5. Related Work

기존에 있었던 연구들 vs RAG → 차이점

하고싶은 말 : RAG가 더 좋다!

Single-Task Retrieval Prior work has shown that retrieval improves performance across a variety of NLP tasks when considered in isolation. Such tasks include open-domain question answering [5, 29], fact checking [56], fact completion [48], long-form question answering [12], Wikipedia article generation [36], dialogue [41, 65, 9, 13], translation [17], and language modeling [19, 27]. Our work unifies previous successes in incorporating retrieval into individual tasks, showing that a single retrieval-based architecture is capable of achieving strong performance across several tasks.

1. Single-Task Retrieval 단일작업 검색

이전의 연구 :

NLP task(자연어 처리 문제) 각각 고려 = task마다 검색 시스템 만듦

NLP task 예시 : open-domain question answering, fact checking, fact completion, long-form question answering,Wikipedia article generation,dialogue, translation,language modeling

RAG :

단일 검색 기반 아키텍처가 다양한 task들 수행함

General-Purpose Architectures for NLP Prior work on general-purpose architectures for NLP tasks has shown great success without the use of retrieval. A single, pre-trained language model has been shown to achieve strong performance on various classification tasks in the GLUE benchmarks [60, 61] after fine-tuning [49, 8]. GPT-2 [50] later showed that a single, left-to-right, pre-trained language model could achieve strong performance across both discriminative and generative tasks. For further improvement, BART [32] and T5 [51, 52] propose a single, pre-trained encoder-decoder model that leverages bi-directional attention to achieve stronger performance on discriminative and generative tasks. Our work aims to expand the space of possible tasks with a single, unified architecture, by learning a retrieval module to augment pre-trained, generative language models.

2. General-Purpose Architectures for NLP 자연어 처리를 위한 범용 아키텍처

이전의 연구 :

검색 X = 사전 학습된 지식으로 과제 수행

RAG :

하나의 통합된 아키텍처가 더 다양한 과제 수행하기 위해서

⇒ 사전 학습된 생성형 언어 모델이 검색 모듈 학습

Learned Retrieval There is significant work on learning to retrieve documents in information retrieval, more recently with pre-trained, neural language models [44, 26] similar to ours. Some work optimizes the retrieval module to aid in a specific, downstream task such as question answering, using search [46], reinforcement learning [6, 63, 62], or a latent variable approach [31, 20] as in our work. These successes leverage different retrieval-based architectures and optimization techniques to achieve strong performance on a single task, while we show that a single retrieval-based architecture can be fine-tuned for strong performance on a variety of tasks.

3. Learned Retrieval 학습된 검색

이전의 연구 :

각각의 task를 수행하기 위해서 서로 다른 검색 모듈과 최적화 기술 사용

RAG :

하나의 검색 기반 아키텍처로 다양한 task들을 수행하도록 미세조정 될 수

있음

어떤 데이터로 미세조정되냐에 따라 아키텍처의 성격 변함

Learned Retrieval There is significant work on learning to retrieve documents in information retrieval, more recently with pre-trained, neural language models [44, 26] similar to ours. Some work optimizes the retrieval module to aid in a specific, downstream task such as question answering, using search [46], reinforcement learning [6, 63, 62], or a latent variable approach [31, 20] as in our work. These successes leverage different retrieval-based architectures and optimization techniques to achieve strong performance on a single task, while we show that a single retrieval-based architecture can be fine-tuned for strong performance on a variety of tasks.

3. Learned Retrieval 학습된 검색

미세조정 : 데이터(질문 - 정답 쌍)로 파라미터를 조금 수정해서 그 문제에 대해 최적화하는 과정

예를들어, Jeopardy Question Generation으로 fine-tuning하면? ⇒ 제퍼티 퀴즈를 만듦

Fact Verification으로 fine-tuning하면? ⇒ 참/거짓으로 대답

⇒ 모델의 구조 변화 X , 검색기 변화 O (QA일 때 답 있는 문서 찾고, Fact Verification일 때 증거있는 문서 찾고 등등)

따라서 범용적 아키텍처다!

Memory-based Architectures Our document index can be seen as a large external memory for neural networks to attend to, analogous to memory networks [64, 55]. Concurrent work [14] learns to retrieve a trained embedding for each entity in the input, rather than to retrieve raw text as in our work. Other work improves the ability of dialog models to generate factual text by attending over fact embeddings [15, 13]. A key feature of our memory is that it is comprised of raw text rather distributed representations, which makes the memory both (i) human-readable, lending a form of interpretability to our model, and (ii) human-writable, enabling us to dynamically update the model's memory by editing the document index. This approach has also been used in knowledge-intensive dialog, where generators have been conditioned on retrieved text directly, albeit obtained via TF-IDF rather than end-to-end learnt retrieval [9].

4. Memory-based Architectures 메모리 기반 아키텍처

이전의 연구 :

fact-embedding 사용 (지식 그래프의 객체와 관계를 컴퓨터가 처리하기 쉬운

저차원 수치 벡터 공간으로 변환하는 기술)

RAG :

메모리가 raw text (원문)로 구성되어 있음 → 사람이 읽고, 편집 가능

장점 - 어떤 문서 선택했는지 사람이 읽을 수 있음, 사람이 데이터베이스 편집하면 모델의 지식 업데이트 됨(다시 학습할 필요 X 굉장히 효율적!)

Retrieve-and-Edit approaches Our method shares some similarities with retrieve-and-edit style approaches, where a similar training input-output pair is retrieved for a given input, and then edited to provide a final output. These approaches have proved successful in a number of domains including Machine Translation [18, 22] and Semantic Parsing [21]. Our approach does have several differences, including less of emphasis on lightly editing a retrieved item, but on aggregating content from several pieces of retrieved content, as well as learning latent retrieval, and retrieving evidence documents rather than related training pairs. This said, RAG techniques may work well in these settings, and could represent promising future work.

5. Retrieve-and-Edit approaches 검색 및 편집

이전의 연구 :

과거의 학습데이터(질문 - 정답 쌍)에서 약간만 바꿔서 답함

RAG :

여러 문서를 종합해서 답함

무엇을 검색해야 정답을 맞힐 수 있을지 그 방향성까지 스스로 학습

6. Discussion & Broader Impact

고찰 및 결론 & 영향

In this work, we presented hybrid generation models with access to parametric and non-parametric memory. We showed that our RAG models obtain state of the art results on open-domain QA. We found that people prefer RAG's generation over purely parametric BART, finding RAG more factual and specific. We conducted an thorough investigation of the learned retrieval component, validating its effectiveness, and we illustrated how the retrieval index can be hot-swapped to update the model without requiring any retraining. In future work, it may be fruitful to investigate if the two components can be jointly pre-trained from scratch, either with a denoising objective similar to BART or some another objective. Our work opens up new research directions on how parametric and non-parametric memories interact and how to most effectively combine them, showing promise in being applied to a wide variety of NLP tasks.

RAG 성과

open-domain QA에 대해 최신의 성능(state of the art SOTA) 가짐

사람들이 RAG가 purely parametric BART보다 사실적이고 구체적이라 평가함

모델의 재학습 없이 인덱스의 hot-swap(교체)으로 모델 업데이트 할 수 있음

Future Work

이 논문에서는 이미 만들어진 검색기(DPR)와 생성기(BART) 사용함

근데 모델을 처음 만들 때부터 검색기와 생성기를 함께 학습시킨다면 유익할까??

이 논문이 parametric(내부)과 non-parametric(외부) 메모리가 서로 어떻게 보완하는지 새로운 연구 방향성을 열어주었다~

- parametric(BART) 보완 : 언어능력은 뛰어남, 근데 팩트 틀림
- non-parametric(위키피디아) 보완 : 지식 많음, 근데 말 못하는 데이터 뭉치

Broader Impact

긍정적인 사회적 영향 :

Hallucination 감소 → 실제 지식에 근거하기 때문

다양한 상황에 사용 가능 : 예를 들어 의료 지식 인덱스 넣으면

잠재적 단점 :

외부 지식이 100% 옳지 않음, 편향되어 있을 수 있음

기술 악용 - 가짜 뉴스 생성, 사칭, 스팸, 피싱 자동 생성에 사용될 수 있음

일자리 줄어들 수 있음