating systems may be preceded by... [SEP]

Evidence Documents

$$\mathcal{D} = \{\boldsymbol{d}_1, \ldots, \boldsymbol{d}_M\}$$

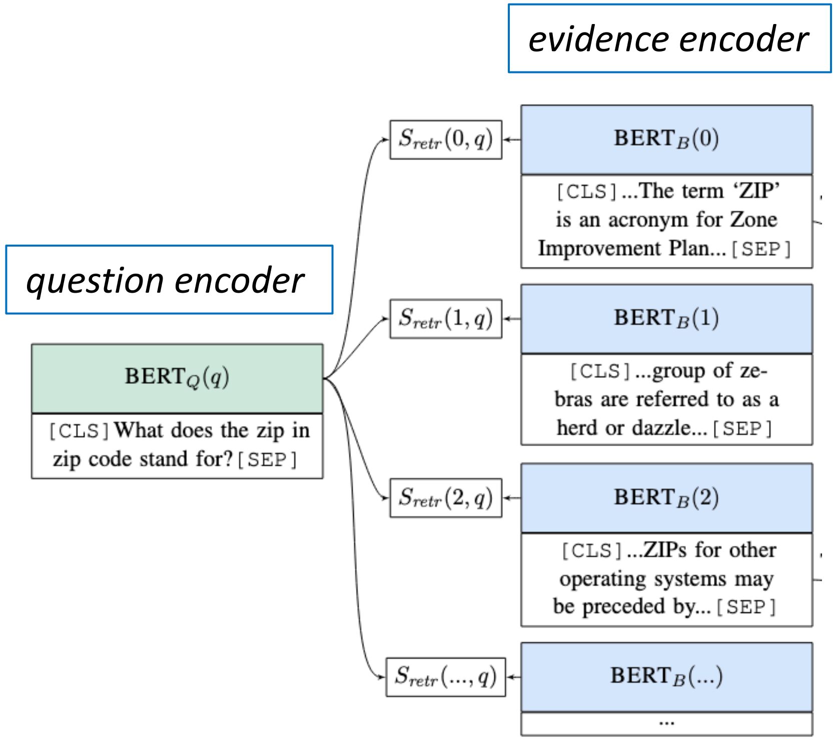
$$\text{score}(\boldsymbol{q}, \boldsymbol{d}_i; \Phi) = f_q(\boldsymbol{q}; \Phi_q)^\top f_d(\boldsymbol{d}_i; \Phi_d)$$

Select Top-K Documents with highest scores

$$\mathcal{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_K\}$$

Image from “Lee et al., 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering”

Dual Encoder Retriever


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

Evidence Documents

$$\mathcal{D} = \{\boldsymbol{d}_1, \ldots, \boldsymbol{d}_M\}$$

$$\mathrm{score}(\boldsymbol{q}, \boldsymbol{d}_i; \Phi) = f_q(\boldsymbol{q}; \Phi_q)^\top f_d(\boldsymbol{d}_i; \Phi_d)$$

Select Top-K Documents with highest scores

$$\mathcal{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_K\}$$

Dual encoder is initialized with **Inverse Cloze Task (ICT)**

Image from "Lee et al., 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering"

Evidence Documents

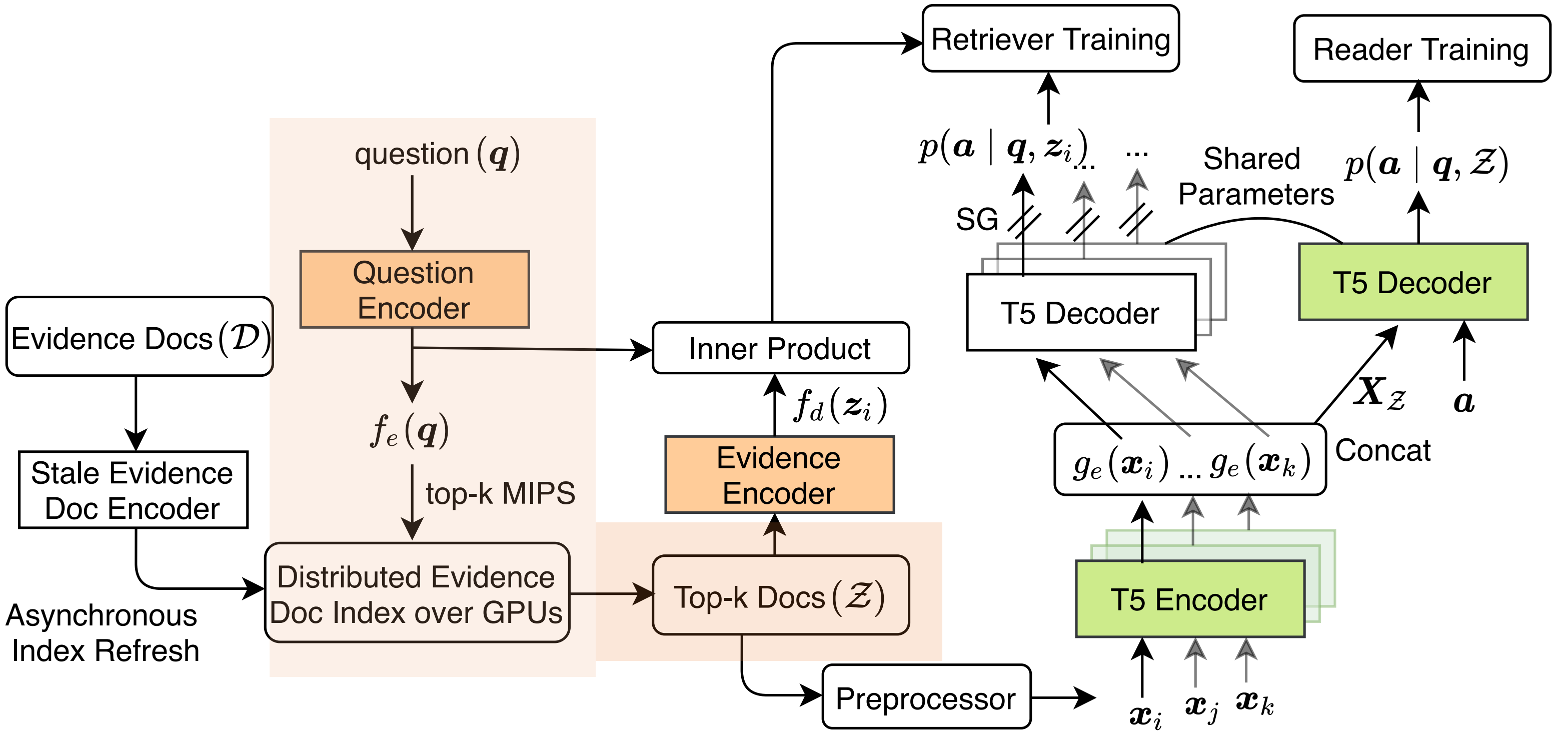
- **Evidence**: document collection containing world knowledge.
- We use **English Wikipedia** from Dec 2018 as evidence.
- Split articles into **100 words** long sequences.
  - Shorter sequences -> higher retrieval accuracy
- Overall size = 21 Million sequences

WIKIPEDIA
The Free Encyclopedia

Top-K Documents Retrieval

- **Pre-compute** evidence embeddings with context encoder.
- Distributed evidence storage over 16 GPUs.
  - 1.3 M document embeddings stored in each GPU
- We perform **online retrieval** at every step.
- Retrieval by asynchronous matrix multiplication in multiple GPUs.

EMDR^2: Top-K Retrieval



question $(\boldsymbol{q})$
Question Encoder
$f_e(\boldsymbol{q})$
top-k MIPS
Evidence Docs $(\mathcal{D})$
Stale Evidence Doc Encoder
Asynchronous Index Refresh
Distributed Evidence Doc Index over GPUs
Inner Product
$f_d(\boldsymbol{z}_i)$
Evidence Encoder
Top-k Docs $(\mathcal{Z})$
Preprocessor
Retriever Training
$p(\boldsymbol{a} \mid \boldsymbol{q}, \boldsymbol{z}_i)$ ... ...
SG
Shared Parameters
T5 Decoder
Reader Training
$p(\boldsymbol{a} \mid \boldsymbol{q}, \mathcal{Z})$
T5 Decoder
$\boldsymbol{X}_{\mathcal{Z}}$
$\boldsymbol{a}$
Concat
$g_e(\boldsymbol{x}_i)$ ... $g_e(\boldsymbol{x}_k)$
T5 Encoder
$\boldsymbol{x}_i$
$\boldsymbol{x}_j$
$\boldsymbol{x}_k$

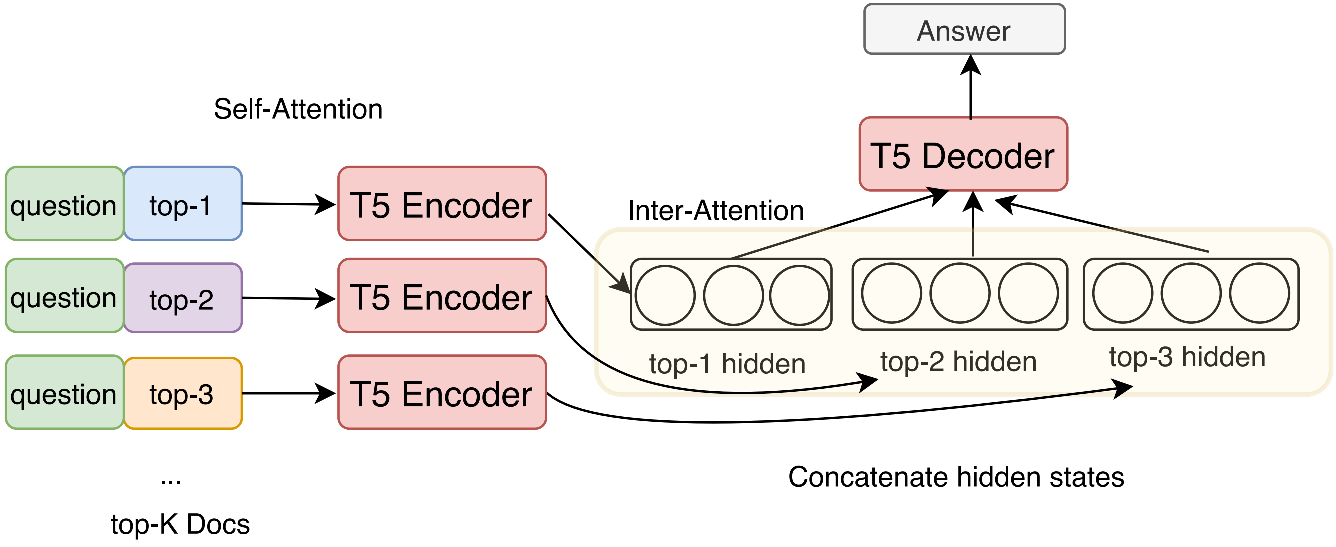
EMDR^2: End-to-End Training of Multi-Document Reader and Retriever

Modeling Components

- **Retriever**: Dual Encoder
- **Reader / Answer Extractor**: **Fusion-in-Decoder (FiD)**

Multi-Document Reader: Fusion-in-Decoder

FiD: generative approach for answer extraction based on T5



Encoder Input

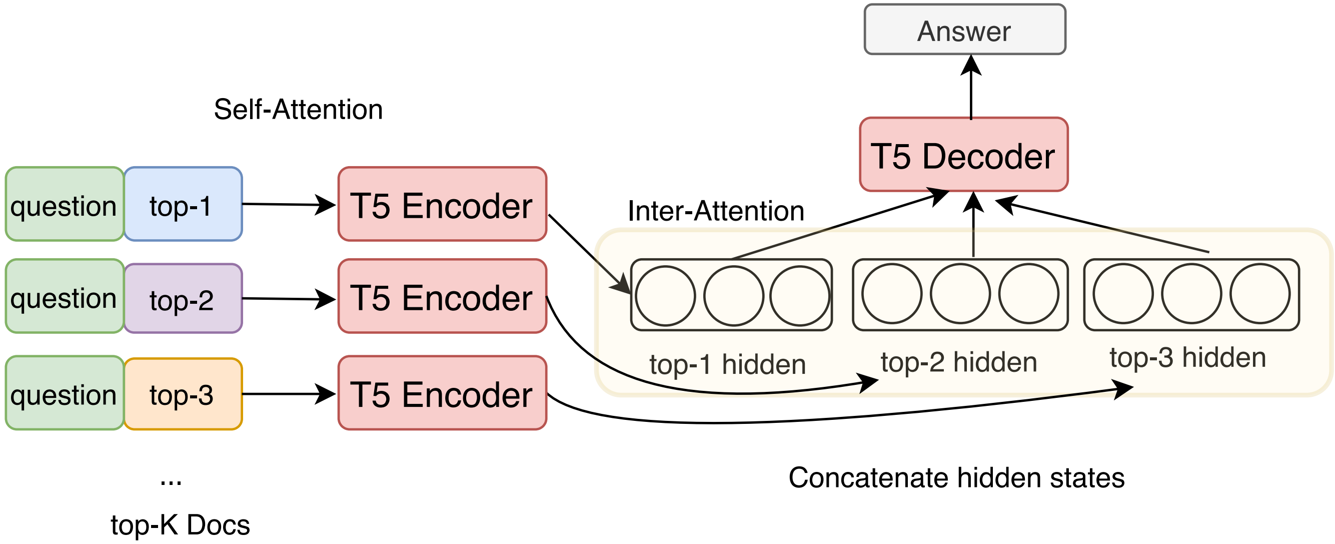
for each Top-K doc:

$$x_k = \texttt{[CLS]}\, q\, \texttt{[SEP]}\, t_{z_k}\, \texttt{[SEP]}\, z_k\, \texttt{[SEP]}$$

q = question
z_k = top-K document
t_z = title of top-K document

"Izacard et al., 2021. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering"

Fusion-in-Decoder: Self-Attention

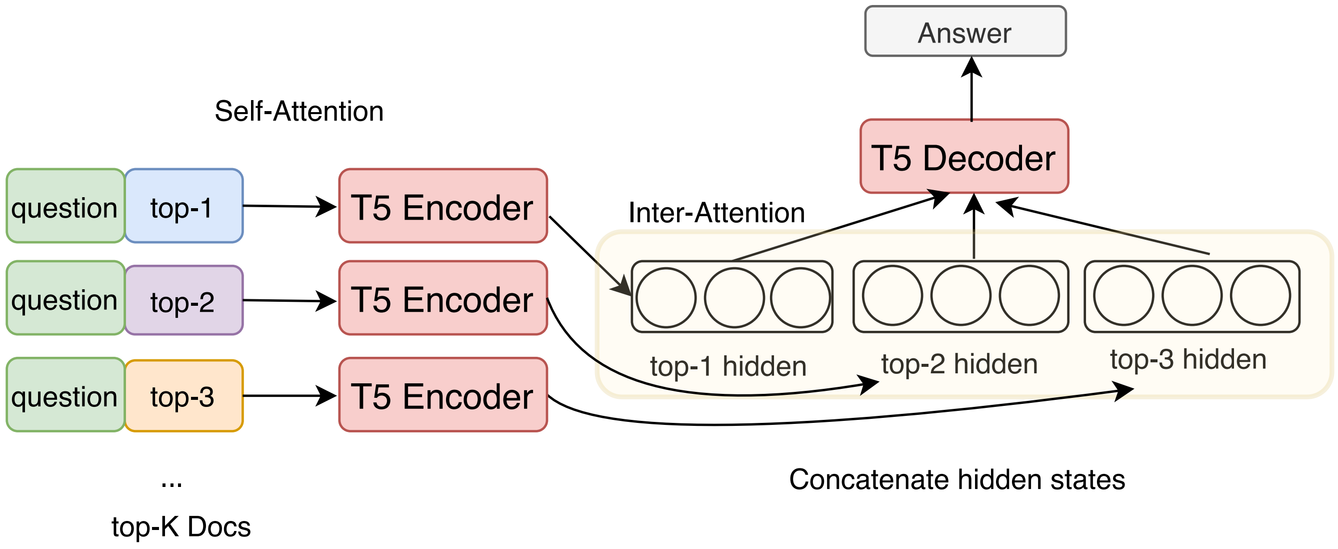


$$\boldsymbol{x}_k = [\mathrm{CLS}]\,\boldsymbol{q}\,[\mathrm{SEP}]\,\boldsymbol{t}_{\boldsymbol{z}_k}\,[\mathrm{SEP}]\,\boldsymbol{z}_k\,[\mathrm{SEP}]$$

Independent self-attention over each x_k

"Izacard et al., 2021. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering"

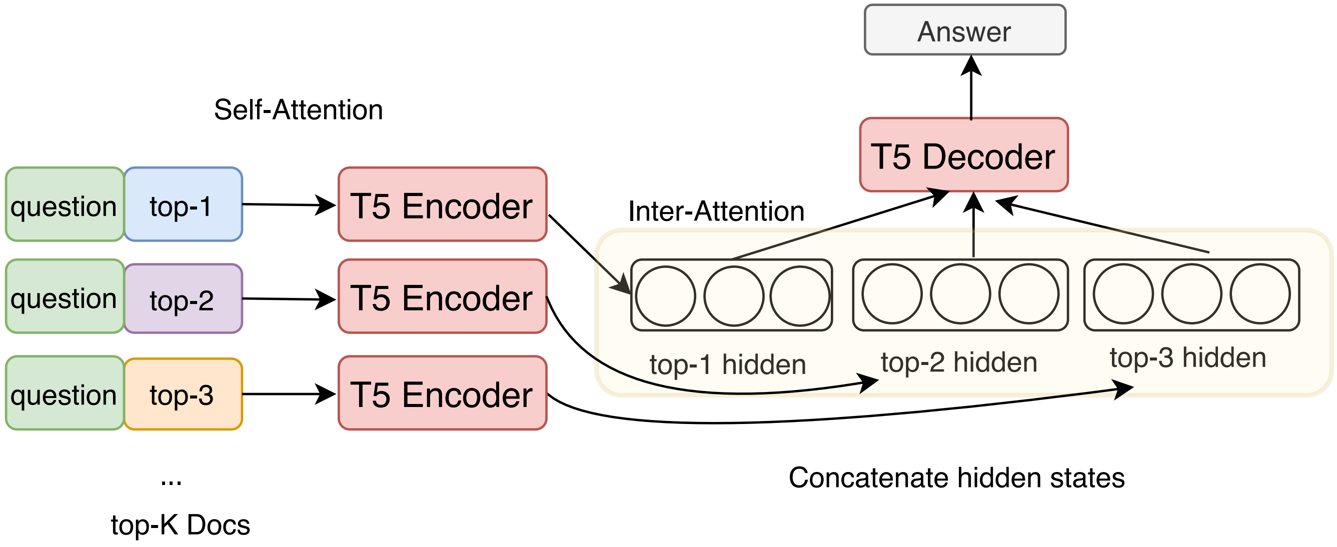
Fusion-in-Decoder: Inter-Attention



Concatenate the encoder representations for decoder’s inter-attention

$$\mathbf{X}_{\mathcal{Z}} = [g_e(\boldsymbol{x}_1); \ldots ; g_e(\boldsymbol{x}_K)] \in \mathbb{R}^{(N \times K) \times H}$$

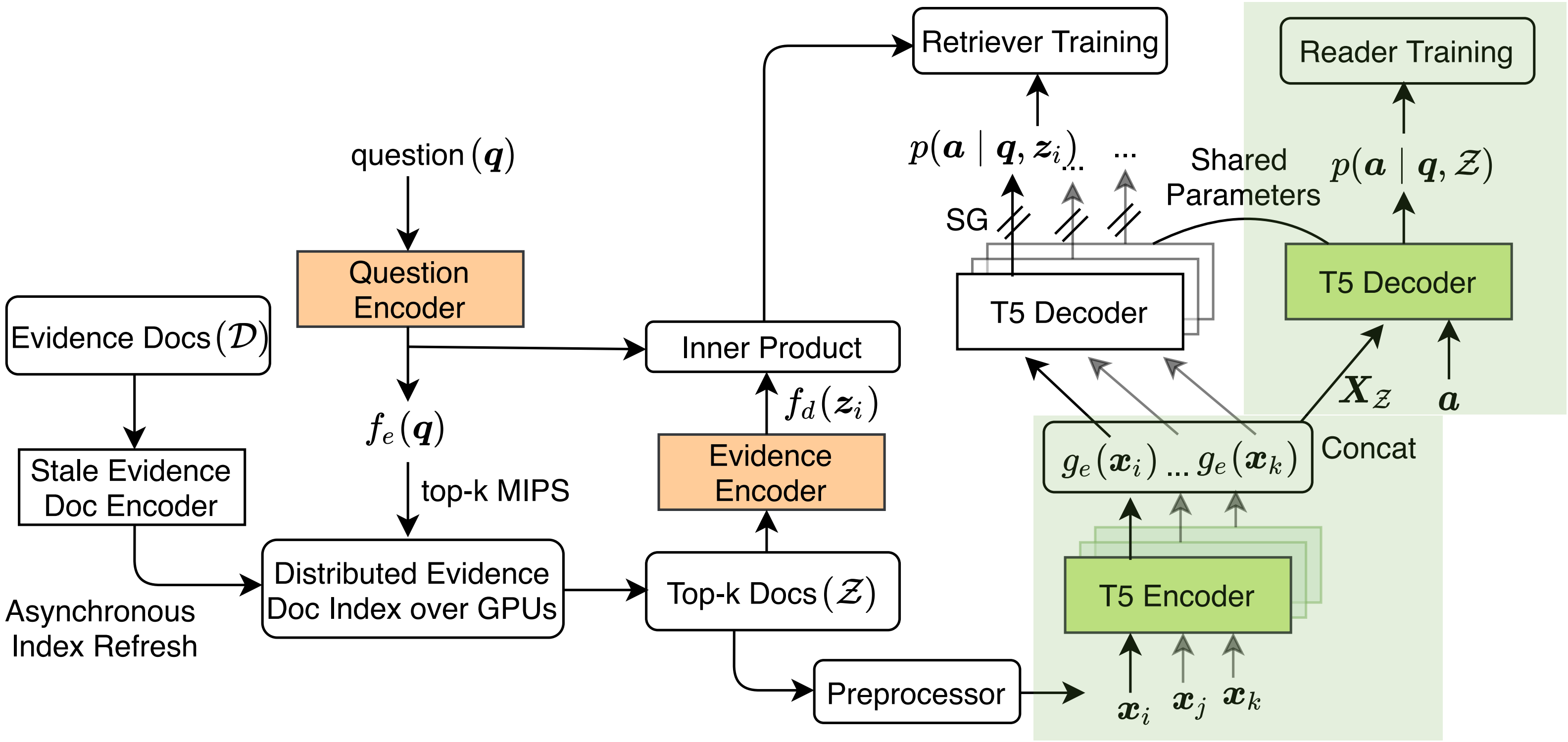
Fusion-in-Decoder: Training



Autoregressively train the model

$$p(\boldsymbol{a} \mid \boldsymbol{q}, \mathcal{Z}; \Theta) = \prod_{t=1}^{T} p\left(a_t \mid \boldsymbol{a}_{<t}, \boldsymbol{q}, \mathcal{Z}; \Theta\right)$$

EMDR^2: Fusion-in-Decoder



Retriever Training
Reader Training
question $(\boldsymbol{q})$
Question Encoder
Evidence Docs $(\mathcal{D})$
Inner Product
$p(\boldsymbol{a} \mid \boldsymbol{q}, \boldsymbol{z}_i)$ ... ...
SG
Shared Parameters
$p(\boldsymbol{a} \mid \boldsymbol{q}, \mathcal{Z})$
T5 Decoder
T5 Decoder
$f_e(\boldsymbol{q})$
$f_d(\boldsymbol{z}_i)$
Stale Evidence Doc Encoder
Evidence Encoder
$\boldsymbol{X}_{\mathcal{Z}}$
$\boldsymbol{a}$
top-k MIPS
$g_e(\boldsymbol{x}_i)$ ... $g_e(\boldsymbol{x}_k)$
Concat
Distributed Evidence Doc Index over GPUs
Top-k Docs $(\mathcal{Z})$
T5 Encoder
Asynchronous Index Refresh
Preprocessor
$\boldsymbol{x}_i$ $\boldsymbol{x}_j$ $\boldsymbol{x}_k$

End-to-End Training: Formulation

Marginal Likelihood

$$p(\boldsymbol{a} \mid \boldsymbol{q}; \Theta, \Phi) = \sum_{\mathscr{Z} \in \mathscr{S}} \underbrace{p(\boldsymbol{a} \mid \boldsymbol{q}, \mathscr{Z}; \Theta)}_{\text{FiD}} \underbrace{p(\mathscr{Z} \mid \boldsymbol{q}; \Phi)}_{\text{Dual Encoder}}$$

- The **set** of retrieved documents $\mathscr{Z}$ is a latent variable.

End-to-End Training: Formulation

Marginal Likelihood

$$p(\boldsymbol{a} \mid \boldsymbol{q}; \Theta, \Phi) = \sum_{\mathcal{Z} \in \mathscr{S}} \underbrace{p(\boldsymbol{a} \mid \boldsymbol{q}, \mathcal{Z}; \Theta)}_{\text{FiD}} \underbrace{p(\mathcal{Z} \mid \boldsymbol{q}; \Phi)}_{\text{Dual Encoder}}$$

- The **set** of retrieved documents $\mathcal{Z}$ is a latent variable.
- All possible values of $\mathcal{Z}$ are **combinatorial** in nature.

$$\mathscr{S} = \binom{M}{K}$$

End-to-End Training: Formulation

For **one** particular value of $\mathcal{Z}$, log-likelihood becomes

$$\begin{aligned}\log p(\boldsymbol{a} \mid \boldsymbol{q}; \Theta) &\approx \log p(\boldsymbol{a} \mid \boldsymbol{q}, \mathcal{Z}; \Theta) p(\mathcal{Z} \mid \boldsymbol{q}; \Phi) \\ &\approx \log p(\boldsymbol{a} \mid \boldsymbol{q}, \mathcal{Z}; \Theta) + \log p(\mathcal{Z} \mid \boldsymbol{q}; \Phi)\end{aligned}$$

End-to-End Training: Formulation

Log-Likelihood

$$\mathcal{L} = \log \underbrace{p(\boldsymbol{a} \mid \boldsymbol{q}, \mathcal{Z}_{\text{reader}}; \Theta)}_{\text{FiD training}} + \log \underbrace{p(\mathcal{Z}_{\text{retriever}} \mid \boldsymbol{q}; \Phi)}_{\text{Dual Encoder training}}$$

EMDR^2: FiD Training

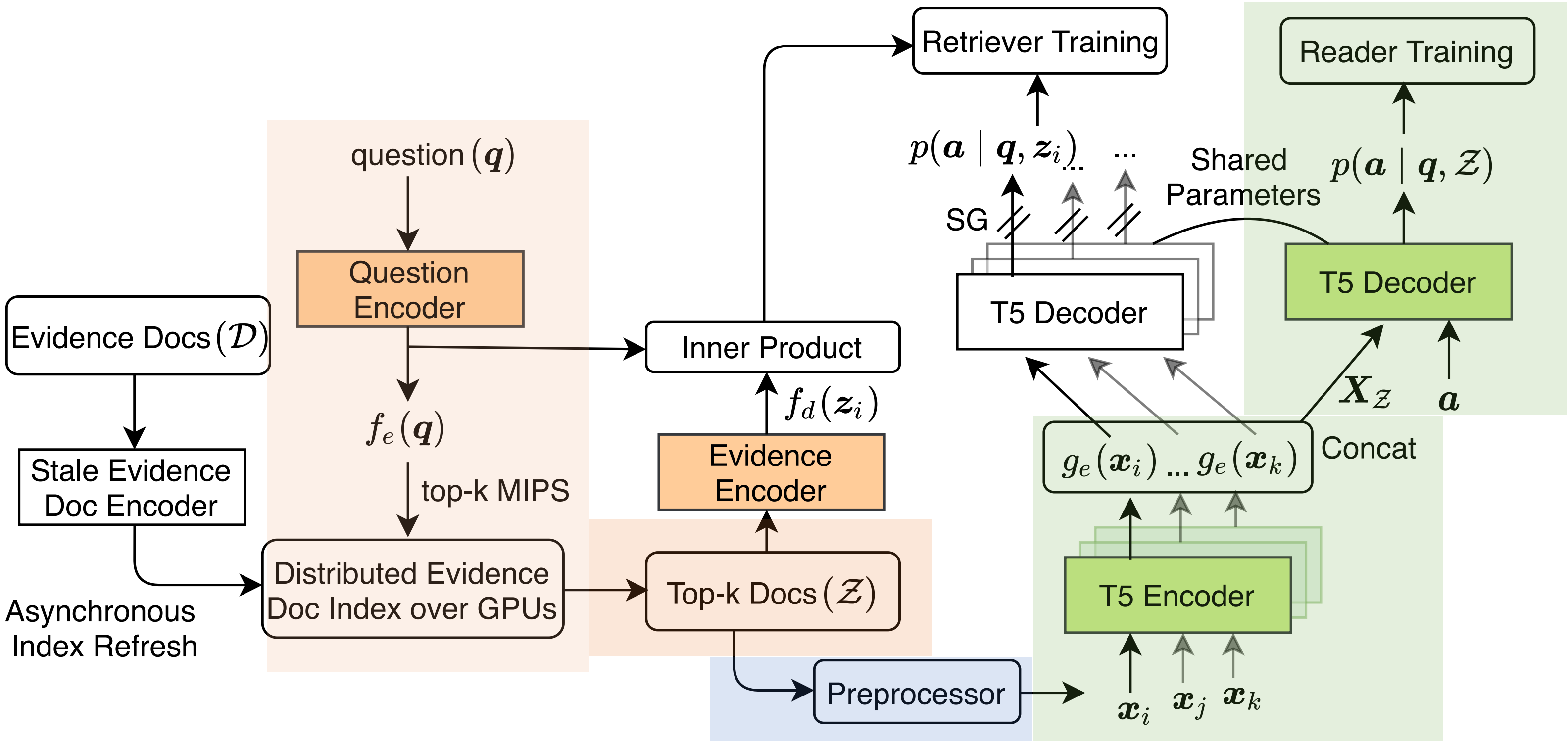
$$\mathcal{L} = \log \underbrace{p(\boldsymbol{a} \mid \boldsymbol{q}, \mathcal{Z}_{\text{reader}}; \Theta)}_{\text{FiD training}} + \log \underbrace{p(\mathcal{Z}_{\text{retriever}} \mid \boldsymbol{q}; \Phi)}_{\text{Dual Encoder training}}$$

- Obtain top-K documents of $\mathcal{Z}$ based on Maximum Inner Product Search (MIPS)

$$\text{score}(\boldsymbol{q}, \boldsymbol{d}_i; \Phi) = f_q(\boldsymbol{q}; \Phi_q)^\top f_d(\boldsymbol{d}_i; \Phi_d)$$

- Teacher-forcing training of FiD

EMDR^2: FiD Training



Retriever Training
Reader Training
question $(\boldsymbol{q})$
Question Encoder
Evidence Docs $(\mathcal{D})$
Inner Product
$f_e(\boldsymbol{q})$
$f_d(\boldsymbol{z}_i)$
top-k MIPS
Stale Evidence Doc Encoder
Evidence Encoder
Distributed Evidence Doc Index over GPUs
Top-k Docs $(\mathcal{Z})$
Asynchronous Index Refresh
Preprocessor
$p(\boldsymbol{a} \mid \boldsymbol{q}, \boldsymbol{z}_i)$ ... ...
SG
Shared Parameters
$p(\boldsymbol{a} \mid \boldsymbol{q}, \mathcal{Z})$
T5 Decoder
T5 Decoder
$\boldsymbol{X}_{\mathcal{Z}}$
$\boldsymbol{a}$
$g_e(\boldsymbol{x}_i)$ ... $g_e(\boldsymbol{x}_k)$
Concat
T5 Encoder
$\boldsymbol{x}_i$ $\boldsymbol{x}_j$ $\boldsymbol{x}_k$

EMDR^2: Retriever Training

$$\mathcal{L} = \log \underbrace{p(\boldsymbol{a} \mid \boldsymbol{q}, \mathcal{Z}_{\text{reader}}; \Theta)}_{\text{FiD training}} + \log \underbrace{p(\mathcal{Z}_{\text{retriever}} \mid \boldsymbol{q}; \Phi)}_{\text{Dual Encoder training}}$$

- Just optimizing the second term leads to poor results
- We need some form of supervision for retriever training, which comes from the answer (a)

EMDR^2: Retriever Training

We use posterior as it contains dependence on answer (a)

Simplifying Assumption: max probability of a set = max of the total probability of its elements

$$\max p(\mathcal{Z}_{\text{retriever}} \mid \boldsymbol{q}, \boldsymbol{a}; \Theta, \Phi) = \max \sum_{k=1}^{K} p(\boldsymbol{z}_k \mid \boldsymbol{q}, \boldsymbol{a}; \Theta, \Phi)$$

EMDR^2: Retriever Training

We use posterior for better training as it contains dependence on answer (a)

From **Conditional Bayes Rule:**

$$p(\boldsymbol{z}_k \mid \boldsymbol{q}, \boldsymbol{a}; \Theta, \Phi) \propto \underbrace{p(\boldsymbol{a} \mid \boldsymbol{q}, \boldsymbol{z}_k; \Theta)}_{\text{T5}} \underbrace{p(\boldsymbol{z}_k \mid \boldsymbol{q}; \Phi)}_{\text{Dual Encoder}}$$

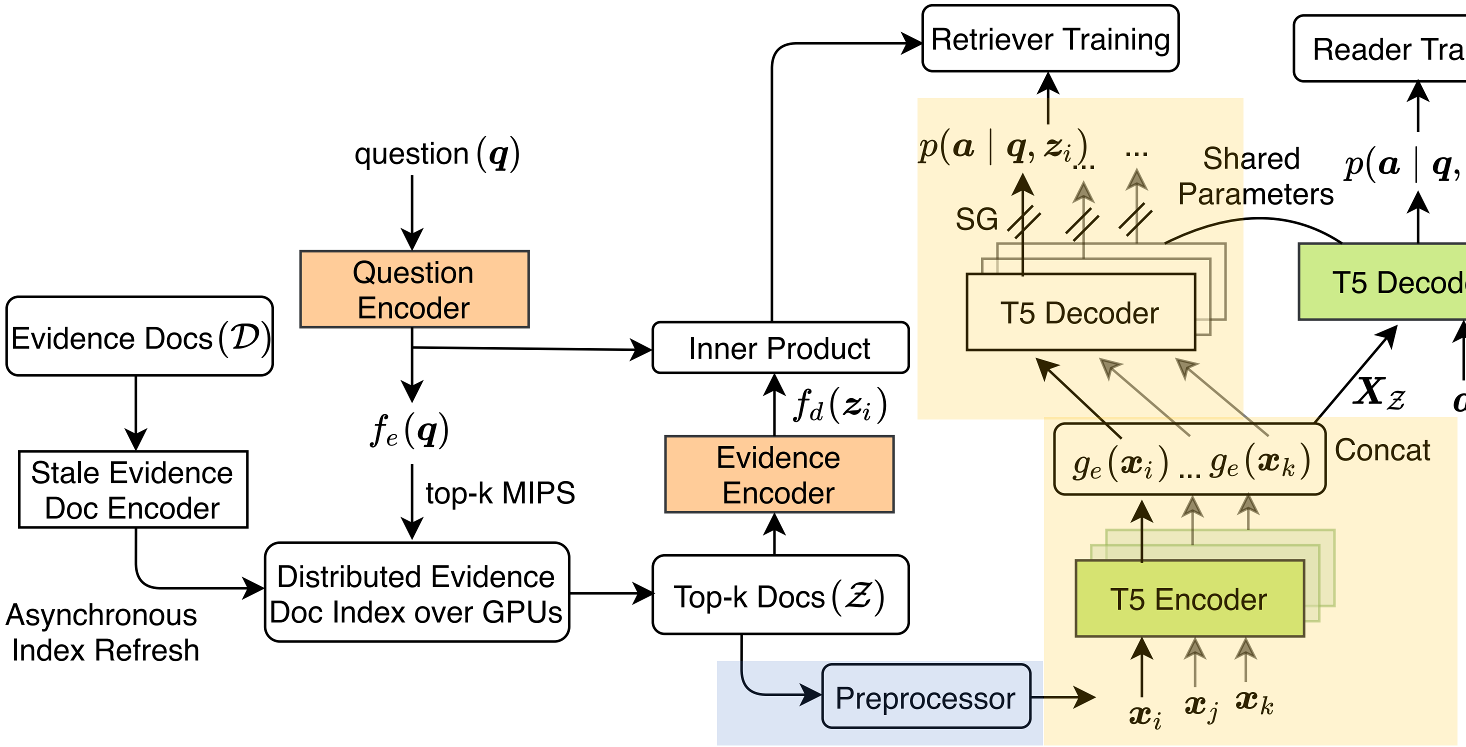
EMDR^2: Retriever Training

From **Conditional Bayes Rule:**

$$p(\boldsymbol{z}_k \mid \boldsymbol{q}, \boldsymbol{a}; \Theta, \Phi) \propto \underbrace{p(\boldsymbol{a} \mid \boldsymbol{q}, \boldsymbol{z}_k; \Theta)}_{\text{T5}} \underbrace{p(\boldsymbol{z}_k \mid \boldsymbol{q}; \Phi)}_{\text{Dual Encoder}}$$

First part can be computed from T5

EMDR^2: Retriever Training


Retriever Training
Reader Training
question $(\boldsymbol{q})$
Question Encoder
Evidence Docs $(\mathcal{D})$
Inner Product
$f_e(\boldsymbol{q})$
$f_d(\boldsymbol{z}_i)$
Stale Evidence Doc Encoder
top-k MIPS
Evidence Encoder
Distributed Evidence Doc Index over GPUs
Top-k Docs $(\mathcal{Z})$
Asynchronous Index Refresh
Preprocessor
$p(\boldsymbol{a} \mid \boldsymbol{q}, \boldsymbol{z}_i)$ ... ...
SG
Shared Parameters
$p(\boldsymbol{a} \mid \boldsymbol{q}, \mathcal{Z})$
T5 Decoder
T5 Decoder
$\boldsymbol{X}_{\mathcal{Z}}$
$\boldsymbol{a}$
$g_e(\boldsymbol{x}_i)$ ... $g_e(\boldsymbol{x}_k)$
Concat
T5 Encoder
$\boldsymbol{x}_i$ $\boldsymbol{x}_j$ $\boldsymbol{x}_k$

EMDR^2: Retriever Training

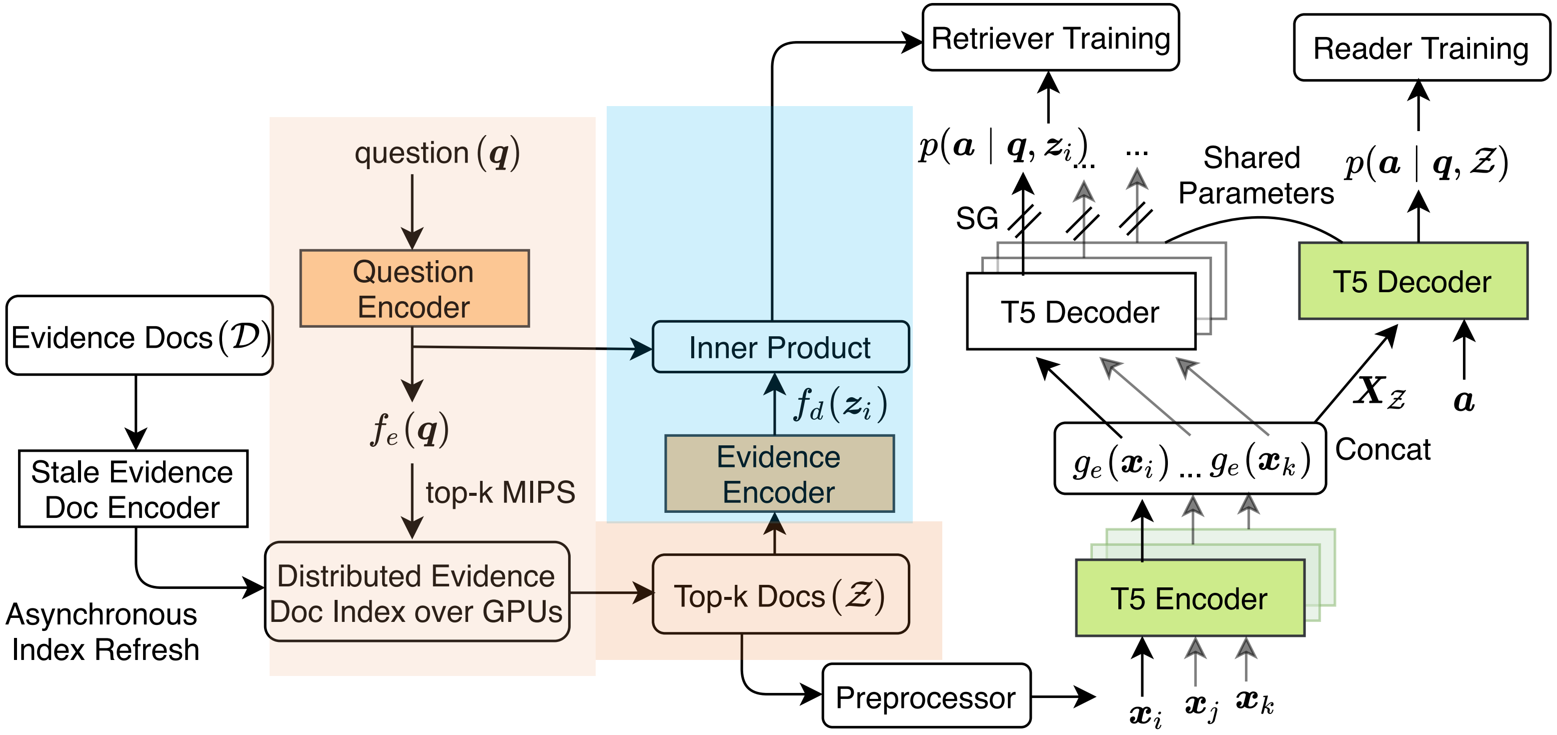
Probability of a document z_k from dual encoder

$$p(\boldsymbol{z}_k \mid \boldsymbol{q}, \boldsymbol{a}; \Theta, \Phi) \propto \underbrace{p(\boldsymbol{a} \mid \boldsymbol{q}, \boldsymbol{z}_k; \Theta)}_{\text{T5}} \underbrace{p(\boldsymbol{z}_k \mid \boldsymbol{q}; \Phi)}_{\text{Dual Encoder}}$$

Apply a softmax with temperature over the top-K scores

$$p(\boldsymbol{z}_k \mid \boldsymbol{q}, \mathcal{Z}_{\text{top-}K}; \Phi) \approx \frac{\exp(\text{score}(\boldsymbol{q}, \boldsymbol{z}_k)/\tau; \Phi)}{\sum_{j=1}^{K} \exp(\text{score}(\boldsymbol{q}, \boldsymbol{z}_j)/\tau; \Phi)}$$

EMDR^2: Retriever Training



Retriever Training
Reader Training
question $(\boldsymbol{q})$
Question Encoder
Evidence Docs $(\mathcal{D})$
Inner Product
$f_d(\boldsymbol{z}_i)$
Evidence Encoder
$f_e(\boldsymbol{q})$
top-k MIPS
Stale Evidence Doc Encoder
Distributed Evidence Doc Index over GPUs
Top-k Docs $(\mathcal{Z})$
Asynchronous Index Refresh
Preprocessor
$p(\boldsymbol{a} \mid \boldsymbol{q}, \boldsymbol{z}_i)$ ... ...
SG
Shared Parameters
$p(\boldsymbol{a} \mid \boldsymbol{q}, \mathcal{Z})$
T5 Decoder
T5 Decoder
$\boldsymbol{X}_{\mathcal{Z}}$
$\boldsymbol{a}$
Concat
$g_e(\boldsymbol{x}_i)$ ... $g_e(\boldsymbol{x}_k)$
T5 Encoder
$\boldsymbol{x}_i$ $\boldsymbol{x}_j$ $\boldsymbol{x}_k$

EMDR^2 Training Objective

$$\mathcal{L} = \underbrace{\log p(\boldsymbol{a} \mid \boldsymbol{q}, \mathcal{Z}_{\text{top-}K}; \Theta)}_{\text{T5 training}} + \underbrace{\log \sum_{k=1}^{K} \mathbb{SG}\left(p(\boldsymbol{a} \mid \boldsymbol{q}, \boldsymbol{z}_k; \Theta)\right) p(\boldsymbol{z}_k \mid \boldsymbol{q}, \mathcal{Z}_{\text{top-}K}; \Phi)}_{\text{Dual Encoder training}}$$

SG: Stop Gradient operation i.e., no backpropagation

EMDR^2 Training Objective: EM View

E Step:

1. Obtain top-K documents $\mathcal{Z}$ based on current retriever parameters

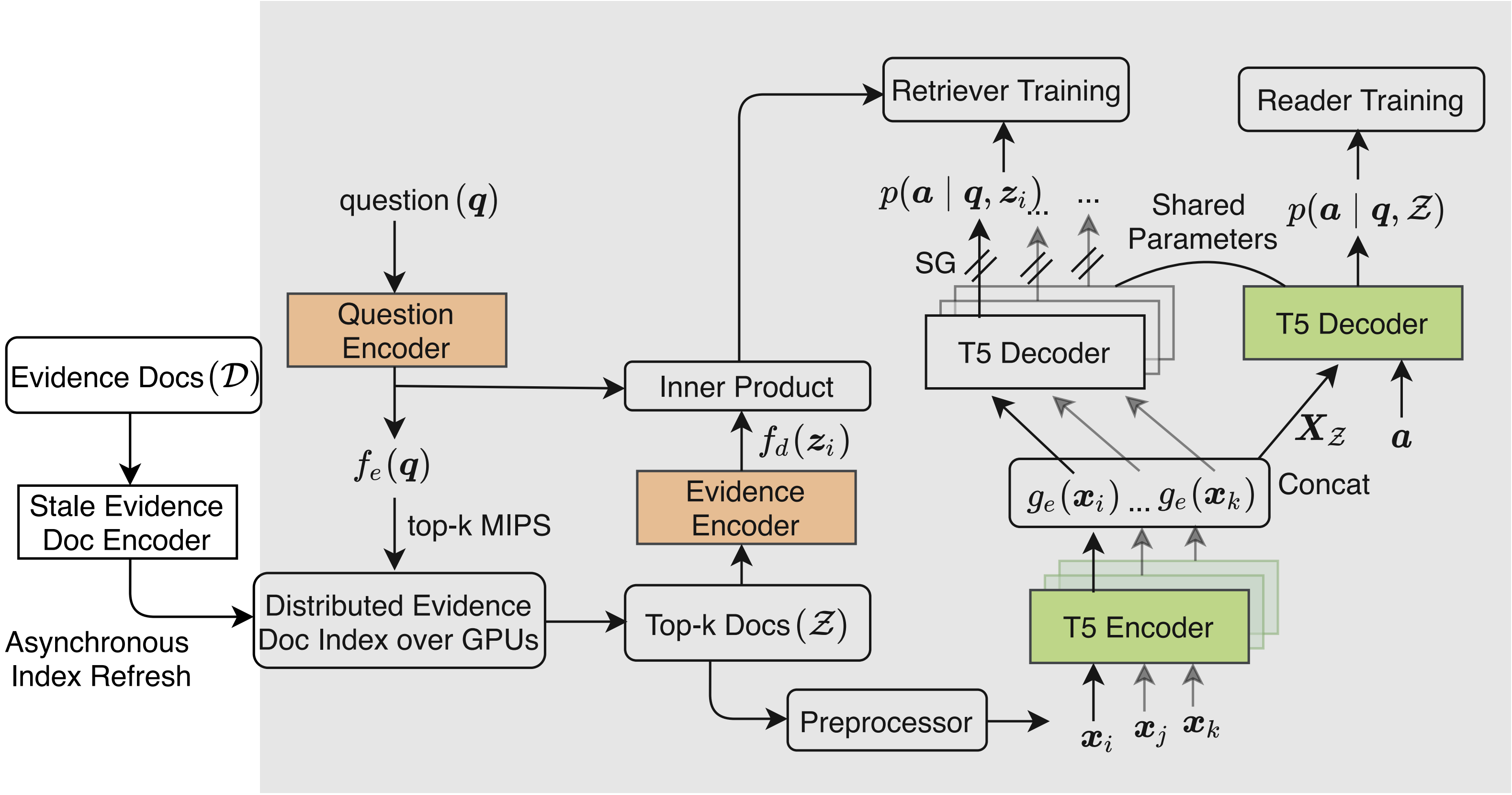
$$\text{score}(\boldsymbol{q}, \boldsymbol{d}_i; \Phi) = f_q(\boldsymbol{q}; \Phi_q)^\top f_d(\boldsymbol{d}_i; \Phi_d)$$

2. Obtain $p(\boldsymbol{a} \mid \boldsymbol{q}, \boldsymbol{z}_k; \Theta)$ based on current T5 parameters

M Step:

$$\mathcal{L} = \underbrace{\log p(\boldsymbol{a} \mid \boldsymbol{q}, \mathcal{Z}_{\text{top-}K}; \Theta)}_{\text{T5 training}} + \underbrace{\log \sum_{k=1}^{K} \mathbb{SG}\left(p(\boldsymbol{a} \mid \boldsymbol{q}, \boldsymbol{z}_k; \Theta)\right) p(\boldsymbol{z}_k \mid \boldsymbol{q}, \mathcal{Z}_{\text{top-}K}; \Phi)}_{\text{Dual Encoder training}}$$

EMDR^2: Modeling Components



Retriever Training
Reader Training
question ($\boldsymbol{q}$)
$p(\boldsymbol{a} \mid \boldsymbol{q}, \boldsymbol{z}_i)$ ... ...
Shared Parameters
$p(\boldsymbol{a} \mid \boldsymbol{q}, \mathcal{Z})$
SG
Question Encoder
T5 Decoder
T5 Decoder
Evidence Docs ($\mathcal{D}$)
Inner Product
$f_e(\boldsymbol{q})$
$f_d(\boldsymbol{z}_i)$
$\boldsymbol{X}_{\mathcal{Z}}$
$\boldsymbol{a}$
Stale Evidence Doc Encoder
Evidence Encoder
$g_e(\boldsymbol{x}_i)$ ... $g_e(\boldsymbol{x}_k)$
Concat
top-k MIPS
Distributed Evidence Doc Index over GPUs
Top-k Docs ($\mathcal{Z}$)
T5 Encoder
Asynchronous Index Refresh
Preprocessor
$\boldsymbol{x}_i$ $\boldsymbol{x}_j$ $\boldsymbol{x}_k$

EMDR^2: Other Implementation Details

- Framework: PyTorch
  - Implemented using “*megatron-lm*” toolkit

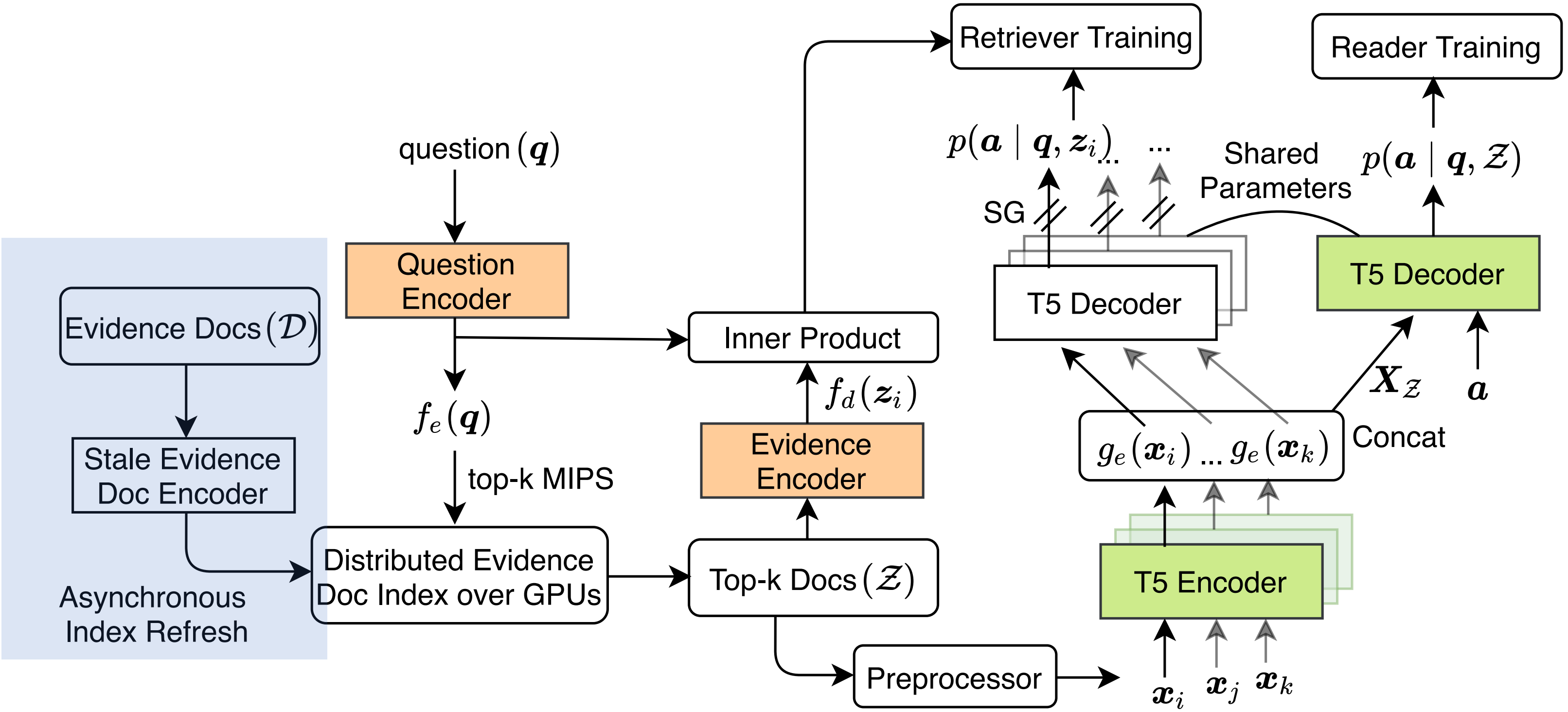
- **Compute**: 16 A100 GPUs, each with 40GB RAM

- 8 GPUs for model training (1st process group)

- 8 GPUs for asynchronous evidence indexing (2nd process group)
  - required because evidence embeddings get stale
  - performed every 500 training steps

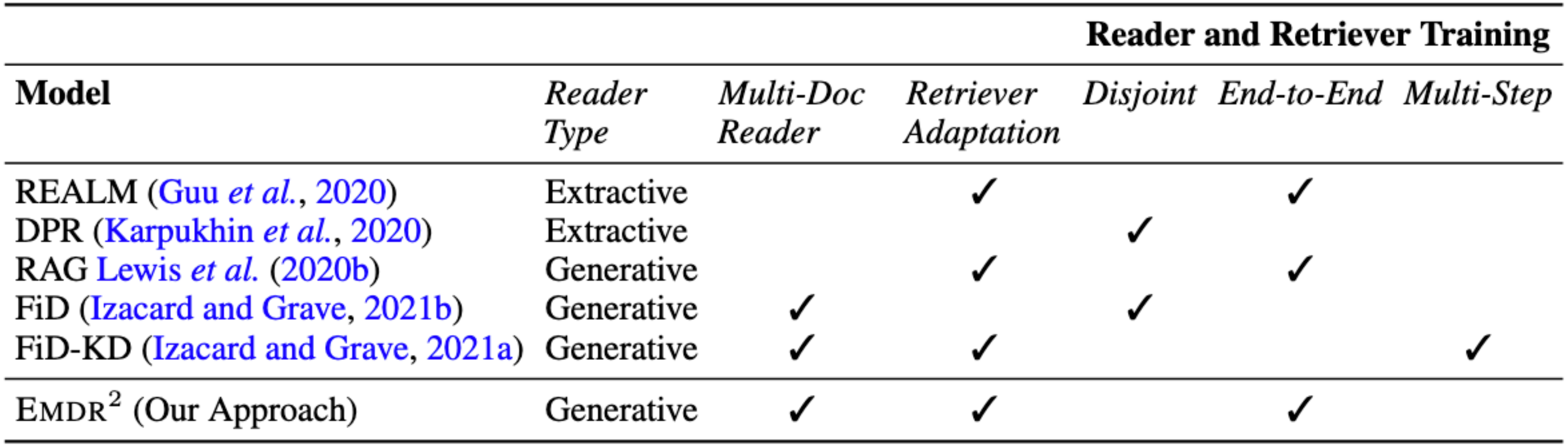
- All 16 GPUs for top-K document retrieval (3rd process group).

EMDR^2: Asynchronous Evidence Indexing



Retriever Training
Reader Training
$p(\boldsymbol{a} \mid \boldsymbol{q}, \boldsymbol{z}_i)$ ...
...
Shared Parameters
$p(\boldsymbol{a} \mid \boldsymbol{q}, \mathcal{Z})$
question $(\boldsymbol{q})$
SG
Question Encoder
T5 Decoder
T5 Decoder
Evidence Docs $(\mathcal{D})$
Inner Product
$f_d(\boldsymbol{z}_i)$
$\boldsymbol{X}_\mathcal{Z}$
$\boldsymbol{a}$
$f_e(\boldsymbol{q})$
Stale Evidence Doc Encoder
Evidence Encoder
$g_e(\boldsymbol{x}_i)$ ... $g_e(\boldsymbol{x}_k)$
Concat
top-k MIPS
Distributed Evidence Doc Index over GPUs
Top-k Docs $(\mathcal{Z})$
T5 Encoder
Asynchronous Index Refresh
Preprocessor
$\boldsymbol{x}_i$ $\boldsymbol{x}_j$ $\boldsymbol{x}_k$

Comparison of Open-Domain QA Approaches



				Reader and Retriever Training		
Model	*Reader Type*	*Multi-Doc Reader*	*Retriever Adaptation*	*Disjoint*	*End-to-End*	*Multi-Step*
REALM (Guu *et al.*, 2020)	Extractive		✓		✓	
DPR (Karpukhin *et al.*, 2020)	Extractive			✓		
RAG Lewis *et al.* (2020b)	Generative		✓		✓	
FiD (Izacard and Grave, 2021b)	Generative	✓		✓		
FiD-KD (Izacard and Grave, 2021a)	Generative	✓	✓			✓
EMDR^2 (Our Approach)	Generative	✓	✓		✓	

Experimental Setting

- Base configuration of T5 and BERT (768 dim hidden size)
- **Total parameters**: 440M (T5-220M + Retriever-220M)
- **Batch Size**: 64
- **Top-K Documents**: 50
- **Evaluation**: Exact Match (EM)

EMDR^2: Unsupervised Pre-Training

- Helps to improve initial retrieval accuracy.
- **Corpus**: sentences containing named entities from Wikipedia.
- **Masked Salient Spans (MSS)**
    - Question: sentence with named entities masked
    - Answer: named entities
- Train for 82K steps with ICT initialized retriever.

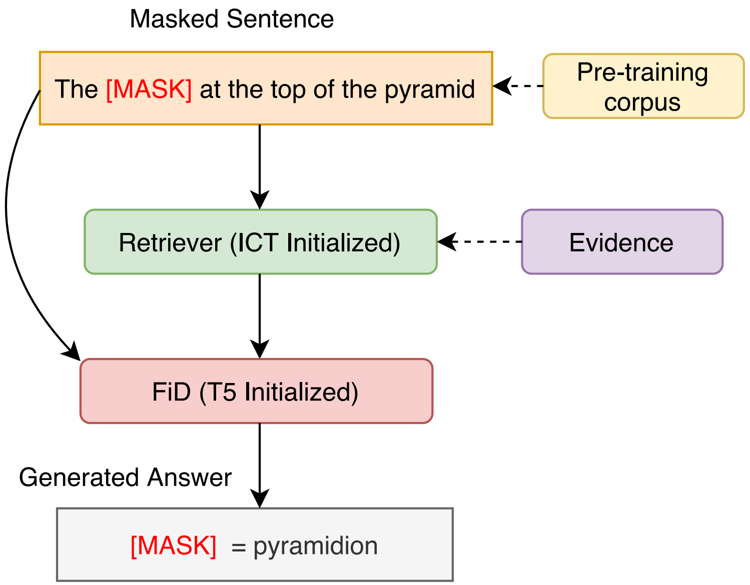


Figure derived from "Guu et al., 2020. REALM: Retrieval-Augmented Language Model Pre-Training"

Datasets

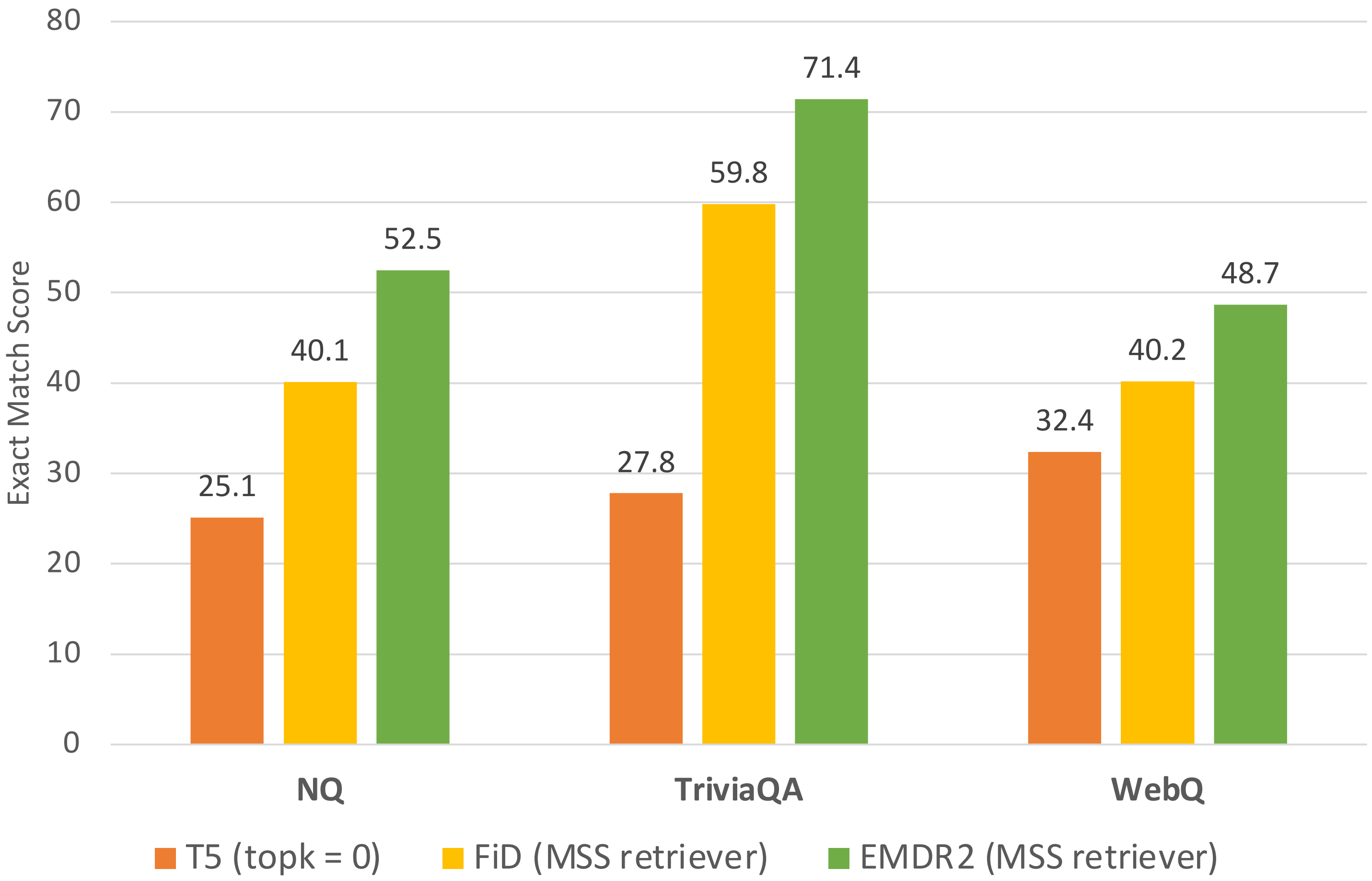
Dataset	Train	Dev	Test
WebQuestions (**WebQ**)	3,417	361	2,032
Natural Questions (**NQ**)	79,168	8,757	3,610
TriviaQA	78,785	8,837	11,313

WebQ: Questions were collected using Google Suggest API.
Freebase IDs in answers are replaced by entity names.

NQ: Real questions asked by users in Google. We use the subset of short answers.

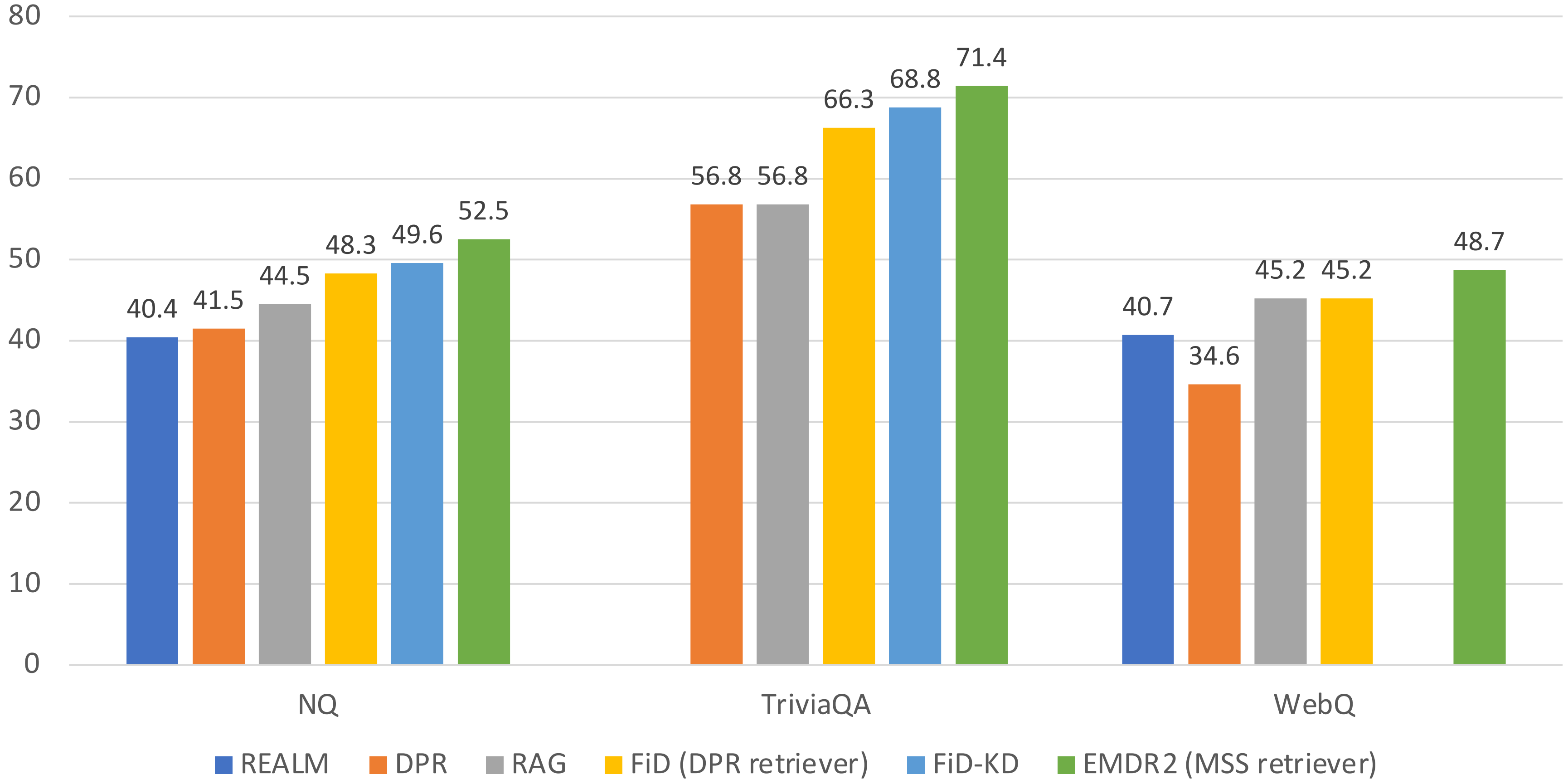
TriviaQA: Collection of trivia question-answer pairs collected from the web.

Results: EMDR^2 Training



End-to-end training provides good performance gains over FiD with MSS retriever

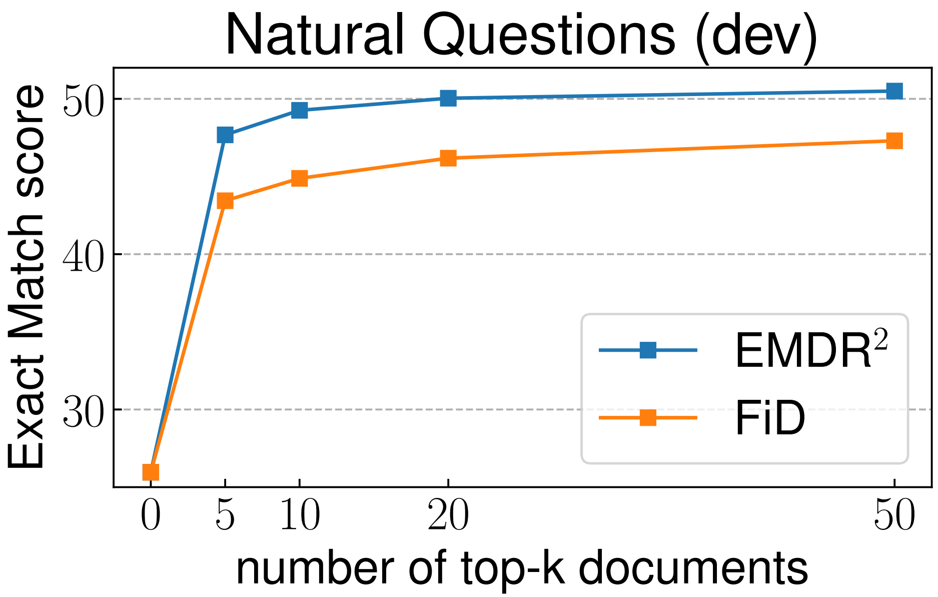
Comparison with Other Approaches

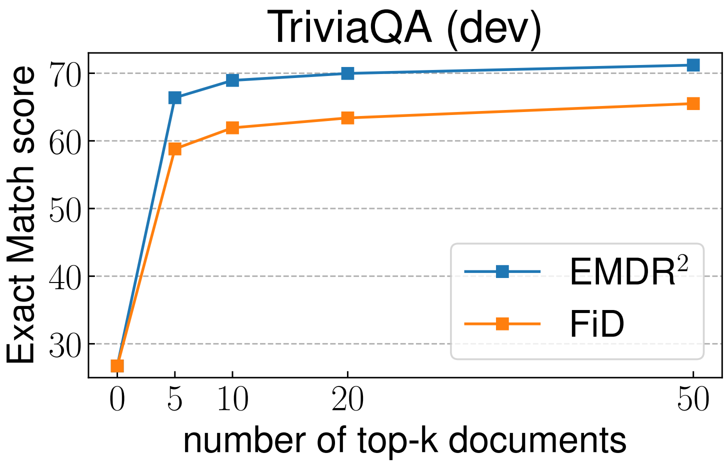


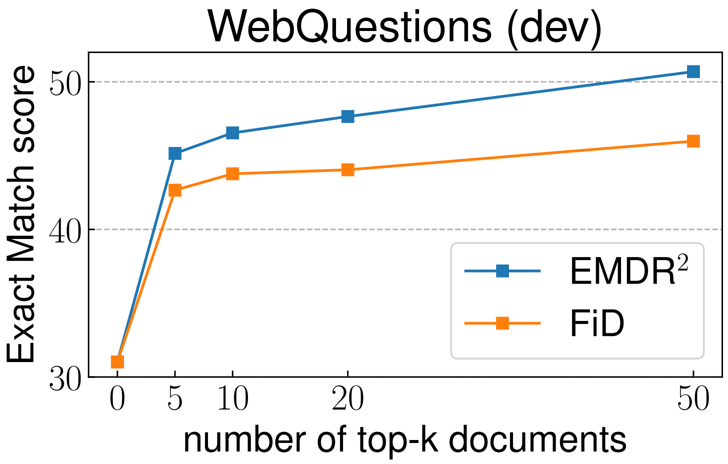
New SOTA Results of EMDR^2

2-3 EM points gain over FiD-KD

Analysis: Effect of the Number of Top-k Documents







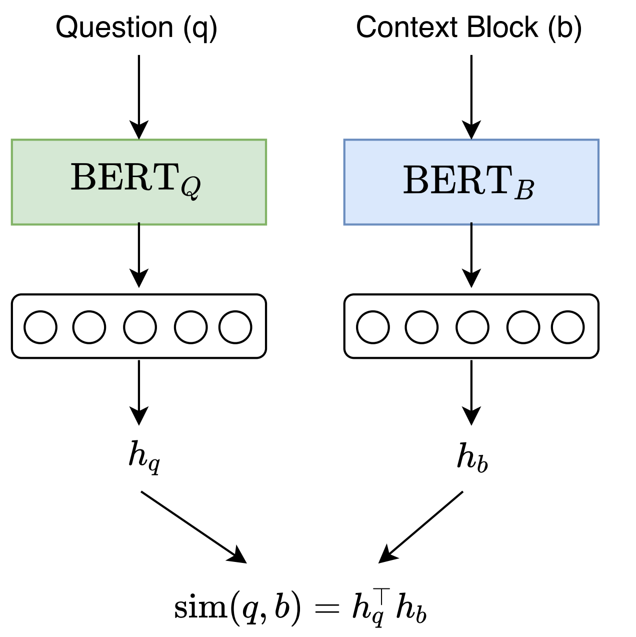
Effect of Retriever Initialization

Approaches compared:

1. Masked Salient Span (MSS) pre-training
2. Dense Passage Retrieval (DPR) training
3. MSS pre-training + DPR training

Review: Dense Passage Retrieval (DPR)

- Dual-encoder model
- Train from **supervised** question-context pairs



$$D = q_i,$$ Question

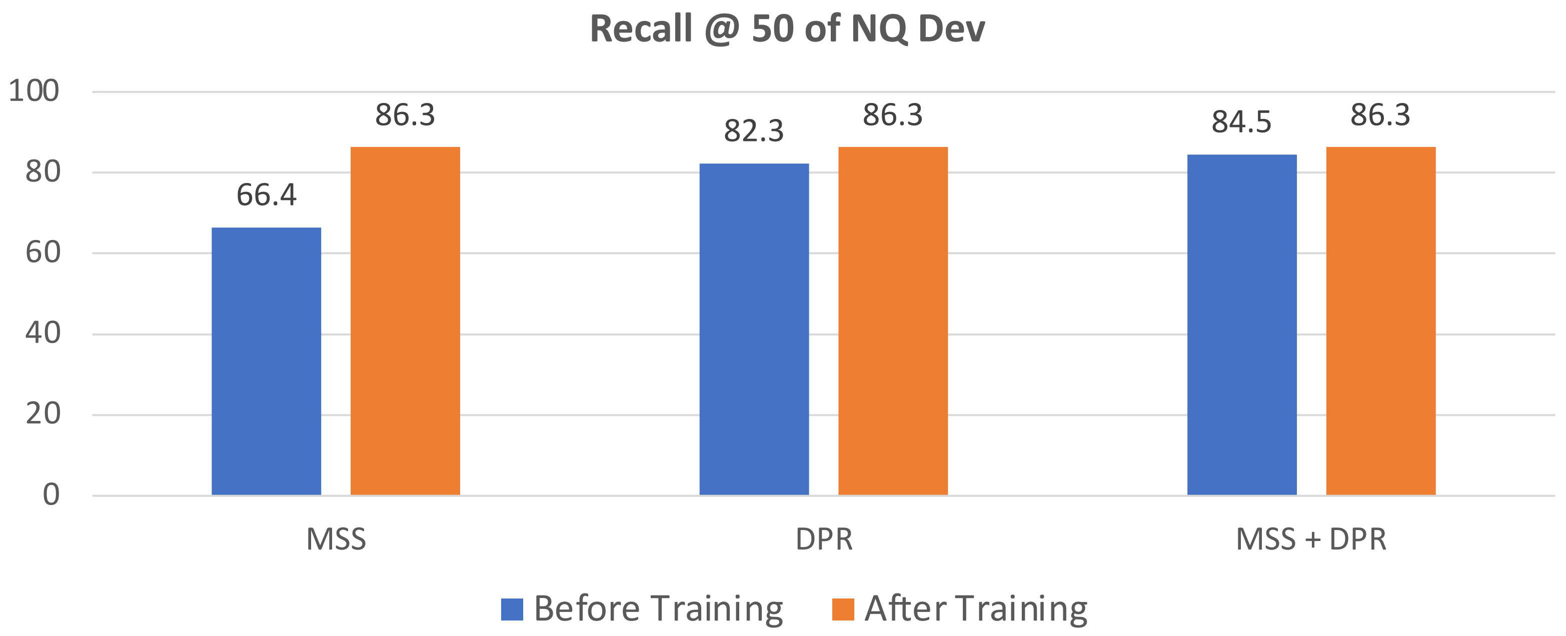
$$p_i^+,$$ One Positive Example

$$p_{i,j}^-$$ Set of Negative Examples

$$\boxed{L = -\log \frac{e^{\mathrm{sim}(q_i, b_i^+)}}{e^{\mathrm{sim}(q_i, b_i^+)} + \sum_{j=1}^{n} e^{\mathrm{sim}(q_i, b_{ij}^-)}}}$$

Karpukhin et al., 2020. Dense Passage Retrieval for Open-Domain Question Answering

Effect of Retriever Initialization



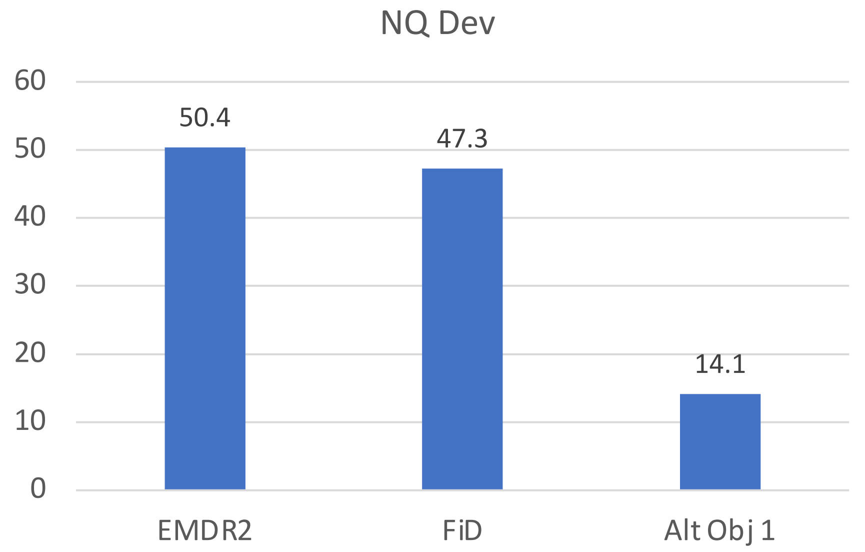
- MSS, DPR, and MSS + DPR retriever initialization results in the same final retrieval accuracy.
- Retriever training by DPR may not be essential for open-domain QA.

Alternative Training Objective 1

$$p(\mathcal{Z} \mid \boldsymbol{q}; \Phi) = \prod_{k=1}^{K} p(\boldsymbol{z}_k \mid \boldsymbol{q}; \Phi)$$

No feedback from FiD reader to retriever

$$\mathcal{L}_{\text{alt-1}} = \underbrace{\log p(\boldsymbol{a} \mid \boldsymbol{q}, \mathcal{Z}; \Theta)}_{\text{FiD reader}} + \underbrace{\sum_{k=1}^{K} \log p(\boldsymbol{z}_k \mid \boldsymbol{q}, \mathcal{Z}; \Phi)}_{\text{retriever}}$$

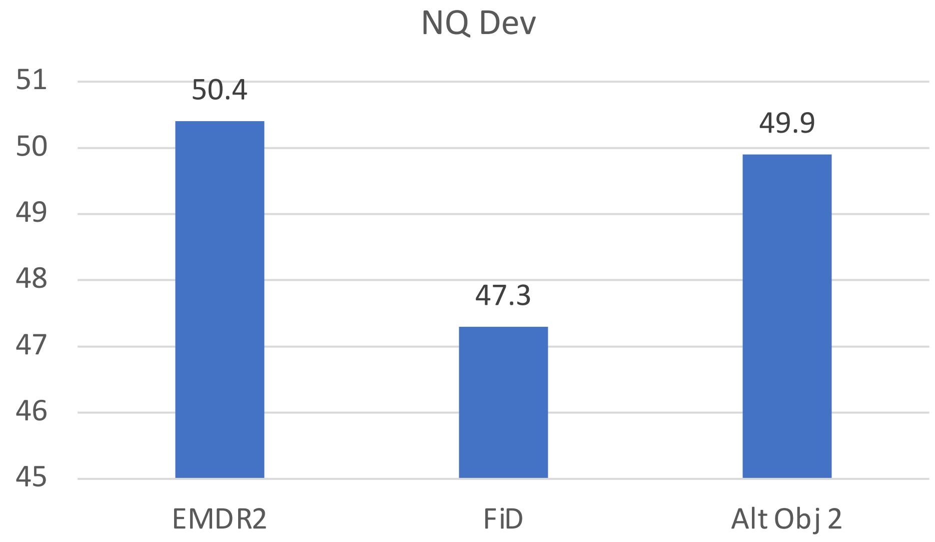


- Poor performance of Alt. Obj.
- Retriever gets stuck in a bad local optimum

Alternative Training Objective 2

$$\tilde{p}(\boldsymbol{a} \mid \boldsymbol{q}, \boldsymbol{z}_k; \Theta) = \frac{p(\boldsymbol{a} \mid \boldsymbol{q}, \boldsymbol{z}_k; \Theta)}{\sum_{j=1}^{K} p(\boldsymbol{a} \mid \boldsymbol{q}, \boldsymbol{z}_j; \Theta)}$$

$$\mathcal{L}_{\text{alt-2}} = \log p(\boldsymbol{a} \mid \boldsymbol{q}, \mathcal{Z}; \Theta) + \mathbb{KL}(\mathbb{SG}\left(\tilde{p}(\boldsymbol{a} \mid \boldsymbol{q}, \boldsymbol{z}_k; \Theta)\right) \mid\mid p(\boldsymbol{z}_k \mid \boldsymbol{q}, \mathcal{Z}; \Phi))$$



This objective offers improvement over the FiD model

Future Work

1. Application to knowledge-grounded dialogue generation.

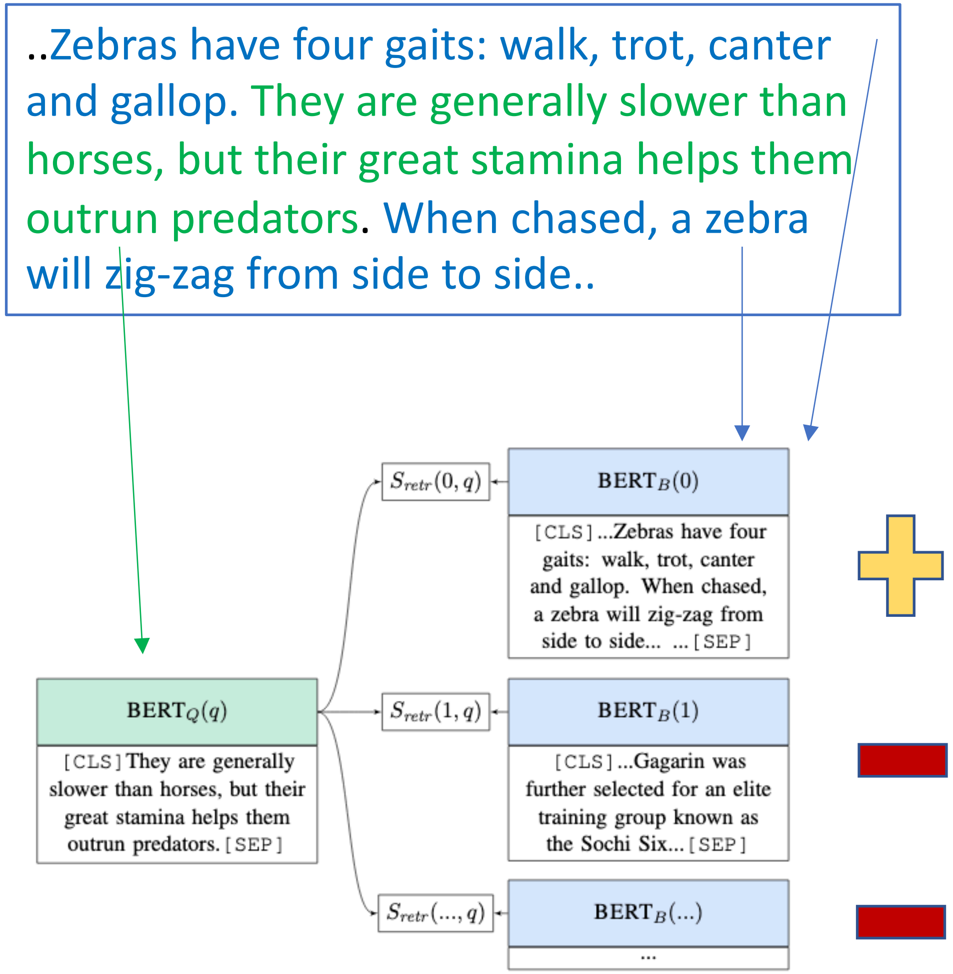
2. Multilingual open-domain question answering.

Thank you and Questions!

Code to be released at: https://github.com/DevSinghSachan/emdr2

Extra Slides: Dual Encoder Initialization by ICT

- Inverse Cloze Task (ICT)
- Sample a sentence from a paragraph.
- Sentence can be considered a *pseudo-query.*
- Remaining sentences can be considered as a *pseudo-context.*
- **Unsupervised** - can use all Wikipedia to train the model.



..Zebras have four gaits: walk, trot, canter and gallop. They are generally slower than horses, but their great stamina helps them outrun predators. When chased, a zebra will zig-zag from side to side..

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

Lee et al., 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering

Things which didn’t work

- FiD: concatenating the top-K documents together and increasing the position embeddings to 12000.
- FiD: concatenating the top-K documents together and introducing K segment embeddings.
- Asynchronous embedding updates with 250 steps not much improvements.