Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

Sangho Daniel Kim

www.linkedin.com/in/danieliscoding

Natural Language Processing

2024/05/21

TRAIN AND TEST

Contents

TRAIN AND TEST

Previously on…

Recap

Previous Problem

"However, these models often express unintended behaviors such as making up facts, generating biased or toxic text, or simply not following user instructions"

(Bender et al., 2021; Bommasani et al., 2021; Kenton et al., 2021; Weidinger et al., 2021; Tamkin et al., 2021; Gehman et al., 2020)

TRAIN AND TEST

Previously on…

Recap

Instruct GPT

"Training language models to follow instructions with human feedback"
"This technique uses human preferences as a reward signal to fine-tune our models."

(Ouyang et al., 2022)

TRAIN AND TEST

Previously on…

Recap

결국 입력값이 결과값에 영향을 미친다면,

하나의 Task를 수행할 수 있는 모델을 만들기 위해
학습 데이터셋을 수집하고 Fine-tuning 하는 방식(기존)과는 달리,

모델에게 질문하는 방식을 바꿔보면 어떨까?

TRAIN AND TEST

Previously on…

Recap

Prompt Engineering

모델에게 질문하는 방법, 즉 프롬프트를 구성하는 방식이 결과물의 퀄리티를 좌우한다!

TRAIN AND TEST

Previously on…

Recap

GPT-3가 방대한 언어 Task를 수행할 수 있는 만큼, 반대로 특정 영역에는 특화되어 있지 않다.

Prompt Engineering을 통해 모델에게 특정 Task를 잘 수행할 수 있도록 조건(Instructions)을 부여해보자!

TRAIN AND TEST

Abstract

Large pre-trained Language models

특징 Feature

Parameter들에 Factual Knowledge를 저장하고 Downstream Task 수행 시 Fine-tuning을 통해 SATA 달성
to store factual knowledge in their parameters, and achieve state-of-the-art results when fine-tuned on downstream NLP tasks

문제점 Problems

1. 특정 전문 지식이 집적된 Task 들에 대해서는 Task-specific 모델들에 비해 성능이 떨어짐
However, their ability to access and precisely manipulate knowledge is still limited,
and hence on knowledge-intensive tasks, their performance lags behind task-specific architectures

2. 판단에 대한 근거를 제공하고 빠르게 변하는 실제 세계의 지식들에 대해 Update하기가 힘듦
providing provenance for their decisions and updating their world knowledge remain open research problems

TRAIN AND TEST

Abstract

Large pre-trained Language models

Parameter들 외에 별도의 Memory를 만들어볼까? (non-parametric memory)

사용자의 질문(Prompt)를 보충해주면 어떨까? (Prompt Engineering)

Pretrained models with a differentiable access mechanism to explicit non-parametric memory have so far been only investigated for extractive downstream tasks.

TRAIN AND TEST

Abstract

R A G

R A G

Retrieval - Augmented Generation

검색 증강 생성

Pre-trained Parametric 메모리와 Non-Praramетric 메모리(DataBase)를 결합해보자!

“models which combine pre-trained parametric and non-parametric memory for language generation”

TRAIN AND TEST

Abstract

Large pre-trained Language models

Pre-trained Parametric 메모리
Pre-trained seq2seq model

Non-Parametric 메모리
the non-parametric memory is a dense vector index of Wikipedia,
accessed with a pre-trained neural retriever

TRAIN AND TEST

Abstract

Large pre-trained Language models

Pre-trained Parametric 메모리 = Language Model (언어 모델)
Pre-trained seq2seq model

Non-Parametric 메모리 = Vector Index via Retriever (검색기)
the non-parametric memory is a dense vector index of Wikipedia, accessed with a pre-trained neural retriever

TRAIN AND TEST

Abstract

Comparison

1. 생성된 하나의 Sequence가 하나의 검색된 문서에서 기반하는 RAG

one which conditions on the same retrieved passages across the whole generated sequence

2. Sequence의 각 토큰이 서로 다른 문서에서 기반하는 RAG

another which can use different passages per token

Introduction

Pre-trained Neural Language Model

Pre-train된 Model들은 Parameter안에 Knowledge를 저장하는데, 이를 통해
해당 Knowledge는 외부 지식에 접근하지 않고도, Model 스스로 다양한 통찰력을 가지게 함

문제점

1. Memory를 확장하거나 업데이트가 쉽지 않음
They cannot easily expand or revise their memory,

2. Output 생성시 Knowledge(Insight)를 직접적으로 활용할 수 없음
can't straightforwardly provide insight into their predictions,

3. **Hallucination 발생** (Model이 사실과 다른 응답을 생성)
and may produce "hallucinations"

TRAIN AND TEST

Introduction

Hybrid Models

Hybrid Model = **Pre-trained model의 Parametric Memory + External DB의 Non-parametric Memory**
* Non-parametric Memory : 모델의 매개변수로 표현되지 않는 외부 메모리

1. Knowledge들이 직접적으로 Revise(업데이트) 되고 Expand(확장) 가능
because knowledge can be directly revised and expanded,

2. Accessed Knowledge가 직접적으로 검증되고 이해될 수 있음
and accessed knowledge can be inspected and interpreted.

EX) REALM, ORQA
Masked Language Model들(BERT)과 다양한 검색기(Retriever)들을 결합한 신모델들이 굉장히 높은 성능을 보임

Q. 하지만 이들은 Open-domain Extractive Q&A Task 관련해서만 주목해왔다. 그 이유가 무엇일까? (Hint : BERT 기반)

TRAIN AND TEST

Introduction

Hybrid Models

Hybrid Model = **Pre-trained model의 Parametric Memory** + **External DB의 Non-parametric Memory**
Sequence-to-Sequence 구조 채택, General-purpose fine-tuning approach 를 통해 기존 **REALM, ORQA** 의 한계 극복

Pre-trained model의 Parametric Memory
- **Pre-trained seq2seq transformer** (**"Workhorse of NLP"**)

External DB의 Non-parametric Memory
- **Dense Vector index of Wikipedia**
- **Accessed with a pre-trained neural retriever (Dense Passage retriever, DPR)**

DPR의 역할
- Input(Query)에 기반하여 Latent document 문서들 추출
- Output을 형성하기 위해 Input과 Latent document를 결합(Concatenate)

TRAIN AND TEST

Introduction

RAG's highlights

기존에도 성능을 증진시키기 위해 Non-parametric memory를 Architecture에 접합하려는 시도가 있었음
Ex) Memory Networks, Stack-Augmented Networks, Memory Layers

What's special about RAG?

1. Non-parametric/Parametric memory all Pre-trained, Pre-loaded!
2. Pre-trained Access mechanism(Retriever)의 도움으로 Additional training 없이 Knowledge로 접근 가능

= Specialization in "Knowledge Intensive Tasks"!

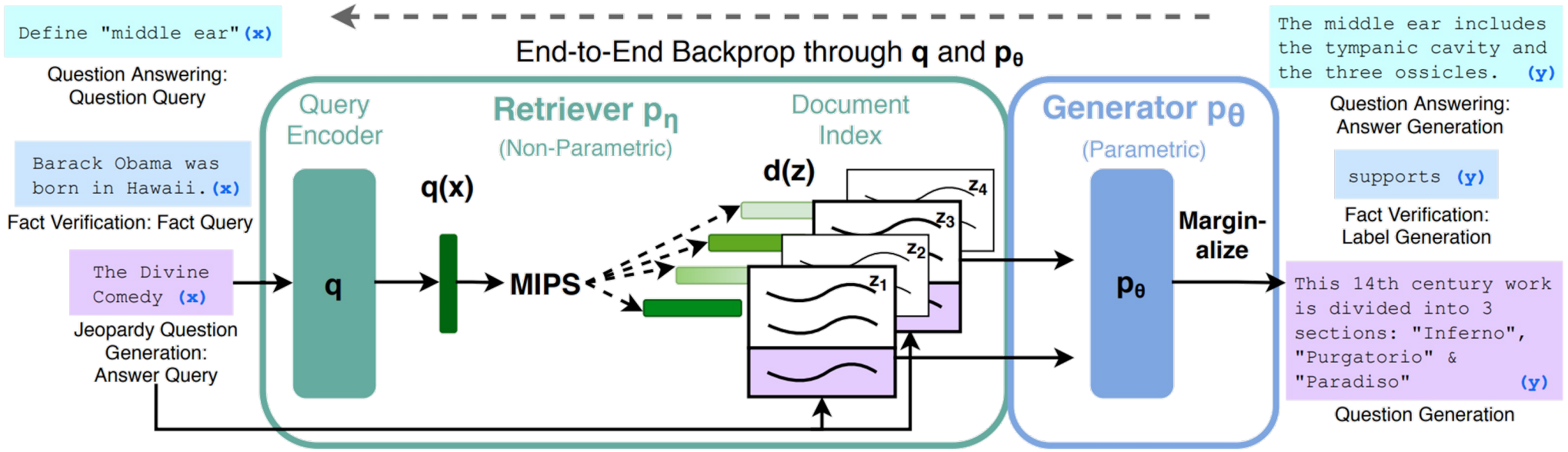
인간조차도 추가적인 자료가 없으면 대답하기 힘든 분야에 대해서도 작동
*"tasks that humans could not reasonably be expected to perform without access to **an external knowledge source**."*

기술이 진보하면 이에 맞추어 바로 대체하여 대응 가능
*"Finally, we demonstrate that the non-parametric memory can be **replaced to update** the models' knowledge as the world changes."*

TRAIN AND TEST

Methods

Overall

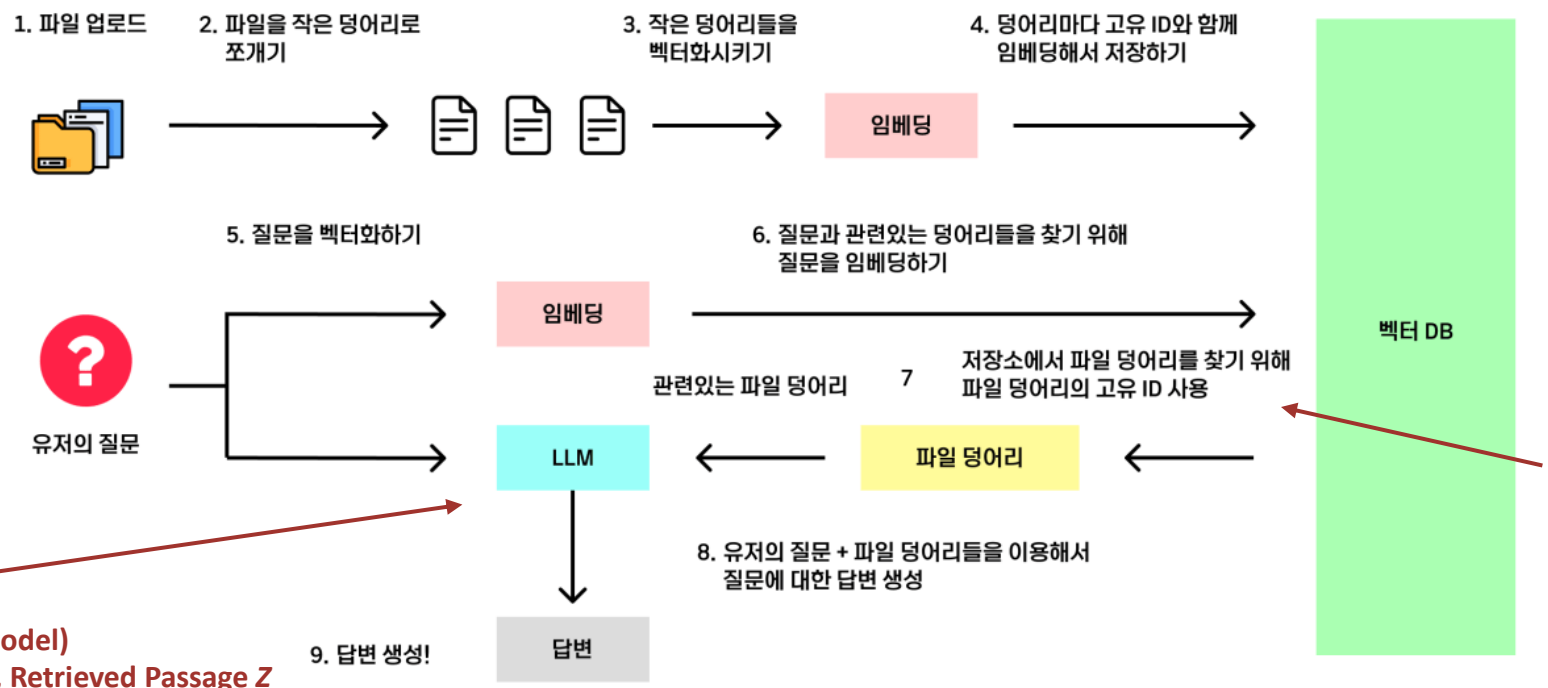


Input Sequence *x* → Retrieve Document *z* → Use as Additional Context → Target Sequence *y*

TRAIN AND TEST

Methods

Overall



Generator
(Pre-Trained Seq2Seq Model)
Previous Token, Input *X*, Retrieved Passage *Z*
기반하여 현재 토큰 생성

Retriever
(Pre-Trained Retriever)

주어진 Query *X* 기반,
텍스트 문서에서 뽑아낸
Top-K truncated distributions

출처 : 비개발자도 알기 쉬운 RAG(검색 증강 생성)에 대한 설명
글쓴이 : Seoyoung Park / Feb 15, 2024

TRAIN AND TEST

Methods

2.1 Models

서로 다른 두 개의 모델로 나눈 이유는
생성 과정에서 참조하는 문서의 다양성과 유연성 때문

1. RAG Sequence Model : 한 문서를 사용하여 전체 시퀀스(모든 토큰)를 생성

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

- x : 입력 시퀀스
- y : 생성된 출력 시퀀스
- z : 검색된 문서
- $p\eta(z|x)$: 주어진 입력 x에 대해 문서 z가 선택될 확률 **(retriever 확률)**
- $p\theta(y|x,z)$: 문서 z와 입력 x를 기반으로 시퀀스 y가 생성될 확률 **(generator 확률)**

단일 문서로 충분히 답변할 수 있는 질문에 적합

2. RAG Token Model : 각 토큰 생성시 서로 다른 문서를 참조하여 정보 집계

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

- N : 출력 시퀀스의 길이
- yi : 시퀀스의 i번째 토큰
- $y1{:}i{-}1$: 시퀀스의 첫 번째부터 i–1번째 토큰까지의 부분 시퀀스
- $p\eta(z|x)$: 주어진 입력 x에 대해 문서 z가 선택될 확률
- $p\theta(yi|x,z,y1{:}i{-}1)$: 문서 z와 입력 x, 그리고 앞선 토큰들 $y1:i$–1을 기반으로 yi 토큰이 생성될 확률

여러 문서에서 정보를 종합해야 하는 복잡한 질문에 적합

TRAIN AND TEST

Methods

2.2 Retriever

DPR Dense Passage Retrieval for Open-Domain QnA

문서 검색 시스템 : 주어진 쿼리에 대해 관련 문서들을 효율적으로 검색하기 위해 사용

1. 쿼리(Query)와 문서(Document)를 벡터 공간에 매핑
2. 이 벡터들 간의 유사도를 계산
3. 가장 관련성이 높은 문서를 검색 (MIPS Algorithm)

Bi-Encoder Architecture

$$p_\eta(z|x) \propto \exp\left(\mathbf{d}(z)^\top \mathbf{q}(x)\right) \qquad \mathbf{d}(z) = \mathrm{BERT}_d(z), \quad \mathbf{q}(x) = \mathrm{BERT}_q(x)$$

쿼리와 문서 벡터 간의 내적을 계산, 유사도를 평가 **벡터 표현 $d(z)$, $q(x)$ 로 변환**

***d*(*z*)** = Dense representation of a document produced by a BERT-base document encoder
***q*(*x*)** = Query representation produced by a query encoder, also based on BERT-base encoder

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

Maximum Inner Product Search (MIPS) 알고리즘
상위 K개의 문서 선택 시, Sub-Linear time 안에 쿼리에 관련성이 높은 문서 탐색

TRAIN AND TEST

Methods

2.3 Generator

BART Denoising Sequence-to-Sequence Pre-training

최종 응답 생성기 : 입력 쿼리와 검색된 문서들을 결합하여 최종 응답을 생성 (400M parameter의 BART-large 사용)

1. 입력 쿼리 x와 검색된 문서들 z를 결합 (“Simply Concatenate them”)
2. BART Encoder – 결합된 입력과 문서를 Embedding Vector로 변환
3. Decoder는 Vector를 기반으로 시퀀스 생성
4. RAG-Token과 RAG-Sequence 모델은 각각 토큰별, 시퀀스별로 문서를 참조, 응답 생성

Encoder

$$\text{EncoderOutput} = \text{BART}_{\text{Encoder}}([x; z])$$

Decoder

$$p_\theta(y_i|x, z, y_{1:i-1}) = \text{BART}_{\text{Decoder}}(y_{1:i-1}, \text{EncoderOutput})$$

Generation by Tokens

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

Generation by Sequence

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

TRAIN AND TEST

Methods

2.4 Training

Adam Adaptive Moment Estimation

Adam Optimizer를 사용하여 Retriever/Generator를 효과적으로 훈련

- **확률적 경사 하강법(SGD)**을 기반으로 두 가지 모멘텀(일/이차)의 추정치 활용, 학습 속도와 안정성을 높임
- 학습 과정에서 각 매개변수의 학습률을 **개별적으로 조정**

$$\mathcal{L} = -\sum_{j} \log p(y_j|x_j)$$

*yj*는 정답 시퀀스의 j번째 토큰
*xj*는 입력 쿼리, *z*는 검색된 문서

1. 생성된 출력 시퀀스 y^와 정답 시퀀스 y를 비교하여 손실을 계산
2. **음의 로그 우도(NLL, Negative Log Likelihood) 최소화**
3. Adam Optimizer를 사용하여 **손실을 최소화**하도록 모델의 매개변수 업데이트

학습 중 Document Encoder마저 업데이트하면 Document Indexing을 정기적으로 업데이트해야 되므로 비용이 많이 소모됨
= Query Encoder와 Generator만 Fine-tuning하고 Document Encoder는 고정 상태로 유지

TRAIN AND TEST

Methods

2.5 Decoding

1. RAG-Token 디코딩 (Standard)

- 각 토큰별로 다른 문서를 참조하여 확률을 계산 후, 빔 서치(Beam Search)를 통해 최종 출력 생성

1. 각 토큰 yi에 대해, 상위 K개의 문서 z를 참조하여 확률을 계산 $p'_\theta(y_i|x, y_{1:i-1}) = \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z_i|x) p_\theta(y_i|x, z_i, y_{1:i-1})$
2. 확률에 기반하여 빔 서치(beam search)로 최종 출력 생성 *후보 시퀀스를 유지하며 가장 높은 점수의 시퀀스를 선택하는 탐색 방법

2. RAG-Sequence 디코딩

- 하나의 문서로 전체 시퀀스를 생성, 각 문서별로 시퀀스를 생성한 후, 이를 종합하여 최종 출력 결정

1. 각 문서 z에 대해 빔 서치 수행, 후보 시퀀스 생성
2. 최종 시퀀스를 선택하기 위해 각 문서에서 생성된 시퀀스의 확률을 합산
3. 빔 서치를 통해 생성된 후보 시퀀스 종합, 최종 출력 결정

(각 후보 시퀀스 y에 대해 문서 z의 확률 $p\eta(z|x)$ 를 곱하여 최종 확률 계산)

TRAIN AND TEST

Experiments

3. Experiments

3.1 Open-domain Question Answering

- 비교 대상 : Extractive QA Paradigms(Non-parametric), ‘Closed-book QA’ (only Parametric)

3.2 Abstractive Question Answering

- 간단한 추출 QA를 넘어 자유롭고 추상적인 텍스트 생성을 통해 질문에 답할 수 있는가?

3.3 Jeopardy Question Generation

- Open-domain 질문 생성 능력 평가 *Jeopardy 질문 생성 : 대상에 대한 사실로부터 역으로 질문을 추측하는 작업

- 3.4 Fact Verification

- 생성문이 Wikipedia에 제대로 기반하는지, 반박하는지, 혹은 정보가 불충분한지 분류

TRAIN AND TEST

Results

4. Results

3.1 Open-domain Question Answering

- 모든 오픈 도메인 QA 작업에서 RAG는 SATA 달성,
- "closed-book" 접근 방식과 "open-book" 검색 기반 접근 방식의 장점을 결합

3.2 Abstractive Question Answering

- BART보다 사실적으로 더 정확한 텍스트 생성

3.3 Jeopardy Question Generation

- RAG-Token > RAG-Sequence
- 두 모델 모두 Q-BLEU-1에서 BART를 능가, 인간 평가(Human Assessment)에서도 성능 인정

3.4 Fact Verification

- 최첨단 모델들 (복잡한 파이프라인) 의 상위 4.3% 이내의 성능

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**

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"

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 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%

TRAIN AND TEST

Discussion

5. Discussion

주요 성과 및 기여

Parametric과 Non-Parametric Memory에 접근할 수 있는 하이브리드 생성 모델 RAG 제안

- 순수 Parametric-based BART보다 사실적이고 구체적
- Additional Training 없이도 지식 인덱스 업데이트 가능

연구 방향

- 공동 사전 훈련 : Parametric과 Non-Parametric Memory Component들을 처음부터 함께 훈련해보자
- 메모리 상호 작용 : Parametric과 Non-Parametric Memory 결합에 대한 새로운 Insight

사회적 영향

긍정적인 영향 : 챗봇에 대한 적용 가능성

부정적인 영향 : Non-Parametric Memory에 편향된 정보가 있으면 위험

TRAIN AND TEST

TRAIN AND TEST