13주차 논문 리뷰: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

0. Abstract

- 대규모 언어 모델은 매개변수에 사실적 지식을 저장하고 NLP 과제에서 최첨단 성능을 보여주지만, 지식 집약적 과제에서는 한계를 드러냄

  → **검색-강화 생성(RAG)** 모델 제안

  - **매개변수 메모리**: 사전 학습된 seq2seq 모델(BART)
  - **비매개변수 메모리**: 위키백과를 벡터화한 외부 인덱스

  → 지식을 검색하고 이를 활용해 **정확하고 다양한 언어 생성**을 가능하게 함

  → 실험 결과, RAG는 기존 **parametric seq2seq 모델** 및 **검색-추출 모델** 대비 더 나은 성능을 보임
  → 질문 응답(QA) 및 언어 생성 과제에서 정확도와 표현력이 향상됨

1. Introduction

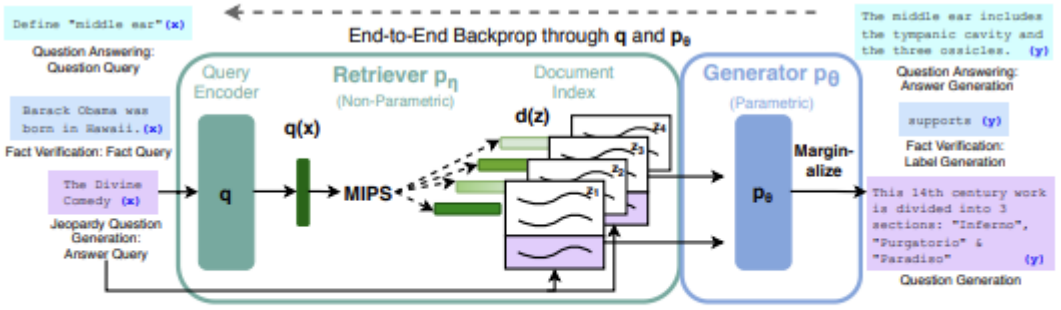

Figure 1: Overview of our approach. We combine a pre-trained retriever (*Query Encoder + Document Index*) with a pre-trained seq2seq model (*Generator*) and fine-tune end-to-end. For query x, we use Maximum Inner Product Search (MIPS) to find the top-K documents z_i. For final prediction y, we treat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

- 사전 학습된 언어 모델은 외부 메모리 없이도 방대한 지식을 활용할 수 있지만, **지식을 수정하거나 확장하는 데 한계**가 있으며, 예측에 근거를 제공하지 못하고, 잘못된 출력을 생성할 가능성이 있음
- **RAG 모델**은 이를 해결하기 위해 매개변수 메모리와 비매개변수 메모리를 결합한 하이브리드 접근법을 사용

1. **검색기(Retriever)**: 질문에 가장 관련성 높은 문서를 데이터베이스(위키백과)에서 검색
2. **생성기(Generator)**: 검색된 문서를 바탕으로 최적의 응답을 생성

→ **end-to-end 학습**으로 검색과 생성 과정을 동시에 최적화하며, 다음과 같은 장점을 가짐

- 지식의 확장성과 업데이트 가능.
- 더 나은 결과의 언어 생성.
- 기존 **검색-추출(retrieve-and-extract)** 모델 대비 성능 향상.

→ 실험에서는 RAG가 지식 집약적 과제(질문 응답, 언어 생성)에서 높은 성과를 기록

2. Methods

- RAG 모델은 입력 텍스트 x를 기반으로 관련 문서 z를 검색하고, 이를 추가적인 컨텍스트로 활용해 출력 시퀀스 y를 생성
- RAG의 주요 컴포넌트
  - **Retriever**: 입력 x를 기반으로 가장 관련성 높은 상위 K개의 문서를 검색
  - **Generator**: 검색된 문서 z와 이전 토큰 y1:i−1을 바탕으로 현재 토큰을 생성

- RAG 모델 설계 방식
  - **RAG-Sequence**
    - 동일한 문서 z를 활용해 전체 출력 시퀀스를 생성
    - p(y ∣ x)를 계산하기 위해 상위K개의 문서 z를 검색하고 이를 기반으로 생성기를 실행하여 확률을 결합

$$P_{RAG-Sequence}(y|x) \approx \sum_{z \in \text{Top-K}(p_r(z|x))} p_r(z|x) p_\theta(y|x,z)$$

- **RAG-Token**
  - 각 타겟 토큰을 생성할 때마다 다른 문서 z를 활용
  - 각 토큰마다 문서 z를 새로 선택하므로, 더 세밀한 조정이 가능

$$P_{RAG-Token}(y|x) \approx \prod_i \sum_{z \in \text{Top-K}(p_r(z|x))} p_r(z|x) p_\theta(y_i|x, z, y_{1:i-1})$$

- **Retriever: DPR (Dense Passage Retriever)**
  - bi-encoder 구조를 따름
  - 문서 표현 d(z)와 쿼리 표현 q(x)를 별도로 계산하고, 둘 간의 내적(dot product)으로 문서-쿼리의 유사성을 측정
  - d(z)=BERTdoc(z) & q(x)=BERTquery(x)
  - 상위 K개의 문서를 검색하는 과정은 MIPS (Maximum Inner Product Search)를 사용
  - DPR은 훈련 시, TriviaQA 및 Natural Questions와 같은 데이터셋으로 학습되어 비매개변수 메모리(non-parametric memory)로 작동

- **Generator: BART**
  - **Generator**는 사전 학습된 **BART 모델(seq2seq 구조)**을 사용
  - BART는 400M 매개변수를 가지며, 다양한 노이즈 추가 기법과 디노이징 목표로 사전 학습
  - 검색된 문서 z와 입력 x를 결합해 최종 출력 y를 생성
  - BART 기반 생성기는 기존 T5 모델을 포함한 유사한 규모의 다른 모델들보다 뛰어난 성능을 기록

- **Training**
  - RAG 모델은 **retriever와 generator를 end-to-end로 학습**
  - 문서 검색 과정은 명시적 지도가 없으며, 입력/출력 쌍 (x,y)를 기반으로 모델이 스스로 최적화

    - 손실 함수는 출력 시퀀스의 음의 로그 가능도 (negative log-likelihood)를 최소화하도록 설계됨.

- **Decoding**
    - **RAG-Sequence**와 **RAG-Token**은 디코딩 시 서로 다른 방식을 사용

    1. **RAG-Token**:
        - 각 토큰마다 상위 K 문서를 활용하여 확률을 계산한 뒤, **beam search로 디코딩**.
        - 더 효율적인 디코딩이 가능
    2. **RAG-Sequence**:
        - 전체 시퀀스의 확률 p(y | x)를 계산하기 위해 각 문서 z에 대해 추가적인 디코딩 단계를 수행.
        - 디코딩 시간이 길어질 수 있으나, 높은 정확도를 보장
        - 이를 개선한 방식으로 **Fast Decoding**을 도입

3. Experiments

- RAG 모델은 다양한 **지식 집약적 과제**에서 실험
- **Wikipedia (2018년 덤프)** 데이터를 비매개변수 지식 소스로 사용

→ 위키백과의 각 문서를 100단어 단위로 나눠 약 **2,100만 개의 문서**로 구성된 데이터셋을 생성

→ FAISS(Fast Approximate Nearest Neighbors)를 사용해 문서의 효율적인 검색을 위한 MIPS(Maximum Inner Product Search) index 구축

3.1 Open-Domain Question Answering (ODQA)

- ODQA는 **질문에 대해 외부 데이터베이스를 기반으로 답변을 생성**하는 지식 집약적 과제
- RAG는 입력-출력 쌍 (x,y)을 통해 학습되며, 음의 로그 가능도 (negative log-likelihood)**를 최소화하는 방식으로 최적화됨
- 비교 모델

1. **Closed-Book QA**: 외부 검색 없이 매개변수 지식만 활용하여 답변 생성
2. **비교 데이터셋**:
   - Natural Questions (NQ)
   - TriviaQA (TQA)
   - WebQuestions (WQ)
   - CuratedTrec (CT)

- 결과

: RAG는 비매개변수 메모리를 활용한 검색-생성 방식으로 기존 ODQA 모델보다 더 높은 성능을 보여줌

3.2 Abstractive Question Answering

- 추출적(extractive) QA가 아닌, 자연어 생성(NLG)을 기반으로 한 **추상적 QA**
- **MSMARCO NLG** 데이터셋을 사용해 질문에 대해 더 자연스럽고 명확한 답변을 생성하는 과제를 수행
- 주의점: MSMARCO 질문 중 일부는 위키백과만으로 답변이 불가능하므로 RAG는 매개변수 지식을 기반으로 응답

3.3 Jeopardy Question Generation

- Jeopardy 형식의 질문 생성: 정답(entity)을 기반으로, 해당 정답에 대한 **사실적 질문**을 생성
- 예시: 정답이 "월드컵"이라면, 생성된 질문은 "1986년 멕시코가 국제 스포츠 대회를 개최한 나라로 기록된 것은?"과 같은 형태
- 데이터셋: **SearchQA** 데이터셋을 사용 (100K 훈련 데이터, 14K 검증 데이터, 27K 테스트 데이터)
- 평가 방법:

1. **SQuAD-tuned Q-BLEU-1**: BLEU의 변형으로, **정답 엔티티와의 매칭 정도**를 평가.
2. **사람 평가**
   - 질문의 사실성(factuality)과 특정성(specificity)을 평가.

- RAG와 BART 모델이 생성한 질문을 비교하여 어느 쪽이 더 나은지 선택.

3.4 Fact Verification

- 자연어 문장이 **위키백과의 내용과 일치하는지 여부**를 확인
- FEVER(Fact Extraction and VERification) 데이터셋을 사용하여, 문장을 참(true), 거짓(false), 확인 불가(NEI)로 분류
- **주요 과제**
  - 정답 및 관련 문서를 검색하고, 이 문서에서 주장된 사실이 참인지 평가
  - RAG는 추론(reasoning)에 적합하며, FEVER 태스크에서도 강력한 성능을 보임

1. **AISS (Facebook AI Similarity Search)**
   - 대규모 벡터 데이터셋에서 빠르게 유사한 항목을 검색하는 라이브러리
   - RAG에서는 FAISS를 사용해 위키백과 색인을 구축하고, MIPS를 통해 관련 문서를 검색
2. **MIPS (Maximum Inner Product Search)**
   - 쿼리 벡터와 데이터셋 벡터 간의 내적 값이 최대인 항목을 검색하는 알고리즘
   - 문서 검색의 핵심 기술로, RAG가 효율적으로 관련 문서를 선택할 수 있도록 지원
3. **FEVER 태스크**
   - 사실 검증(Fact Verification)을 위한 표준 데이터셋 및 과제
   - 자연어 문장이 위키백과의 내용과 얼마나 일치하는지 평가하는 데 사용
4. **Q-BLEU**
   - BLEU 점수의 변형으로, 단순 문장 유사성 대신 정답 엔티티와의 매칭 정확도를 강조

4. Results

4.1 Open-Domain Question Answering (ODQA)

- RAG는 Open-Domain QA(ODQA)에서 기존의 최첨단 모델보다 우수한 성능을 기록
- 특히, RAG는 **"Closed-Book"** 접근법(매개변수 지식만 활용)과 **"Open-Book"** 접근법(검색 기반) 모두의 장점을 결합
- **RAG의 주요 장점**:
  1. 문서에 정확한 답이 없어도 단서를 통해 올바른 답을 유추 가능
  2. 추출적(extractive) 모델로는 불가능한 경우에도 정확한 답을 생성 가능
  3. Natural Questions(NQ), TriviaQA(TQA), WebQuestions(WQ), CuratedTrec(CT) 데이터셋에서 더 높은 성과를 보임

4.2 Abstractive Question Answering

- RAG-Sequence는 Open MS-MARCO NLG에서 BART 모델을 BLEU 점수 기준으로 2.6점, Rouge-L 점수 기준으로 2.6점 초과.
- **RAG의 강점**:
  1. Gold passage(정확한 문서)가 없어도 높은 성능을 발휘.
  2. 일부 질문이 Wikipedia 외부의 정보를 요구하더라도 합리적인 답변 생성.
  3. RAG는 BART보다 더 사실적이고 적합한 응답을 생성하며, **"환각(hallucinations)" 문제를 줄임.**

4.3 Jeopardy Question Generation

- **RAG-Token**이 Jeopardy 질문 생성에서 RAG-Sequence보다 더 나은 성과를 보임.
- **평가 결과**:
  - 인간 평가에서 RAG는 **factuality과 specificity**에서 BART를 초과
  - RAG는 42.7%의 경우에서 더 사실적인 응답을 생성 (BART: 7.1%).
  - Jeopardy 질문 생성에서 RAG의 응답이 BART보다 구체적이고 사실적
- **예시**:
  - 정답 "The Sun Also Rises"에 대해, RAG는 관련된 사실적 질문을 생성함으로써 높은 정확도를 보여줌.

4.4 Fact Verification

- FEVER 데이터셋에서 RAG는 최첨단 모델 대비 4.3% 내외의 성능 차이.
- **RAG의 특징**:
  - 문서 검색 없이도 응답 생성이 가능.
  - 71%의 경우에서 상위 10개의 검색된 문서 안에 Gold article(정답 문서)을 포함

4.5 Additional Results

- **Generation Diversity**:
  - RAG-Sequence와 RAG-Token 모두 BART보다 더 다양한 언어 생성을 보여줌
- **Retrieval Ablations**:
  - RAG의 성능은 검색 기법에 의존하며, 비매개변수 메모리를 기반으로 더욱 효과적인 검색을 가능케 함
- **Index Hot-Swapping**:
  - 비매개변수 메모리(index)를 교체해 모델의 세계 지식을 효율적으로 업데이트할 수 있음

💡

1. **Closed-Book QA**
    - 검색 없이 매개변수로 저장된 지식만 활용해 질문에 답변
    - RAG는 이 방식을 결합해 더 높은 성능을 달성

2. **Fact Verification**
    - 문장이 사실인지, 거짓인지, 확인 불가능한지 분류하는 작업
    - RAG는 검색된 문서의 내용을 근거로 답변을 생성

3. **Jeopardy Question Generation**
    - 주어진 정답(예: "World Cup")에 기반한 사실적 질문 생성
    - 일반적인 질문 생성보다 도전적인 과제

4. **Index Hot-Swapping**
    - 새로운 데이터를 기반으로 RAG의 비매개변수 메모리를 교체해 지식을 업데이트하는 기법
    - 예: 2016년 Wikipedia 덤프를 2018년 데이터로 교체

5. Related Work

5.1 Single-Task Retrieval

- 기존 연구에서는 검색(retrieval)이 특정 NLP 작업(예: ODQA, Fact Verification, 언어 생성 등)에서 성능을 향상시킨다고 보고
- RAG는 다양한 태스크에서 검색을 통합해 **단일 검색 기반 아키텍처**로도 강력한 성능을 발휘함을 보여줌

5.2 General-Purpose Architectures for NLP

- 사전 학습된 언어 모델은 검색 없이도 강력한 성능을 달성할 수 있음

- 대표적으로
  - GPT-2: 왼쪽에서 오른쪽으로 진행되는 단일 구조로 분류 및 생성 작업에서 성과
  - BART, T5: 양방향 주의를 사용하여 더 강력한 성능을 발휘
- RAG는 이러한 일반 구조를 확장하여 **검색 기반 모듈**을 추가, 성능을 더욱 강화

5.3 Learned Retrieval

- 학습된 검색(learned retrieval)은 문서 검색을 통해 성능을 최적화
- 기존 연구:
  - 강화 학습, 잠재 변수 접근법 등이 특정 태스크 최적화에 사용됨
  - RAG는 단일 검색 기반 아키텍처를 사용해 다수의 태스크에서 성능을 달성

5.4 Memory-Based Architectures

- 문서 색인은 모델의 **외부 메모리 역할**을 함
- 주요 특징:
  - 인간이 읽고 수정할 수 있는 메모리 형태
  - 검색된 정보를 통해 모델의 지식을 동적으로 업데이트 가능

5.5 Retrieve-and-Edit Approaches

- **기존의 Retrieve-and-Edit 방식**과 유사점
  - 특정 입력-출력 쌍을 검색하고 이를 편집하여 최종 출력 생성
- 차이점:
  - RAG는 여러 문서를 종합적으로 활용하며, 학습된 검색 및 다양한 증거를 활용해 더 정교한 출력을 생성

6. Discussion

- **RAG의 기여**:
  - 매개변수 및 비매개변수 메모리를 결합한 하이브리드 생성 모델을 제시
  - Open-Domain QA에서 최첨단 성능 기록
  - BART 대비 사실적이고 구체적인 응답 생성

- **주요 실험**:
  - 검색된 문서를 변경(Index Hot-Swapping)을 통해 추가 훈련 없이 지식 업데이트 가능
- **향후 연구 방향**:
  - 두 메모리 구성 요소를 처음부터 공동으로 사전 학습하는 가능성
  - 매개변수 및 비매개변수 메모리 간 상호작용 연구