Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

RAG

2024.11.12

SAI NLP

김수효

RAG?

원인

기존의 사전학습된 언어모델 : parameter안에 지식을 저장

1. 지식에 대한 업데이트나 확장이 어려움
2. output을 생성할 때 관련된 지식을 직접적으로 활용하지 못하여 환각 발생

결과

검색 증강 생성(Retrieval-Augmented Generation, RAG)

Parametric memory와 non-parametric memory를 결합한 하이브리드 모델

RAG 구조

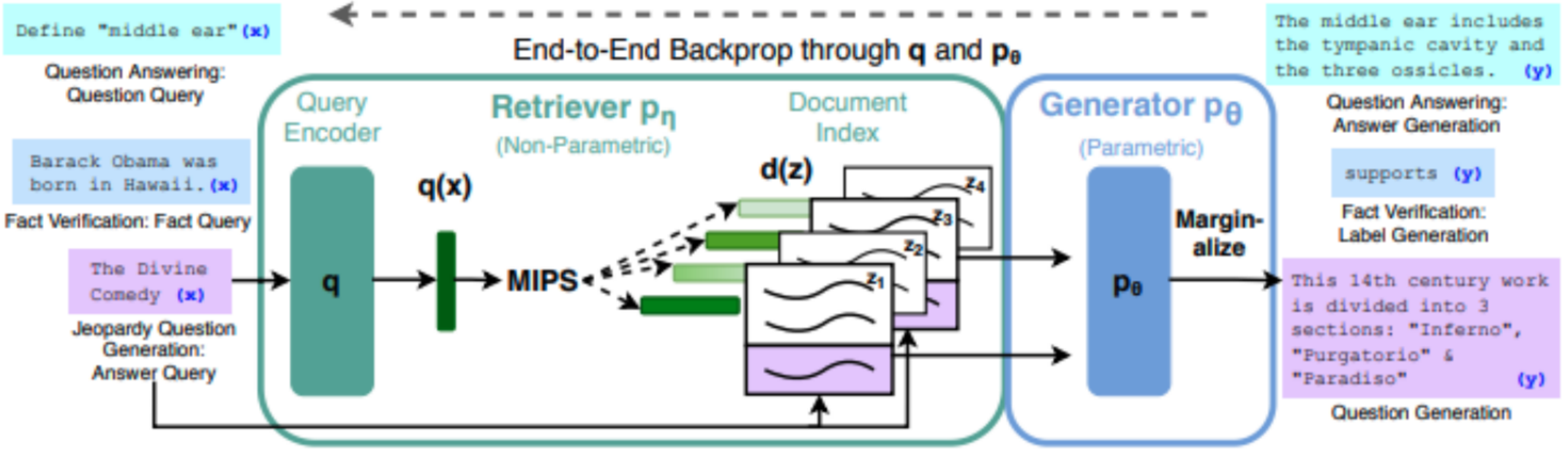

Figure 1: Overview of our approach. We combine a pre-trained retriever (*Query Encoder + Document Index*) with a pre-trained seq2seq model (*Generator*) and fine-tune end-to-end. For query x, we use Maximum Inner Product Search (MIPS) to find the top-K documents z_i. For final prediction y, we treat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

1. Retriever : question x를 기반으로 유사한 document z를 retrieve하는 모델
2. Seq2Seq Generator : Retriever가 반환한 유사한 document z와 question x를 encoder에 넣고 decoder에서 answer y를 generate 하는 모델

모델 종류

RAG-Sequence Mode

하나의 검색된 문서를 사용하여 전체 출력 시퀀스를 생성

1. 검색기(retriever)를 통해 상위 K개의 문서를 검색
2. 각 문서에 대해 생성기(generator)가 출력 시퀀스의 확률을 계산
3. 이 확률들을 marginalization(주변화)하여 최종 확률 계산

RAG-Token Model

각각의 토큰(단어)마다 다른 문서를 참조할 수 있는 모델

1. 각 출력 토큰마다 상위 K개의 문서를 새로 검색
2. 각 문서에 대해 다음 토큰의 확률 분포를 계산
3. 이를 marginalization하여 최종 토큰 확률 계산
4. 이 과정을 다음 토큰에 대해 반복

실험 1

Open-Domain QA

Q: "에펠탑의 높이는 얼마인가요?"
A: "324미터입니다."

	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**

실험 2

Jeopardy Question Generation

A: "주기율표에서 가장 가벼운 금속"
Q: "리튬은 무엇인가요?"

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%