Lost in the Middle: How Language Models Use Long Contexts

Nelson F. Liu[1], Kevin Lin[2], John Hewitt[1], Ashwin Paranjape[3,4], Michele Bevilacqua[3], Fabio Petroni[3], Percy Liang[1]

[1]Stanford University, [2]University of California, Berkeley, [3]Samaya AI, UK, [4]Samaya AI, USA

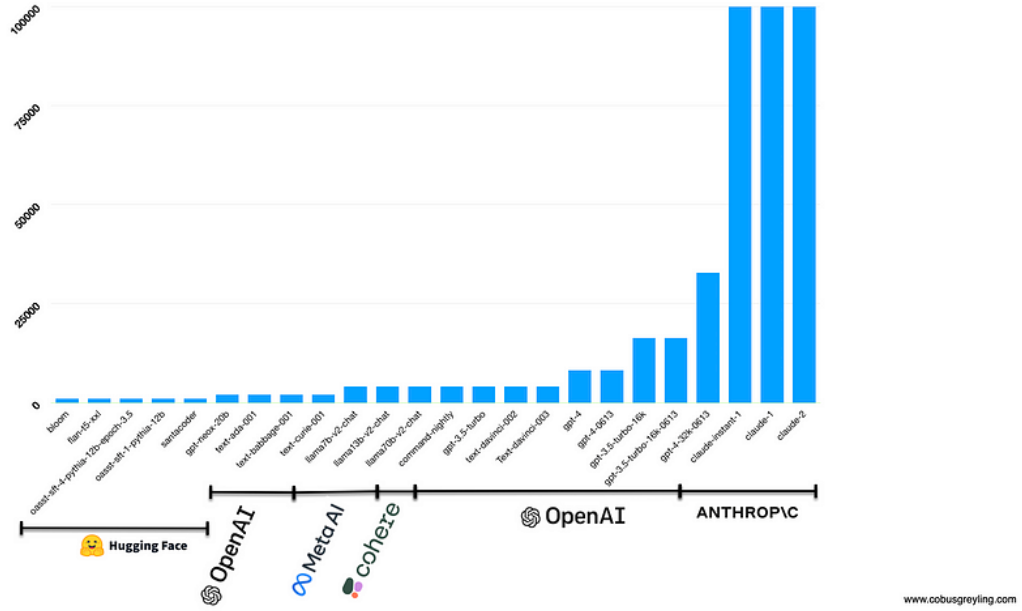
TACL 2024
2024.07.01

발제자: 윤예준

Background

- 최근 LM은 long contexts를 입력으로 받을 수 있지만, 이를 얼마나 잘 사용하는지에 대해서는 잘 알고 있지 않음

➔ Input text 내에 관련된 정보가 있는 2가지 task를 통해 LM 성능 분석
  - Multi-document question answering
  - Key-value retrieval



https://cobusgreyling.medium.com/rag-llm-context-size-6728a2f44beb

Experiment – Multi-Document Question Answering

- Goal: LM이 input context를 어떻게 사용하는지 더 잘 이해하는 것

- Setup
  - Dataset: NaturalQuestions-Open
  - Input
    - a question to answer
    - k documents (Wikipedia)
      - document: 최대 100 tokens로 이루어진 passage
      - k-1 "distractor": Contriever 이용하여 query와 가장 연관되나 정답은 포함되지 않은 document 선택
  - Evaluation metric: Accuracy
  - Decoding strategy: greedy

Experiment – Multi-Document Question Answering

Input Context

```
Write a high-quality answer for the given question using only the provided search
results (some of which might be irrelevant).

Document [1](Title: Asian Americans in science and technology) Prize in physics for
discovery of the subatomic particle J/ψ. Subrahmanyan Chandrasekhar shared...
Document [2](Title: List of Nobel laureates in Physics) The first Nobel Prize in
Physics was awarded in 1901 to Wilhelm Conrad Röntgen, of Germany, who received...
Document [3](Title: Scientist) and pursued through a unique method, was essentially
in place. Ramón y Cajal won the Nobel Prize in 1906 for his remarkable...

Question: who got the first nobel prize in physics
Answer:
```

Desired Answer

```
Wilhelm Conrad Röntgen
```

위치 변경

Input Context

```
Write a high-quality answer for the given question
using only the provided search results (some of
which might be irrelevant).

Document [1](Title: List of Nobel laureates in
Physics) ...
Document [2](Title: Asian Americans in science and
technology) ...
Document [3](Title: Scientist) ...

Question: who got the first nobel prize in physics
Answer:
```

Desired Answer

```
Wilhelm Conrad Röntgen
```

문서 수 변경

Input Context

```
Write a high-quality answer for the given question
using only the provided search results (some of
which might be irrelevant).

Document [1](Title: Asian Americans in science and
technology) ...
Document [2](Title: List of Nobel laureates in
Physics) ...
Document [3](Title: Scientist) ...
Document [4](Title: Norwegian Americans) ...
Document [5](Title: Maria Goeppert Mayer) ...

Question: who got the first nobel prize in physics
Answer:
```

Desired Answer

```
Wilhelm Conrad Röntgen
```

Result – Multi-Document Question Answering

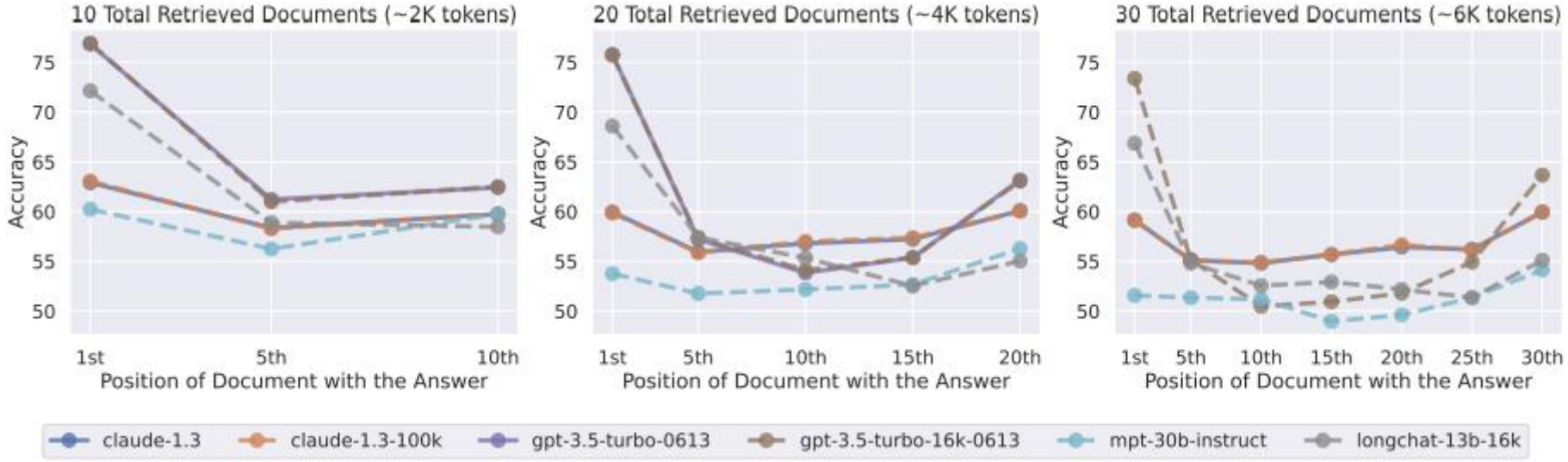
- Closed-Book: document 주어지지 않음 –> parametric memory 의존
  - GPT-3.5-Turbo > Claude-1.3 > LongChat-13B > MPT-30B-Instruct
- Oracle: 1개의 gold document 주어짐
  - GPT-3.5-Turbo > LongChat-13B > MPT-30B-Instruct > Claude-1.3

Model	Closed-Book	Oracle
LongChat-13B (16K)	35.0%	83.4%
MPT-30B-Instruct	31.5%	81.9%
GPT-3.5-Turbo	56.1%	88.3%
GPT-3.5-Turbo (16K)	56.0%	88.6%
Claude-1.3	48.3%	76.1%
Claude-1.3 (100K)	48.2%	76.4%

Result – Multi-Document Question Answering

- 관련 정보가 input context의 시작(primacy bias) 또는 끝(recency bias)에 위치할 때 높은 성능을 보여줌

- Extended-context models이 input context를 처리하는데 있어 항상 좋다고 할 수 없음
  - Context window에 input context가 fit한 경우



Experiment – Key-Value Retrieval

- Goal: LM은 입력 컨텍스트에서 검색을 얼마나 잘 수행하는지 분석

- Setup
  - String-serialized JSON object with k key-value pairs
  - Each of the keys and values are unique, randomly generated UUIDs

Input Context

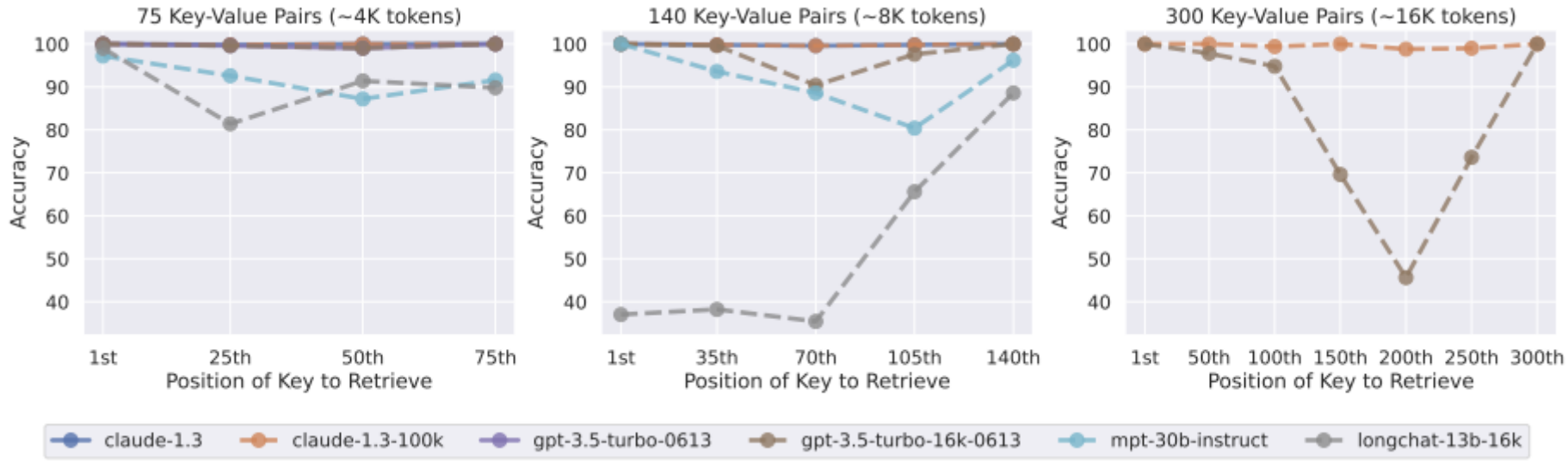
```
Extract the value corresponding to the specified key in the JSON object below.

JSON data:
{"2a8d601d-1d69-4e64-9f90-8ad825a74195": "bb3ba2a5-7de8-434b-a86e-a88bb9fa7289",
 "a54e2eed-e625-4570-9f74-3624e77d6684": "d1ff29be-4e2a-4208-a182-0cea716be3d4",
 "9f4a92b9-5f69-4725-ba1e-403f08dea695": "703a7ce5-f17f-4e6d-b895-5836ba5ec71c",
 "52a9c80c-da51-4fc9-bf70-4a4901bc2ac3": "b2f8ea3d-4b1b-49e0-a141-b9823991ebeb",
 "f4eb1c53-af0a-4dc4-a3a5-c2d50851a178": "d733b0d2-6af3-44e1-8592-e5637fdb76fb"}

Key: "9f4a92b9-5f69-4725-ba1e-403f08dea695"
Corresponding value:
```

Desired Output

```
703a7ce5-f17f-4e6d-b895-5836ba5ec71c
```

Result – Key-Value Retrieval

- Claude는 모든 input context lengths에서 거의 완벽하게 수행
- 다른 모델의 경우 140, 300 Key-Value pairs에서 성능 저하
- GPT-3.5-Turbo와 MPT-30B-Instruct는 key, value값이 중간에 존재할 때 성능이 가장 낮음



Experiment - Why Are LM Is Not Robust to Changes in the Position of Relevant Information?

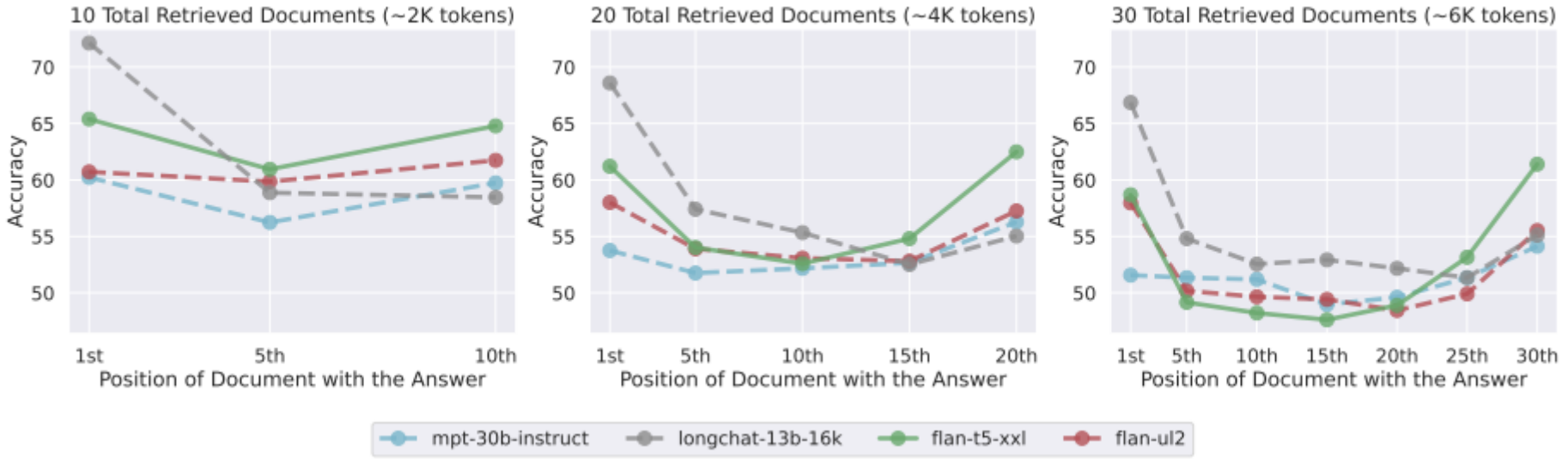
LM이 관련 정보 위치 변화에 robust하지 않은 이유 분석을 위해 아래 실험 진행

- Model Architecture
- Query-aware contextualization
- Instruct fine-tuning

Result – Effect of Model Architecture

- Decoder-only: mpt-30b-instruct, longchat-13b-16k
- Encoder-Decoder: flan-t5-xxl, flan-ul2

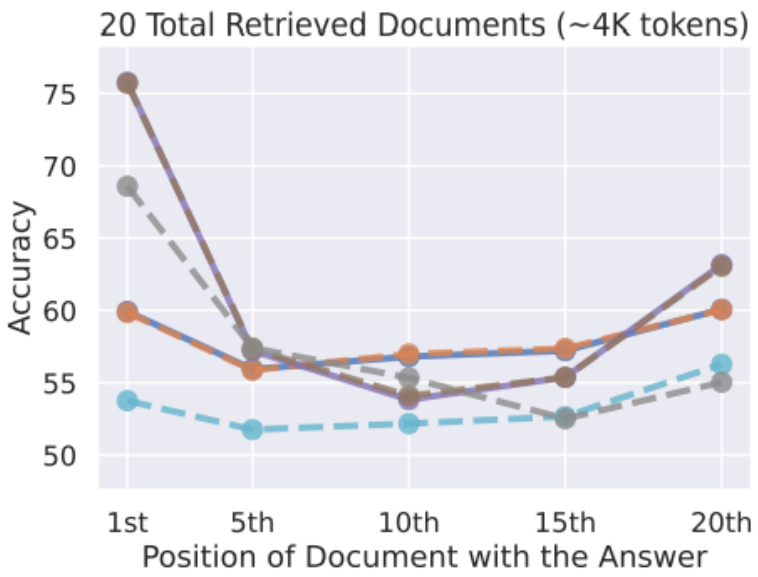
- Encoder-Decoder 모델이 학습한 context window 내에 있는 경우 입력 컨텍스트 내의 관련 정보 위치
  변화에 비교적 robust함
- 그러나 중앙 및 오른쪽 그림처럼 context가 늘어나 학습한 context window를 벗어나는 경우 관련 정보가
  중간에 배



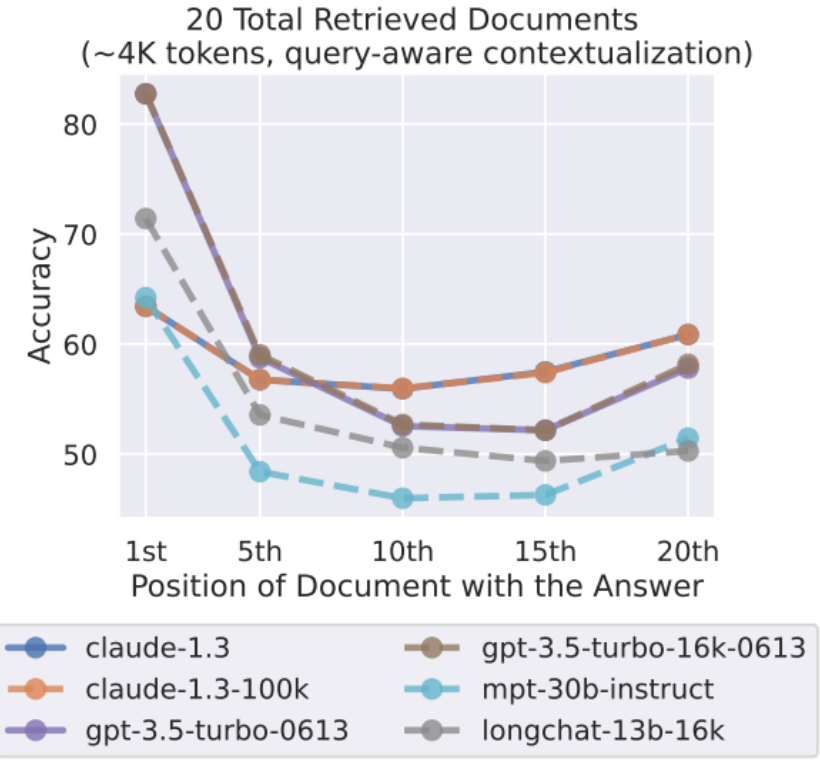
Result – Effect of Query-Aware Contextualization

- Multi-document QA에서 관련 문서를 우선적으로 주고 뒤에 query를 줌
  ➔ Decoder-only 모델은 문서 또는 key-value pairs를 contextualizing 할 때 query 토큰에 attend 할 수 없음
- 쿼리를 이전에 배치하여 decoder-only 모델을 개선할 수 있는지 분석

기존



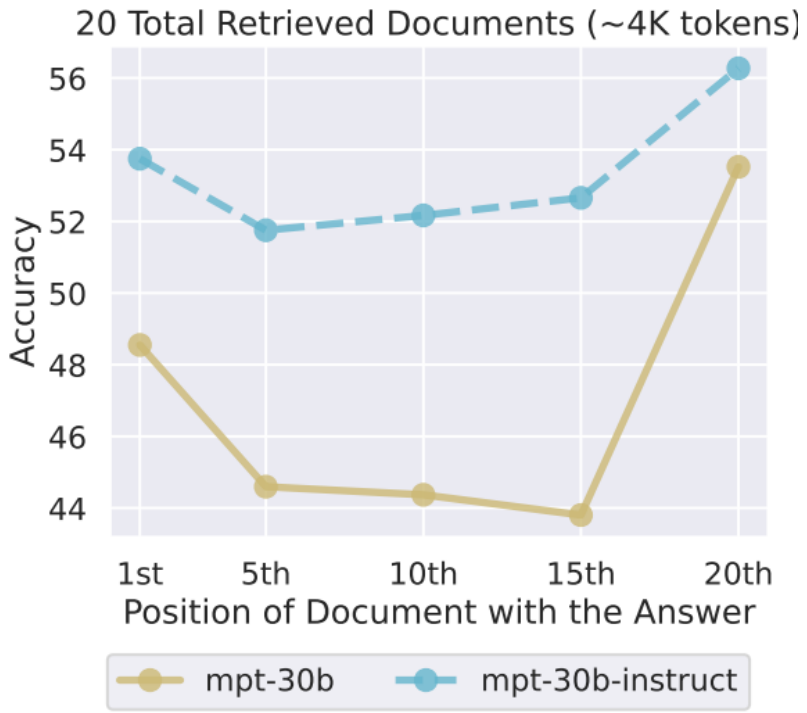
변경

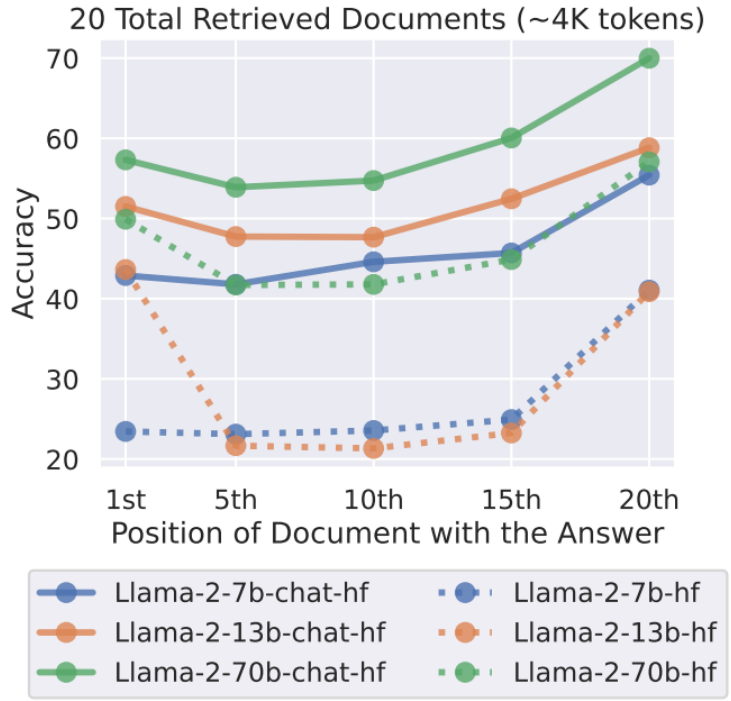


Experiment – Effect of Instruction Fine-Tuning

- Goal: instruct fine-tuning의 potential effect를 이해하기 위함
  - Instruct fine-tuned 모델은 미세조정 데이터에서 instruct prompt가 시작 부분에 배치되어 이로 인해 입력 컨텍스트의 시작 부분에 더 많은 가중치를 부여할 수도 있음

- Instruct fine-tuning시: 성능↑, best case와 worst case 성능 차이를 줄여줌
- non-instruction fine-tuned LM은 특히 관련정보가 맨 끝에 발생시 가장 성능이 높음
  ➔ 최근 토큰에 편향되어 있음을 나타냄

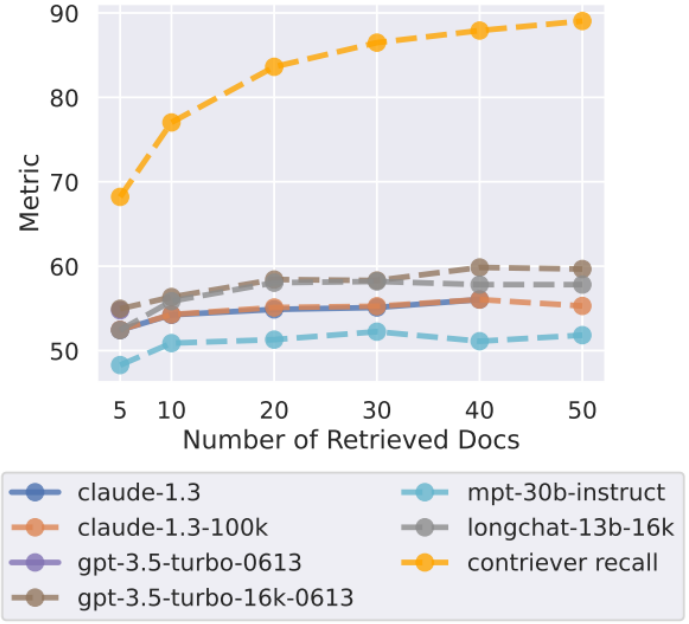




Case study

- Goal: 더 많은 context가 항상 더 좋은지 분석
  - 더 많은 정보를 제공하면 다운스트림 수행하는데 도움이 될 수 있지만 추론해야하는 content의 양도 늘어나 정확도가 떨어질 수 있음

- Standard retriever-reader setup으로 모델 평가
  - 추가 컨텍스트를 효과적으로 사용하지 않는 것을 보여줌



- 검색된 문서를 관련 정보를 시작부분에 넣어줄 수 있도록 reranking하고 적절한 문서 수를 선택하는 것이
  좋을 것이라 판단됨

결론

- LM이 긴 입력 컨텍스트를 사용하는 방식을 경험적으로 연구
- 긴 입력 컨텍스트에서 정보를 robust하게 액세스하고 사용하는데 어려움을 겪음
  특히, 관련 정보가 입력 컨텍스트 중간에 배치될 때 성능 저하가 큼
- Context window가 넘지 않는 경우 Input context 처리량이 커진다고 input context 처리 능력 또한 향상되는 것은 아님
- Decoder only는 맨 앞에 query가 나왔을 때 더 좋은 성능을 보임
- Instruct fine-tuning 했을 때 성능이 더 좋으며 best-case와 worst case 성능차가 적음
- Input context를 무작정 늘리는 것보다 적절한 문서 수를 선택하고
  맨 앞이나 맨 끝에 관련 문서가 올 수 있도록 reranking하는 것이 좋음

Open Questions

- Retriever-reader setup에서 더 잘하기 위해 어떤 시도들을 해볼 수 있을지?
- Lost in the middle 해결 방안?
  - Found in the Middle: Permutation Self-Consistency Improves Listwise Ranking in Large Language Models

감사합니다.