고려대학교 KOREA UNIVERSITY

Natural Language Processing & Artificial Intelligence

연구실 세미나

고려대학교 NLP&AI 연구실
발표자: 손준영

◆ Introduction

ICLR 2025

Published as a conference paper at ICLR 2025

RETRIEVAL HEAD MECHANISTICALLY EXPLAINS LONG-CONTEXT FACTUALITY

Wenhao Wu[λ] **Yizhong Wang**[δ] **Guangxuan Xiao**[σ] **Hao Peng**[π] **Yao Fu**[μ]
[λ]Peking University [δ]University of Washington [σ]MIT [π]UIUC [μ]University of Edinburgh
waynewu@pku.edu.cn haopeng@illinois.edu yao.fu@ed.ac.uk
https://github.com/nightdessert/Retrieval_Head

EMNLP 2025

Query-Focused Retrieval Heads Improve Long-Context Reasoning and Re-ranking

Wuwei Zhang[♠] **Fangcong Yin**[♢] **Howard Yen**[♠] **Danqi Chen**[♠] **Xi Ye**[♠]
[♠] Princeton Language and Intelligence, Princeton University
[♢] The University of Texas at Austin
[♠] {wuwei.zhang,hyen,danqic}@cs.princeton.edu xi.ye@princeton.edu
[♢] fangcongyin@utexas.edu

LLM 내부 attention head가 생성하는 query-to-document attention을 문서 관련도 신호로 해석하고,
이를 in-context document ranking에 활용하는 방법에 대한 논문들

◆ Retrieval Head Mechanistically Explains Long-Context Factuality

Motivation

- Transformer-based Language Model의 Long-context Capabilities는 매우 중요함:
  extremely-long context로부터 관련 있는 정보를 정확히 가져오는 능력과 이를 활용하여 효과적으로 downstream tasks를 해소하는 방법 등

- 특히, 입력에 주어진 내용 중 관련 있는 정보를 식별하고, 그 정도들을 기반으로 multi-step reasoning을 수행하는 능력에 대하여,
  How do these model acquire such long-context capabilities ?

➔ Large Language Model의 내부 내커니즘을 조사하겠다!

◆ Retrieval Head Mechanistically Explains Long-Context Factuality

Hypothesis

- CopyNet[1] (Gu et al., 2016):

  "생성 모델은 입력의 정보를 복사해서 출력할 수 있다."

- Induction Heads[2] (Olsson et al., 2022):

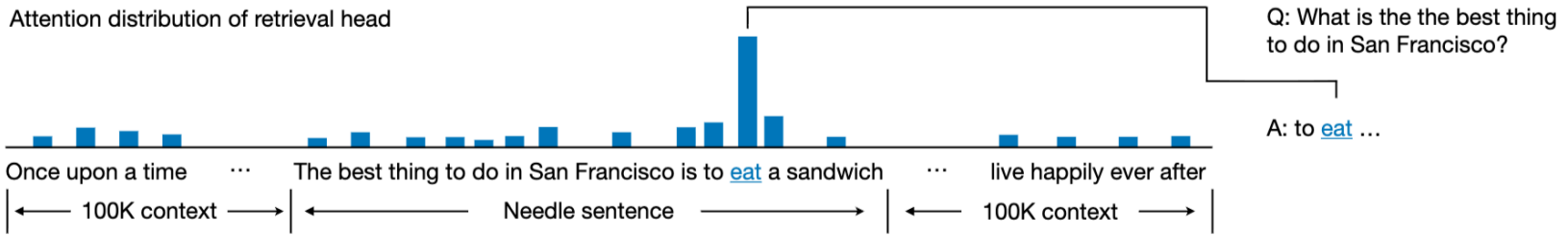
  "Transformer 내부에는 이전 context의 패턴을 찾아 현재 예측에 활용하는 attention heads가 있다."

➔ "LLM의 long-context capability도 단순한 블랙박스 능력이 아니라, 입력에서 필요한 정보를 찾아오고 재사용하는 특정 attention mechanism으로 설명될 수 있다"는 가설로 시작함

1) Gu, Jiatao, et al. "Incorporating copying mechanism in sequence-to-sequence learning." Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. 2016.
2) Olsson, Catherine, et al. "In-context learning and induction heads." arXiv preprint arXiv:2209.11895 (2022).

◆ Retrieval Head Mechanistically Explains Long-Context Factuality

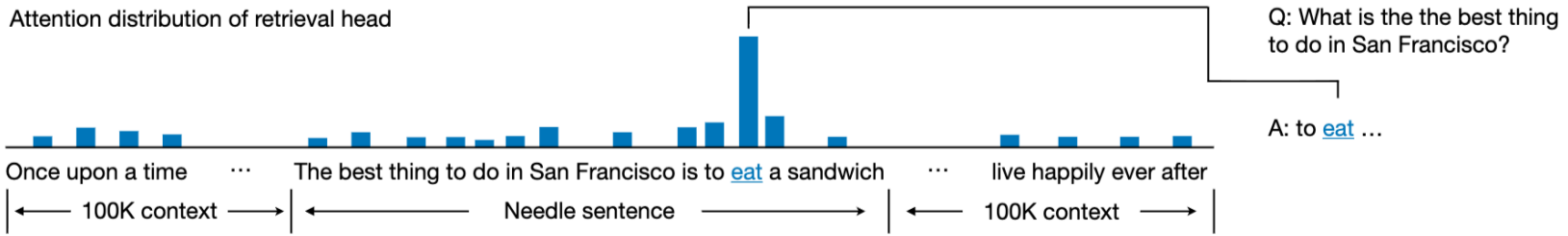
Detecting Retrieval Head



- Needle-in-a-Haystack (NIAH) 세팅 기반
  - question q and its corresponding answer k (the "needle") → insert k into a context x (the "haystack")
  - The language model is tasked with answering q based on the haystack with the inserted needle
  - The needle k is semantically irrelevant to the context x

◆ Retrieval Head Mechanistically Explains Long-Context Factuality

Detecting Retrieval Head

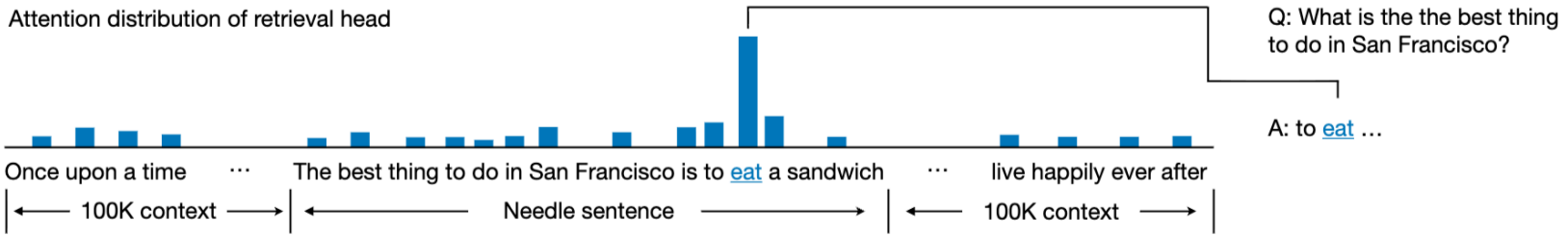


- Retrieval Score for Attention Heads:
  the current token being generated as $\boldsymbol{w}$ and the attention scores of $\boldsymbol{a}$ head as $\boldsymbol{a} \in \mathbb{R}^{|x|}$
  - An attention head h copy-paste operation for a token w from the needle if it satisfies two conditions:
    1. Token Inclusion: The token $\boldsymbol{w}$ must belong to the needle sentence, i.e.,
    2. Maximal Attention: The token $\boldsymbol{w}$ must correspond to the input position $\boldsymbol{j}$,
  i.e., $\boldsymbol{x_j = w}$ where $\boldsymbol{j = argmax(a)}$ and $\boldsymbol{j \in i_q}$, meaning that the highest attention score aligns with $\boldsymbol{w}$.

- Let $\boldsymbol{g_h}$ represent the set of tokens copied from the needle, The retrieval score for head $\boldsymbol{h}$ is then defined as:
  - Retrieval score for head $\boldsymbol{h} = |\boldsymbol{g_h}|/|\boldsymbol{k}|$

◆ Retrieval Head Mechanistically Explains Long-Context Factuality

Detecting Retrieval Head



- Let g_h represent the set of tokens copied from the needle, The retrieval score for head h is then defined as:
  - Retrieval score for head $h = \frac{|g_h|}{|k|}$
- 모든 attention heads에 대하여 retrieval score 계산
  - 각 모델에 대해 약 600개의 NIAH instances (다양한 길이, 위치 포함)에 대하여 계산
  - 각 head의 retrieval score를 계산한 뒤, 인스턴스들에 대하여 평균을 냄 → 각 head의 평균 retrieval score
  - threshold: 평균 retrieval score 〉 0.1인 경우 해당 head를 retrieval head로 분류
    (최소 10%의 사례에서 copy-paste 동작을 수행)
  - retrieval head 식별을 위해 약 600개 인스턴스가 필요

◆ Retrieval Head Mechanistically Explains Long-Context Factuality

Detecting Retrieval Head

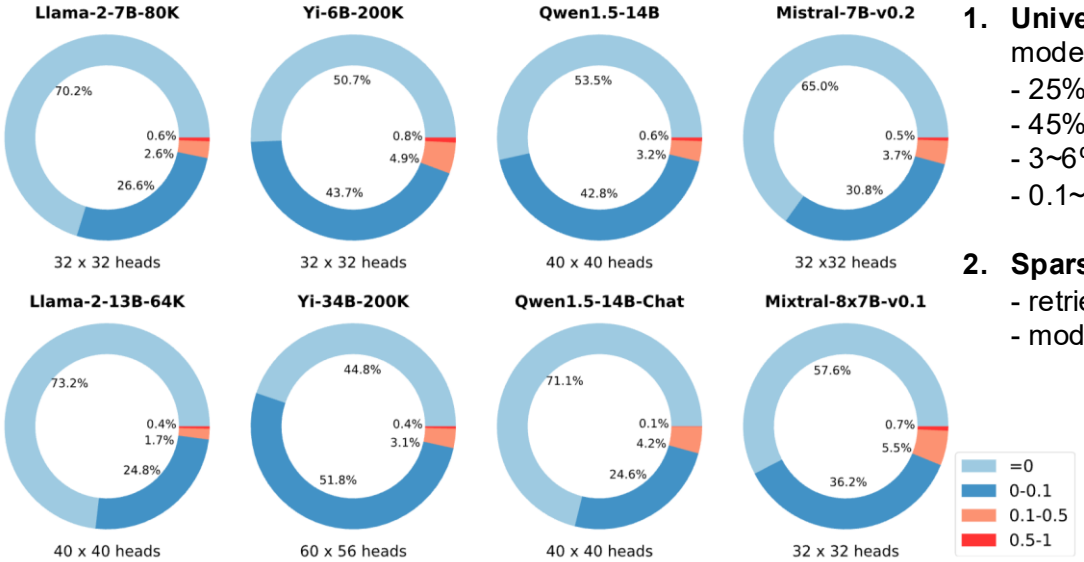
Base Model	Variant	Variation Type
Llama-2-7B	Llama-2-7B-80K	Length Extension via Continue Pretrain
	Llama-2-13B-64K	Model Scaling and Length Extension
Mistral-7B-v0.2	Mistral-7B-Instruct-v0.2	SFT and RLHF
	Mixtral-8x7B-v0.1	Sparse Upcycling to Mixture of Experts
Yi-6B	Yi-6B-200K	Length Extension via Continual Pretraining
	Yi-34B-200K	Model Scaling and Length Extension
Qwen1.5-14B	Qwen1.5-14B-Chat	SFT and RLHF

- context length extension에 대한 continued pretraining의 영향 분석
- Mistral/Qwen: alignment 의 효과에 따른 retrieval head 분석
- Mixtral: MoE architecture에서의 분석

... 등을 위한 다양한 model variation 고려

◆ Retrieval Head Mechanistically Explains Long-Context Factuality

Basic Properties of Retrieval Heads: (1) University and Sparsity

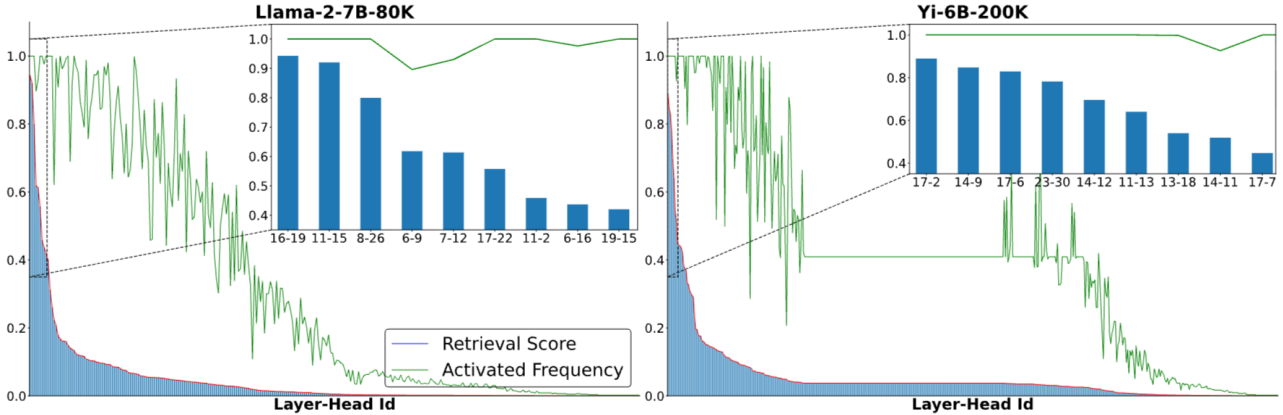


1. **University**
   model variations에 상관 없이,
   - 25%~52%: copy-paste behavior 관측
   - 45%~73%의 attention heads는 retrieval score가 0
   - 3~6%의 attention heads: retrieval score > 0.1
   - 0.1~0.8%: retrieval score > 0.5

2. **Sparsity**: depend on the task
   - retrieval-head tasks → higher level sparsity
   - model's internal knowledge tasks → low level sparsity

◆ Retrieval Head Mechanistically Explains Long-Context Factuality

Basic Properties of Retrieval Heads: (2) Dynamic Activation

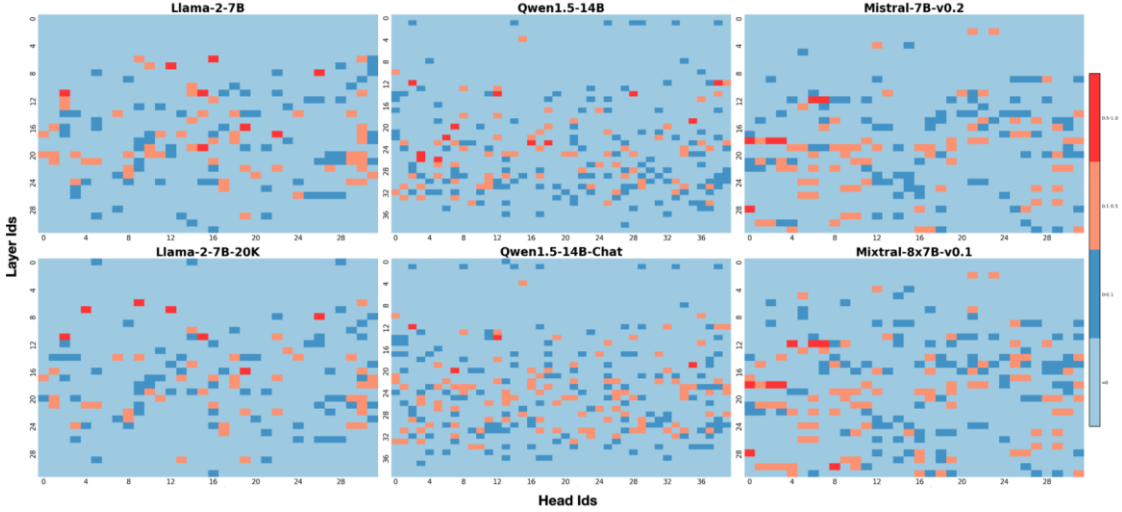


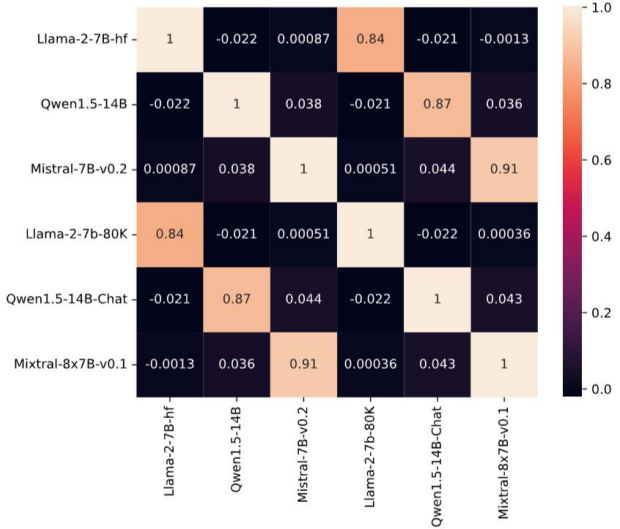
- Activation Frequency: “특정 attention head가 주어진 입력에서 적어도 한 토큰에 대해 활성화되는가?”에 대한 빈도 (한 샘플에서 해당 head가 한 토큰이라도 copy-paste 동작을 수행하면 그 샘플에서 head는 활성화 된 것으로 간주)
- Head의 맥락 의존성 발견: 1) Retrieval Score가 낮고, 2) Activation Frequency가 높은 경우
  ➔ 특정 토큰/contexts에서만 특정 head가 활성화 됨을 의미함
- Llama-2-7B-80K (left)에서는 12개 attention heads는 모든 샘플에서 activation frequency = 1
- Yi-6B-200K (right)에서는 36개 attention heads가 모든 샘플에서 activation frequency = 1
- 즉, 일부 retrieval heads는 context-agnostic하게 전역적으로 동작하지만 (초록 곡선과 파란 곡선이 둘 다 높은 경우), 일부 retrieval heads는 context-dependent하게 특정 맥락에서만 동작함 (초록 곡선은 높으나 파란 곡선이 낮은 경우)

◆ Retrieval Head Mechanistically Explains Long-Context Factuality

Basic Properties of Retrieval Heads: (3) Intrinsic Nature

(left) model architecture에 따른 retrieval heads의 layer별 분포
(right) model variants와 Spearman correlation (first row vs. second row)





- Model variants (first row vs. second row)에 따라서 consistency in retrieval patterns
  ➔ contionued pretraining (Llama-2-7B), chat fine-tuning (Qwen1.5-14B), Architecture variant (Mistral vs. Mixtral) 에 따라서 retrieval patterns 변화가 거의 없이 consistent 함
- 실제 Spearman correlation (right figure)도 같은 family (LLama2 vs. Llama2-80K) 내 fine-tuned counterpart와 상당히 높은 correlation을 보임
  ➔ Retrieval Heads는 Model Pre-training 과정에서 자연스럽게 발생하는 Intrinsic Nature 성질을 가졌다고 말할 수 있음

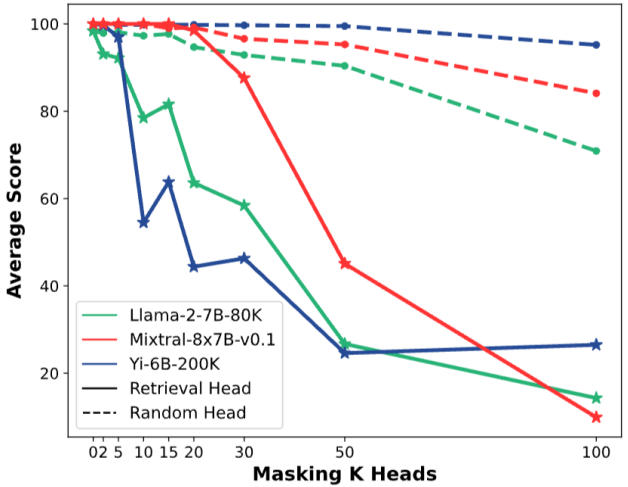
◆ Retrieval Head Mechanistically Explains Long-Context Factuality

Retrieval Heads and Factuality in NIAH

- retrieval score 기준으로 정렬했을 때, top-K retrieval heads를 masking out 하여 NIAH 성능을 측정

- Prunning 이 들어가도 성능이 어느정도 유지가 됨 (~10 Masking K Heads)
  ➔ retrieval heads 끼리 상호보완적인 작용을 하도록 동작함

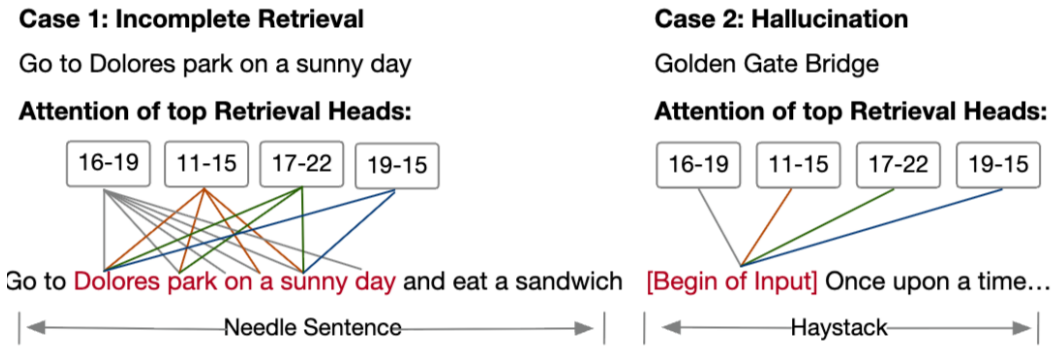
- Masking out Random head보다 Masking out Retrieval Head가 NIAH에서 치명적인 성능 감소를 보임
  ➔ 실제 long-context scenario에서 in-context retrieval에 관여



◆ Retrieval Head Mechanistically Explains Long-Context Factuality

Retrieval Heads and Factuality in NIAH

- Masking out Top-K retrieval heads이 어떤 에러를 보이는가?: fail case 분석



- **Incomplete retrieval (left)**: 필요한 정보의 일부분만 retrieval, 일부 중요한 내용 retrieval fail
- **Hallucination/wrong extraction (right)**: Initial tokens에 retrieval heads가 attention을 하는 현상

➔ *The most effective retrieval heads (Top-K)가 없이 weaker heads만 남았을 때, partial information만 참조하거나, 잘못된 정보를 참조하는 경우가 다수 발생*
특히 retrieval score > 0.4인 retrieval heads가 masking out된 경우 이런 오류 유형이 자주 발생하고, 더 많은 retrieval heads가 masking out될 수록 빈도가 급격하게 증가하는 경향성 있다고 함

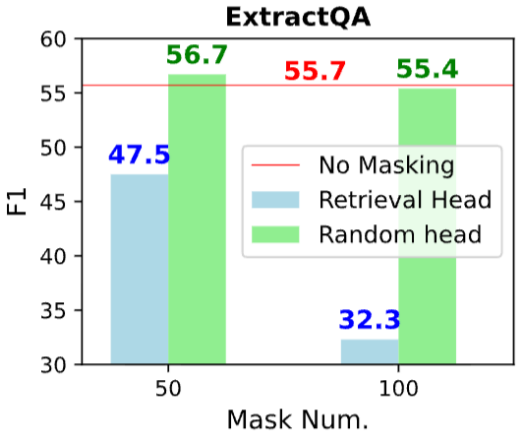
◆ Retrieval Head Mechanistically Explains Long-Context Factuality

Impact on Extractive Question Answering

- Retrieval heads가 다른 downstream tasks에 어떤 영향을 주는지 분석 (extractive QA)
- Model internal parametric knowledge의 개입을 방지하기 위해서 최근 news articles로 extractive QA 데이터 구성

- Masking out Random heads는 Extractive QA 성능에 거의 영향을 주지 않으나 (No Masking과 비슷), Retrieval Heads를 Masking한 경우 성능이 크게 감소함
  (9.2% for 50 and 23.1% for 100 감소)

  ➔ Retrieval Heads가 Real-world long-context QA tasks에서도 매우 중요한 역할을 함



◆ Retrieval Head Mechanistically Explains Long-Context Factuality

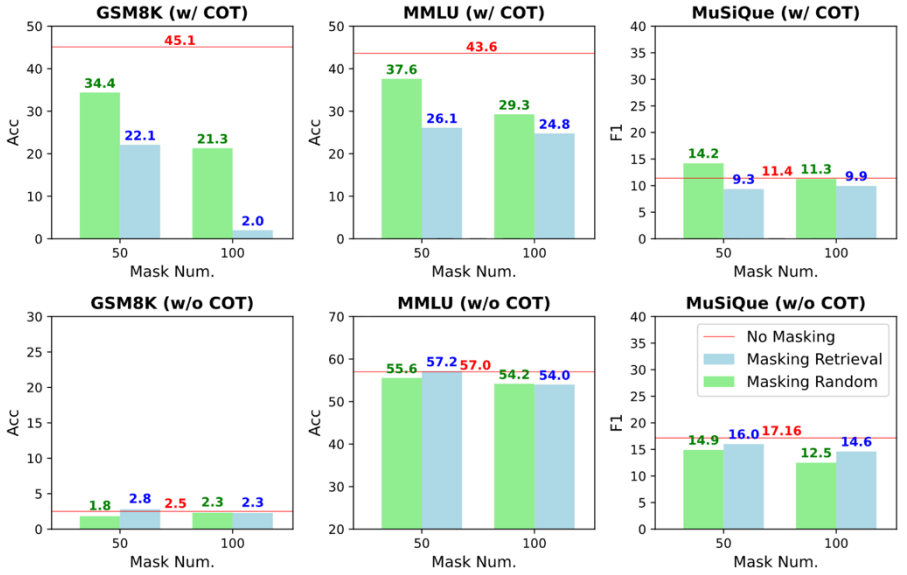
Chain-of-Thought Reasoning Also Requires Retrieval Heads

- Retrieval heads가 reasoning tasks에 주는 영향을 분석: Mistrial-7B-Instruct-v0.2 on MMLU, MuSiQue, and GSM8K에 대하여 CoT reasoning 수행
- MMLU는 model's parametric knowledge 평가가 목적이므로, CoT reasoning 영향을 덜 받을 것이고, MuSiQue (Multi-hop QA)와 GSM8K (Math)는 CoT가 중요한 역할을 할 것임

- CoT 가 없을 때는 (w/o COT) masking out retrieval head/random이 성능에 영향을 거의 주지 않음

- CoT를 사용할 때 (upper, first row)는 Retrieval Heads가 Task 수행에 많은 영향을 주는 것을 확인

➔ Chain-of-Thought를 수행할 때 retrieval heads가 입력을 자주 참조(copy-paste)하므로, retrieval heads가 중요한 역할을 수행함



◆ Retrieval Head Mechanistically Explains Long-Context Factuality

Chain-of-Thought Reasoning Also Requires Retrieval Heads

Question 1:
Mishka bought 3 pairs of shorts, 3 pairs of pants, and 3 pairs of shoes. One pair of shorts costs $16.50. One pair of pants costs $22.50 and one pair of shoes costs $42.
How many dollars did Mishka spend on all the clothing items?

Masking Top 100 Retrieval Heads:

Let's calculate the cost of each type of clothing for Mishka:
1. For shorts, there are 3 pairs, so the total cost is 3 * $16.5 = $59.10.
2. For pants, the cost is the same, so 3 pairs cost $3 * $22 = $66.
3. In total, Mishka spent $59 + $66 = $125.
So, the answer is $125.

Misunderstood Information

Question 2:
James decides to run 3 sprints 3 times a week. He runs 60 meters each sprint. How many total meters does he run a week ?

Masking Top 100 Retrieval Heads:

Let's think through this step by step:
1. James runs 3 times a week for each sprint session.
2. During each session, he covers a distance of 60 meters (3 times the length of a standard football field goal post).
3. Therefore, the total distance James covers in a week is:
6 (sessions in a week) x1 (number of sprints in session)
x 60(meters in a sprint)= 360 meters.
So, James runs a total of 360 meters every week.

- Left: Masking out retrieval heads 이후, pants, shoes에 대한 costs를 retrieve 실패 → hallunication 발생
- Right: "3 sprints, 3 times a week"을 retrieve 하지 못함
  ➔ new rules를 hallucination 함 (6 sessions in a week, 1 sprints)

◆ Query-Focused Retrieval Heads Improve Long-Context Reasoning and Re-ranking

Query-Focused Retrieval Heads Improve Long-Context Reasoning and Re-ranking

Wuwei Zhang[♠] **Fangcong Yin**[◇] **Howard Yen**[♠] **Danqi Chen**[♠] **Xi Ye**[♠]
[♠] Princeton Language and Intelligence, Princeton University
[◇] The University of Texas at Austin
[♠] {wuwei.zhang,hyen,danqic}@cs.princeton.edu xi.ye@princeton.edu
[◇] fangcongyin@utexas.edu

- 바로 직전에 발표한 Retrieval Head 논문이 조금 더 LLM Behavior에 집중했다면,
  이 논문은 1) Retrieval Head의 더 효과적인 식별과,
  2) Information Retrieval (Reranking) 관점에 더 접목하여 실험을 수행하고 전개하는 것에 더 집중한 논문이다.

◆ Query-Focused Retrieval Heads Improve Long-Context Reasoning and Re-ranking

Motivation

- 이전 Retrieval Heads 연구의 두 가지 문제점 지적

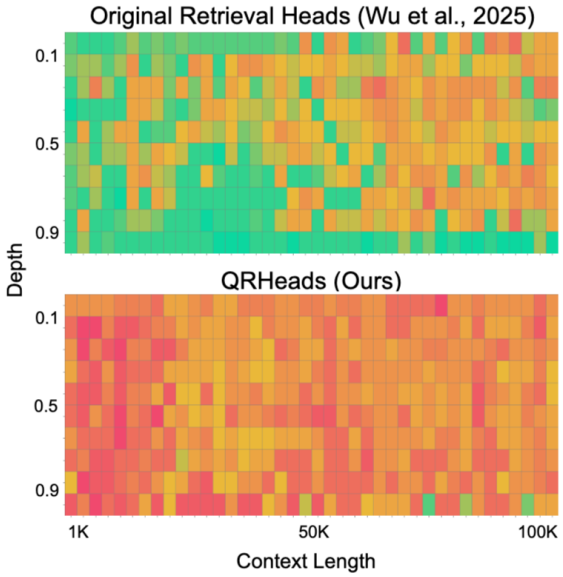
  1. Copy-paste operation에 기반한 방법: decoding 단계까지 가서 계산할 수 있음

  2. NIAH 시나리오에 기반한 Synthetic data setting으로 Retrieval Head set을 식별하는 점:
     - Real-world tasks 와 거리가 멂
     ➔ 실제 언어 모델이 입력에 주어진 문서 집합에서 중요한 정보를 식별하는 시나리오와 맞지 않음

◆ Query-Focused Retrieval Heads Improve Long-Context Reasoning and Re-ranking

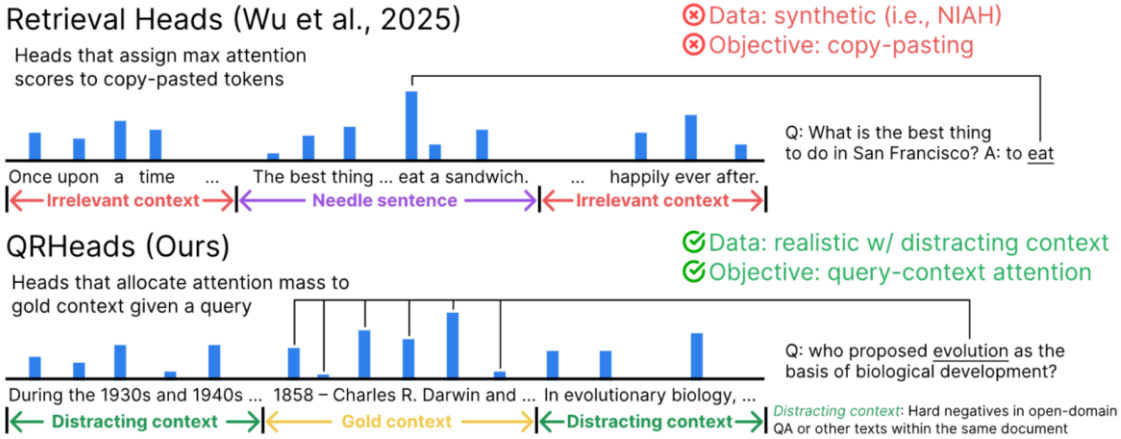
Two key changes and findings

- **A query-context scoring function:**
  - input query에 대응하는 적절한 context spans에 할당된 attention mass를 측정하도록 함
- **More natural data from real-world tasks:**
  - Synthetic data가 아닌, question answering over long texts와 같은 실제 real-world tasks에 기반한 데이터를 활용하여 Retrieval Heads를 식별하도록 함
- 이전 연구의 in-domain scenario인 NIAH에서 Original Retrieval Heads/QRHead를 masking out했을 때 More natural data (out-domain)에서 식별한 head set을 사용함에도 불구하고, Original Retrieval Heads보다 더 큰 영향을 보였음 (우측 Figure)



◆ Query-Focused Retrieval Heads Improve Long-Context Reasoning and Re-ranking

QRHead: Identifying Query-Focused Retrieval Heads

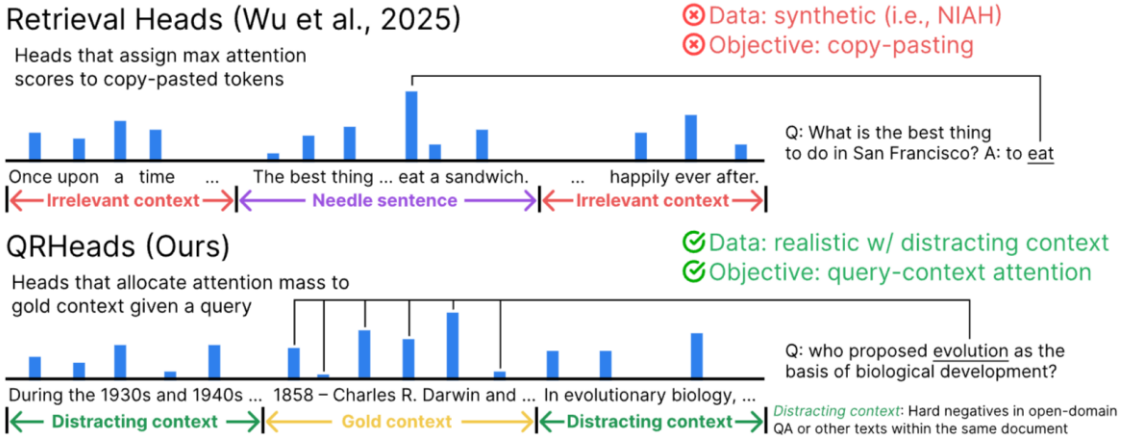


- **Original Retrieval Heads:** Decoding 단계에서 생성할 target token이 1) Needle 에 있는 token 이고, 2) attention head에서 가장 많이 attend 하는 token인 경우를 조건으로 Retrieval Heads를 식별

- **QRHead:** For all $doc \in D$에 대하여 Query - doc attention mass를 계산하고,
  전체 context documents 중 Gold context에 가는 attention mass가 높은 attention heads를 식별하여 retrieval heads로 사용

◆ Query-Focused Retrieval Heads Improve Long-Context Reasoning and Re-ranking

QRHead: Identifying Query-Focused Retrieval Heads



- **QRHead**: For all $doc \in D$에 대하여 Query - doc attention mass를 계산하고,
  전체 context documents 중 Gold context에 가는 attention mass가 높은 attention heads를 식별하여 retrieval heads로 사용

- **Query-focused retrieval score:** query tokens → document tokens에 대한 attention weights의 합을 query 길이로 나눈 값

$$QRscore_h(q, d_i) = \frac{1}{|q|} \sum_{t_q \in q} \sum_{t_d \in d_i} A_h^{t_q \to t_d}$$

◆ Query-Focused Retrieval Heads Improve Long-Context Reasoning and Re-ranking

QRRetriever: Using QRHead to Build A General-Purpose Retriever

- **Query - document relevancy:** A selected set of QRHead = $\mathcal{H}^{select}$

$$R(q, d_i) = \frac{1}{|\mathcal{H}^{select}|} \sum_{h \in \mathcal{H}^{select}} QRscore_h(q, d_i)$$

- **Calibration:** LM의 attention weights에 대한 intrinsic biases를 해소하기 위해 제안된 방법으로,
  $R(q, d_i)$에서 $R(q_{null}, d_i)$ (q_{null}("N/A")) 를 빼주는 방식으로 모델에 내제된 attention biases를 제거하는 방법 (Chen et al. 2025[1])

- **Final Query-document relevancy:**

$$R(q, d_i) - R(q_{null}, d_i)$$

1) Chen, Shijie, Bernal Jimenez Gutierrez, and Yu Su. "Attention in large language models yields efficient zero-shot re-rankers." International Conference on Learning Representations. 2025.

◆ Query-Focused Retrieval Heads Improve Long-Context Reasoning and Re-ranking

Experimental setup

- **데이터셋: Long-Context Multi-Hop Reasoning**

1. **LongMemEval**[1]: LLM-based chat assistant의 장기 상호작용 기억 능력을 평가하기 위한 데이터셋
   - Information Extraction, Multi-Session Reasoning, Temporal Reasoning, Knowledge Update, Abstention 등을 평가
   - 사용자-어시스턴트 대화 이력에서 evidence를 찾아 질문에 답하는 NIAH-style memory evaluation 데이터셋
   - 표준 설정 기준 각 질의는 약 115k tokens, 30-40 sessions의 사용자-어시스턴트 대화 히스토리로 구성

   ** head detection에 LongMemEval의 70 examples를 사용했다고 함*

2. **CLIPPER**[2]: 책에 대한 Chapter 별 outlines와 summaries를 기반으로 합성된 claim (참/거짓)에 대하여, claim이 참인지 거짓인지 실제 책의 내용을 참조하여 판별해야하는 데이터셋
   - 19K 규모의 Book-Claim(참/거짓) 쌍 구성으로,
   - 각 example prompt는 book text를 포함 → 평균 90k tokens의 long-context task
   - Book-level claims는 여러 book chapters의 사건을 종합해야 하므로, long-context multi-hop reasoning을 요구

1) Wu, Di, et al. "Longmemeval: Benchmarking chat assistants on long-term interactive memory." arXiv preprint arXiv:2410.10813 (2024).
2) Pham, Chau Minh, Yapei Chang, and Mohit Iyyer. "Clipper: Compression enables long-context synthetic data generation." arXiv preprint arXiv:2502.14854 (2025).

◆ Query-Focused Retrieval Heads Improve Long-Context Reasoning and Re-ranking

Experimental setup

- Base LMs

    1. *Llama-3.2 (3B)*
    2. *Llama-3.1 (8B and 70B)*
    3. *Qwen2.5 (7B)*

- 10B보다 작은 모델은 Top-16 Attention Heads 사용, 70B 모델은 Top-32 Attention Heads 사용
  ➔ Approximately 1-2% of the total attention heads

1) Wu, Di, et al. "Longmemeval: Benchmarking chat assistants on long-term interactive memory." arXiv preprint arXiv:2410.10813 (2024).
2) Pham, Chau Minh, Yapei Chang, and Mohit Iyyer. "Clipper: Compression enables long-context synthetic data generation." arXiv preprint arXiv:2502.14854 (2025).

◆ Query-Focused Retrieval Heads Improve Long-Context Reasoning and Re-ranking

Experimental setup

- Baselines
  - **RankGPT**[1]
    - LLM을 list-wise reranker로 활용하는 training-free reranking 방법
    - Query와 candidate documents를 함께 입력하고, LLM이 관련도 순서로 문서 리스트를 재정렬 (생성 기반)
    - 별도 학습 없이 강한 reranking 성능을 보이지만, generation 기반이므로 inference cost가 큼
  - **In-Context-Re-ranking (ICR)**[2]
    - LLM 내부 attention weights를 활용하는 training-free zero-shot reranking 방법
    - Query와 candidate documents를 하나의 context로 입력
    - Query-to-document attention mass를 문서 관련도 점수로 해석
    - Generation 없이 attention score만으로 candidate documents를 재정렬
    - QRHead와의 차이점은 retrieval heads를 식별하는 방법은 없고, attention heads를 모두 사용함
  - **그 외 Contriever**[3]**, 1.5B Stella V5**[4] **등의 dense Retriever**

1) Sun, Weiwei, et al. "Is ChatGPT good at search? investigating large language models as re-ranking agents." Proceedings of the 2023 conference on empirical methods in natural language processing. 2023.
2) Chen, Shijie, Bernal Jimenez Gutierrez, and Yu Su. "Attention in large language models yields efficient zero-shot re-rankers." International Conference on Learning Representations. 2025.
3) Izacard, Gautier, et al. "Unsupervised dense information retrieval with contrastive learning." arXiv preprint arXiv:2112.09118 (2021).
4) Zhang, Dun, et al. "Jasper and stella: distillation of sota embedding models." arXiv preprint arXiv:2412.19048 (2024).

◆ Query-Focused Retrieval Heads Improve Long-Context Reasoning and Re-ranking

Experimental Results

Retriever	LongMemEval				CLIPPER			
	Retrieval Recall@k		End-to-End Performance		Retrieval Recall@k		End-to-End Performance	
	k = 5	k = 10	Top-5	Top-10	k = 3	k = 5	Top-3	Top-5
Base LM: Llama-3.2-3B-Instruct								
Full context	-	-	28.1		-	-	25.2	
BM25	57.5	67.5	46.1	44.9	74.6	83.7	20.0	22.8
Contriever	62.7	79.2	**48.6**	46.5	60.2	78.9	12.6	18.4
Stella	63.9	77.6	44.9	**47.7**	83.3	90.0	21.3	25.1
RankGPT	1.8	3.4	23.5	23.3	16.8	27.3	3.6	8.8
RankGPT^{Bubble}	2.1	3.8	24.0	24.4	17.0	27.4	3.8	8.8
ICR	68.7	78.8	46.5	45.1	72.8	83.6	19.4	23.6
QRRetriever (Ours)	**77.6**	**86.6**	47.4	**47.7**	**85.5**	**93.4**	23.4	**26.9**
Base LM: Llama-3.1-8B-Instruct								
Full context	-	-	46.5		-	-	31.3	
BM25	57.5	67.5	48.8	50.9	74.6	83.7	37.9	37.9
Contriever	62.7	79.2	52.6	55.4	60.2	78.9	28.2	31.1
Stella	63.9	77.6	50.9	58.4	83.3	90.0	38.8	39.6
RankGPT	2.1	4.0	26.7	24.2	30.0	39.4	15.9	19.4
RankGPT^{Bubble}	8.3	9.0	28.1	27.0	36.7	44.3	19.7	20.4
ICR	77.0	84.4	59.3	56.1	89.3	94.7	43.8	**42.5**
QRRetriever (Ours)	**85.5**	**91.7**	**59.8**	**60.2**	**93.8**	**96.9**	**47.6**	41.9
Base LM: Llama-3.1-70B-Instruct								
Full context	-	-	34.2		-	-	63.9	
BM25	57.5	67.5	52.8	53.0	74.6	83.7	60.1	66.5
Contriever	62.7	79.2	53.7	60.5	60.2	78.9	38.5	49.7
Stella	63.9	77.6	56.3	62.3	83.3	90.0	65.9	71.2
RankGPT	1.8	3.5	21.2	27.4	57.0	63.4	44.7	50.4
RankGPT^{Bubble}	47.9	49.0	44.0	42.6	74.3	78.8	58.4	61.5
ICR	32.1	46.5	36.5	39.8	86.2	93.1	68.9	71.6
QRRetriever (Ours)	**80.4**	**88.5**	**66.7**	**67.7**	**95.0**	**98.2**	**74.2**	**73.3**

- **QRRetriever는 long context에서 retrieval / end-to-end performances 모두에서 강력한 성능을 보임.**
  - LongMemEval, CLIPPER 모두에서 강력한 성능 (retrieval recall and end-to-end performance)
  - Llama-3.1-8B-Instruct Full context 기준 both tasks에서 10% 이상의 end-to-end performance 개선
- **QRRetriever generalizes across domains.**
  - QRRetriever는 off-the-shelf dense retrievers (Contriever and Stella)를 압도함
  - 별도 학습 없이도, dense retrievers보다 더 잘 작동했다는 점에서 cross domain generalization이 강하다고 해석
- **QRRetriever scles with model sizes.**
  - RankGPT나 ICR의 경우 model scale에 따라서 성능 차이가 심한 반면, QRRetriever는 model size가 개선될 수록 성능이 개선이 됨

◆ Query-Focused Retrieval Heads Improve Long-Context Reasoning and Re-ranking

Experimental Results

Retriever	LongMemEval				Clipper			
	Retrieval Recall@k		End-to-End Performance		Retrieval Recall@k		End-to-End Performance	
	k = 5	k = 10	Top-5	Top-10	k = 3	k = 5	Top-3	Top-5
Base LM: Llama-3.2-3B-Instruct								
Full context	-	-	28.1		-	-	25.2	
BM25	57.5	67.5	46.1	44.9	74.6	83.7	20.0	22.8
Contriever	62.7	79.2	**48.6**	46.5	60.2	78.9	12.6	18.4
Stella	63.9	77.6	44.9	**47.7**	83.3	90.0	21.3	25.1
RankGPT	1.8	3.4	23.5	23.3	16.8	27.3	3.6	8.8
RankGPT^{Bubble}	2.1	3.8	24.0	24.4	17.0	27.4	3.8	8.8
ICR	68.7	78.8	46.5	45.1	72.8	83.6	19.4	23.6
QRRetriever (Ours)	**77.6**	**86.6**	47.4	**47.7**	**85.5**	**93.4**	23.4	**26.9**
Base LM: Llama-3.1-8B-Instruct								
Full context	-	-	46.5		-	-	31.3	
BM25	57.5	67.5	48.8	50.9	74.6	83.7	37.9	37.9
Contriever	62.7	79.2	52.6	55.4	60.2	78.9	28.2	31.1
Stella	63.9	77.6	50.9	58.4	83.3	90.0	38.8	39.6
RankGPT	2.1	4.0	26.7	24.2	30.0	39.4	15.9	19.4
RankGPT^{Bubble}	8.3	9.0	28.1	27.0	36.7	44.3	19.7	20.4
ICR	77.0	84.4	59.3	56.1	89.3	94.7	43.8	**42.5**
QRRetriever (Ours)	**85.5**	**91.7**	**59.8**	**60.2**	**93.8**	**96.9**	**47.6**	41.9
Base LM: Llama-3.1-70B-Instruct								
Full context	-	-	34.2		-	-	63.9	
BM25	57.5	67.5	52.8	53.0	74.6	83.7	60.1	66.5
Contriever	62.7	79.2	53.7	60.5	60.2	78.9	38.5	49.7
Stella	63.9	77.6	56.3	62.3	83.3	90.0	65.9	71.2
RankGPT	1.8	3.5	21.2	27.4	57.0	63.4	44.7	50.4
RankGPT^{Bubble}	47.9	49.0	44.0	42.6	74.3	78.8	58.4	61.5
ICR	32.1	46.5	36.5	39.8	86.2	93.1	68.9	71.6
QRRetriever (Ours)	**80.4**	**88.5**	**66.7**	**67.7**	**95.0**	**98.2**	**74.2**	**73.3**

- **Compact LMs의 제한된 generation abilities에도 불구하고, 강력한 Retrieval 성능을 보였음**
  - Llama-3.2-3B-Instruct 기준 Recall@10 (LongMemEval) 86.6점
  ➔ Llama-3.1-70B-Instruct의 88.5점과 유사
  - 그러나 Llama-3.2-3B-Instruct는 Generation abilities가 부족해서, End-to-End performance는 Top-10 기준 47.7점 (vs. 67.7점 70B)
  ➔ 성능 저하의 주된 원인은 어떤 토큰/문서가 중요한지 식별하는 것에 있는 것이 아니라, 복잡한 추론/문장생성을 제대로 못 하는 Generation 측면에 있다고 분석
  ➔ 실용적 제안: 경량 모델을 빠르고 비용 효율적인 retriever로 활용하고, 상위 모델을 generator로 쓰는 파이프라인이 효율적일 것

◆ Query-Focused Retrieval Heads Improve Long-Context Reasoning and Re-ranking

Passage Re-Ranking: BEIR[1] benchmark

- QRRetriever는 IR Benchmark 에서도 ICR/RankGPT, 그리고 Dual- and Cross-Encoder baselines 보다 consistently 능가하는 성능을 보였음

  ➔ Information Retrieval Benchmark로 평가해도 QRRetriever의 방법론이 일관적으로 좋다!

	NQ	COVID	NFCorpus	FiQA	Scifact	Scidocs	FEVER	Climate	DBPedia	Robust04	News	Avg
BM25	30.5	59.5	32.2	23.6	67.9	14.9	65.1	16.5	31.8	40.7	39.5	38.4
Base LM: Llama-3.2-3B-Instruct												
RankGPT	30.0	59.5	32.2	23.6	67.9	14.9	65.9	17.1	31.8	40.7	39.5	38.5
RankGPT^{Bubble}	33.2	61.8	32.0	22.4	66.1	14.8	65.8	17.1	34.8	40.5	40.2	39.0
ICR	49.2	72.3	33.8	31.8	73.3	17.4	82.6	24.2	34.7	47.2	44.7	46.5
QRRETRIEVER (Ours)	**54.9**	**77.4**	**35.1**	**35.1**	**74.7**	**18.3**	**83.7**	**24.5**	**36**	**49.7**	**45.1**	**48.6**
Base LM: Llama-3.1-8B-Instruct												
RankGPT	30.0	59.5	32.2	23.6	67.9	14.9	65.9	16.8	31.8	40.7	39.5	38.4
RankGPT^{Bubble}	53.7	75.5	34.3	31.4	69.3	17.4	67.5	23.8	**42.9**	47.8	**46.2**	46.3
ICR	54.0	73.3	34.8	35.6	75.5	19.0	**85.8**	**24.8**	36.9	49.0	44.5	48.5
QRRETRIEVER (Ours)	**58.6**	**77.5**	**35.3**	**39.1**	**76.2**	**19.4**	85.3	23.9	37.2	**51.4**	**46.2**	**50.0**
Base LM: Qwen-2.5-7B-Instruct												
RankGPT	30.0	59.5	32.2	23.6	67.9	14.9	65.9	16.8	31.8	40.7	39.5	38.4
RankGPT^{Bubble}	42.7	**70.5**	**34.1**	**29.5**	69.3	**16.6**	70.5	19.7	**37.1**	**46.4**	**43.6**	43.6
ICR	43.1	66.1	32.7	27.0	**71.1**	16.4	79.2	19.6	35.3	43.0	40.0	43.0
QRRETRIEVER (Ours)	**49.9**	67.7	33.1	29.2	71	15.3	**80.7**	**20.1**	35.7	43.7	39.8	**44.2**
Base LM: Llama-3.1-70B-Instruct												
RankGPT	45.4	62.7	33.6	28.6	71.3	16.1	74.2	18.9	37.6	41.3	39.8	42.7
RankGPT^{Bubble}	58.4	**81.2**	**36.1**	41.0	76.1	20.2	80.0	**25.1**	**45.5**	**59.0**	**48.5**	**51.9**
ICR	57.9	72.3	34.2	38.9	74.6	19.4	**86.4**	22.0	38.3	42.6	40.4	47.9
QRRETRIEVER (Ours)	**62.1**	73.9	34.7	**43.8**	**76.8**	**20.4**	85.9	23.1	35.7	51.6	43.5	50.1
Dual Encoder and Cross Encoder												
Contriever	44.6	67.5	32.8	28.4	67.1	18.9	64.2	28.0	39.5	45.7	41.7	43.5
GTR-t5-base	51.4	74.8	32.5	34.7	62.1	15.8	72.9	26.8	37.1	46.1	42.8	45.2
BGE-Reranker-base	55.2	66.4	31.0	31.7	70.8	15.7	88.6	36.5	42.5	39.9	37.0	46.8
msmarco-MiniLM	55.8	74.3	35.2	35.1	68.5	17.5	80.4	25.5	45.3	47.9	43.0	48.0

1) *Thakur et al., "BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models," NeurIPS, 2021.*

◆ Query-Focused Retrieval Heads Improve Long-Context Reasoning and Re-ranking

Impact of Head Selection

1. QRHead
2. All Attention Heads
3. Original Retrieval Heads
4. Randomly Selected Heads

- BEIR and LongMemEval
- 두 모델 패밀리에서 QRHead가 가장 좋음

		BEIR$_{\text{SHUFFLED}}$ NDCG@10	LONGMEMEVAL RECALL
Llama-8B	RANDOM HEADS	37.5	59.8
	FULL HEADS	42.8	77.0
	RETRIEVAL HEAD	43.4	81.5
	QRHEAD	**47.5**	**85.5**
Qwen-7B	RANDOM HEADS	19.9	57.2
	FULL HEADS	22.6	67.1
	RETRIEVAL HEAD	27.4	70.7
	QRHEAD	**31.9**	**83.2**

1) Thakur et al., "BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models," NeurIPS, 2021.

◆ Query-Focused Retrieval Heads Improve Long-Context Reasoning and Re-ranking

Generalizability Across Lengths

- 32K context length에서 QRHead 를 선택하고, 128K (or 64K)에 적용하는 경우 선택된 heads가 longer context lengths에서 generalize 할지? (혹은 그 반대도)
  → short-to-long and long-to-short

- short-to-longer 에서 generalizable,
  . long-to-short이 더 잘 generalize 되는 경향성 있음

	Model: LLama-3.1-8B-Instruct			
	NQ+FEVER		LongMemEval	
	32K	128K	32K	128K
ICR	66.7	56.5	85.2	78.2
QRRETRIEVER^{32K}	**70.1**	63.9	89.2	85.2
QRRETRIEVER^{128K}	68.8	**67.2**	**89.2**	**85.6**
	Model: Qwen-2.5-7B-Instruct			
	NQ+FEVER		LongMemEval	
	32K	64K	32K	64K
ICR	40.0	17.4	83.4	67.1
QRRETRIEVER^{32K}	51.9	25.3	**90.2**	**77.9**
QRRETRIEVER^{64K}	**54.1**	**29.1**	90.1	77.0

1) Thakur et al., "BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models," NeurIPS, 2021.

◆ Query-Focused Retrieval Heads Improve Long-Context Reasoning and Re-ranking

Ablations on Scoring Function and Task Selection For Head Detection

- Two key factors의 중요성을 평가하기 위해서, (**query-context attention objective** and **using realistic data**)
  1) NIAH (using Synthetic data) 에서 QRscore로 head detection 를 수행해봄
- NIAH로 detected head가 Original Retrieval Head보다 consistently improved
  ➔ Synthetic data에서도 QRscore가 잘 작동함. 즉, query-context attention objective가 꽤 중요한 factor임을 강조
- 그러나 NQ/LongMemEval 데이터로 식별한 head set이 NIAH (synthetic)에서 식별한 head set보다 성능이 더 좋음
  ➔ "using realistic data" factor도 중요함
- 즉, QRHead를 구성하는 two key factors 모두 중요한 요소임

	Data	BEIR NDCG@10	LongMem Recall
	Model: LLama-3.1-8B-Inst		
QRHead	NQ	**47.5**	83.9
QRHead	LongMem	47.1	**85.6**
QRHead	NIAH	46.8	83.4
Retrieval Head	NIAH	43.4	81.5
	Model: Qwen-2.5-7B-Inst		
QRHead	NQ	31.9	80.2
QRHead	LongMem	**32.1**	**83.2**
QRHead	NIAH	30.9	79.7
Retrieval Head	NIAH	27.4	70.7

◆ Query-Focused Retrieval Heads Improve Long-Context Reasoning and Re-ranking

Head Detection Data의 Sensitivity에 대하여

- head detection process의 robustness를 평가하기 위해, 서로 다른 random samples를 구성하여 성능을 평가해봄
- 그럼에도 우측 Table의 Overlap 부분을 보면, 50 heads 이상이 Top 64 heads에서 overlap이 됨을 발견함
- 게다가, different subset 사이의 BEIR 성능이 거의 비슷함을 확인

➔ QRHead는 small samples of data로 충분히 신뢰할 수 있는 성능에 도달할 수 있고, dataset sensitivity도 그렇게 크지 않고 reliable하다!

	Overlap (Top 64) Set0	Set1	Set2	BEIR nDCG@10
	Model: LLama-3.1-8B-Inst			
QRHEAD^{Set0}	64	51	51	49.8
QRHEAD^{Set1}	51	64	53	49.7
QRHEAD^{Set2}	51	53	64	49.9
	Model: Qwen-2.5-7B-Inst			
QRHEAD^{Set0}	64	50	53	44.2
QRHEAD^{Set1}	50	64	57	44.4
QRHEAD^{Set2}	53	57	64	44.5

감사합니다.