Dense Passage Retrieval for Open-Domain Question Answering

15조 캠퍼 최성욱

1. Introduction

Dense Representation

1. **Dense Representation**은 Sparse Representation에 비해 **synonyms/paraphrases를 더 잘 Mapping**한다.

2. **Learnable -> Task-Specific** 만들 수 있다.

Q: Who is the bad guy in the Lord of the Rings?

P(Ground Truth): Sala Baker is best known for portraying the villain Sauron in the Lord of the Rings trilogy

1. Introduction

Why do we need to use the Dense Retrieval in paper?

1. **Good Dense Vector Representation -> 방대한 양의 Labeling Q-P Pair**
2. **오직 ORQA**(Dense Retrieval)가 **TF-IDF/BM25를 능가**.
3. **But, ORQA는 2가지 Weakness**를 가진다. -> 이를 해소하며, Multiple ODQA dataset에서 SOTA를 달성

- ✓ **ICT pretraining 활용** -> computationally intensive / regular sentence에 대해 좋은지 명확하지 않다.
- ✓ **Passage Encoder를 fine-tuning 하지 않는다.** → Point
- ✓ **Relatively Small # of Q-P pair로 학습한다.**

1. Introduction

DPR introduced in this Paper

1. **Dual Encoder** Architecture -> Question Encoder / Passage Encoder
2. **Question Encoder / Passage Encoder 모두 Fine-tuning**
3. **Similarity** between Q vector and P vector -> **DP 활용** (Dot Product)
4. **In-batch Negative Sampling**
5. 추가적인 Pretraining은 필요하지 않을 것이다.

✓ **Dual Encoder -> Cross Encoder / Poly Encoder 활용해볼만 하다.**
✓ **Similarity -> DP가 가장 좋다. (Performance / Computational)**
✓ **Negative Sampling -> In-batch가 좋다.**

2. Dense Passage Retriever

Overview

1. k개의 최종 passage를 reader로 보내준다. 보통 **k는 20~100을 활용**한다.
2. 유사도 -> **DP** $\text{sim}(q,p) = E_Q(q)^\intercal E_P(p).$
3. Loss : **NLL** is better than Triplet
4. Encoder
   1. Dual Encoder (Q/P)
   2. Encoder -> BERT-base-uncased 활용
   3. 각 Passage의 **[CLS] token을 Representation 활용**
   4. hidden dimension -> 768
5. Inference -> **FAISS** 적용

Sim	Loss	Retrieval Accuracy			
		Top-1	Top-5	Top-20	Top-100
DP	NLL	**44.9**	**66.8**	**78.1**	**85.0**
	Triplet	41.6	65.0	77.2	84.5
L2	NLL	43.5	64.7	76.1	83.1
	Triplet	42.2	66.0	**78.1**	84.9

✓ **[CLS] token을 Representation으로 활용했다 -> 평균 혹은 다양한 방법을 시도**
✓ **FAISS 적용**

2. Dense Passage Retriever

Training

1. **Training Dataset 형태**

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

2. **Loss Function -> NLL**

$$L(q_i, p_i^+, p_{i,1}^-, \cdots, p_{i,n}^-) \qquad (2)$$

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

2. Dense Passage Retriever

Training – Negative Sampling

1. Question Vector가 Positive Passage와는 가깝도록 / Negative Passage와는 멀도록 학습하기 위해 활용.

2. Random -> Corpus에서 랜덤으로 선택

3. BM25 -> Question에 대한 정답은 없으면서 BM25 기반 높은 점수를 가지는 Passage

4. **Gold -> Training Set에서 Q-P pair에 대하여 다른 Q에 대한 P를 활용**

# of Negative Samples : 7 / Not In Batch
Random / BM25 / Gold 사이의 큰 차이는 없다.

Type	#N	IB	Top-5	Top-20	Top-100
Random	7	✗	47.0	64.3	77.8
BM25	7	✗	50.0	63.3	74.8
Gold	7	✗	42.6	63.1	78.3

Original Setting
Gold가 BM25보다 1점 정도의 약간 높은 성능을 보인다. -> Gold 채택

	Top-1	Top-5	Top-20	Top-100
Gold	44.9	66.8	78.1	85.0
Dist. Sup.	43.9	65.3	77.1	84.4

2. Dense Passage Retriever

Training – In Batch Negative Sampling

1. **계산비용의 절감**
    1. In Batch -> B개의 Question / Passage Pair
    2. Negative Sampling을 위해 embedding -> Q metrics / P metrics (B x h)
    3. DP 유사도 계산 -> $S = QP^T$ -> S(B x B) matrix
    4. 대각원소 -> Q-PP score/ 비대각원소 -> Q-NP score
    5. 행렬 연산을 통한 In-Batch에서의 Negative Sampling을 할 경우 계산 비용의 절감

2. **성능 향상**

Type	#N	IB	Top-5	Top-20	Top-100
Random	7	✗	47.0	64.3	77.8
BM25	7	✗	50.0	63.3	74.8
Gold	7	✗	42.6	63.1	78.3
Gold	7	✓	51.1	69.1	80.8

✓ **In-Batch Negative Sampling 활용**

3. Passage Retrieval

Training

1. **In-Batch Negative Sampling**
2. **Batch_size: 128**
3. Question별로 In-Batch에서 BM25기반 가장 유사도가 높으면서 정답을 포함하지 않는 Passage를 추가로 Negative Sample로 활용
4. Large Dataset(NQ, TriviaQA, SQuAD) -> 40 epoch
5. **Small Dataset(TREC, WQ) -> 100 epoch**
6. Learning Rate -> 10^{-5}
7. Optimizer -> AdamW
8. Scheduler -> Linear Scheduling with Warm-up
9. Dropout Ratio -> 0.1
10. **BM25 / DPR / BM25+DPR 실험**

✓ **Small Dataset -> 100 epoch**
✓ **Batch size 128 is better than smaller ones**
✓ **다양한 HyperParameter Tuning**

3. Passage Retrieval

Training - BM25+DPR

1. BM25/DPR에서 **각각 top-2000개의 passage**를 뽑는다.

2. 다음의 식을 활용하여 **top-k개의 최종 Passage**를 선택한다.

$$\mathrm{BM25}(q,p) + \lambda \cdot \mathrm{sim}(q,p)$$

3. 위 식에서 **Lambda는 가중상수이며, 본 논문에서는 Retrieval(BM25, DPR) Accuracy**에 기반하여 1.1을 활용.

✓ **BPS+DPR 적용해보기**

3. Passage Retrieval

Main Results

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

1. SQuAD를 제외화고는 DPR이 BM25에 비하여 성능이 우수하다.

2. 상대적으로 작은 Dataset의 경우 Multiple 학습을 진행할 경우 성능이 눈에 띄게 개선된다.

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

✓ **TREC/WQ Dataset에 비교 -> DPR/BM25+DPR 시도해보기**

3. Passage Retrieval

Main Results – Why does BM25 perform better in SQuAD Dataset

1. Annotation Bias
    1. 데이터 셋을 제작할 때 생긴 Bias
    2. SQuAD Dataset을 제작할 때, Passage를 확인하고 이를 바탕으로 Question/Answer을 제작한다. -> Question이 Passage에 Dependent하다.
    3. 이로 인해, BM25가 DPR보다 성능이 좋을 것으로 추측

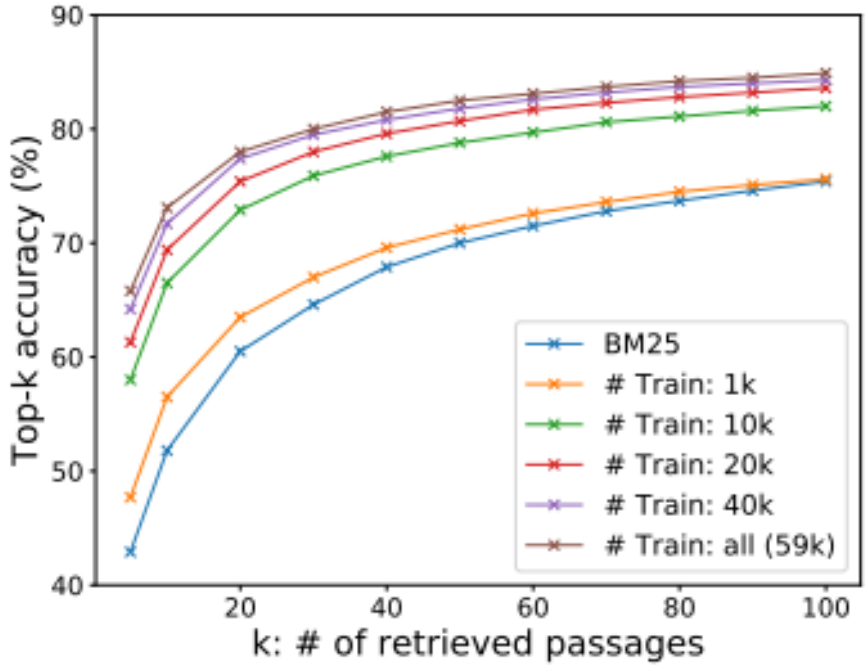
2. Biased Example
    1. SQuAD Dataset은 사람들이 많이 보는 약 500개의 문서로부터 만들어진 Passage들을 가공하여 만든 Dataset. -> 이로부터 Bias가 포함되어 있을 확률이 높다.

3. Passage Retrieval

Ablation Study on Model Training – Sample Efficiency

1. **Training Dataset이 많으면 많을수록 Retrieval Accuracy가 증가한다.**

2. **Top-k**, 즉 Retrieval할 **최종 Passage의 개수를 늘리면 늘릴수록 Retrieval Accuracy가 증가**한다.



✓ **Training Dataset 증가 -> DataAugmentation 적용**
✓ **K를 늘려보면서 실험하기**

3. Passage Retrieval

Ablation Study on Model Training – In-Batch Negative Training

1. First Block -> Random/Bm25/Gold 간의 차이가 많이 없다.
2. Yellow -> **In-Batch Negative Sampling**의 성능 개선
3. Pink -> **Batch_size가 커짐**에 따라 성능 개선
4. Red -> **BM25 Negative Passage**를 사용하면 성능 개선
5. **BM25 Negative Passage는 1개**만 사용하는 것이 적절.

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

✓ **In-Batch Negative Sampling 활용**
✓ **Batch_size를 늘리면서 실험해보기**
✓ **BM25 Negative Sampling을 활용하기**

3. Passage Retrieval

Ablation Study on Model Training – Cross-Dataset Generalization

1. Cross-Dataset Generalization
   -> 추가적인 Fine-tuning없이도 다른 Dataset에 잘 작동하는가?

2. NQ(Natural Question)에 대하여 학습 -> WQ(WebQuestion) / TREC(CurratedTREC)에 적용

3. 상당히 유의미하게 작동한다.

Training	Top-20 WQ Accuracy	Top-20 TREC Accuracy
DPR(Multiple)	**75.0%**	**89.1%**
DPR(NQ)	**69.9%**	**86.3%**
BM25	**55.0%**	**70.9%**

3. Passage Retrieval

Ablation Study on Model Training – Runtime Efficiency

1. **CPU: Intel Xeon CPU E5-2698 v4 @ 2.20GHz and 512GB Memory**
2. **GPU: 32GB**

1. DPR
    1. 초당 995개의 질문을 처리하고, 각 질문당 top-100개의 passages를 반환
    2. Building an index for dense vector
        - **GPU parallelized**
            - **21백만 개의 Passage를 처리하는데 8개의 GPU로 8.8시간 소요**
        - **FAISS index with Single GPU**
            - **21백만 개의 Passage를 처리하는데 8.5시간 소요**

2. BM25
    1. 23.7개의 질문을 per second per CPU thread로 처리한다.
    2. Building an inverted index -> 30분이 채 걸리지 않는다.

✓ **FAISS 적용**

4. Question Answering

End-to-end QA System

1. Retriever로부터 top-k개의 Passage를 Reader Model받는다.
2. 주어진 Passage들에 대하여 **Passage Selection Score**를 부여하고 Passage들을 Re-rank한다.
3. **선택된 Passage로부터 Answer Span을 추출하고 Span Score**를 부여한다.
4. 가장 높은 점수를 가진 Span을 최종

$$P_{\text{start},i}(s) = \text{softmax}\left(\mathbf{P}_i \mathbf{w}_{\text{start}}\right)_s, \quad (3)$$

$$P_{\text{end},i}(t) = \text{softmax}\left(\mathbf{P}_i \mathbf{w}_{\text{end}}\right)_t, \quad (4)$$

$$P_{\text{selected}}(i) = \text{softmax}\left(\hat{\mathbf{P}}^{\intercal} \mathbf{w}_{\text{selected}}\right)_i, \quad (5)$$

where $\hat{\mathbf{P}} = [\mathbf{P}_1^{[\text{CLS}]}, \ldots, \mathbf{P}_k^{[\text{CLS}]}] \in \mathbb{R}^{h \times k}$ and $\mathbf{w}_{\text{start}}, \mathbf{w}_{\text{end}}, \mathbf{w}_{\text{selected}} \in \mathbb{R}^h$ are learnable vectors.

✓ **다양한 Reader 모델 활용하기**
- ✓ Poly-encoders(https://arxiv.org/pdf/1905.01969.pdf)
- ✓ RECONSIDER(https://aclanthology.org/2021.naacl-main.100.pdf)
- ✓ etc

4. Question Answering

End-to-end QA System

1. Retriever로부터 top-k개의 Passage를 Reader Model받는다.
2. 주어진 Passage들에 대하여 **Passage Selection Score**를 부여하고 Passage들을 Re-rank한다.
3. **선택된 Passage로부터 Answer Span을 추출하고 Span Score**를 부여한다.
4. 가장 높은 점수를 가진 Span을 최종

$$P_{\text{start},i}(s) = \text{softmax}\left(\mathbf{P}_i \mathbf{w}_{\text{start}}\right)_s, \quad (3)$$

$$P_{\text{end},i}(t) = \text{softmax}\left(\mathbf{P}_i \mathbf{w}_{\text{end}}\right)_t, \quad (4)$$

$$P_{\text{selected}}(i) = \text{softmax}\left(\hat{\mathbf{P}}^{\intercal} \mathbf{w}_{\text{selected}}\right)_i, \quad (5)$$

where $\hat{\mathbf{P}} = [\mathbf{P}_1^{[\text{CLS}]}, \ldots, \mathbf{P}_k^{[\text{CLS}]}] \in \mathbb{R}^{h \times k}$ and $\mathbf{w}_{\text{start}}, \mathbf{w}_{\text{end}}, \mathbf{w}_{\text{selected}} \in \mathbb{R}^h$ are learnable vectors.

✓ **다양한 Reader 모델 활용하기**
- ✓ Poly-encoders(https://arxiv.org/pdf/1905.01969.pdf)
- ✓ RECONSIDER(https://aclanthology.org/2021.naacl-main.100.pdf)
- ✓ etc

4. Question Answering

End-to-end QA System - Training

1. Retriever로부터 주어진 Passage들 중에서 **1개를 Positive Passage로 m-1개를 Negative Passage로 Sampling하여 학습한다.**
    1. M은 hyperparameter로서 잘 선택해야하며 본 논문에서는 **24를 활용**

2. 선택된 Positive Passage의 Log-likelihood와 함께, Positive Passage에서의 모든 정답 span의 Marginal Log-likelihood를 최대화하는 방향으로 학습한다.
    - 모든 정답 span이란 passage에서 여러 정답 span이 나올 수 있는데 이들 모두를 의미한다.

3. Batch_size: 16(NQ, TriviaQA, SQuAD) / 4(WQ, TREC)

✓ **Small Dataset의 경우 작은 Batch_Size를 활용하여 학습을 진행했다.**

4. Question Answering

Results

1. ORQA/REALM -> Additional pretraining / Expensive end-to-end training regime를 활용
   > 간단한 (Q, P) pair에 대한 DPR Model만을 활용하여 NQ/TriviaQA에서 더 좋은 성능을 이끌었다.

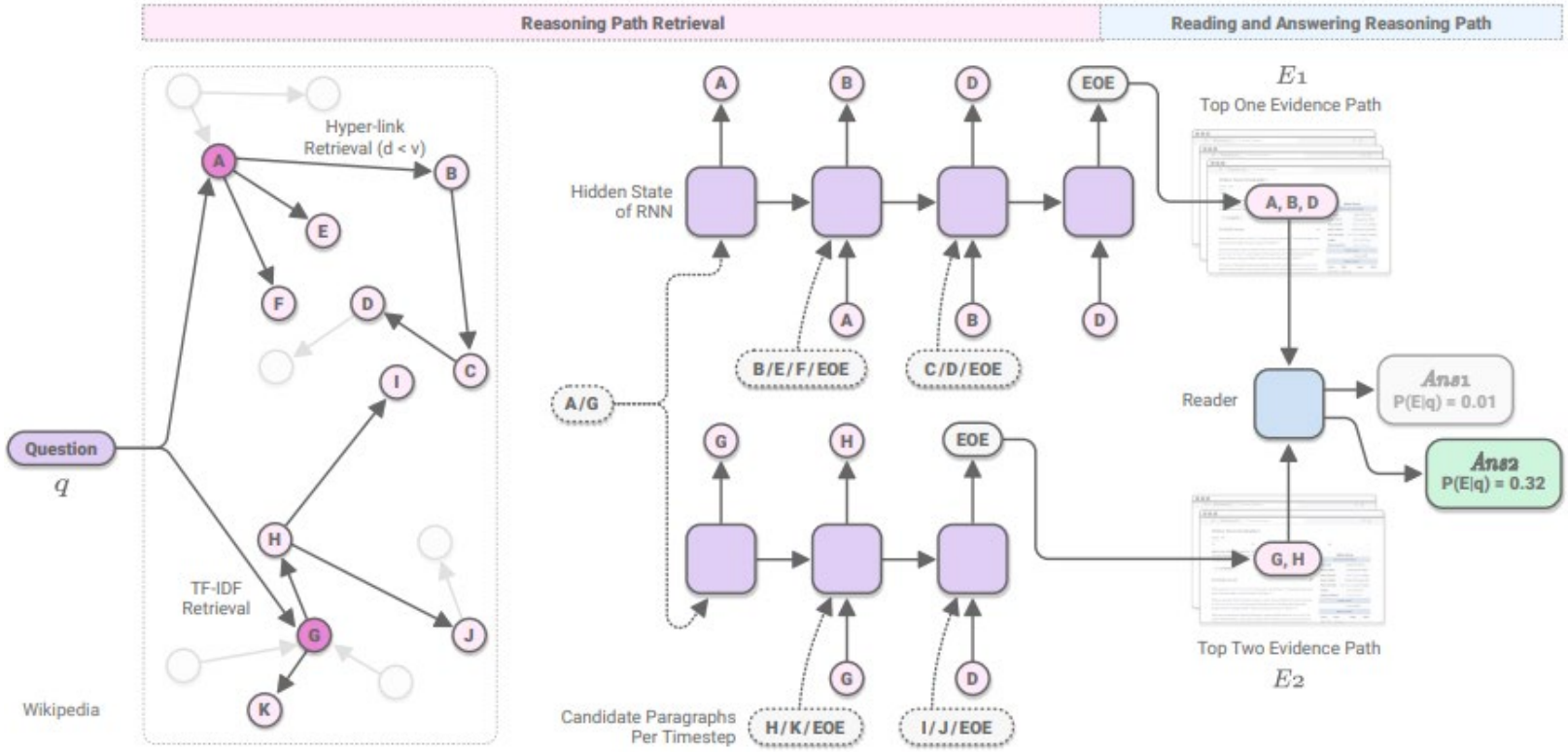
2. Joint Model(Retrieval + Reader)과의 비교 실험을 진행했으며 Joint Model(Latent Retrieval for Weakly Supervised Open Domain Question Answering)에서 EM이 39.8점으로 **Retriever와 Reader를 독립적으로 사용하는 것이 더 좋은 전략**임을 제안한다.

3. **추가적인 Pre-training의 경우, 작은 Dataset에서 더 유용**할 수 있다.

- ✓ **Joint Model < Isolate Retriever/Reader**
- ✓ **If we had small Dataset -> Additional Pre-training is better**

5. Related Work

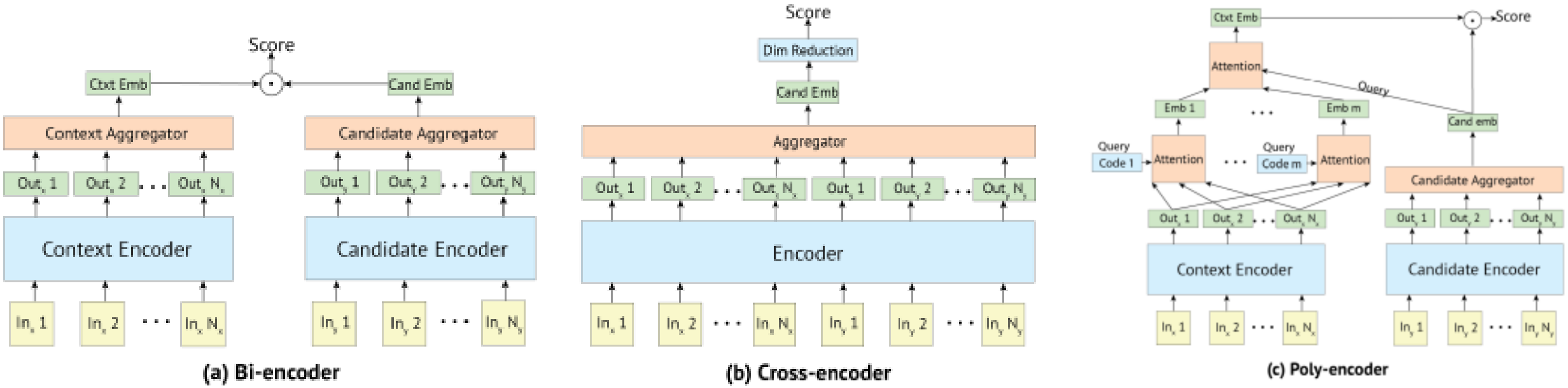
1. LEARNING TO RETRIEVE REASONING PATHS OVER WIKIPEDIA GRAPH FOR QUESTION ANSWERING
   - ➢ 지식 그래프와 위키피디아 하이퍼링크와 같은 외부 구조화된 정보로 텍스트 기반 Retrieval 연구



- LEARNING TO RETRIEVE RASONING PATHS OVER WIKIPEDIA GRAPH FOR QUESTION ANSWERING (https://arxiv.org/pdf/1911.10470.pdf)

5. Related Work

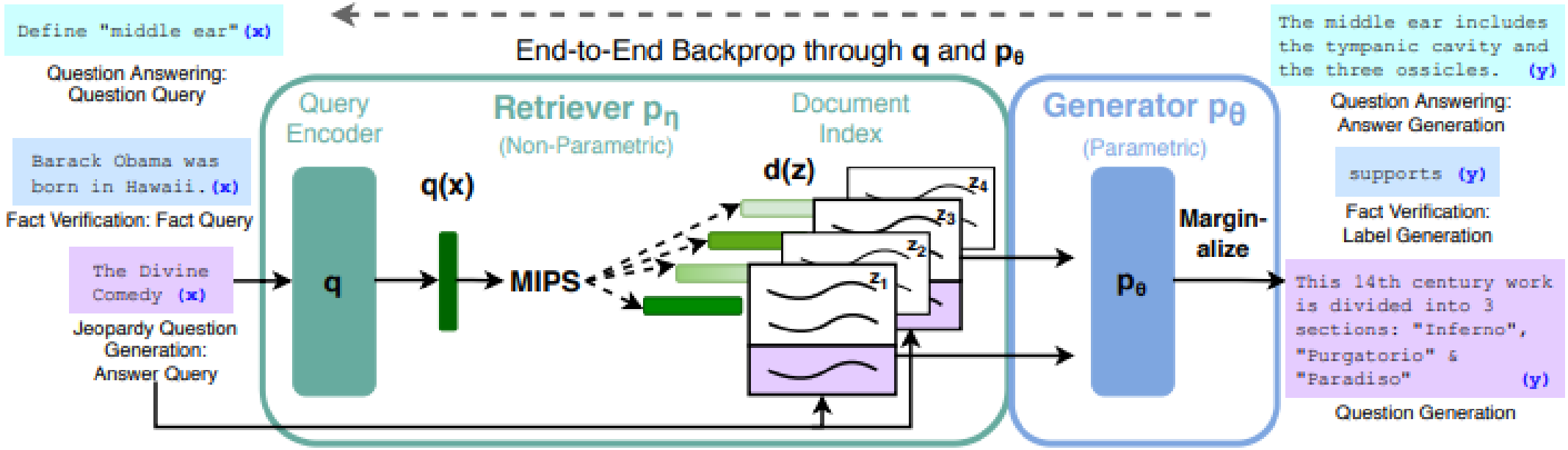
1. POLY-ENCODERS
   - ➢ Bi-Encoder보다는 정확도가 높고, Cross-attention Encoder보다는 빠르다.



- POLY-ENCODERS (https://arxiv.org/pdf/1905.01969.pdf)

5. Related Work

1. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks
   - ➢ Generation Model + DPR
   - ➢ Knowledge-intensive task에서 좋은 성능을 보인다.



- Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks (https://arxiv.org/pdf/2005.11401.pdf)

6. Additional Work

1. RECONSIDER(Facebook AI)
   - Span 내 주석을 사용하여 더 작은 후보 집합에 대해 span 중심 Re-rank 진행

- RECONSIDER(https://aclanthology.org/2021.naacl-main.100.pdf)