

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



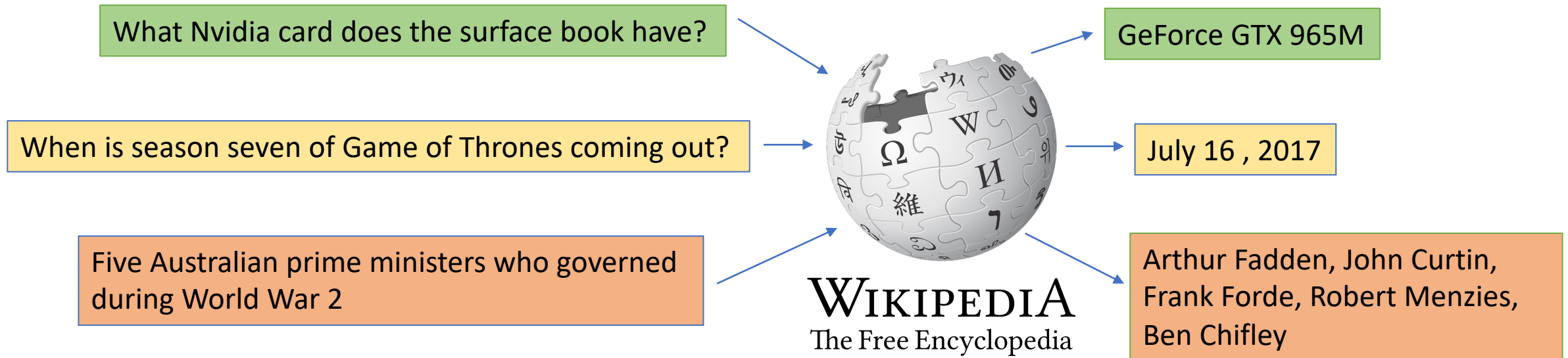
Chris Dyer



Dani Yogatama

Problem Setup: Open-Domain QA

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



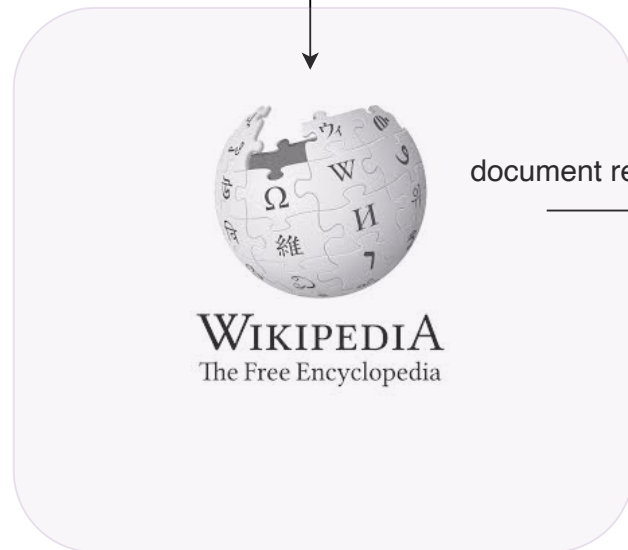
Background: Open-Domain QA

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

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]

- History
- Features

Nvidia
From Wikipedia, the free encyclopedia

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

Nvidia Corporation^m ^{nl} (/nɪˈvidiə en-nidiə-ə/) is an American multinational technology company incorporated in Delaware and based in Santa Clara, California.^[k] 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.^{[R][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.^{[R][7]} In addition to AMD, its competitors include Intel and Qualcomm.

Contents [hide]

- History
 - 1.1 Major releases and acquisitions
 - 1.2 Class action lawsuit
 - 1.3 Apple/Nvidia web driver controversy
 - 1.4 Hardware Unboxed controversy
- Finances
- GPU Technology Conference
- Product families
- Open-source software support
- Deep learning
- 1 DDX
- Inception Program
 - 7.1 2018 winners^[17]
 - 7.2 2017 winners^[17]
- See also
- Notes
- References
- External links

History [edit]

Nvidia was founded on April 5, 1993,^{[R][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 "nvidia", the Latin word for "envious."^[13] Nvidia went public on January 22, 1999.^{[13][14][15]}

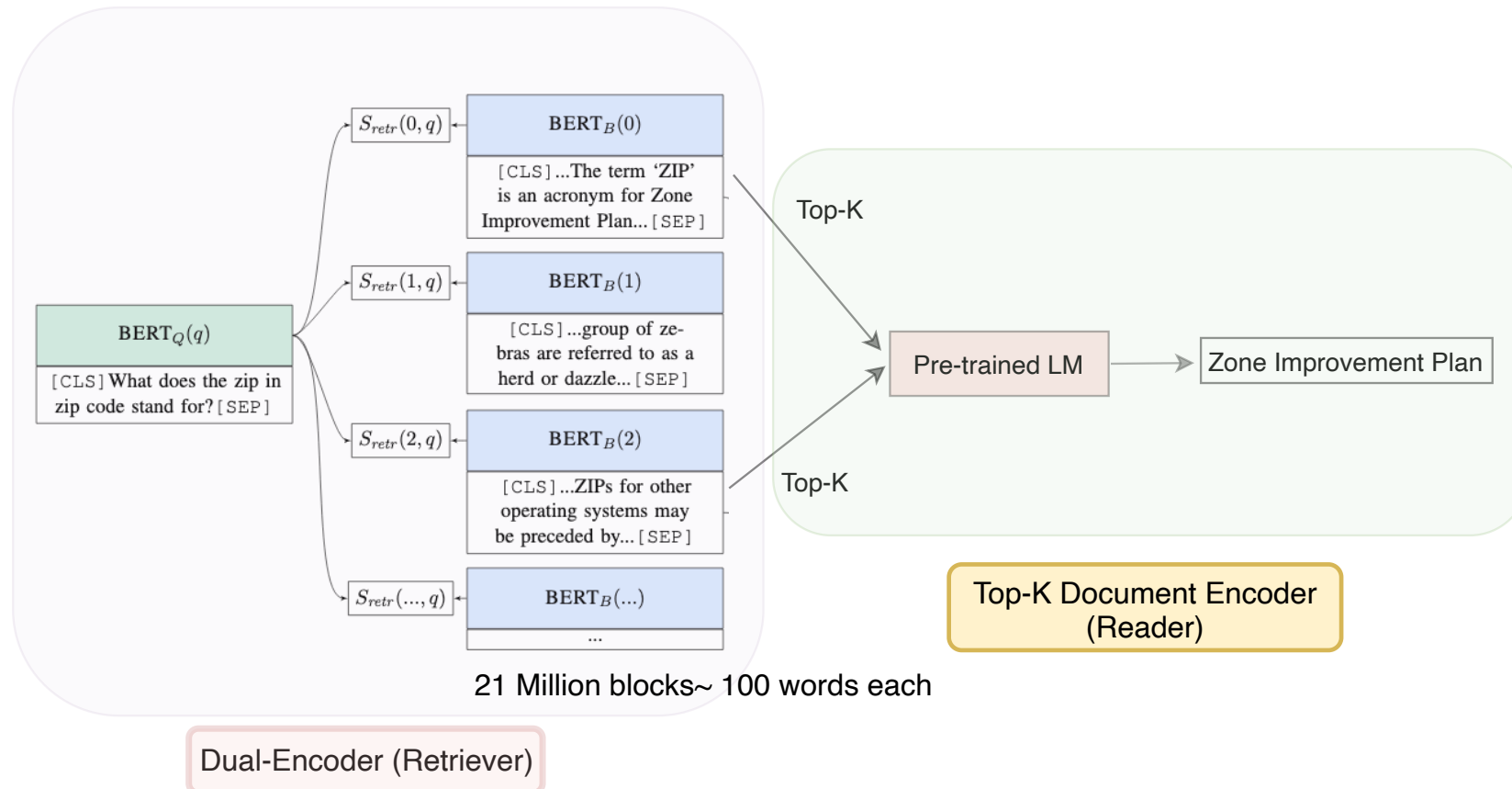
Answer Extraction

GeForce GTX 965 M

Background: Neural Models for Open-Domain QA

Stage 1: Trainable Information Retrieval

Stage 2: Trainable Answer Extraction



EMDR²: End-to-End Training of Multi-Document Reader and Retriever

Modeling Components

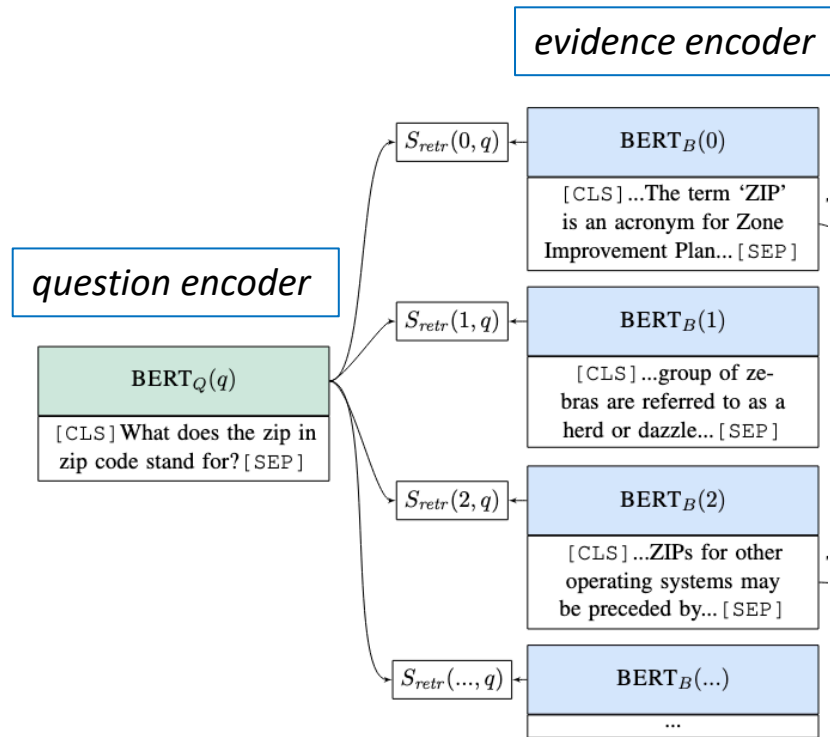
- **Retriever:** Dual Encoder
- **Reader / Answer Extractor:** Fusion-in-Decoder (FiD)

EMDR²: End-to-End Training of Multi-Document Reader and Retriever

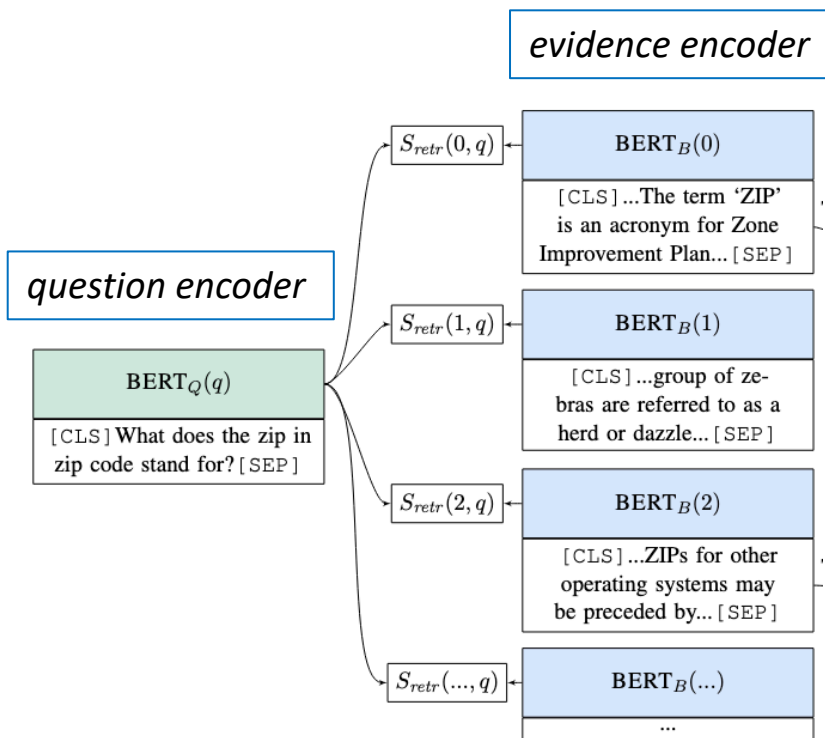
Modeling Components

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

Dual Encoder Retriever



Dual Encoder Retriever



Evidence Documents

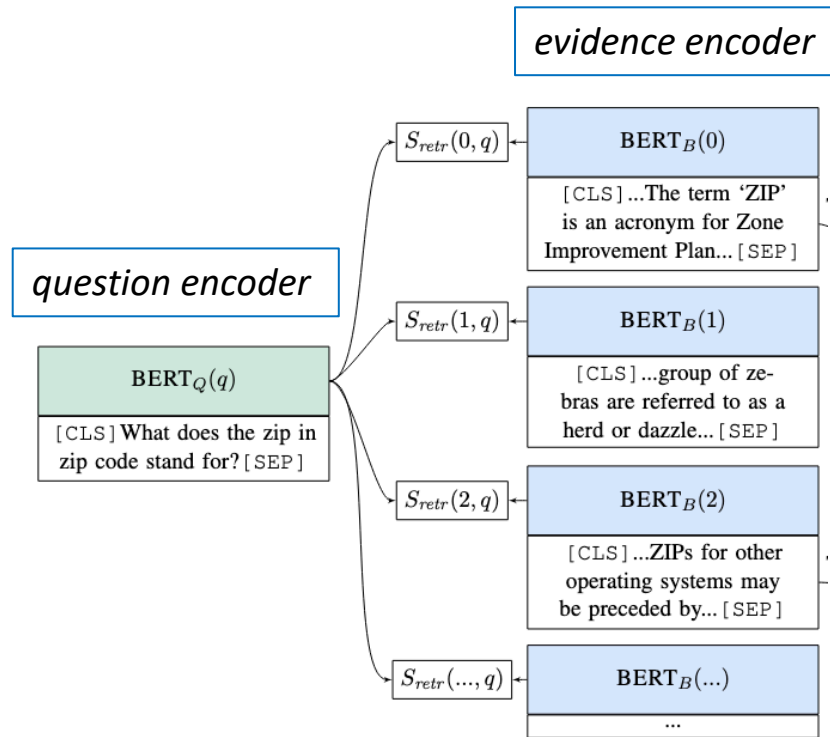
$$\mathcal{D} = \{d_1, \dots, d_M\}$$

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

Select Top-K Documents with highest scores

$$\mathcal{Z} = \{z_1, \dots, z_K\}$$

Dual Encoder Retriever



Evidence Documents

$$\mathcal{D} = \{d_1, \dots, d_M\}$$

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

Select Top-K Documents with highest scores

$$\mathcal{Z} = \{z_1, \dots, z_K\}$$

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

Evidence Documents

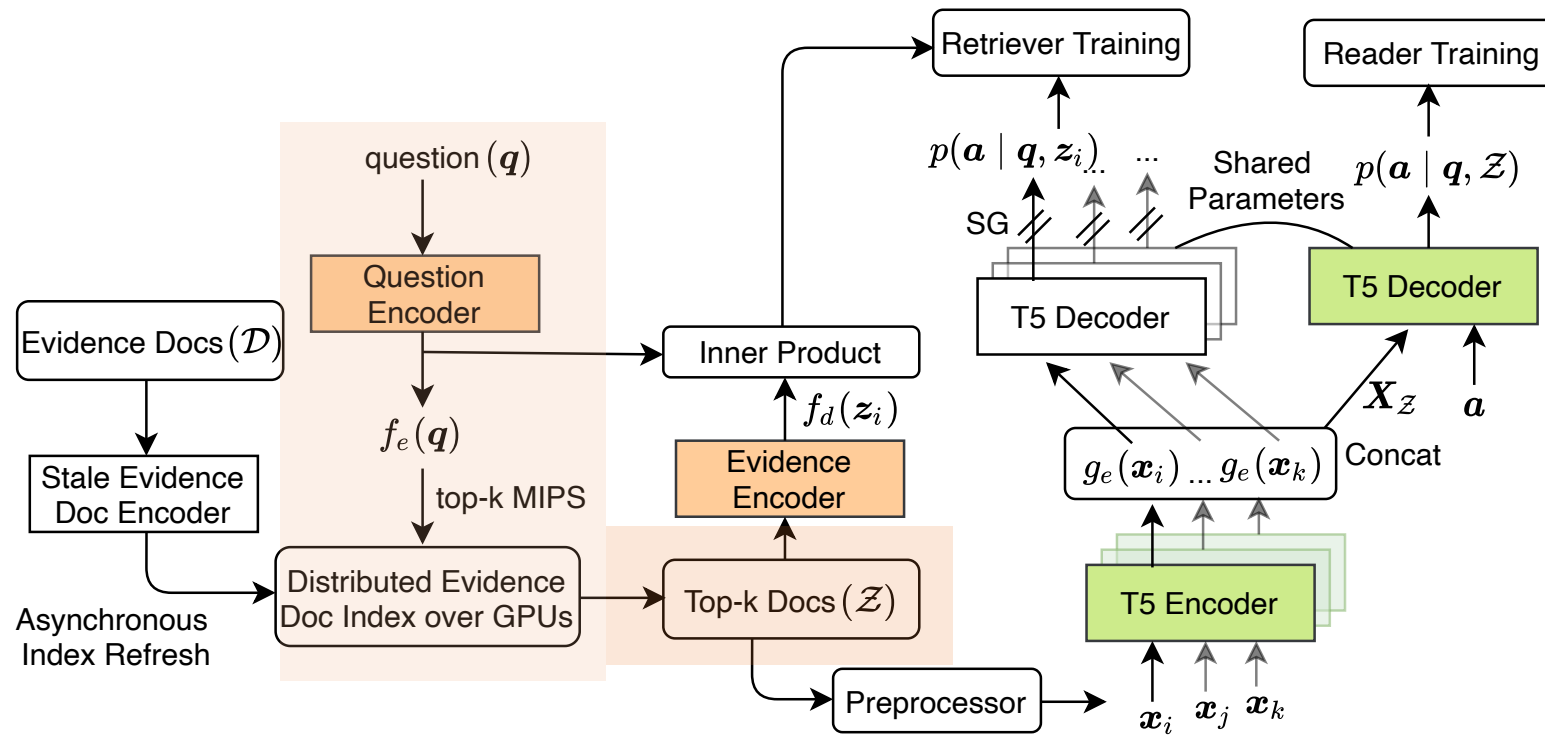
- **Evidence**: document collection containing world knowledge.
- We use **English Wikipedia** from Dec 2018 as evidence.
- Split articles into **100 words** long sequences.
 - Shorter sequences -> higher retrieval accuracy
- Overall size = 21 Million sequences



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.

EMDR²: Top-K Retrieval



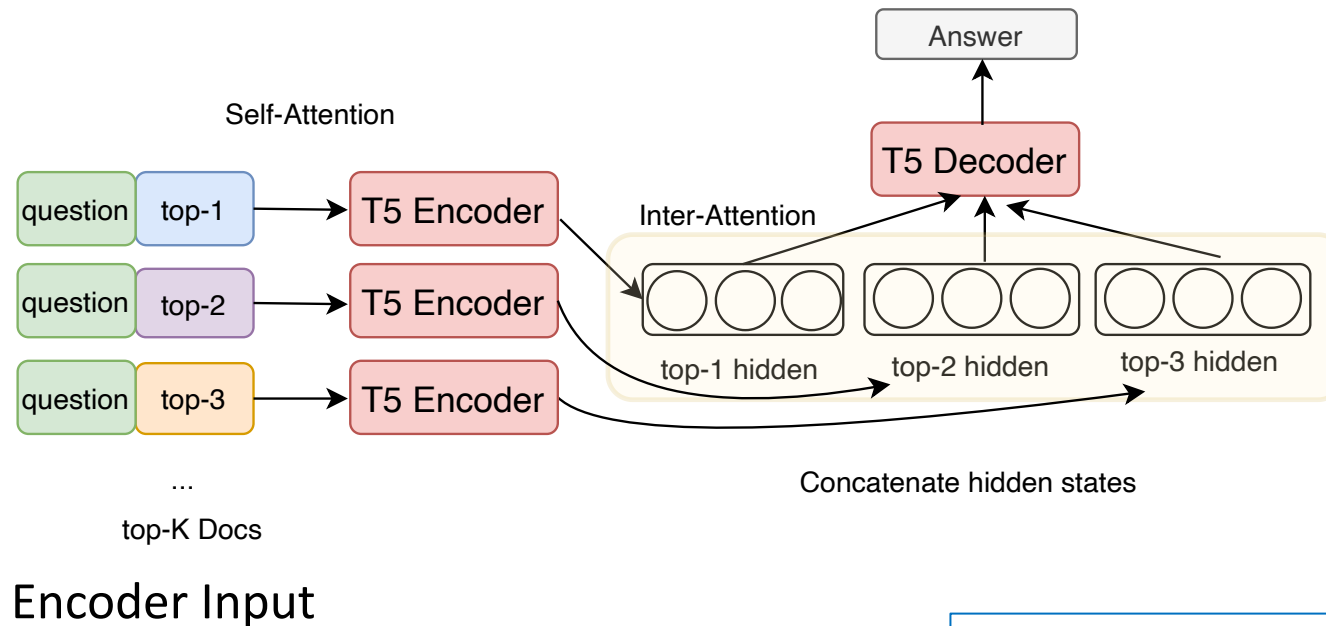
EMDR²: End-to-End Training of Multi-Document Reader and Retriever

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

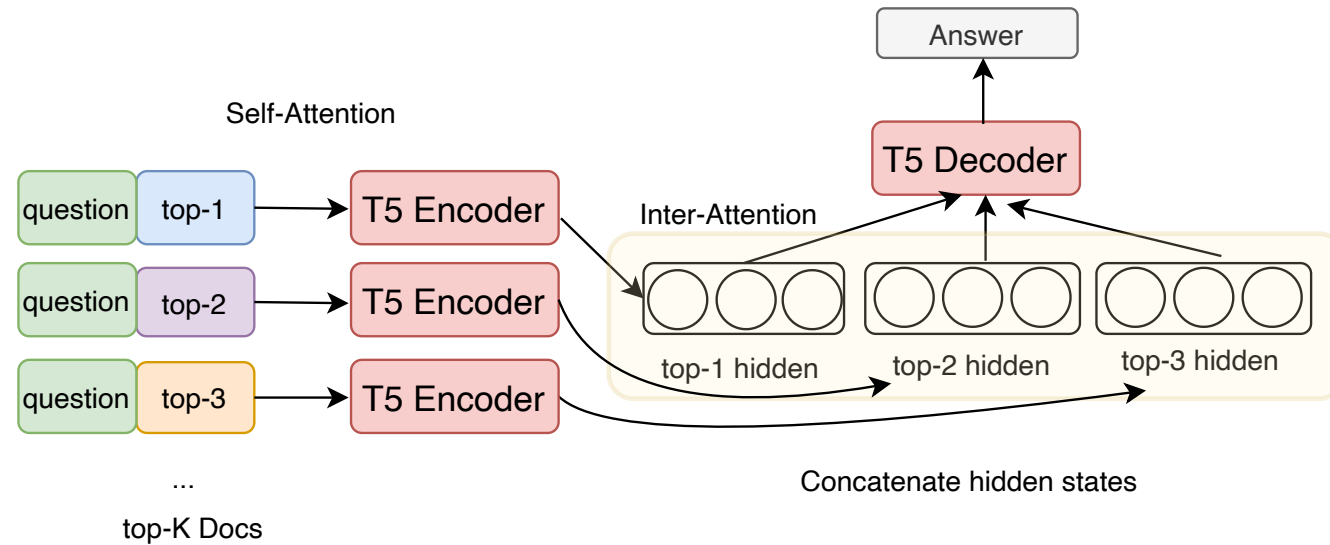


for each Top-K doc:

$$\mathbf{x}_k = [\text{CLS}] \mathbf{q} [\text{SEP}] \mathbf{t}_{z_k} [\text{SEP}] \mathbf{z}_k [\text{SEP}]$$

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

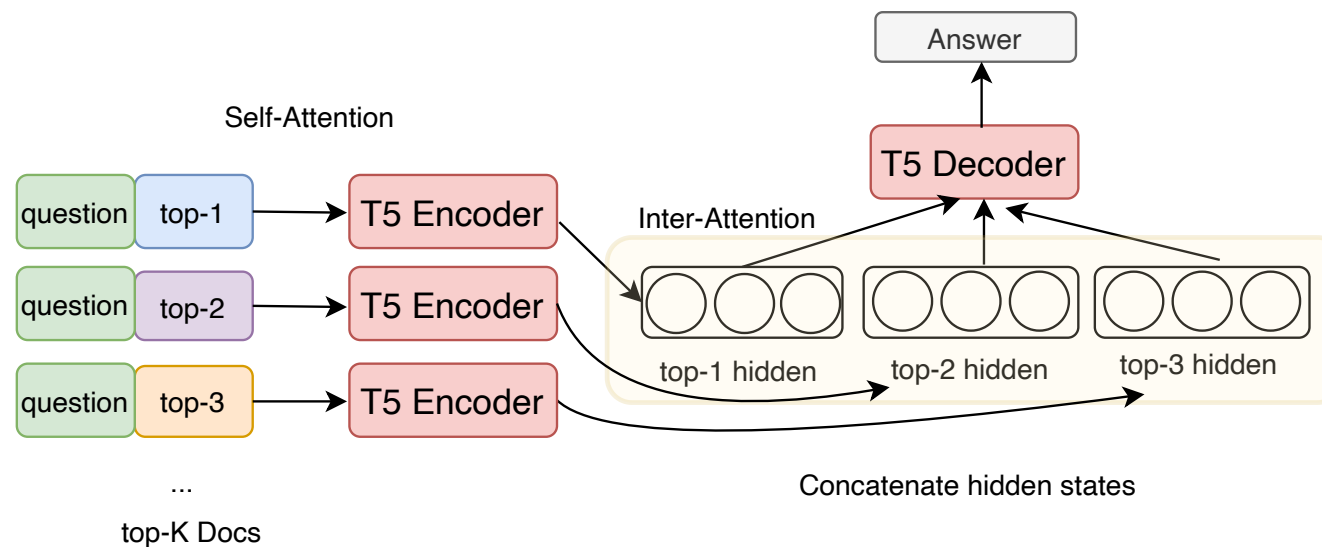
Fusion-in-Decoder: Self-Attention



$$\mathbf{x}_k = [\text{CLS}] \mathbf{q} [\text{SEP}] \mathbf{t}_{z_k} [\text{SEP}] \mathbf{z}_k [\text{SEP}]$$

Independent self-attention over each \mathbf{x}_k

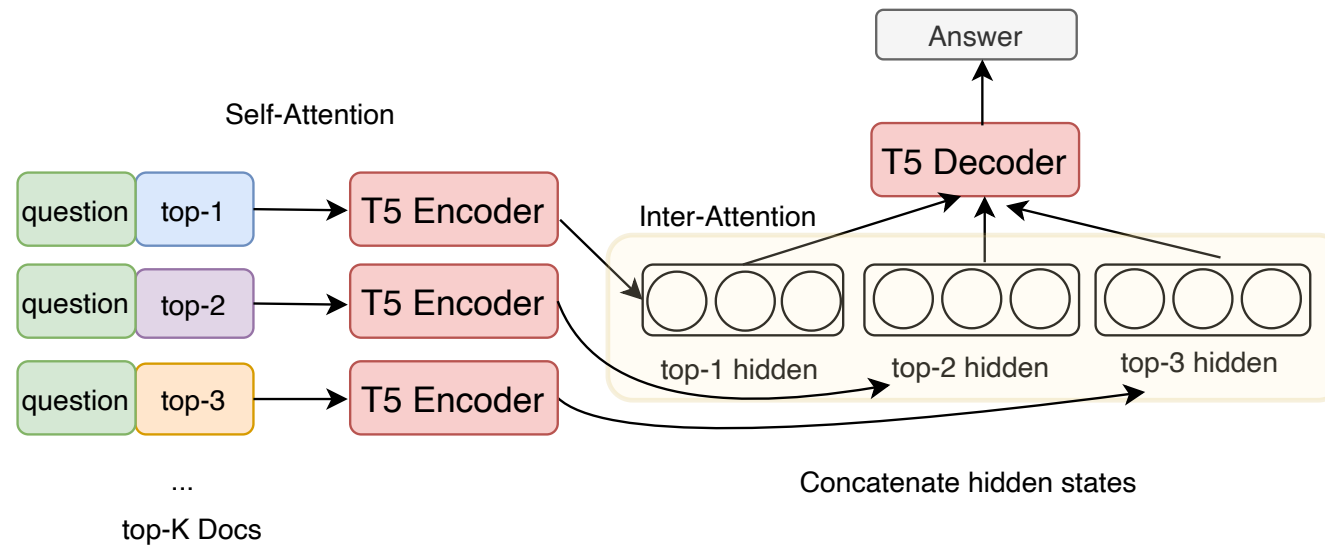
Fusion-in-Decoder: Inter-Attention



Concatenate the encoder representations for decoder's inter-attention

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

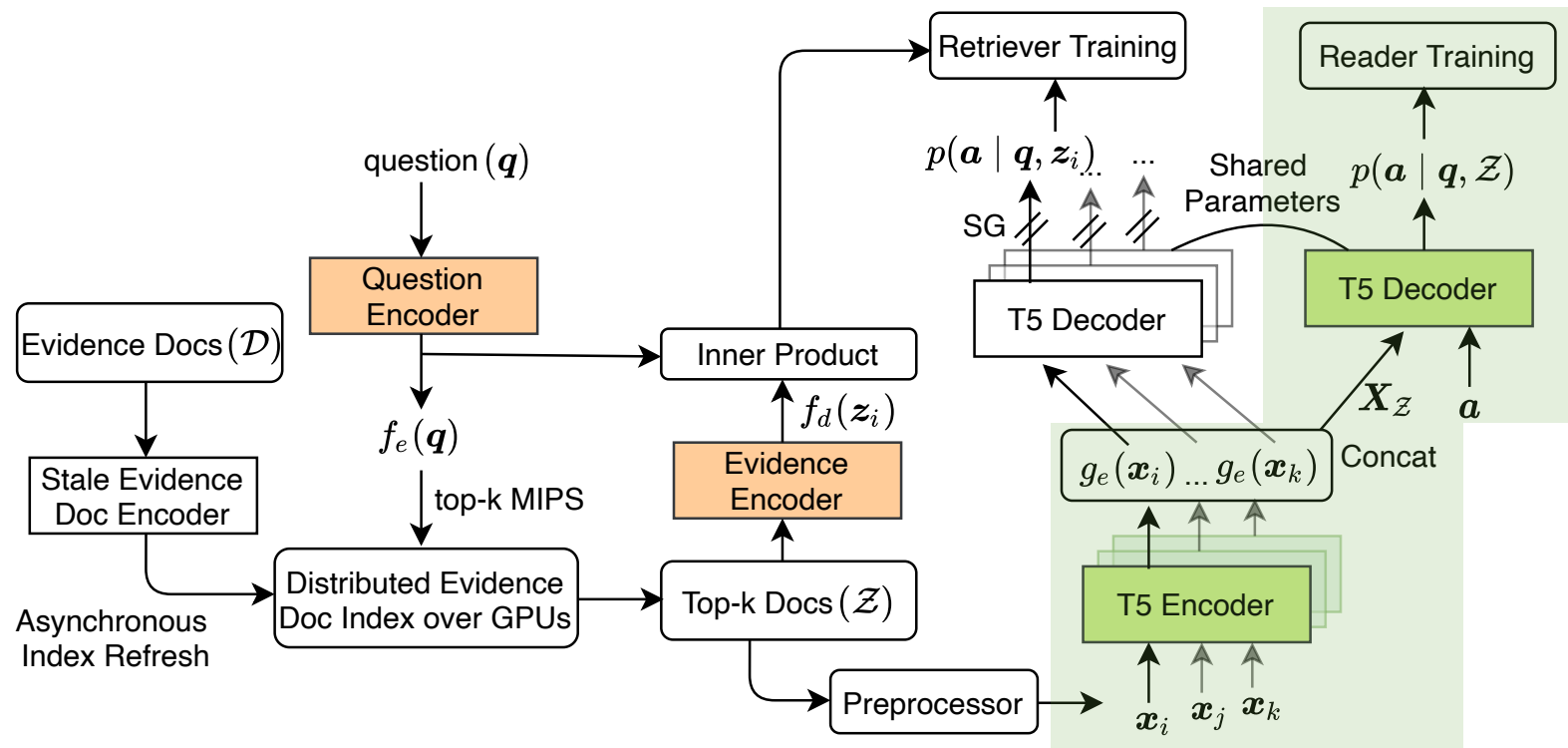
Fusion-in-Decoder: Training



Autoregressively train the model

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

EMDR²: Fusion-in-Decoder



End-to-End Training: Formulation

Marginal Likelihood

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

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

End-to-End Training: Formulation

Marginal Likelihood

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

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

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

End-to-End Training: Formulation

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

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

End-to-End Training: Formulation

Log-Likelihood

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

EMDR²: FiD Training

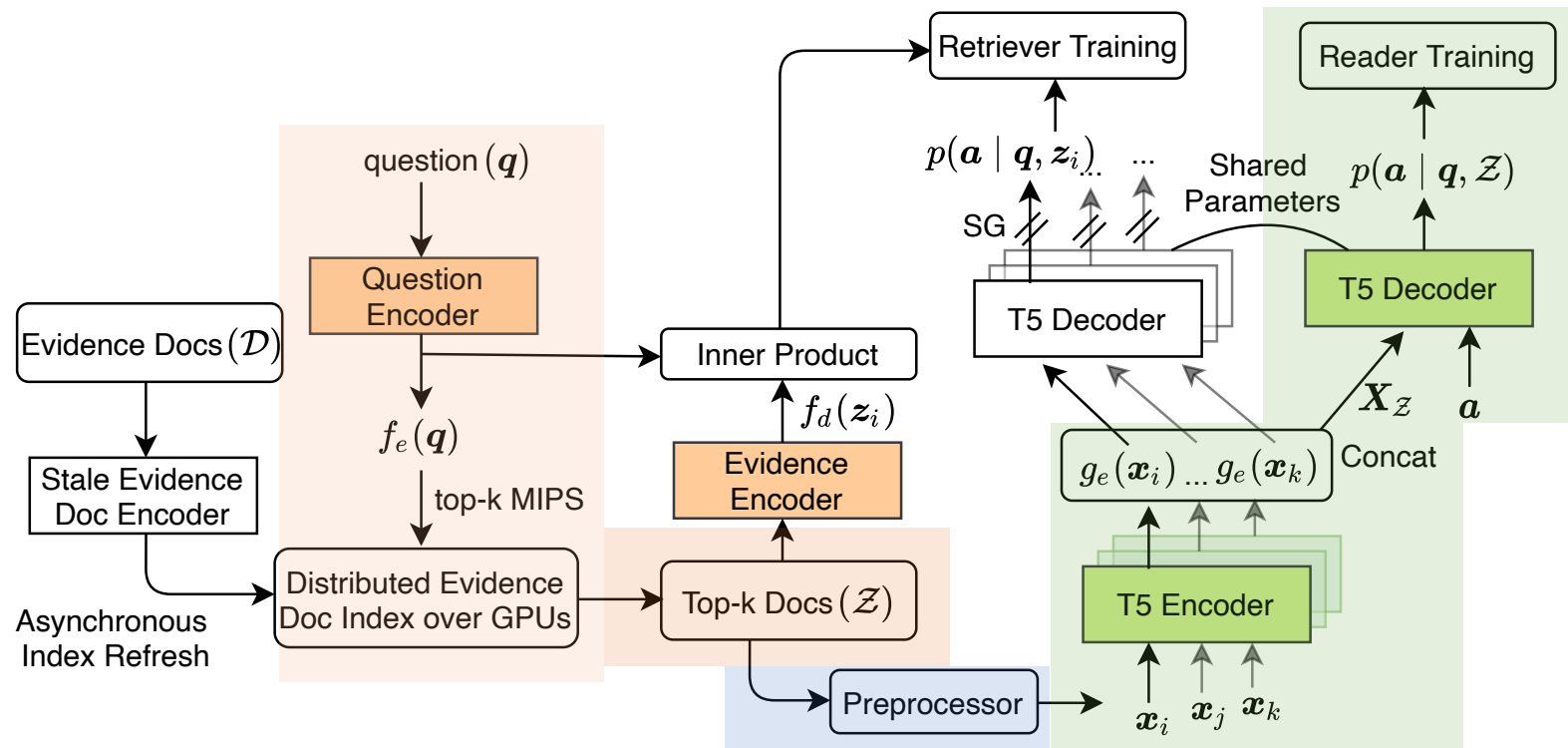
$$\mathcal{L} = \underbrace{\log p(\mathbf{a} \mid \mathbf{q}, \mathcal{Z}_{\text{reader}}; \Theta)}_{\text{FiD training}} + \log \underbrace{p(\mathcal{Z}_{\text{retriever}} \mid \mathbf{q}; \Phi)}_{\text{Dual Encoder training}}$$

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

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

- Teacher-forcing training of FiD

EMDR²: FiD Training



EMDR²: Retriever Training

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

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

EMDR²: 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

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

EMDR²: Retriever Training

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

From **Conditional Bayes Rule**:

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

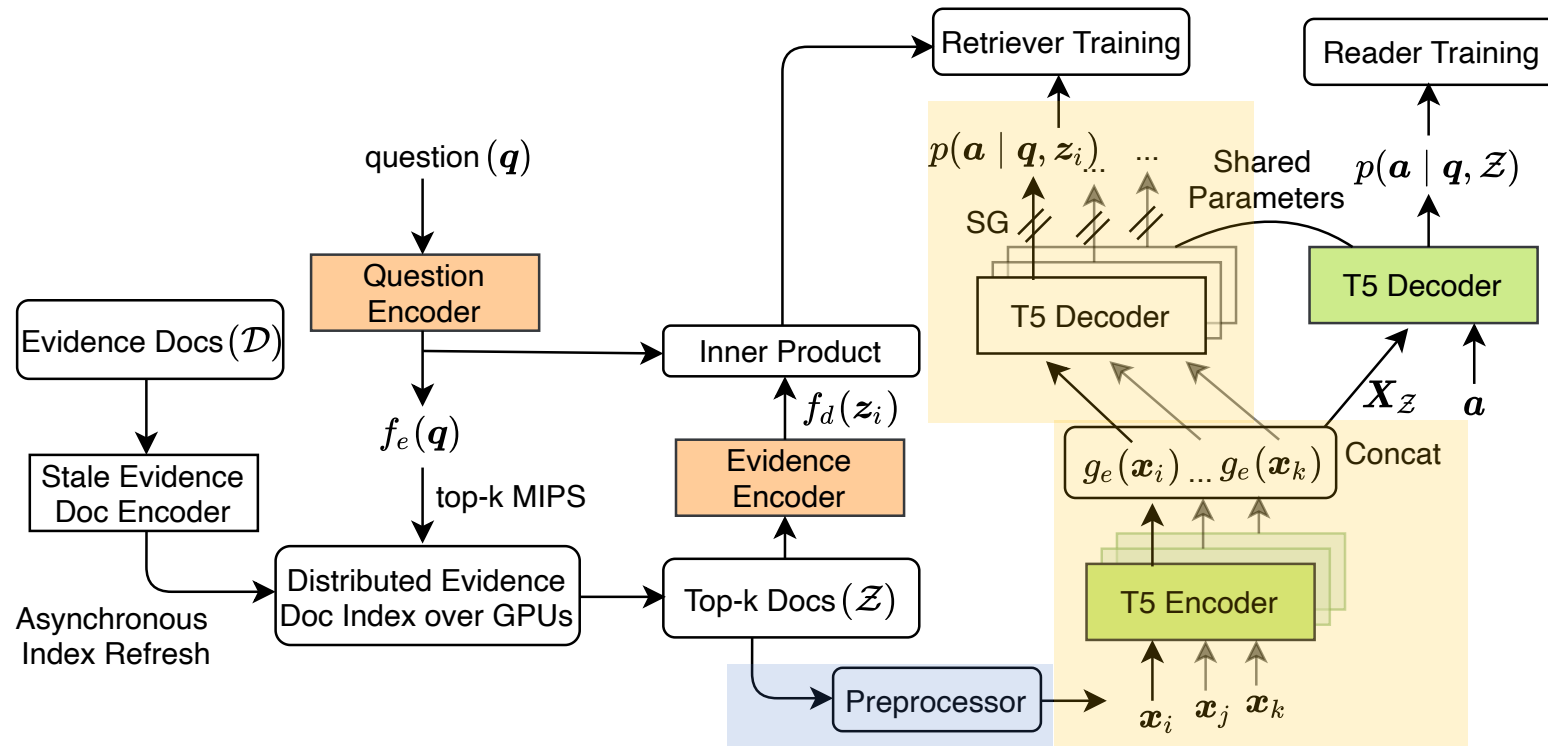
EMDR²: Retriever Training

From **Conditional Bayes Rule**:

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

First part can be computed from T5

EMDR²: Retriever Training



EMDR²: Retriever Training

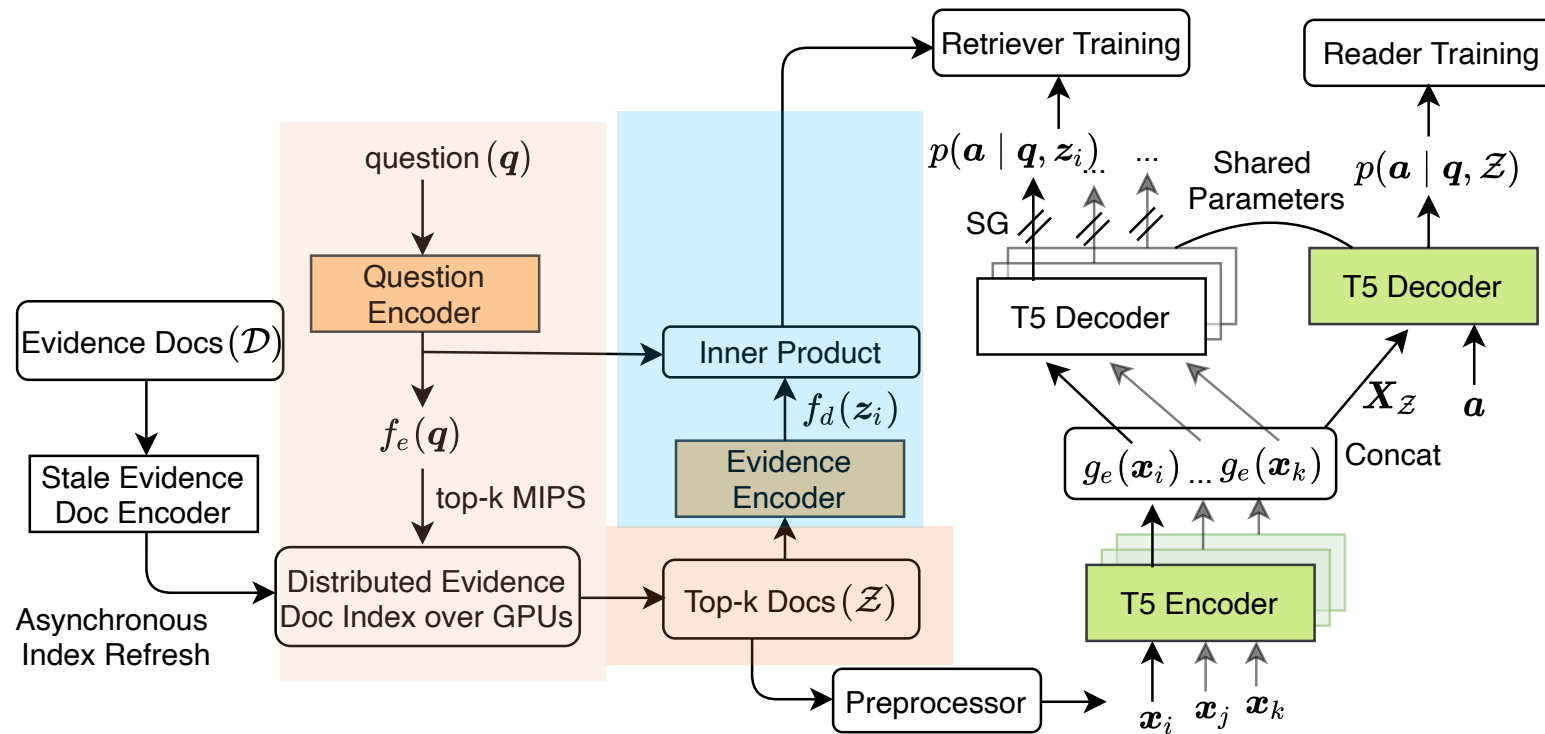
Probability of a document z_k from dual encoder

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

Apply a softmax with temperature over the top-K scores

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

EMDR²: Retriever Training



EMDR² Training Objective

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

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

EMDR² Training Objective: EM View

E Step:

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

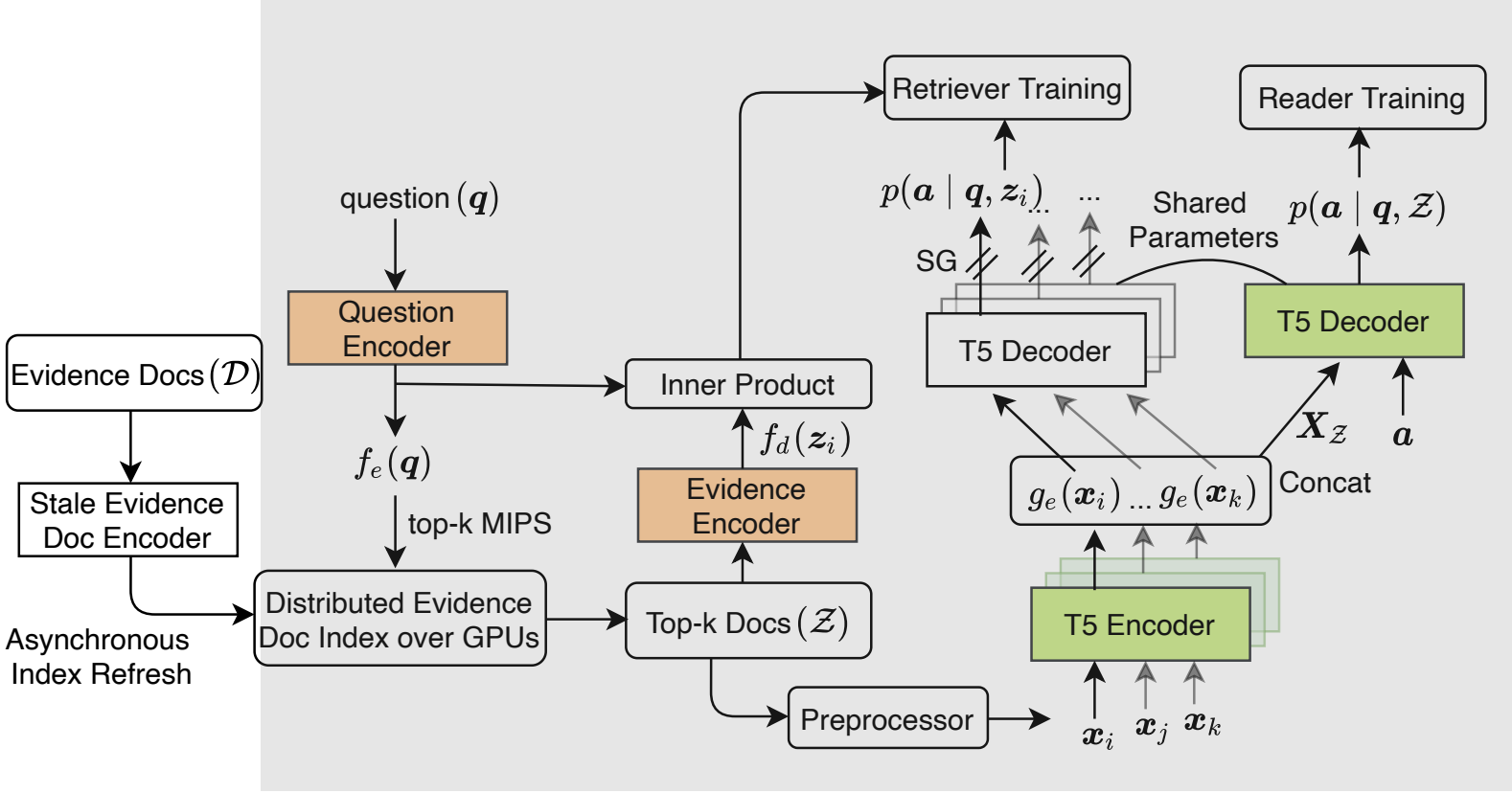
$$\text{score}(\mathbf{q}, \mathbf{d}_i; \Phi) = f_q(\mathbf{q}; \Phi_q)^\top f_d(\mathbf{d}_i; \Phi_d)$$

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

M Step:

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

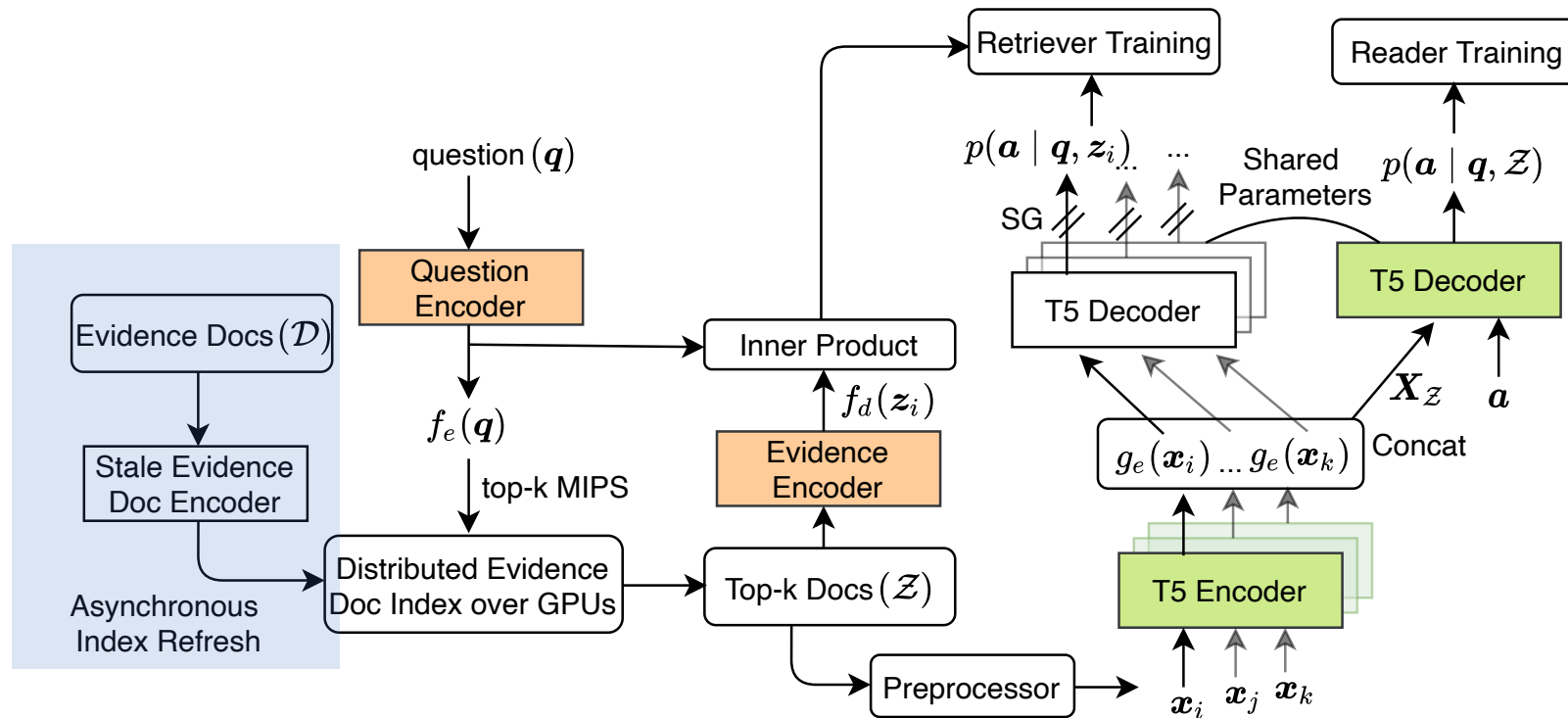
EMDR²: Modeling Components



EMDR²: 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).

EMDR²: Asynchronous Evidence Indexing



Comparison of Open-Domain QA Approaches

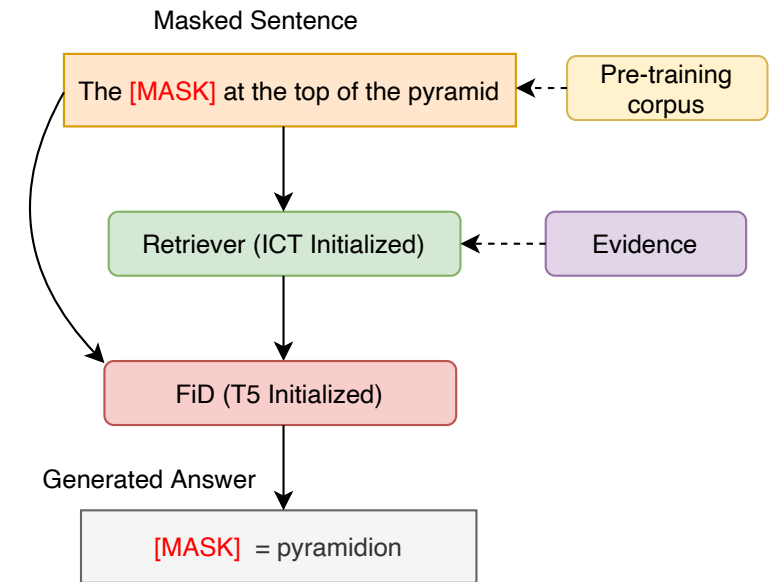
Model	Reader and Retriever Training					
	<i>Multi-Doc Reader</i>	<i>Retriever Adaptation</i>	<i>Disjoint</i>	<i>End-to-End</i>	<i>Multi-Step</i>	<i>Unsupervised Retriever</i>
REALM (Guu <i>et al.</i> , 2020)		✓		✓		✓
DPR (Karpukhin <i>et al.</i> , 2020)			✓			
RAG (Lewis <i>et al.</i> , 2020b)		✓		✓		
FiD (Izacard and Grave, 2021b)	✓		✓			
FiD-KD (Izacard and Grave, 2021a)	✓	✓			✓	
EMDR ² (Our Approach)	✓	✓		✓		✓

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)

EMDR²: 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**.



Datasets

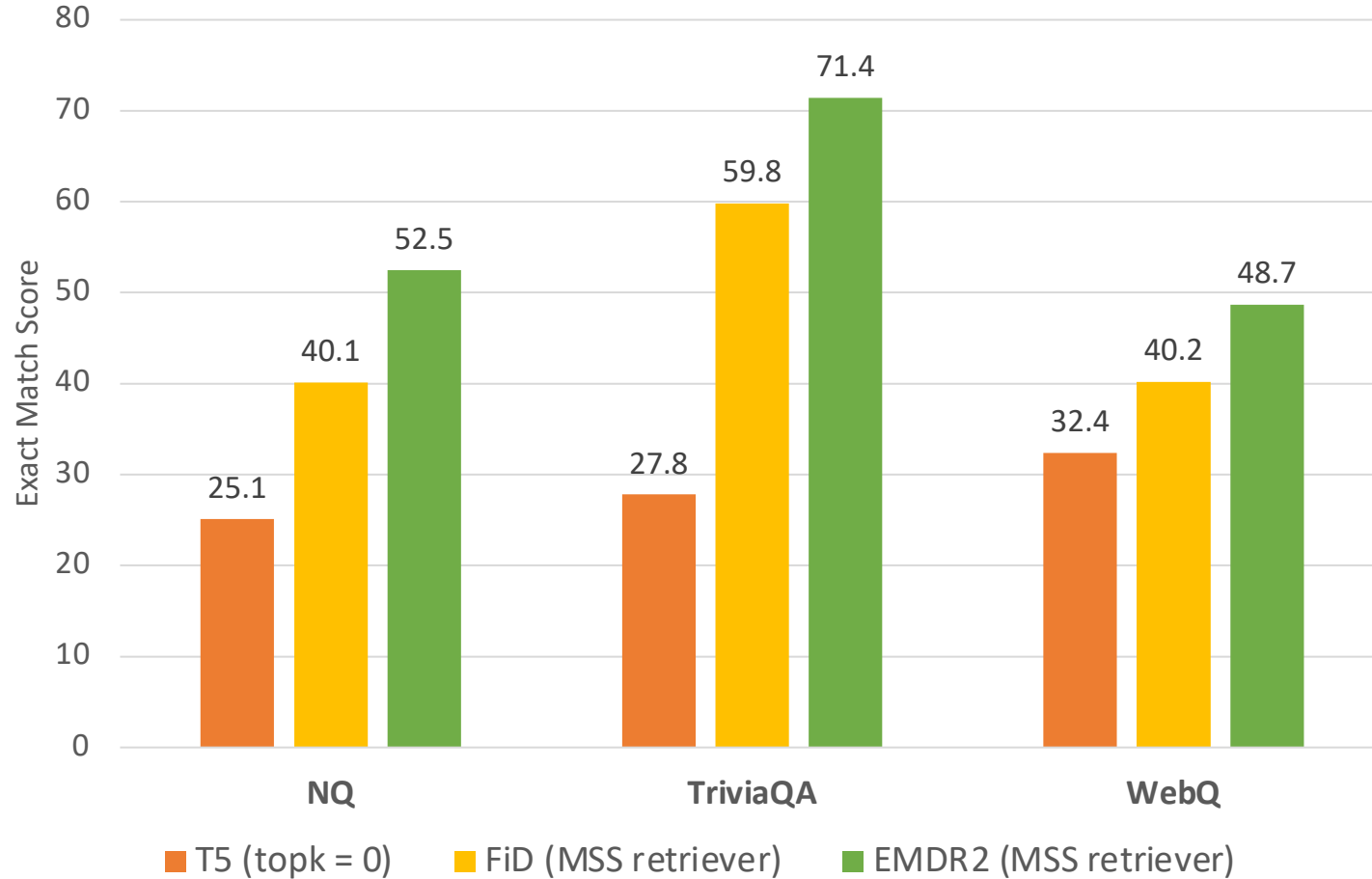
Dataset	Train	Dev	Test
WebQuestions (WebQ)	3,417	361	2,032
Natural Questions (NQ)	79,168	8,757	3,610
TriviaQA	78,785	8,837	11,313

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.

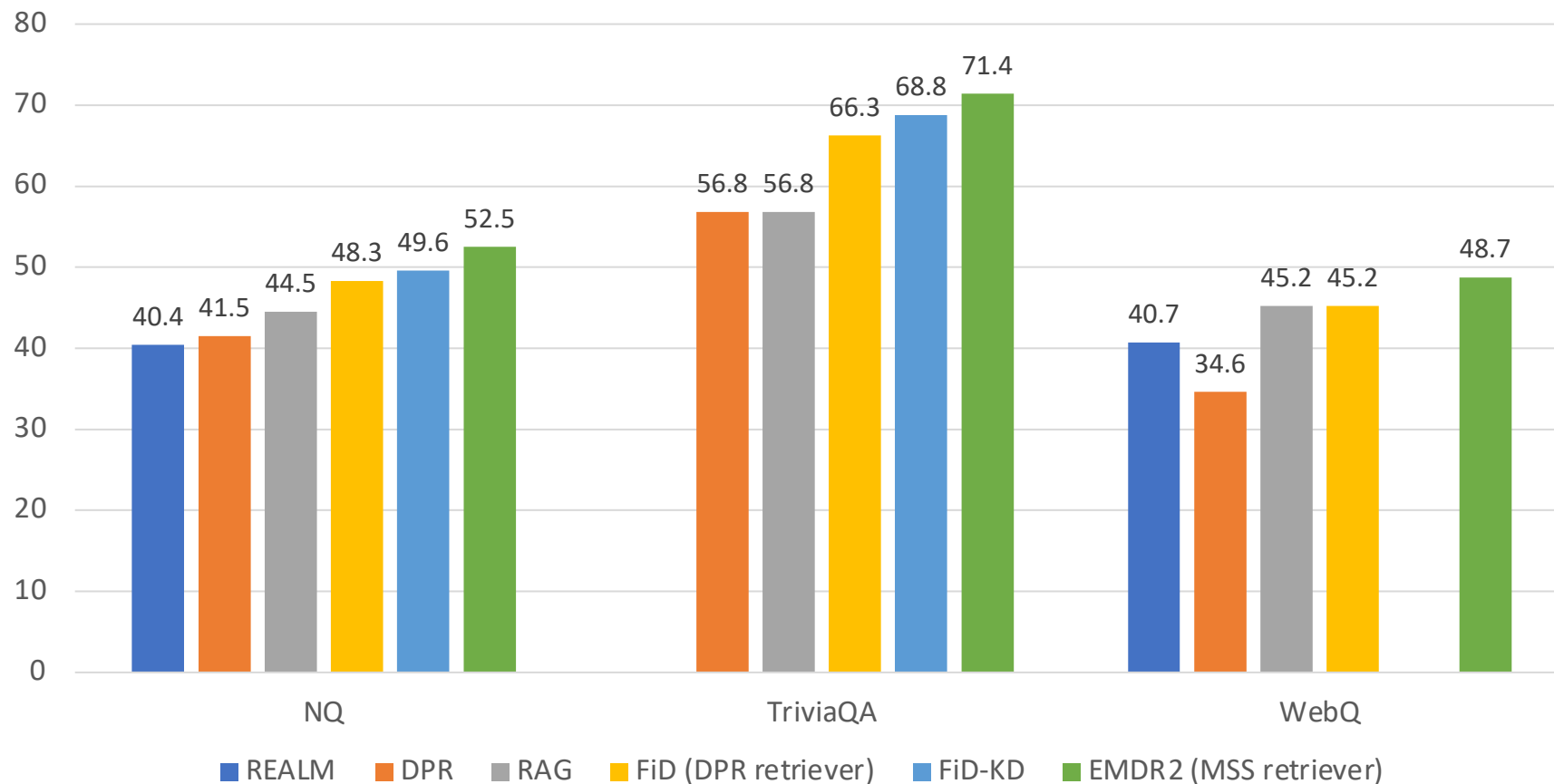
TriviaQA: Collection of trivia question-answer pairs collected from the web.

Results: EMDR² Training



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

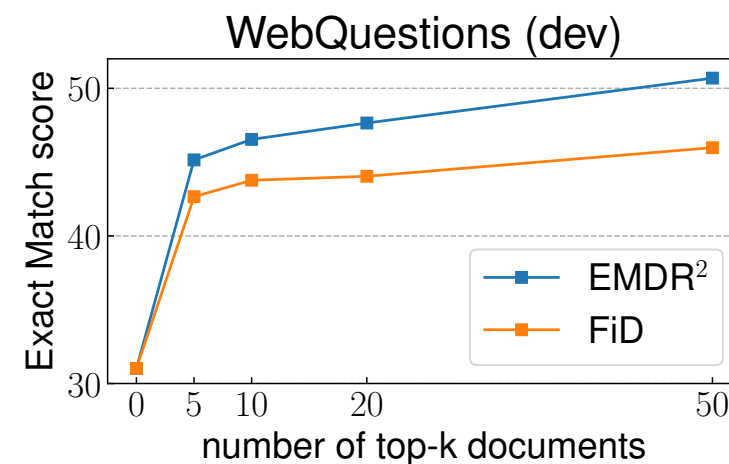
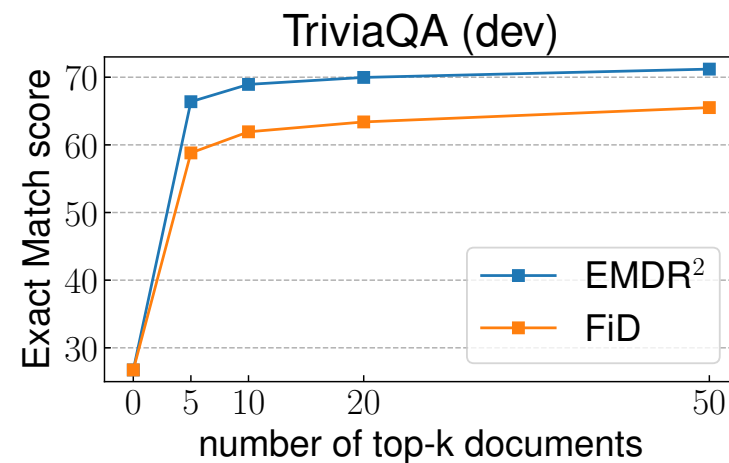
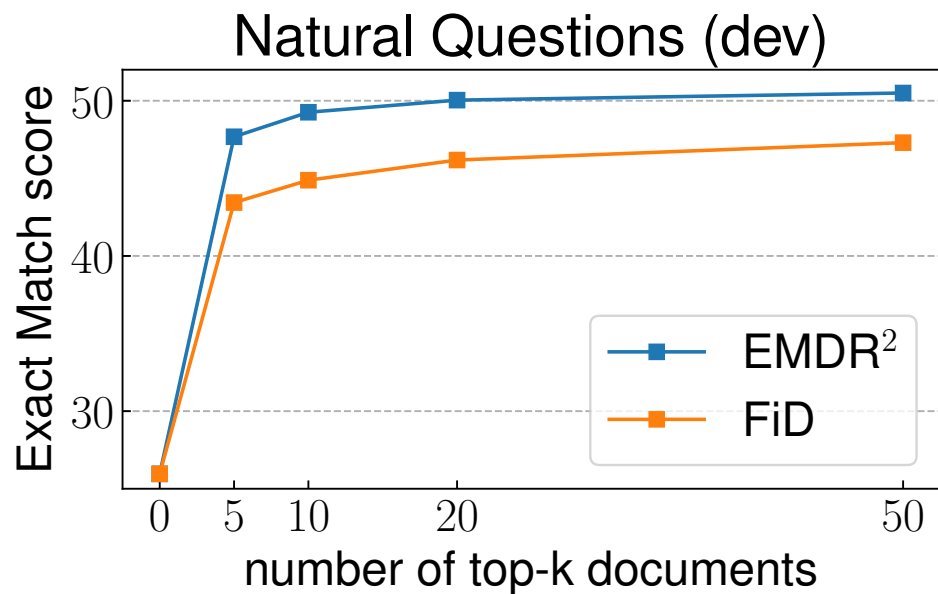
Comparison with Other Approaches



New SOTA Results of EMDR²

2-3 EM points gain over FiD-KD

Analysis: Effect of the Number of Top-k Documents



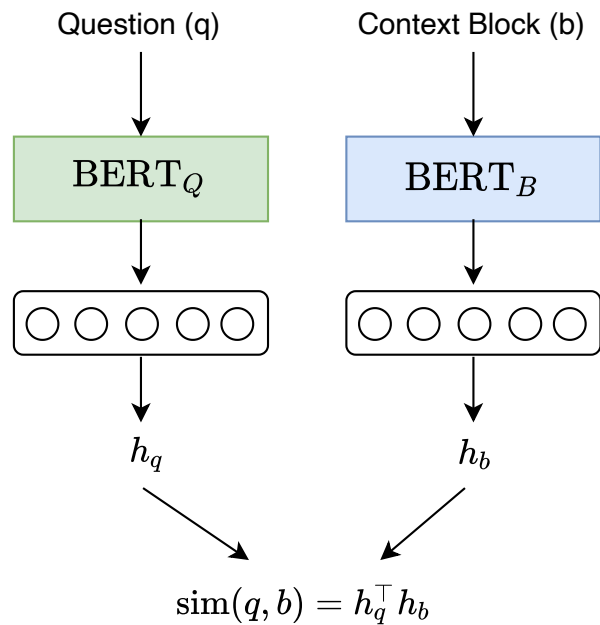
Effect of Retriever Initialization

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)

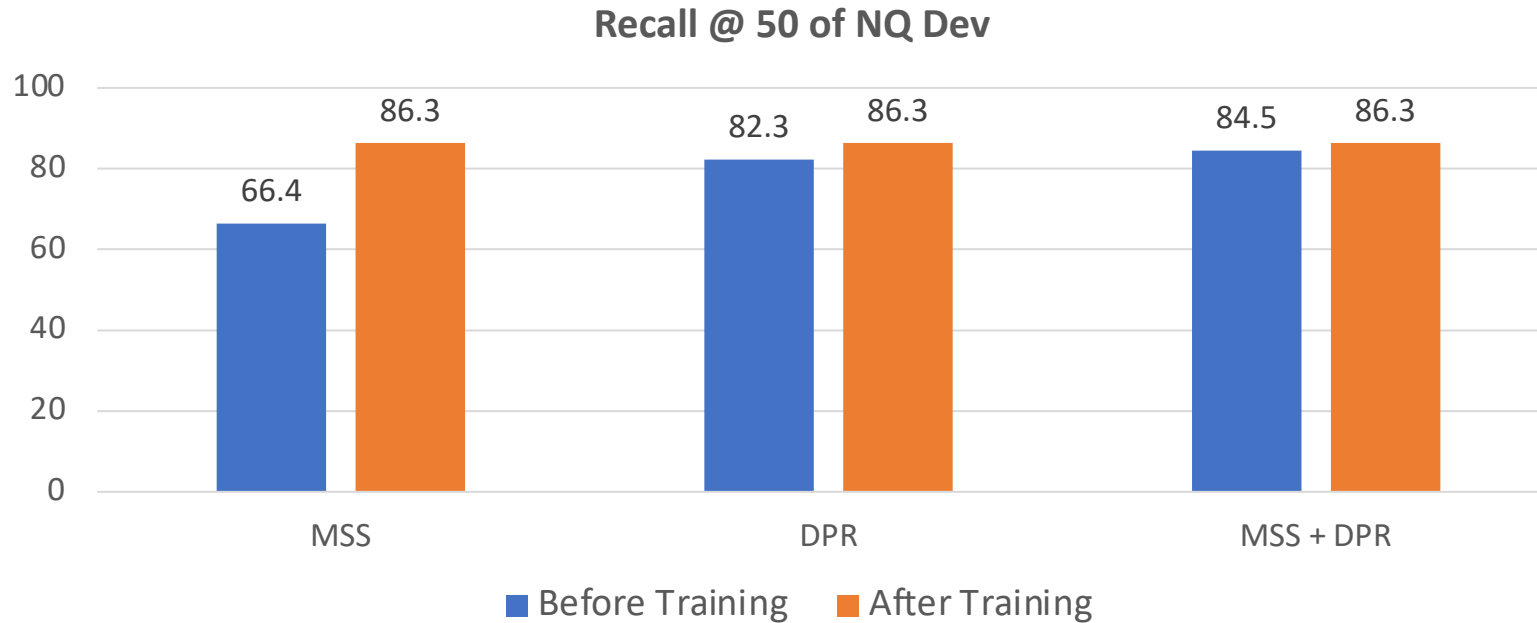
- Dual-encoder model
- Train from **supervised** question-context pairs



$D = q_i,$ Question
 $p_i^+,$ One Positive Example
 $p_{i,j}^-$ Set of Negative Examples

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

Effect of Retriever Initialization



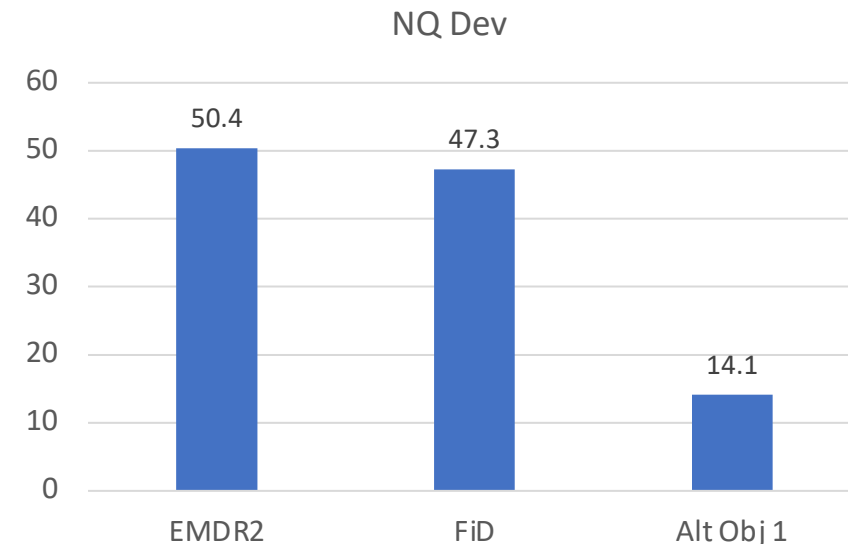
- MSS, DPR, and MSS + DPR retriever initialization results in the same final retrieval accuracy.
- Retriever training by DPR may not be essential for open-domain QA.

Alternative Training Objective 1

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

No feedback from FiD reader to retriever

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

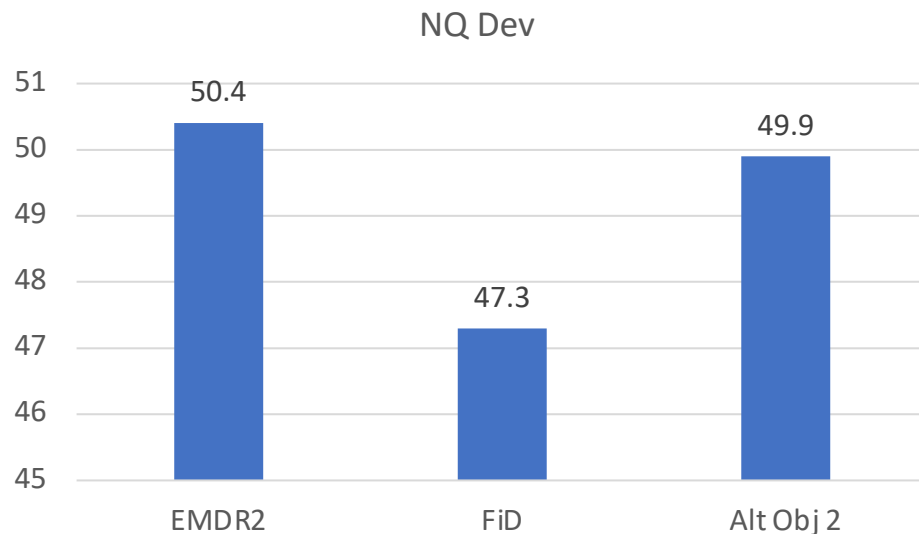


- Poor performance of Alt. Obj.
- Retriever gets stuck in a bad local optimum

Alternative Training Objective 2

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

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



This objective offers improvement over the FiD model

Future Work

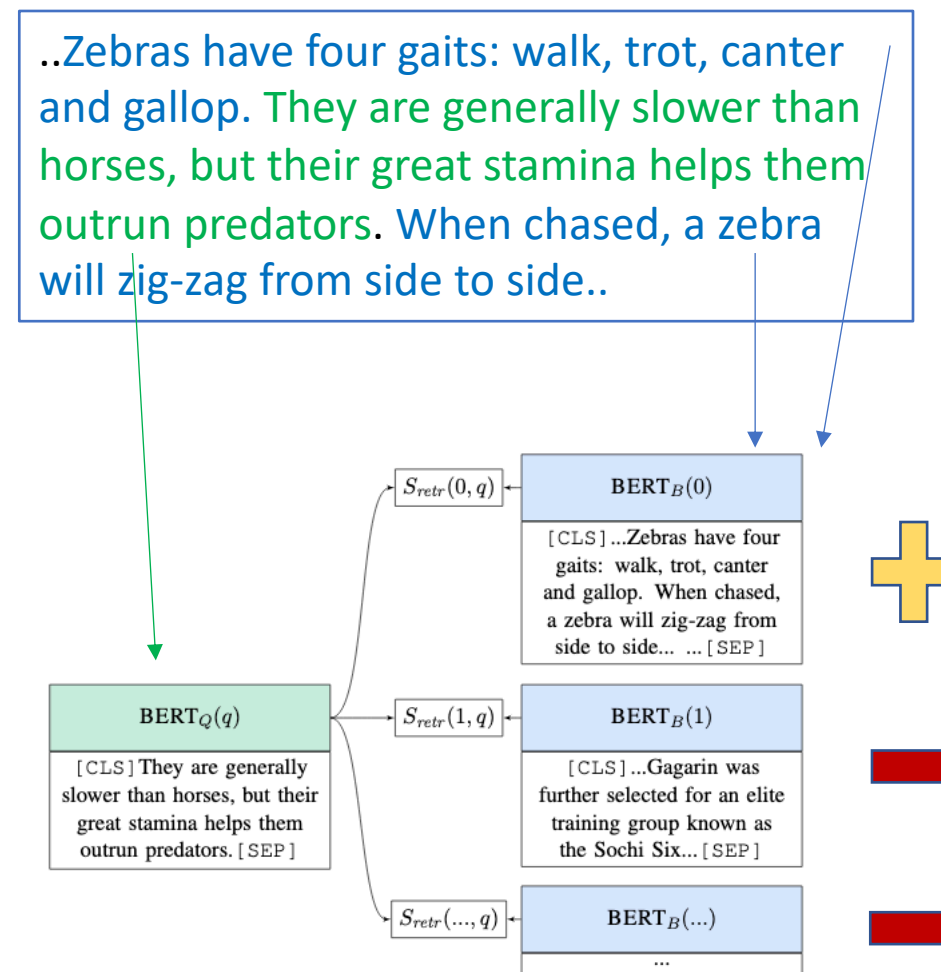
1. Application to knowledge-grounded dialogue generation.
2. Multilingual open-domain question answering.

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.



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.