Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

연세대학교 컴퓨터과학과 김환희

2024년 10월

과제명: IoT 환경을 위한 고성능 플래시 메모리 스토리지 기반 인메모리 분산 DBMS 연구개발

과제번호: 2017-0-00477

과학기술정보통신부
Ministry of Science and ICT

연세대학교
YONSEI UNIVERSITY

정보통신기술진흥센터
Institute for Information & communications Technology Promotion

Table of Contents

1. Abstract & Introduction

- Large pre-trained Language Models
  - parameters에 factual knowledge(사실적 지식)을 저장하고 다운스트림 NLP 작업에 맞춰 finetuning되었을 때 sota 성능을 보임
  - 암시적인 지식기반으로 외부 메모리에 접근하지 않고 Pre-trained 과정을 통해 많은 지식을 배움

  - knowledge-intensive tasks(지식 집약적 작업)에서는 task-specific architectures(작업 특화 아키텍쳐)에 비해 성능 뒤쳐짐
  - 모델의 결정에 대한 출처 제공 불가
  - 메모리를 쉽게 확장하거나 수정할 수 없어 world knowledge를 업데이트하기 힘듦
  - hallucination을 생성

- **Parametric memory와 non-Parametric memory를 결합한 hybrid model**
  - Non-Parametric memory(명시적인 비매개 변수 메모리)에 대한 미분 가능한 접근 매커니즘
    - Non-Parametric Memory (비매개 변수 메모리)
    - Differentiable Access(미분 가능한 접근)
    - 모델이 외부 메모리를 참고하면서도 학습 과정에서 그 메모리 접근 과정 조정 가능

- 사전 학습 모델은 지식을 수정하고 확장할 수 있으며, 해석할 수 있어 이 문제의 일부 해결

1. Abstract & Introduction

- Open-domain extractive downstream task(추출 기반의 다운스트림 작업)에 대해서만 연구됨
  - Encoder-Only 구조의 BERT 기반, 특정 corpus 안에서 알맞은 부분만 추출하는 방식으로 Open-QA task를 수행
  - 의미를 인코딩하고 표현하는 데는 강점이 있지만, 이를 바탕으로 자연스럽게 새로운 **텍스트를 생성하는 기능은 제한적**

- **Hybrid Parametric memory와 Non-Parametric memory를 seq2seq(시퀀스-투-시퀀스) 모델에 적용**
- **Retrieval-Augmented Generation(검색 기반 생성), 범용 fine-tuning 방법**
  - Pre-trained, parametric-memory generation models에 non-parametric memory를 사용

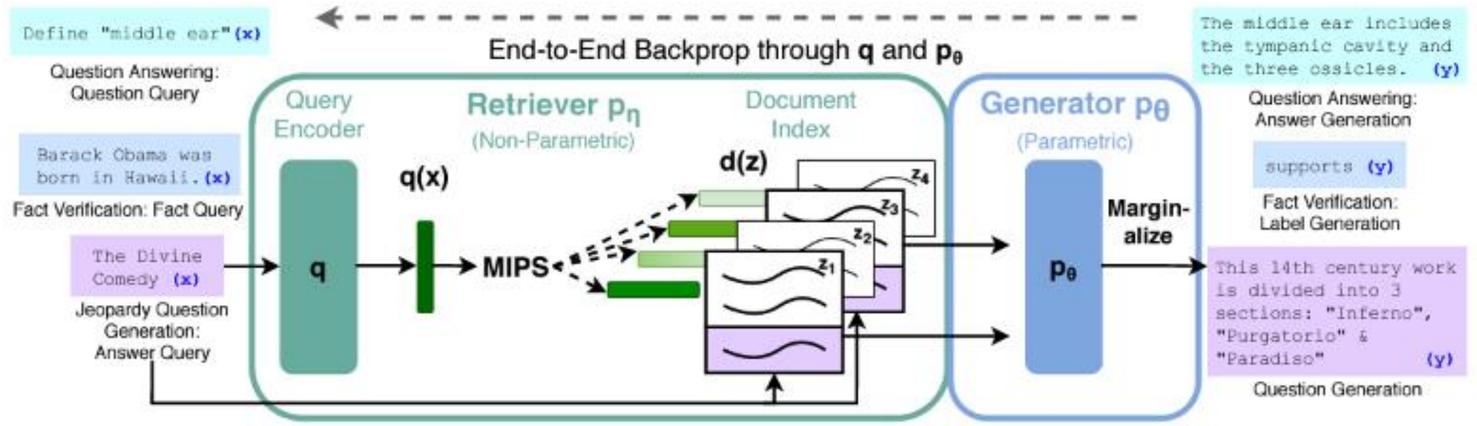


Figure 1: Overview of our approach. We combine a pre-trained retriever (*Query Encoder* + *Document Index*) with a pre-trained seq2seq model (*Generator*) and fine-tune end-to-end. For query x, we use Maximum Inner Product Search (MIPS) to find the top-K documents z_i. For final prediction y, we treat z as a latent variable and marginalize over seq2seq predictions given different documents.

1. Abstract & Introduction

- Non-parametric memory를 활용하는 기존 방법들은 특정 task를 위해 처음부터 훈련됨.

- **반면에 RAG는 방대한 knowledge로 parametric-memory와 non-parametric memory 구성 요소가 모두 pre-trained, pre-load됨**
  - **추가 훈련 없이도 지식에 접근하는 능력을 제공함**

- 논문의 결과는 knowledge-intensive tasks를 위한 생성 과정에서 parametric and non-parametric memory를 결합하는 것의 이점을 강조함
  - non-parametric memory를 대체하여 세계의 변화에 맞춰 모델의 지식을 업데이트할 수 있음

2. Methods

2.1 Models

(i) a retriever $p_\eta(z|x)$ 주어진 쿼리 x에 대하여 텍스트 구절에 대한 (상위 K개의 문서로 축소된) 분포를 반환

(ii) a generator $p_\theta(y_i|x, z, y_{1:i-1})$ 이전 토큰들 $y_{1:i-1}$, 원래 입력 x, 검색된 구절 z의 컨텍스트를 기반으로 현재 토큰 y_i를 생성

- Retriever와 generator를 end-to-end로 학습시키기 위해, retrieved document를 latent variable(잠재 변수) 취급
  - Marginalization(주변화): latent document들을 다른 방식 marginalization하는 두 가지 모델 제안

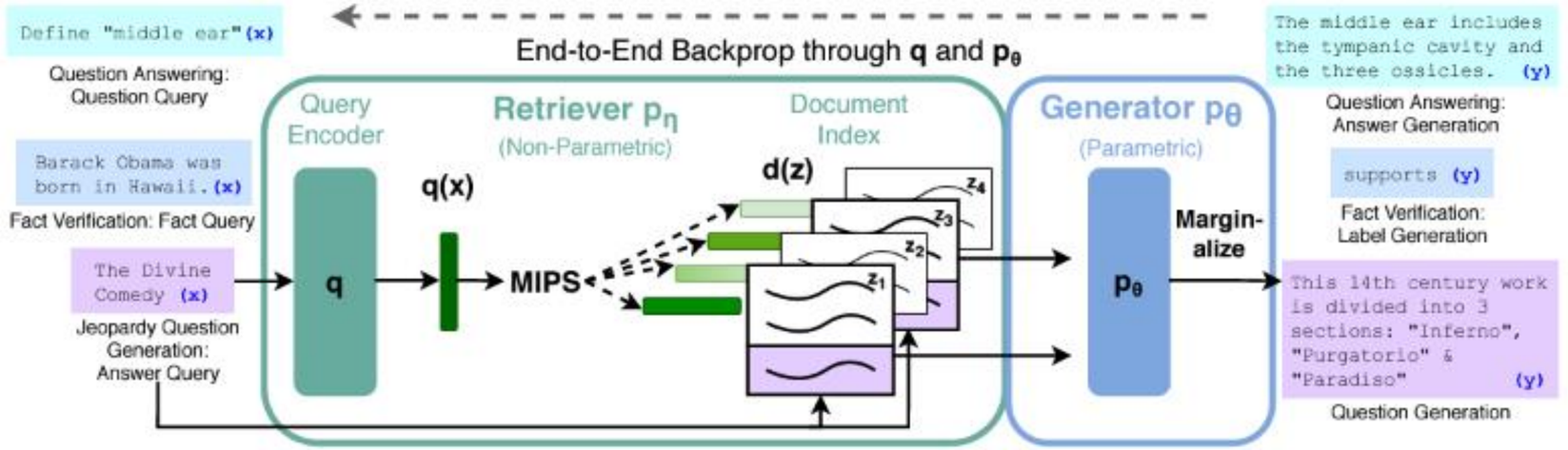


Figure 1: Overview of our approach. We combine a pre-trained retriever (*Query Encoder* + *Document Index*) with a pre-trained seq2seq model (*Generator*) and fine-tune end-to-end. For query x, we use Maximum Inner Product Search (MIPS) to find the top-K documents z_i. For final prediction y, we treat z as a latent variable and marginalize over seq2seq predictions given different documents.

2. Methods

2.1 Models

- RAG-Sequence Model
  - 검색된 문서를 단일 latent variable로 취급, 전체 시퀀스를 생성하기 위해 동일한 검색 문서 사용
  - top-k 근사를 통해 seq2seq 확률 $p(y|x)$를 얻기 위해 marginalize
  - retriever를 통해 상위 K개의 문서 검색, , generator 는 각 문서에 대해 출력 시퀀스 확률을 생성, marginalize, 출력 시퀀스를 생성

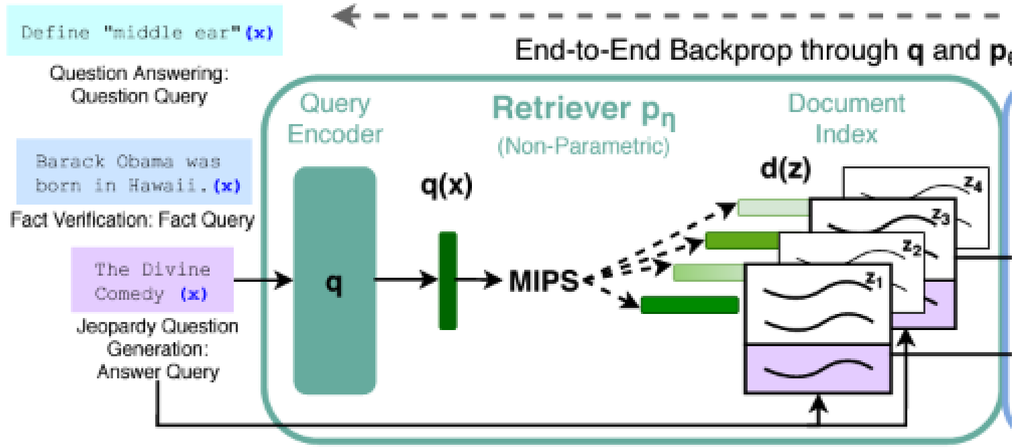
$$p_{\text{RAG-Sequence}}(y|x) \approx \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y|x,z) = \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) \prod_i^N p_\theta(y_i|x,z,y_{1:i-1})$$

- RAG-Token Model
  - 각 target token마다 다른 latent document를 선택하여 이에 따라 marginalize 가능
  - generator 답변을 생성할 때 여러 문서에서 콘텐츠를 선택 가능
  - retriever를 통해 상위 K개의 문서 검색, generator는 각 문서에 대한 다음 출력 토큰에 대한 분포를 생성, marginalize, 이 프로세스를 그 다음 출력 토큰에 대하여 반복

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

2. Methods

2.2 Retriever: DPR



- Retriever $p_\eta(z|x)$는 **Dense Passage Retrieval을 기반**으로 함
  - Dense Passage Retrieval(DPR)
    - 질문 응답(QA) 작업을 위한 검색 시스템, 관련성이 높은 문서를 밀집 벡터(dense vector) 형태로 표현하여 검색하는 방식
    - 쿼리와 문서를 별도로 인코딩하는 양방향 인코더(bi-encoder) 구조

$$\mathbf{d}(z) = \text{BERT}_d(z), \quad \mathbf{q}(x) = \text{BERT}_q(x)$$

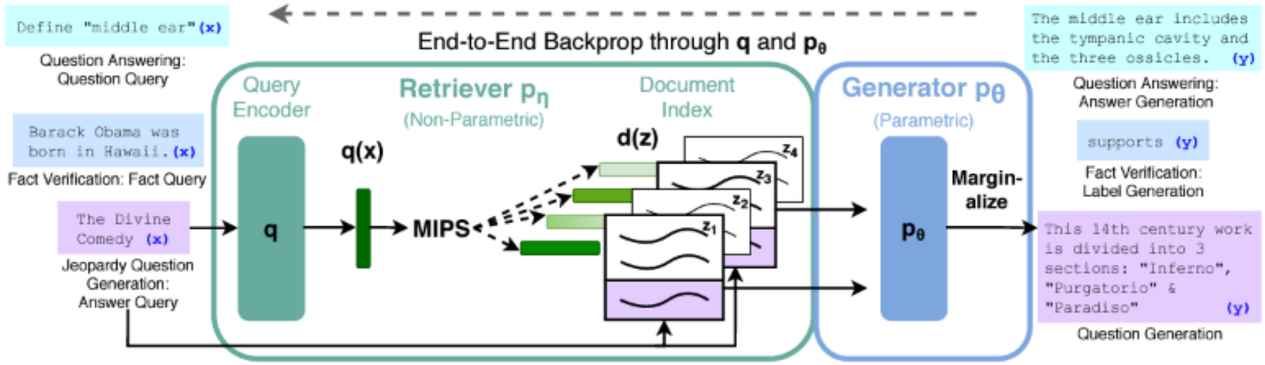
$$p_\eta(z|x) \propto \exp\left(\mathbf{d}(z)^\top \mathbf{q}(x)\right)$$

$$\text{top-k}(p_\eta(\cdot|x))$$

    - 산출된 $d(x)$와 $q(z)$의 내적 연산을 기반으로 유사도를 계산
    - **유사도**($p_\eta(z|x)$)가 높은 순서대로 top-k document를 골라 retrieve
    - Retrieve할 때 **Maximum Inner Product Search(MIPS) 알고리즘** 사용하여 sub-liner time 탐색이 가능

- DPR에서 Pretrain된 bi-encode를 사용하여 Retriever를 초기화하고 문서 인덱스 구축
- 문서 인덱스: **Non-parametric memory**

2. Methods



2.3 Generator: BART

- Generator $p_\theta(y_i|\,\mathrm{x}, \mathrm{z}, y_{i-1})$
  - 다양한 encoder-decoder 아키텍처를 generator 모델로 사용 가능
  - 이 논문에서는 4억 개의 매개변수를 가진 pretrained seq2seq transformer BART-large를 generator로 사용

  - BART에서 시퀀스를 생성할 때, input x와 검색된 콘텐츠 z를 단순히 concatenate해서 결합
  - BART는 denoising 목표와 다양한 잡음 함수들을 사용하여 pre-train 됨
  - BART 생성기의 매개변수 θ: **Parametric memory**

2.4 Training

- DPR기반의 retriever와 BART-large 기반의 generator는 training 과정에서 동시에 학습됨
- 어떤 문서가 검색되어야 한다는 direct supervision은 주어지지 않음

- Fine-tuning training corpus
- 각 타켓의 Negative marginal log-likelihood를 최소화하는 방향으로 Adam을 사용한 확률적 경사 하강법으로 학습됨
- Document encoder($BERT_d$, index)는 고정하고, Query encoder($BERT_q$)와 BART Generator만 fine-tuning함

input/output pairs (x_j, y_j)

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

2. Methods

2.5 Decoding

- 테스트 시점에서 RAG-Sequence와 RAG-Token은 $argmax_y p(y|x)$를 근사화하는 방법이 다름

- **RAG-Token**
  - Transition possibility를 가지는 표준 autoregressive seq2seq generator
  - Decoding을 위해 $p'_\theta(y_i|x, y_{1:i-1})$을 표준 beam decoder에 그대로 적용할 수 있음

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

- **RAG-Sequence**
  - $p(y|x)$를 토큰별 확률로 나눌 수 없어서, 단일 beam search로 해결할 수 없음
  - 상위 k개의 검색된 문서 z 각각에 대해 beam search를 실행하고, 각 가설을 $p_\theta(y_i|x, z, y_{i:i-1})$를 사용하여 점수화 -> 가설 집합 Y 생성
    - 일부 가설은 모든 문서의 beam search에서 나타나지 않을 수 있음
    - Thorough Decoding
      - 특정 가설 y의 확률을 추정하기 위해, y가 beam에 나타나지 않은 문서 z에 대해 추가적인 forward pass를 실행
      - generator 확률에 $p_\eta(z|x)$를 곱한 후 beam 간 확률을 합산하여 marginalize
    - Fast Decoding
      - x, z_i로부터 y가 생성되지 않은 경우, $p_\theta(y|x, z_i) \approx 0$으로 근사화하는 추가적인 접근

3. Experiments

- 다양한 knowledge-intensive tasks에서 RAG를 실험
- Non-parametric knowledge source로 단일 Wikipedia dump 파일을 사용
  - 문서 인코더를 사용하여 각 문서의 임베딩 계산
  - 빠른 검색을 위해 Hierarchical Navigable Small World(HNSW) 근사를 사용하는 FAISS(Facebook AI Similarity)를 이용
  - 단일 MIPS 인덱스(Maximum Inner Product Search Index)를 생성

3.1 Open-domain Question Answering

- 질문과 답변을 입력-출력 텍스트 쌍 (x,y)으로 처리하면서 RAG를 정답의 음의 로그 가능도를 직접 최소화여 학습

- RAG를 검색된 문서 내에서 정답을 포함하는 구절을 추출하는 추출형 QA 패러다임(Extractive QA)과 비교
  - Non-parametric knowledge(비매개 변수 지식), 검색된 문서의 정보에 주로 의존
- RAG와 폐쇄형 QA(Closed-Book QA) 방식과도 비교
  - 외부 문서나 검색 과정 없이 모델의 내부 지식(Parametric knowledge)에만 의존하여 답변을 생성하는 방식

- 네 가지 오픈 도메인 QA 데이터셋을 사용
  - Natural Questions (NQ), TriviaQA (TQA). WebQuestions (WQ), CuratedTrec (CT)
  - Exact Match(EM) 점수로 평가: 모델이 생성한 답변이 정답과 완전히 일치할 때만 정답으로 간주하는 엄격한 평가 기준
  - TriviaQA(TQA) 데이터셋에서, T5 모델과 비교하기 위해 TQA Wiki 테스트 세트를 사용하여 성능을 비교

3. Experiments

3.2 Abstractive Question Answering (추상적 질문 응답)

- RAG 모델은 단순한 추출형 QA를 넘어, 자유 형식의 생성형(Abstractive) 텍스트 생성으로 질문에 답변 가능

- 자연어 생성 능력을 테스트하기 위해 MSMARCO NLG 작업 사용
  - 각 질문에 대해 검색 엔진에서 검색된 10개의 정답 구절(gold passage)과 검색된 구절에서 작성된 완전한 문장 형태의 답변으로 구성됨
  - 제공된 정답 구절은 사용하지 않고 질문과 답변만 사용
  - "캘리포니아, 볼케이노의 날씨는 어떻습니까?"와 같은 질문이 포함됨
  - 일부는 Wikipedia만으로는 답변할 수 없는 경우도 있음, 이 경우 RAG는 매개 변수 지식(parametric knowledge)을 활용하여 합리적인 답변을 생성

3. Experiments

3.3 Jeopardy Question Generation

- 일반적으로 짧고 단순한 질문으로 구성된 기존의 오픈 도메인 QA 과제 대신, Jeopardy 질문을 생성하는 더 어려운 작업을 제안
- Jeopardy는 특정 사실에 기반하여 해당 사실과 관련된 엔티티를 추측하는 방식
  - Jeopardy 질문: 1986년에 멕시코가 이 국제 스포츠 대회를 두 번 개최한 첫 국가로 기록되었습니다.
  - 답변: The World Cup

  - SearchQA 데이터셋을 사용, RAG와 비교를 위해, BART 모델도 Jeopardy 질문 생성을 위해 학습
  - SQuAD에 최적화된 Q-BLEU-1 메트릭을 사용하여 평가
  - 두 가지 인간 평가
    - Factuality(사실설): 생성된 질문이 신뢰할 수 있는 외부 출처로부터 검증 가능한지 평가
    - Specificity(특이성): 생성된 질문이 주어진 답변과 높은 상호 의존성을 가지고 있는지를 평가
    - Pairwise comparative evaluation
      - 평가자들에게 RAG와 BART에서 각각 생성된 질문을 제시하고, 둘 중 어느 질문이 더 나은지, 혹은 둘 다 좋거나 둘 다 좋지 않은지를 선택하도록 함
      - 질문 A가 더 좋다, 질문 B가 더 좋다, 둘 다 좋다, 혹은 둘 다 좋지 않다.

3. Experiments

3.4 Fact Verification (사실 검증)

- FEVER 데이터셋
  - RAG 모델이 주어진 주장에 대해 Wikipedia의 관련 증거를 검색
  - 주장이 지지(supports)되는지, 반박(refutes)되는지, 또는 판단할 정보가 충분하지 않음(not enough info)을 분류하도록 함
  - 복잡한 포함 추론(entailment reasoning) 과제를 포함, 생성(generation)이 아닌 분류(classification)를 다루는 능력을 탐구하기 적절함

- 대부분의 FEVER 접근 방식과 달리, 검색된 증거에 대한 감독(supervision)을 사용하지 않음

- 표준 3가지 분류 작업: 지지(supports), 반박(refutes), 또는 판단할 정보가 충분하지 않음(not enough info)
- 2가지 분류 작업: 지지(supports), 반박(refutes)
  - Thorne과 Vlachos의 연구에서 사용된 방식

- label accuracy(레이블 정확도)를 통해 모델의 성능을 평가

4. Results

4.1 Open-domain Question Answering

- RAG는 폐쇄형(Closed-Book) 접근 방식의 생성 유연성과 오픈북(Open-Book) 검색 기반 접근 방식의 성능(정확도)가 결합
- 기존의 REALM이나 T5+SSM 모델들은 Salient Span Masking과 같은 특수 사전 학습이 필요하지만, RAG는 이러한 추가인 사전 학습 없이도 우수한 성능
- BERT 기반 크로스 인코더로 문서를 재순위(re-rank)하는 DPR QA 시스템과 비교할 때도 경쟁력 있는 성능을 보여줌
    - RAG가 최첨단 성능을 달성하기 위해 재순위자나 추출형 리더(extractive reader)가 필요하지 않음을 증명

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**

- 답변을 추출할 수 있는 경우에도, 생성 방식을 사용하는 것의 장점
    - 답변을 직접 포함하지 않더라도 단서를 포함하는 문서들을 활용하여 정답 생성에 기여 가능
    - 검색된 문서에 정답이 포함되어 있지 않은 경우에도 RAG는 정답을 유추하여 생성 가능

4. Results

4.2 Abstractive Question Answering

- RAG-Sequence는 Open MS-MARCO NLG 작업에서 BART를 BLEU 점수 2.6과 ROUGE-L 점수 2.6 차이로 능가

- RAG는 다음과 같은 점을 고려했을 때(3가지)도 최신 성능에 근접한 성과를 보여줘 인상적인 결과

1. 다른 최신 모델들은 정답 구절(gold passages)을 활용해 참조 답변을 생성하는 데 필요한 특정 정보를 이용
2. 많은 질문이 정답 구절 없이는 답할 수 없음
3. 모든 질문이 Wikipedia만으로는 답변될 수 없음

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 B-1	Jeopardy QB-1	MSMARCO R-L	MSMARCO B-1	FVR3 Label Acc.	FVR2 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 3: Examples from generation tasks. RAG models generate more specific and factually accurate responses. '?' indicates factually incorrect responses, * indicates partially correct responses.

Task	Input	Model	Generation
MS-MARCO	define middle ear	BART	?The middle ear is the part of the ear between the middle ear and the nose.
		RAG-T	The middle ear is the portion of the ear internal to the eardrum.
		RAG-S	The middle ear includes the tympanic cavity and the three ossicles.
	what currency needed in scotland	BART	The currency needed in Scotland is Pound sterling.
		RAG-T	Pound is the currency needed in Scotland.
		RAG-S	The currency needed in Scotland is the pound sterling.
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"

4. Results

4.3 Jeopardy Question Generation

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 B-1	Jeopardy QB-1	MSMARCO R-L	MSMARCO B-1	FVR3 Label Acc.	FVR2 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 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%

Table 3: Examples from generation tasks. RAG models generate more specific and factually accurate responses. '?' indicates factually incorrect responses, * indicates partially correct responses.

Task	Input	Model	Generation
MS-MARCO	define middle ear	BART	?The middle ear is the part of the ear between the middle ear and the nose.
		RAG-T	The middle ear is the portion of the ear internal to the eardrum.
		RAG-S	The middle ear includes the tympanic cavity and the three ossicles.
	what currency needed in scotland	BART	The currency needed in Scotland is Pound sterling.
		RAG-T	Pound is the currency needed in Scotland.
		RAG-S	The currency needed in Scotland is the pound sterling.
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"

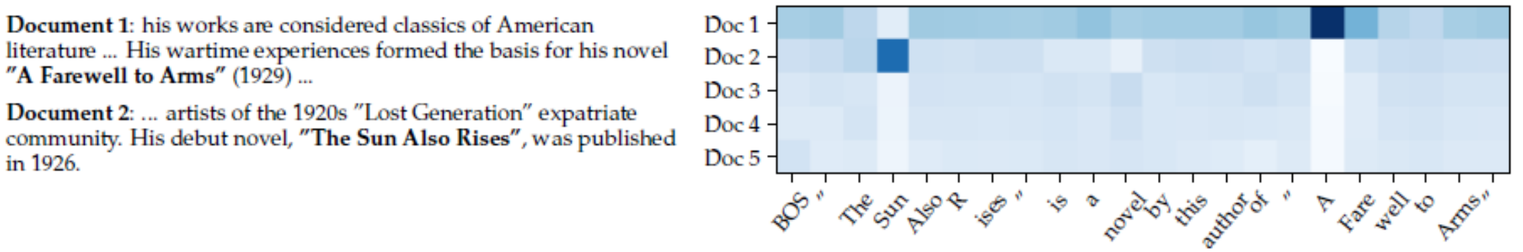


Figure 2: RAG-Token document posterior $p(z_i|x, y_i, y_{-i})$ for each generated token for input "Hemingway" for Jeopardy generation with 5 retrieved documents. The posterior for document 1 is high when generating "A Farewell to Arms" and for document 2 when generating "The Sun Also Rises".

4. Results

4.4 Fact Verification

- 3가지 분류(3-way classification) 작업에서 RAG는 SOTA 모델보다 4.3% 낮은 정확도를 기록
  - SOTA 모델은 도메인 특화 아키텍처와 상당한 엔지니어링이 적용된 복잡한 파이프라인 시스템, 중간 검색 감독(intermediate retrieval supervision)을 사용해 학습됨, RAG는 이러한 감독 없이 학습됨

- 2가지 분류(2-way classification)에서는 Thorne과 Vlachos의 모델과 비교
  - 이들은 RoBERTa를 사용해, 골드 증거(gold evidence) 문장이 주어진 상황에서 주장을 참(true) 또는 거짓(false)으로 분류하도록 학습함.
  - RAG는 주장만을 입력으로 받아 스스로 증거를 검색했음에도 불구하고 이 모델보다 2.7% 낮은 정확도를 기록함

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 B-1	Jeopardy QB-1	MSMARCO R-L	MSMARCO B-1	FVR3 Label Acc.	FVR2 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>		

4. Results

4.5 Additional Results

- 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%

- 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		

- Index hot-swapping
  - DrQA의 2016년 12월 Wikipedia dump의 인덱스를 사용하는 RAG와 2018년 12월 최신 인덱스를 사용한 결과 비교
  - 2016년 인덱스를 사용하여 2016년 지도자에 대해 70%의 정확도
  - 2018년 인덱스를 사용하여 2018년 지도자에 대해 68%의 정확도

- Effect of Retrieving more documents

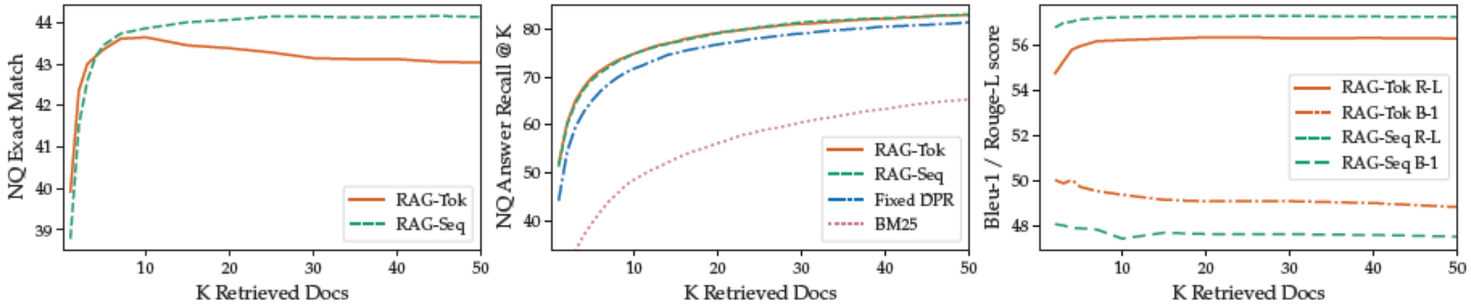


Figure 3: Left: NQ performance as more documents are retrieved. Center: Retrieval recall performance in NQ. Right: MS-MARCO Bleu-1 and Rouge-L as more documents are retrieved.

5. Related Work

- Single-Task Retrieval
  - 검색이 다양한 NLP 작업에서 성능을 향상시킨다는 것을 보여줌
  - 이전의 검색 기반 작업 성공 사례들을 통합하여, 단일 검색 기반 아키텍처가 여러 작업에서 강력한 성능을 달성할 수 있음을 보여줌

- General-Purpose Architectures for NLP
  - 검색 모듈을 학습하여 사전 학습된 생성 언어 모델을 보완함으로써, 단일 통합 아키텍처로 처리할 수 있는 작업의 범위를 확장

- Learned Retrieval
  - 단일 작업에서 강력한 성능을 달성하기 위해 서로 다른 검색 기반 아키텍처와 최적화 기법을 활용했지만, 논문에서는 단일 검색 기반 아키텍처가 여러 작업에서 우수한 성능을 위해 파인 튜닝될 수 있음을 보여줌

- Memory-based Architectures
  - 사람이 읽을 수 있고, 모델에 해석 가능성을 부여, 문서 인덱스를 편집하여 모델의 메모리를 동적으로 업데이트할 수 있는 장점

- Retrieve-and-Edit approaches
  - 검색된 항목을 간단히 편집하는 데 중점을 두기보다는, 여러 검색 결과의 내용을 통합하고, 잠재 검색(latent retrieval)을 학습하며, 관련 학습 쌍 대신 증거 문서를 검색하는 방식에서 차이

6. Discussion

- 매개 변수(parametric) 및 비매개 변수(non-parametric) 메모리에 접근할 수 있는 하이브리드 생성 모델(RAG)을 제시

- RAG 모델이 오픈 도메인 QA에서 State-of-the-art 성능을 달성함을 보여줌
- 사람들이 순수 매개 변수 기반인 BART보다 RAG의 생성 결과를 선호하며, 이를 더 사실적이고 구체적이라고 평가
- 학습된 검색 구성 요소를 조사하여 구성 요소의 효과 검증, 검색 인덱스를 교체(hot-swap)하여 재학습 없이 모델을 업데이트할 수 있는 방법을 제시

- 향후 연구에서는 두 구성 요소(Retriever, Generator)를 BART와 유사한 Denoising 목표나 다른 목표를 통해 처음부터 함께 학습할 가능성을 탐구하는 것이 유익할 수 있음

감사합니다.