Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal,
Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, Douwe Kiela

Facebook AI Research, University College London, New York University

NeurIPS 2020
2024.05.27

발제자:
윤예준

연구 배경

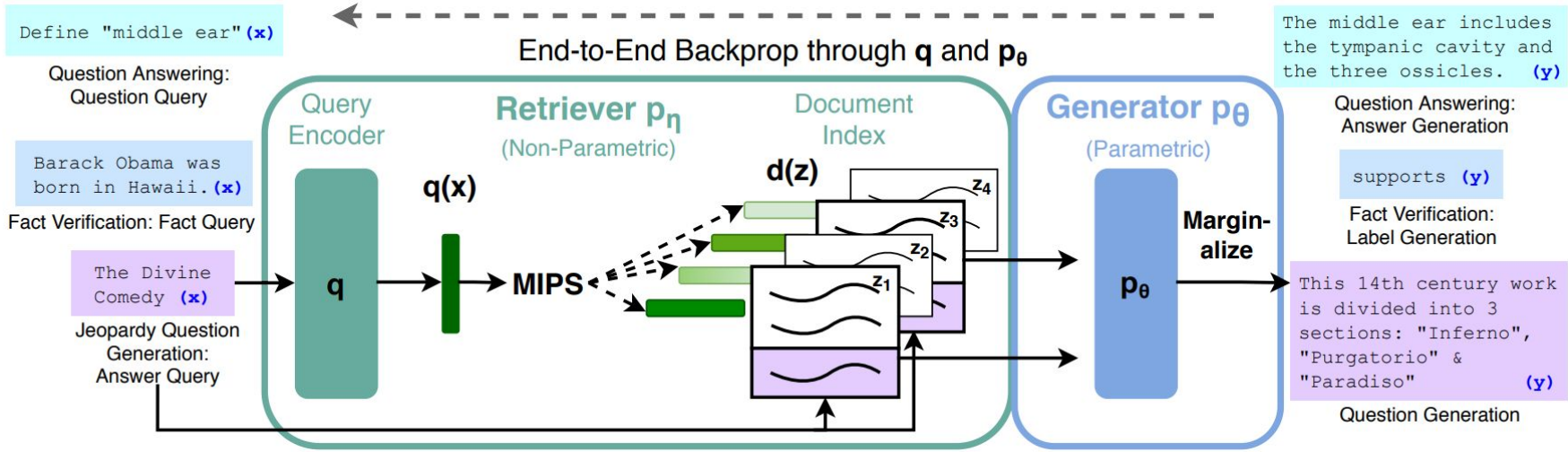
- Parametric knowledge Bases: T5, GPT-3
  → 외부 지식 접근 없이 추론 가능
  → 지식 확장 및 수정 어려움
  → hallucination 발생 가능

- Non-Parametric Knowledge Bases: REALM, ORQA
  → 지식 수정 가능
  → answer를 생성하는 것이 아니기 때문에 answer 범위 제한

Pre-trained parametric와 Non-parametric memory를 결합한 Language Generation Model 제안

Method - Overview

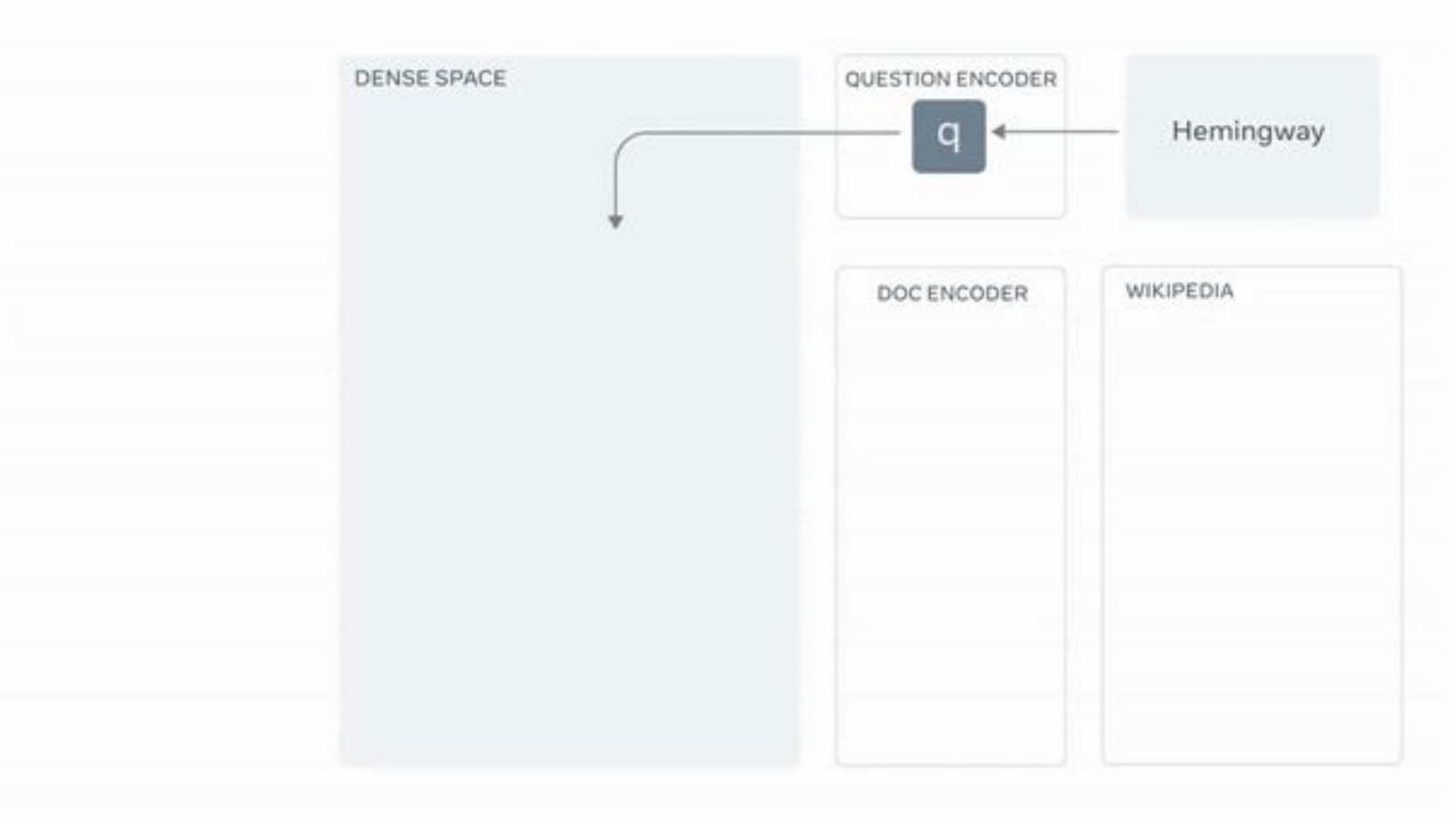
- RAG(Retrieval-Augmented Generation) Model
  - Step 1: Query가 들어오면 Query Encoder를 통해 q(x) 생성
  - Step 2: MIPS(Maximum Inner Product Search) 이용 q(x)와 가장 가까운 Top-k개의 Document 탐색
  - Step 3: Query와 Document를 concat하여 Generator 입력으로 사용
  - Step 4: 각 concat(Query, Document)간 Generator 출력을 Marginalize하여 최종 출력 생성



- Query Encoder: DPR Query Encoder – BERT Base
- Retriever: DPR Document Encoder – BERT Base
- Generator: BART Large

DPR(Dense Passage Retrieval) – bi-encoder architecture

Method - Overview



Method - Model

- RAG-Sequence Model

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

- RAG-Token Model

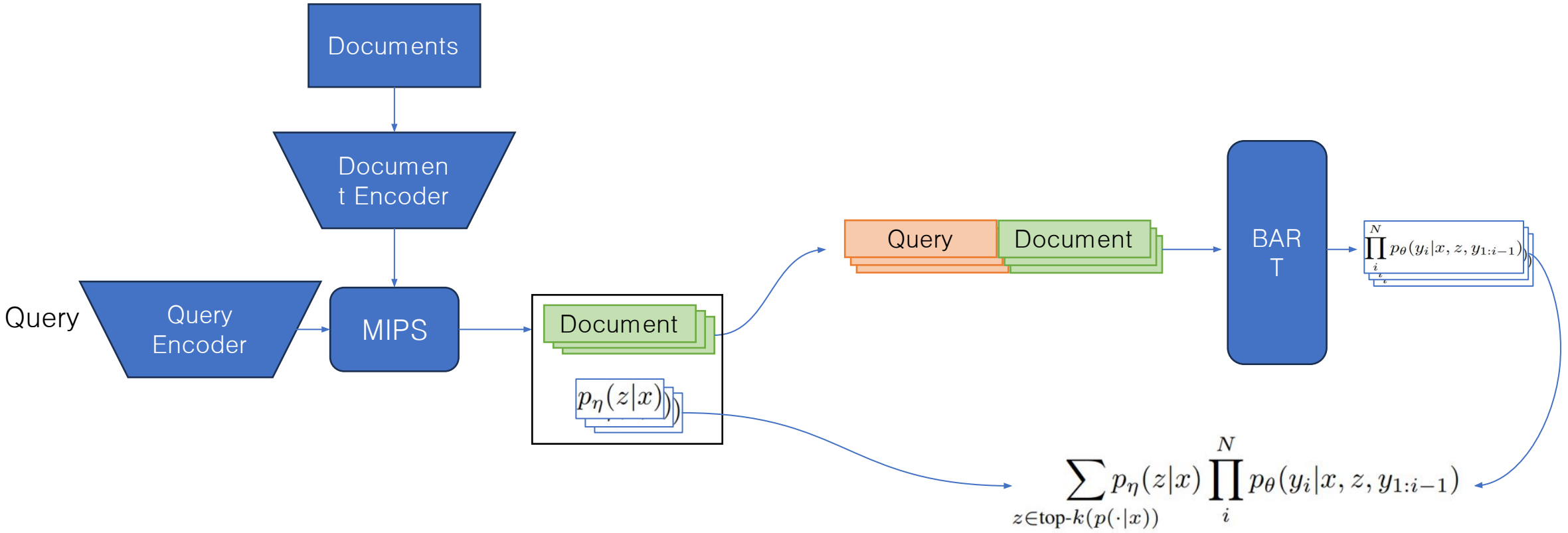
$$p_{\text{RAG-Token}}(y|x) \approx \prod_i^N \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y_i|x,z_i,y_{1:i-1})$$

- Marginalize 방법에 따라 두 가지 모델 제안
- RAG-Sequence: 각 document마다 sequence 생성 후 marginalize
- RAG-Token: 토큰 생성 마다 marginalize

Method – RAG-Sequence Model

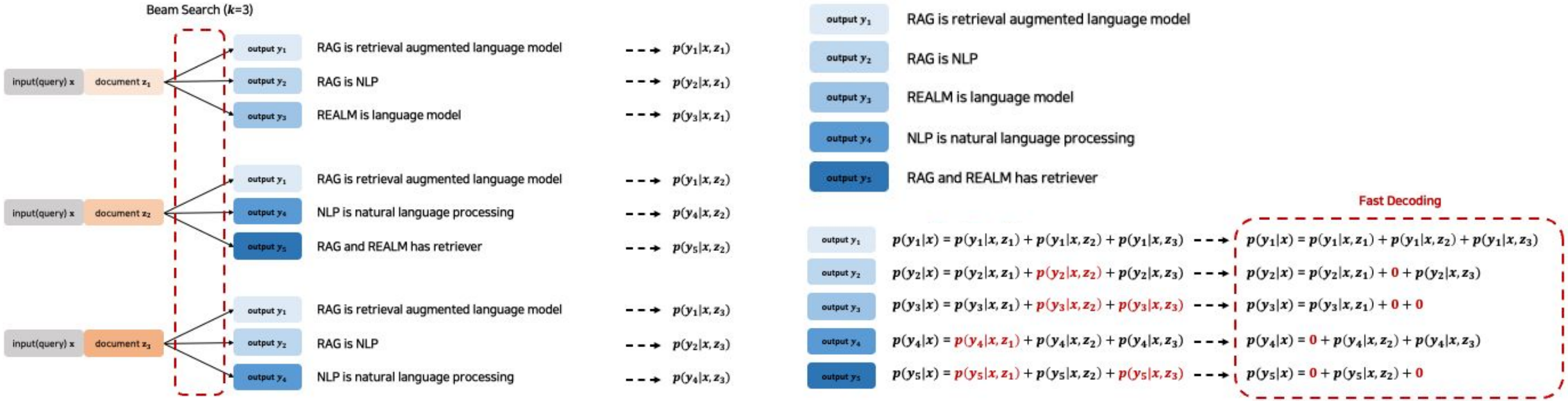
- Train

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



Method – RAG-Sequence Model

- Decoding
  - Sequence를 끝까지 생성 후 document에 대해 marginalize 진행 -> 기존의 beam search 적용 불가
  - 해결 방법: 각 document 별로 beam search 적용
  - 문제점: beam search 과정에서 생성되지 않은 확률 존재
    → 존재하지 않은 경우 additional forward pass 진행



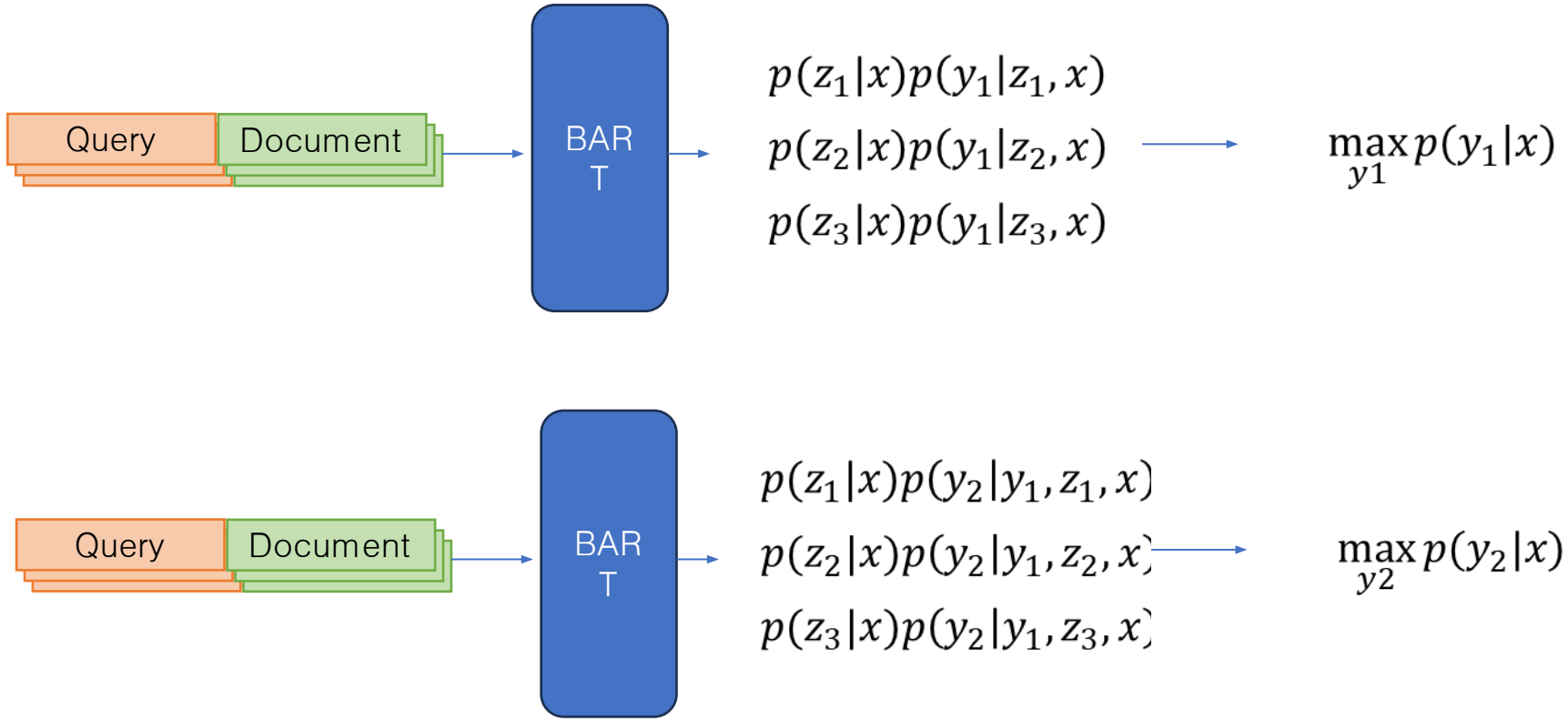
- 위 방법은 긴 sequence일수록 매우 비효율적 → Fast Decoding 제안

Figure: https://gbdai.tistory.com/67

Method – RAG-Token Model

- Train

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



- 기존 Decoding 방법 사용 가능

결과

- Open-Domain Question Answering

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

	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**

- 외부 지식을 사용하지 않은 Closed-Book setting보다 월등히 좋은 성능
- Retrieval과 Extractive system에 의존하는 Open Book setting보다 좋은 성능
- 검색된 문서에 정답이 없는 경우에도 정답 생성 가능을 보임 (RAG: 11.8% vs Extractive model: 0%)

결과

- Knowledge Intensive Tasks
  - Jeopardy Question Generation: 주어진 정답 entity를 가지고 그 정답에 상응하는 질문을 생성하는 task

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		

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"

- Abstractive Question Answering & Jeopardy Question Generation & Fact Verification
  - BART보다 좋은 성능 + 적은 hallucination + 다양성↑
  - SotA: 정답과 관련성이 높은 문서를 직접 입력 -> Retrieval 없이 gold passage가 입력됨

결과

- Ablation Study

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		

- RAG의 Retriever를 BM25로 교체
  → 대부분 task에서 낮은 성능
  → FEVER에선 높은 성능: FEVER의 claim이 entity-centric이기 때문
- Retriever Freeze: Generator만 학습 → Retriever로 같이 학습하는 것이 효과적
- Knowledge base 변경에 따른 성능 변화
  - 각국 지도자를 맞추는 task: “Who is {position}?” (e.g. Who is the President of Pero?”)
  - 단순히 non-parametric memory만 변경함으로써 RAG의 world knowledge를 변경할 수 있음을 보임

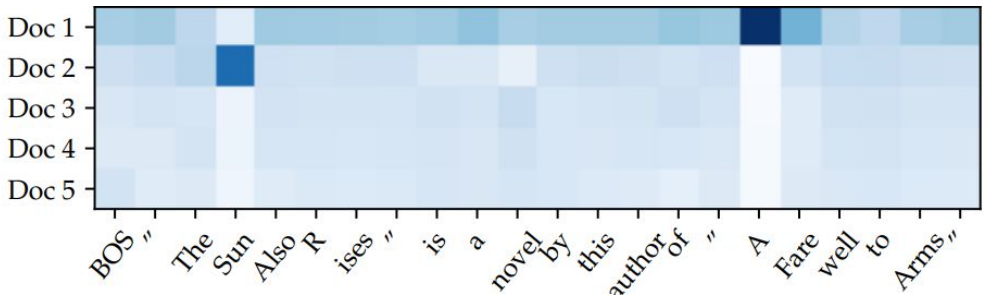
ACC	2016 wiki	2018 wiki
2016 leader	70%	4%
2018 leader	12%	68%

결과

- RAG-Token model로 Hemingway를 입력으로 질문 생성 시 posterior $p(z_i|x, y_i, y_{-i})$ 시각화

Document 1: his works are considered classics of American literature ... His wartime experiences formed the basis for his novel **"A Farewell to Arms"** (1929) ...

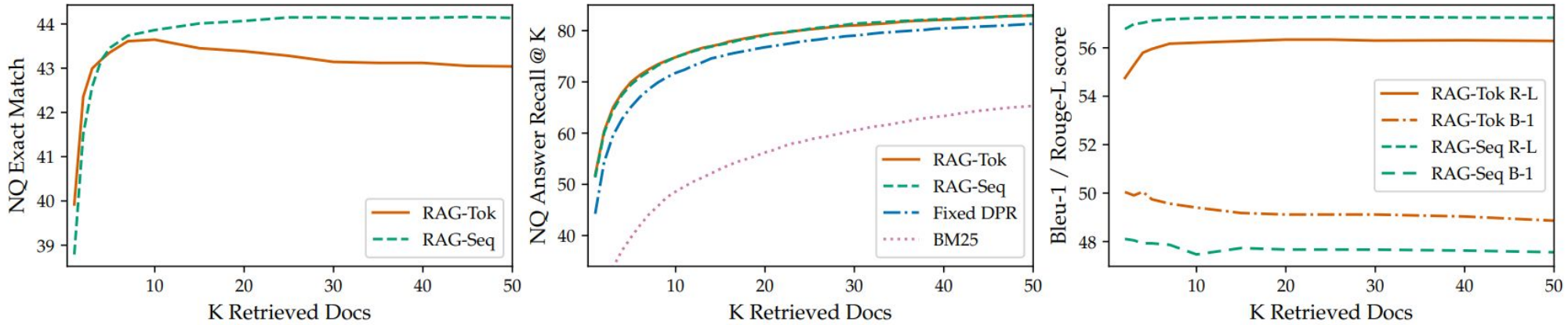
Document 2: ... artists of the 1920s "Lost Generation" expatriate community. His debut novel, **"The Sun Also Rises"**, was published in 1926.



- 각 책 제목을 생성할 때 관련된 문서에 높은 점수 부여
- 책 제목의 첫 토큰을 생성하면 특정 문서에 덜 집중 → parametric knowledge 이용 추정

결과

- Retrieving more documents 효과 (왼쪽부터 A, B, C)



- (A,B): RAG-Seq의 경우 Retrieved Docs가 많을수록 성능 개선
- (A,B): RAG-TOK의 경우 Retrieved Docs가 일정 수 이상되면 성능 저하
- (C): Abstractive Answer Generation의 경우 많은 Docs가 필요하지 않음

결론

- Pre-trained parametric와 Non-parametric memory를 결합한 Language Generation Model 제안
- Non-parametric memory를 통해 LM의 지식 수정 가능
- Generator로 인해 다양한 출력 생성 가능

Open Questions

- 당연 데이터셋마다 RAG-Seq, RAG-Token이랑 성능이 다른데 어떻게 이를 선택하는 것이 좋을까?
- Retrieved Docs가 많을 수록 성능 개선이 될 것 같은데 왜 안되는가?

감사합니
다.