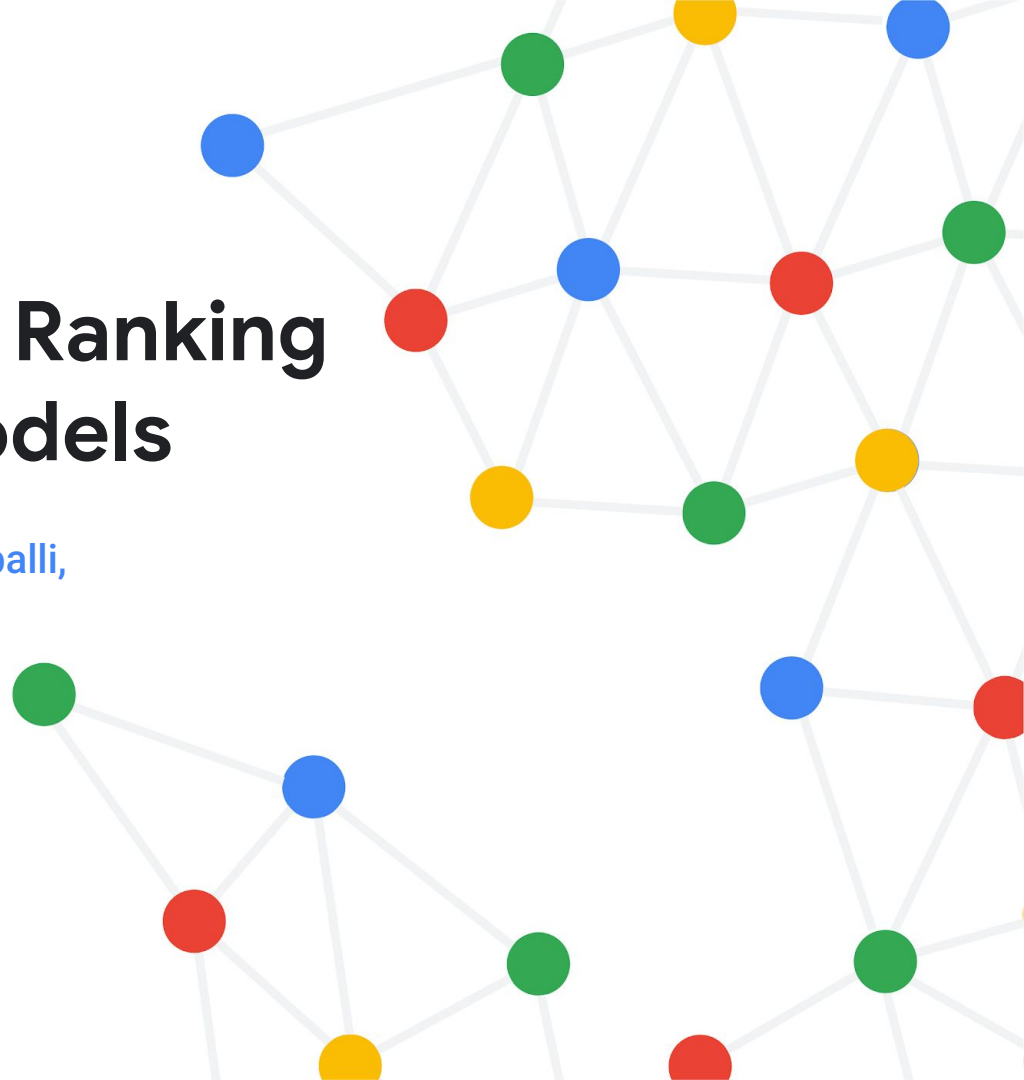
Scalable In-context Ranking with Generative Models

Nilesh Gupta, Chong You, Srinadh Bhojanapalli,
Sanjiv Kumar, Inderjit Dhillon & Felix Yu

Project Link



Agenda

01 IR in the age of LLMs

02 ICR Attention Analysis

03 BlockRank - scalable ICR

04 Future Directions

Agenda

01 IR in the age of LLMs

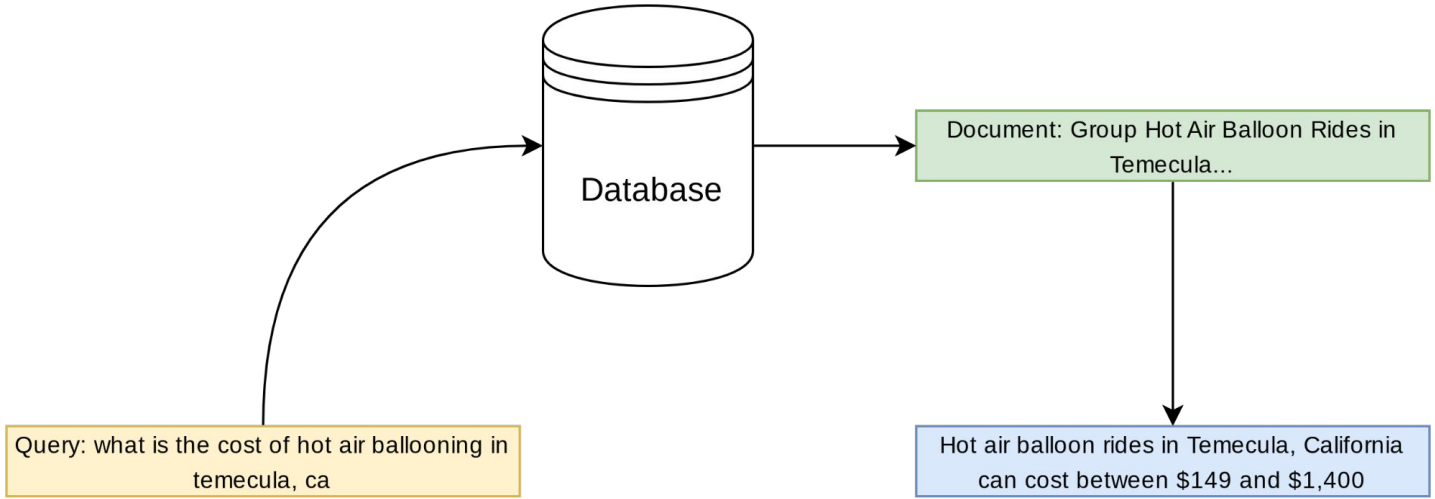
02 ICR Attention Analysis

03 BlockRank - scalable ICR

04 Future Directions

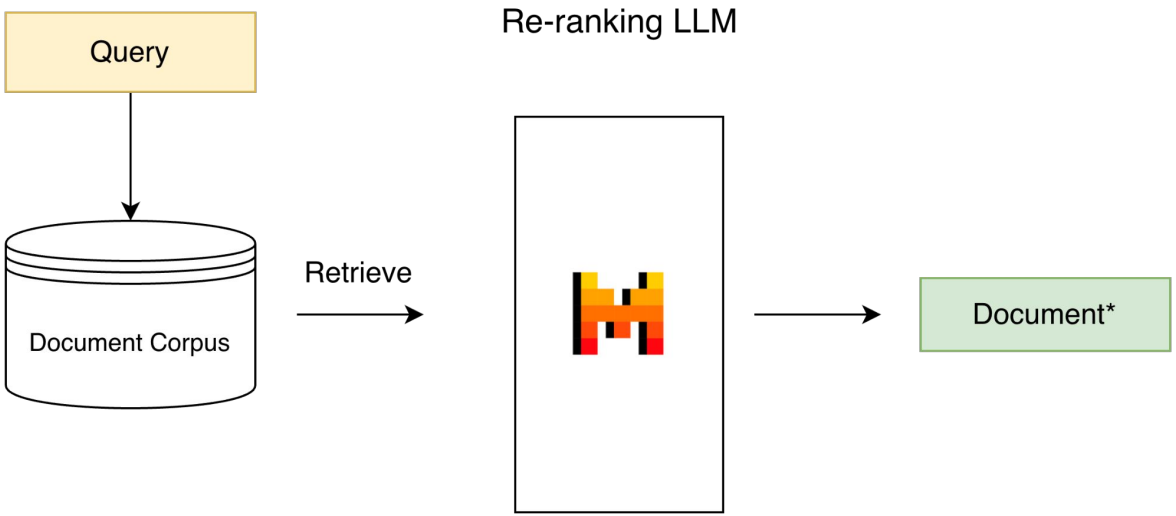
Information Retrieval (IR)

Seeking answer for a query from a database



Typical IR Pipeline

Retrieve (embedding based) then Rerank (strong language model)



Role of LLMs in IR

LLMs have proven to be strong re-rankers especially as in-context listwise ranker

Reference

Instruction

```
You will be given a list of documents. You need to read carefully and understand all of them. Then you
will be given a query that may require you to use 1 or more documents to find the answer. Your goal is
to find all documents from the list that can help answer the query.
```

Corpus Formatting

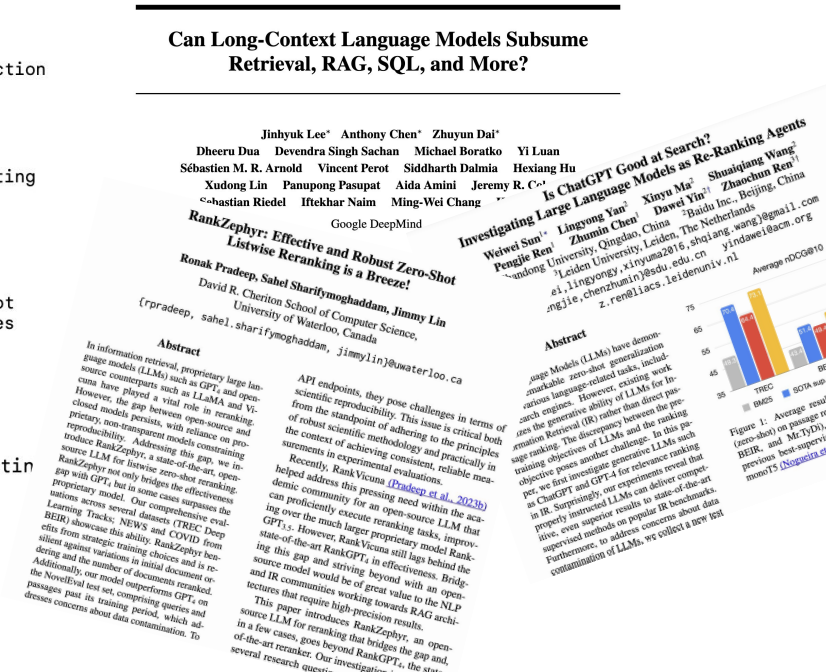
```
ID: 0 | TITLE: Shinji Okazaki | CONTENT: Shinji Okazaki is a Japanese … | END ID: 0
…
ID: 53 | TITLE: Ain't Thinkin' 'Bout You | CONTENT: "Ain't Thinkin' 'Bout You" is a song … | END ID: 53
ID: 54 | TITLE: Best Footballer in Asia 2016 | CONTENT: … was awarded to Shinji Okazaki … | END ID: 54
…
```

Few-shot Exemples

```
====== Example 1 ======
Which documents are needed to answer the query? Print out the TITLE and ID of each document. Then format
the IDs into a list.
query: What year was the recipient of the 2016 Best Footballer in Asia born?
The following documents are needed to answer the query:
TITLE: Best Footballer in Asia 2016 | ID: 54
TITLE: Shinji Okazaki | ID: 0
Final Answer: [54, 0]
…
```

Query Formattin

```
====== Now let's start! ======
Which documents are needed to answer the query? Print out the TITLE and ID of each document. Then format
the IDs into a list.
query: How many records had the team sold before performing "aint thinkin bout you"?
The following documents are needed to answer the query:
```



Can Long-Context Language Models Subsume Retrieval, RAG, SQL, and More?

Jinhyuk Lee* Anthony Chen* Zhuyun Dai*
Dheeru Dua Devendra Singh Sachan Michael Boratko Yi Luan
Sébastien M. R. Arnold Vincent Perot Siddharth Dalmia Hexiang Hu
Xudong Lin Panupong Pasupat Aida Amini Jeremy R. Co
Sebastian Riedel Iftekhar Naim Ming-Wei Chang
Google DeepMind

RankZephyr: Effective and Robust Zero-Shot Listwise Reranking is a Breeze!

Ronak Pradeep, Sahel Sharifymoghaddam, Jimmy Lin
David R. Cheriton School of Computer Science, University of Waterloo, Canada
{rpradeep, sahel.sharifymoghaddam, jimmylin}@uwaterloo.ca

Abstract

In information retrieval, proprietary large language models (LLMs) such as GPT-4 and open-source counterparts such as LLaMA and Vicuna have played a vital role in reranking. However, the gap between open-source and closed models persists, with reliance on proprietary, non-transparent models constraining reproducibility. Addressing this gap, we introduce RankZephyr, a state-of-the-art, open-source LLM for listwise zero-shot reranking. RankZephyr not only bridges the effectiveness gap with GPT-4 but in some cases surpasses the proprietary model. Our comprehensive evaluations across several datasets (TREC Deep Learning Tracks; NEWS and COVID from BEIR) showcase this ability. RankZephyr benefits from strategic training choices and is resilient against variations in initial document ordering and the number of documents reranked. Additionally, our model outperforms GPT-4 on the NovelEval test set, comprising queries and passages past its training period, which addresses concerns about data contamination. To

API endpoints, they pose challenges in terms of scientific reproducibility. This issue is critical both from the standpoint of adhering to the principles of robust scientific methodology and practically in the context of achieving consistent, reliable measurements in experimental evaluations.

Recently, RankVicuna (Pradeep et al., 2023b) helped address this pressing need within the academic community for an open-source LLM that can proficiently execute reranking tasks, improving over the much larger proprietary model RankGPT3.5. However, RankVicuna still lags behind the state-of-the-art RankGPT4 in effectiveness. Bridging this gap and striving beyond with an open-source model would be of great value to the NLP and IR communities working towards RAG architectures that require high-precision results.

This paper introduces RankZephyr, an open-source LLM for reranking that bridges the gap and, in a few cases, goes beyond RankGPT4, the state-of-the-art reranker. Our investigation of several research questi

Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agents

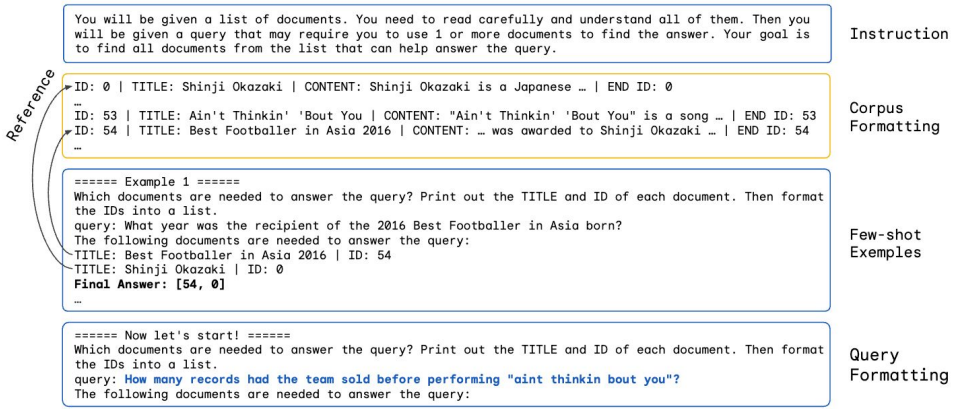
Weiwei Sun[1*] Lingyong Yan[2] Zhumin Chen[1] Shuaiqiang Wang[2] Zhaochun Ren[1†]
Pengjie Ren[1] Xinyu Ma[2] Dawei Yin[2]
[1]Shandong University, Qingdao, China [2]Baidu Inc., Beijing, China
[1]Leiden University, Leiden, The Netherlands
{sunweiwei,lingyongy,xinyuma2016,shqiang.wang}@gmail.com
{renpengjie,chenzhumin}@sdu.edu.cn yindawei@acm.org
z.ren@liacs.leidenuniv.nl

Abstract

Average nDCG@10

Figure 1: Average resul (zero-shot) on passage r BEIR, and Mr.TyDi). previous best-supervi monoT5 (Nogueira et

Challenges



- Inefficient with ranking list size O(N^2) attention
- Positional bias of in-context documents - lost in the middle!
- Interpretability - does the model retrieve the right information?

BlockRank

Can we use the special structure in an ICR task for efficient training and inference

Agenda

01 IR in the age of LLMs

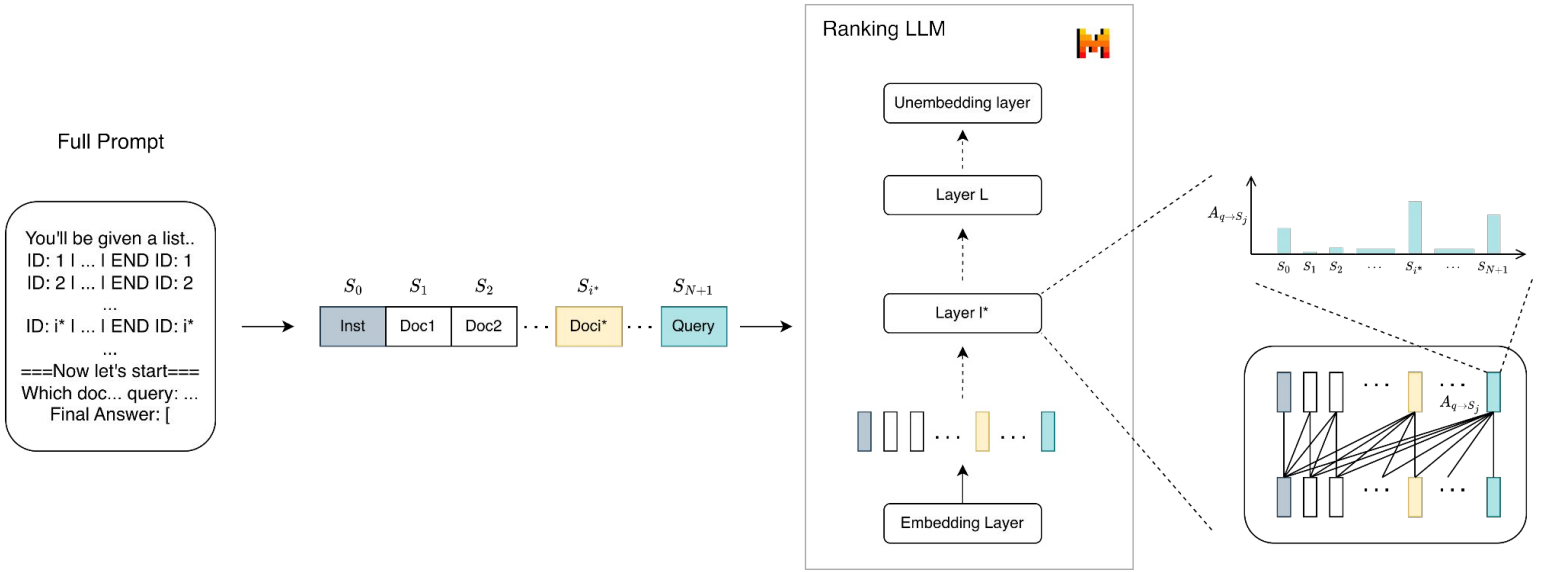
02 ICR Attention Analysis

03 BlockRank - scalable ICR

04 Future Directions

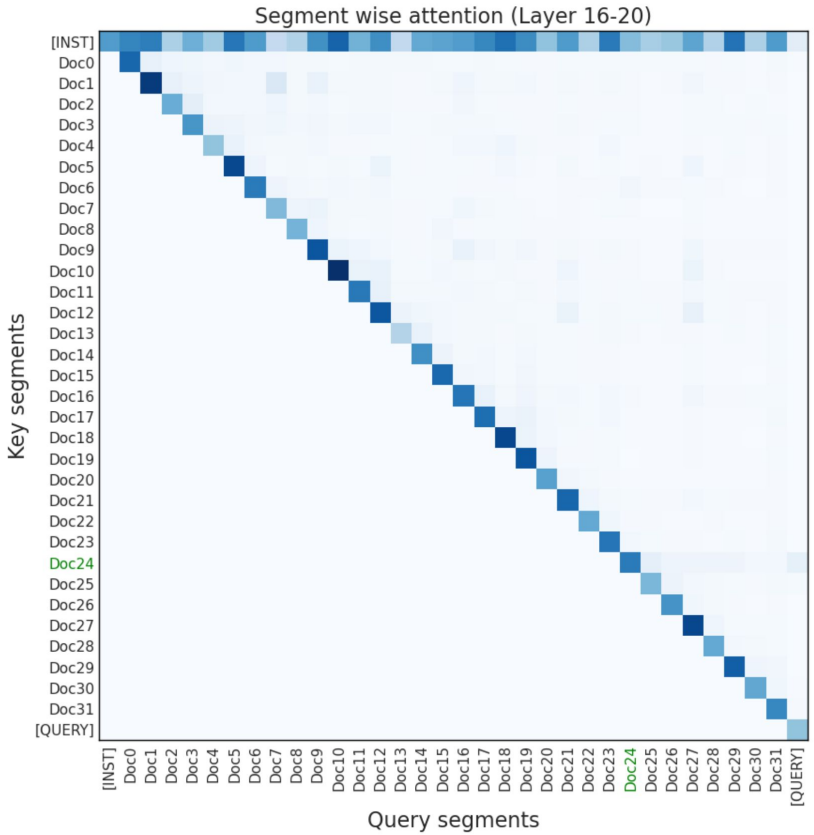
Analysis Setup

We analyze blockwise attention patterns inside a ranking LLM performing ICR

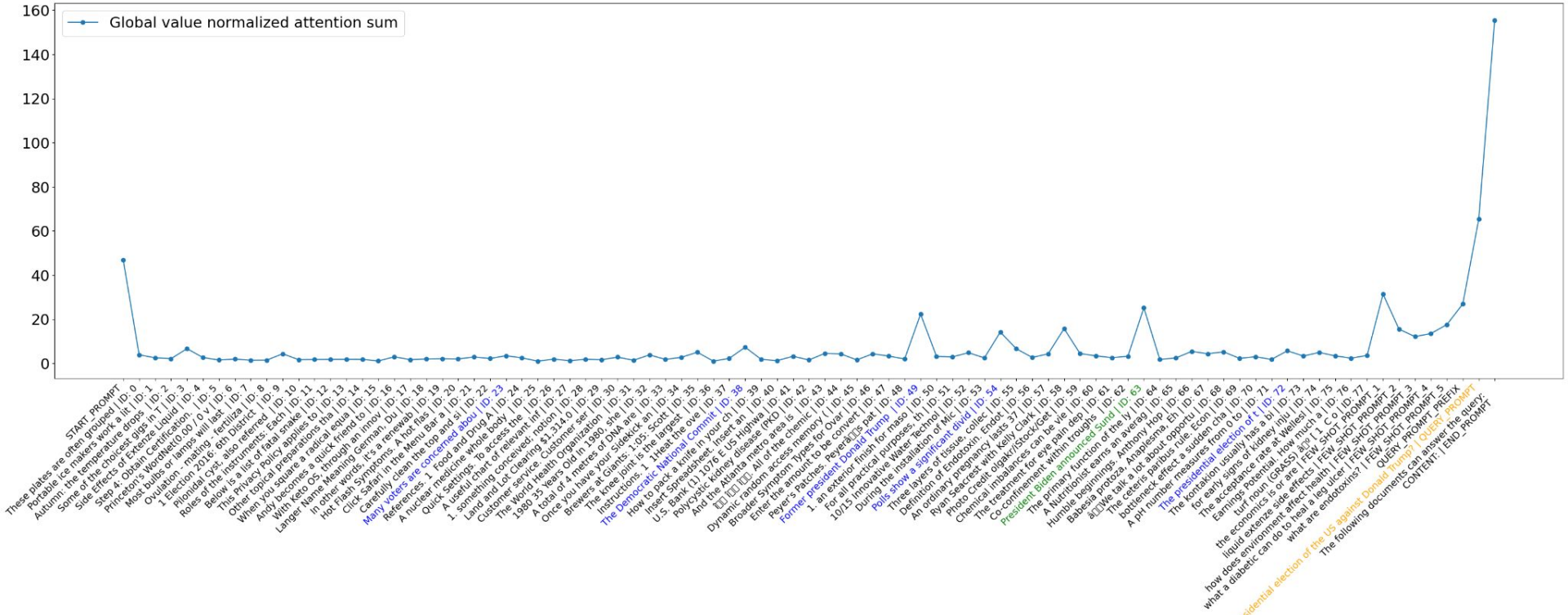


Inter-document Block Sparsity

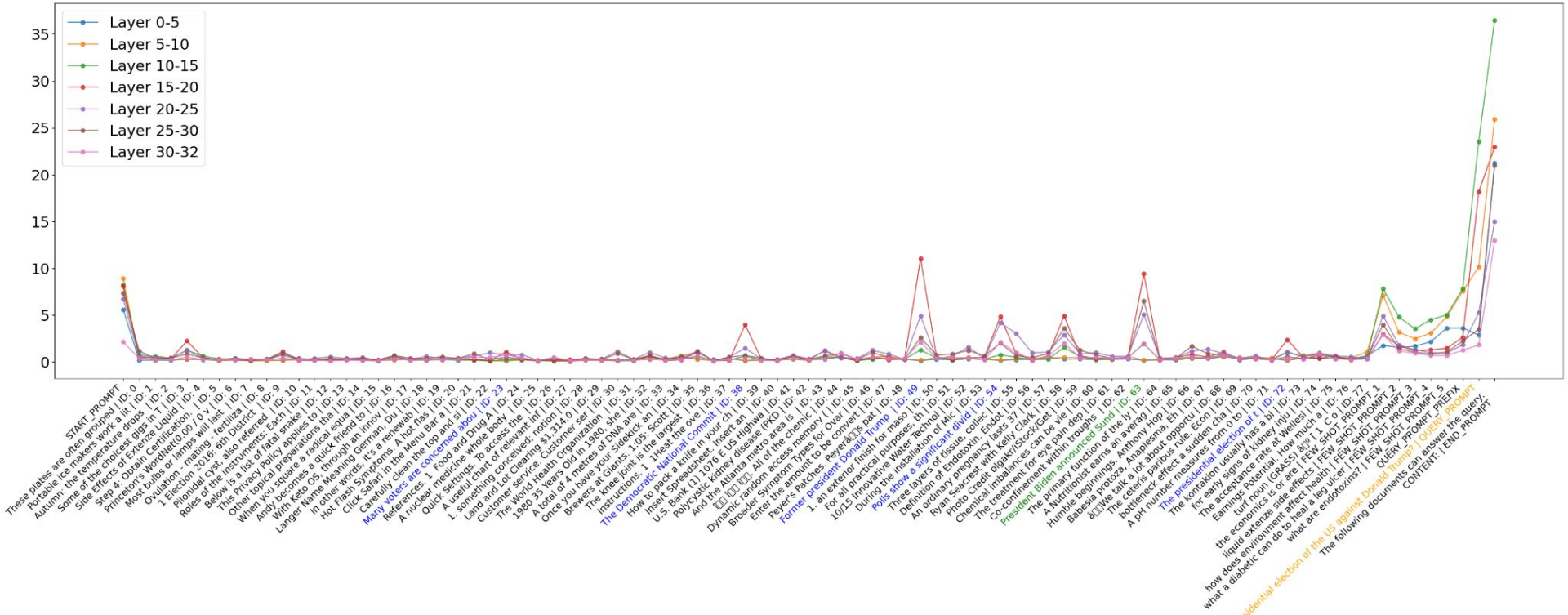
- Attention has structured sparsity
- Documents tend to attend to themself and instruction



Query-document Block Relevance



Query-document Block Relevance



Query-document Block Relevance

- Certain tokens (“:”, “[”) have strong correlation between attention concentration and relevance
- With SFT on ICR data this correlation improves significantly

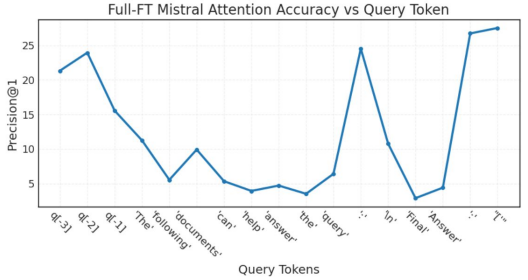


Figure 5 | Performance of Full-FT model’s attention-based inference vs the query token for which attention scores are extracted from.

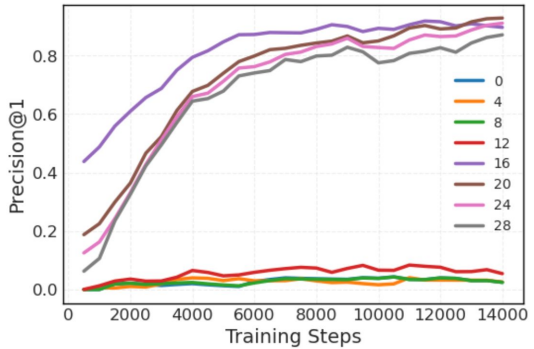


Figure 7 | Layerwise Attention Precision@1 on a held-out subset of MSMarco training data vs training steps for Full-FT model

Agenda

01 IR in the age of LLMs

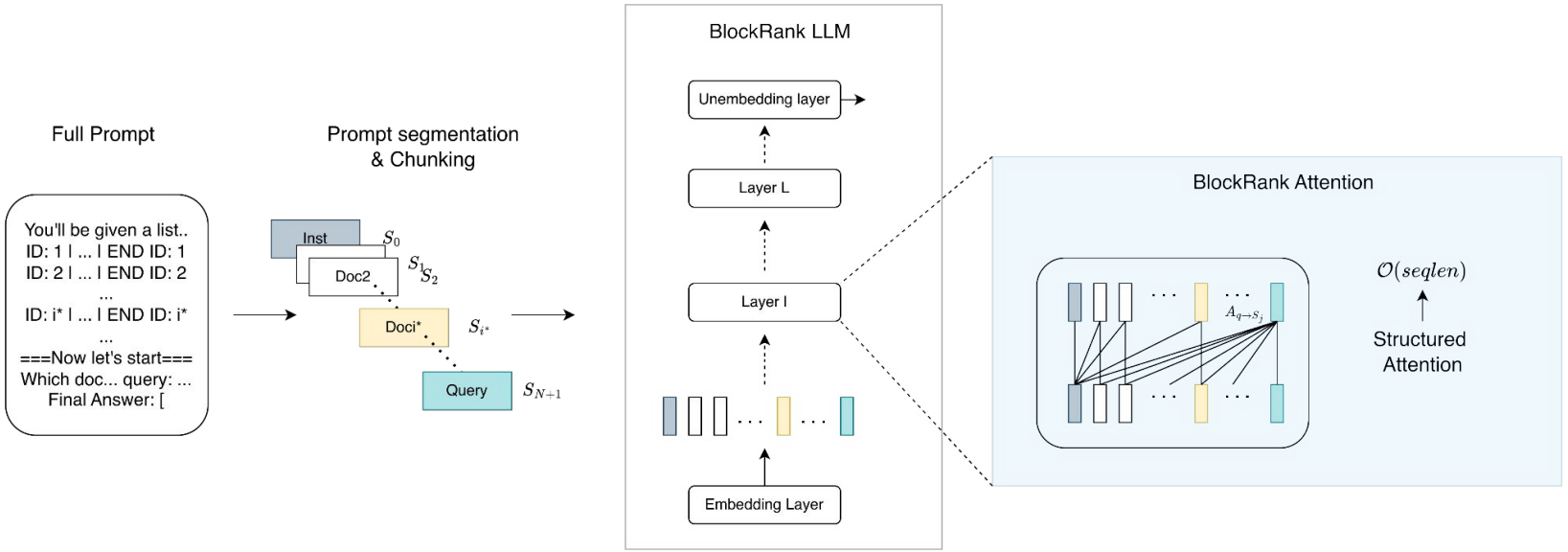
02 ICR Attention Analysis

03 BlockRank - scalable ICR

04 Future Directions

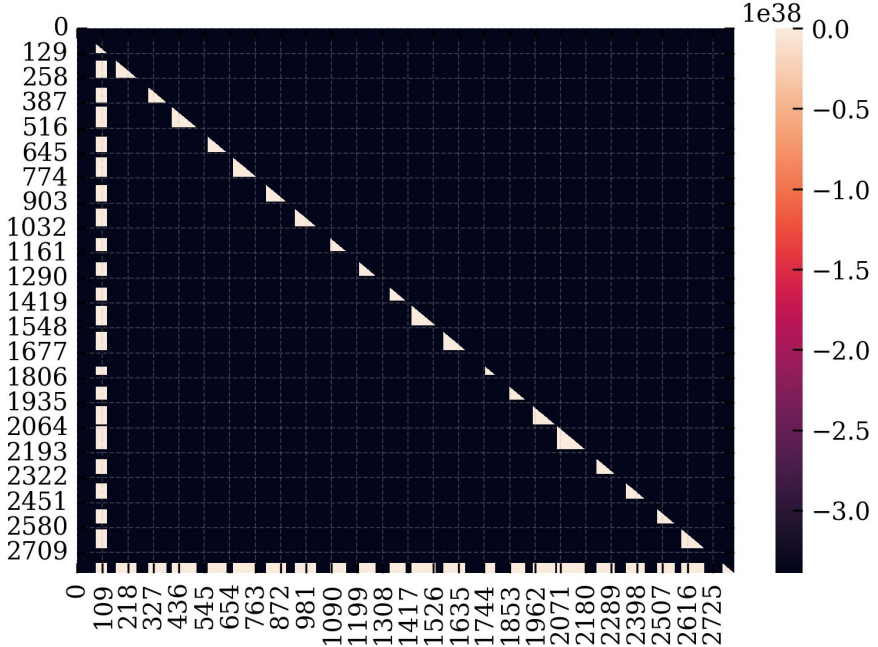
Attention Architecture

Enforce Inter-document sparsity - doc blocks only attend to self + instruction block



Attention Architecture

Resulting effective attention mask ~ triangular boundary



Permutation-invariant Position Encoding

"lost-in-the-middle" problem in ICR

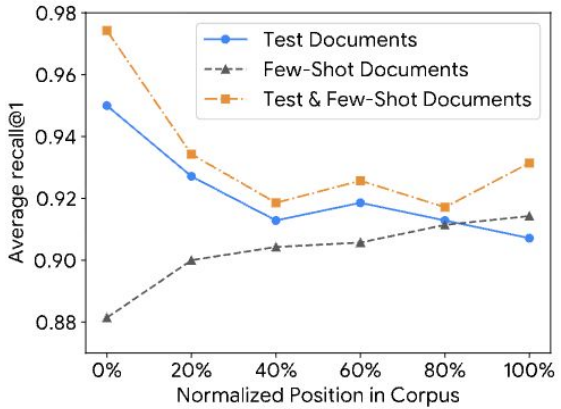
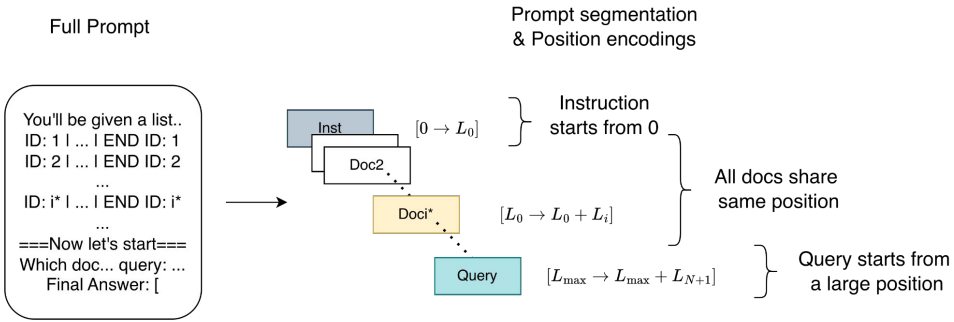


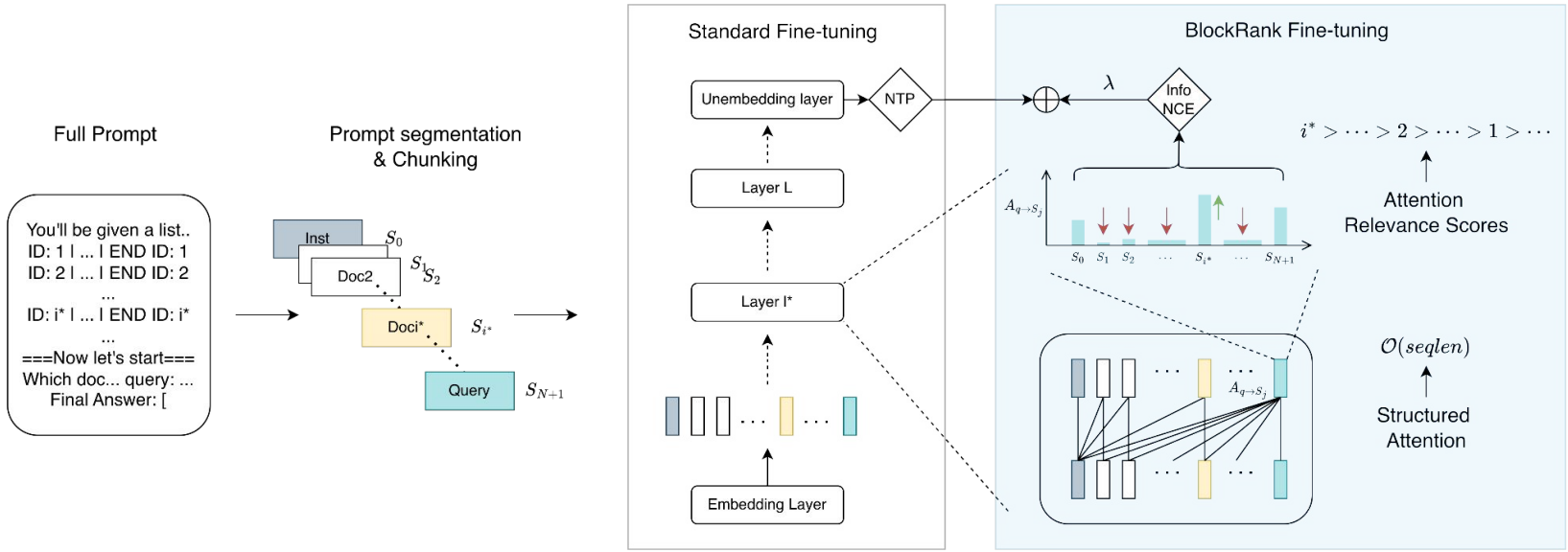
Figure 4: **Positional Analysis.** We vary gold document positions of queries within the corpus (0% = beginning, 100% = end).

BlockRank allows/employs permutation invariant position encodings



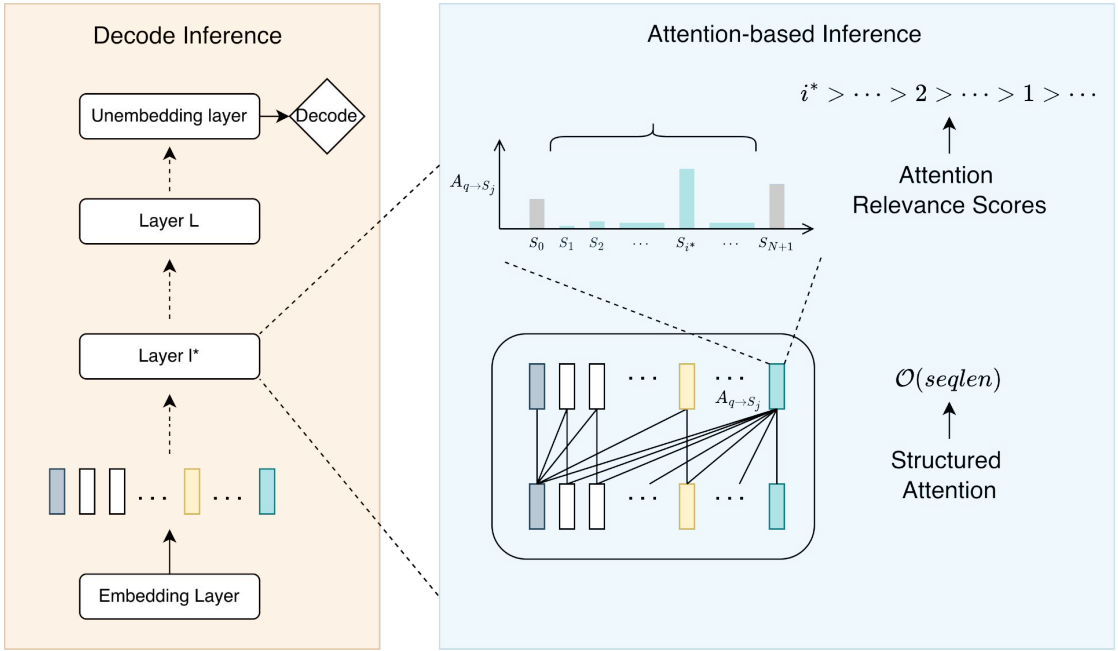
Loss

Explicitly optimize query-document block attention using a contrastive loss



Inference

Two modes of inference: (1) decode answer; (2) measure attention concentration



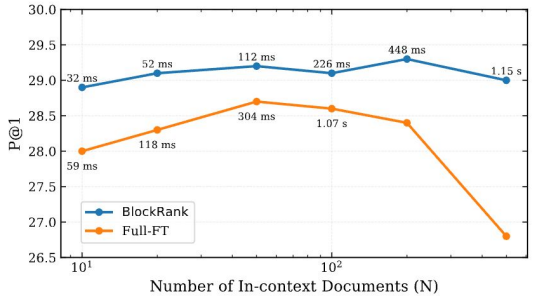
Results on BEIR

BlockRank achieves strong results while being much efficient and scalable

Table 1 | nDCG@10 on BEIR benchmark, all re-ranker rank top-100 documents retrieved from Contriever retrieval model. **Bold** indicates the best numbers.

Reranker	Train Data	Avg.	Climate-FEVER	DB-Pedia	FEVER	FiQA	Hotpot QA	MS Marco	NF-Corpus	NQ	Sci-docs	Sci-fact	Trec-COVID
None (Contriever)	MS Marco	45.9	23.7	41.3	75.8	32.9	63.8	40.7	32.8	49.8	16.5	67.7	59.6
Cross-Encoder	MS Marco	50.7	25.5	47.0	81.9	35.6	71.8	47.0	34.5	57.6	17.0	69.1	71.0
Rank Vicuna	GPT 3.5	50.7	**28.2**	50.0	81.0	35.9	73.5	36.7	33.1	58.6	18.4	70.5	71.3
Rank Zephyr	GPT 3.5+4	53.7	25.6	50.0	80.1	42.2	71.6	42.7	**37.7**	65.6	**20.5**	**76.7**	78.4
FIRST	GPT-4	54.3	26.7	**50.9**	81.7	42.2	74.2	44.4	37.4	**66.4**	20.4	74.6	**78.8**
BlockRank Mistral	MS Marco	**54.8**	26.8	49.7	**87.3**	**44.9**	**75.5**	**48.6**	36.6	62.4	18.7	**76.5**	76.2

Figure 4 | P@1 and Latency (annotated) of BlockRank vs Full-FT Mistral, scaling N on MSMarco.



Ablation

- Block sparse attention doesn't hurt decoding performance
- Contrastive loss important for attention-based inference
- Attention-based inference is better for inferring multiple predictions

Table 3 | Impact of training loss on Attention-based (Attn) and Decoding (Decode) Inference.

Training Configuration	Precision@1	
	Decode	Attn
Full-FT	28.7	27.6
Full-FT (w/ aux)	28.7	28.1
BlockRank (w/o ntp)	15.8	28.6
BlockRank (w/o aux)	28.4	27.8
BlockRank (full)	**28.7**	**29.1**

Table 4 | Ablation: Inference Method Effectiveness & Latency (MSMarco, N=50).

Model	Inference Method	P@1	MRR@10
Full-FT	Decode	28.7	38.4
Full-FT	Attn	27.6	38.8
BlockRank	Decode	28.7	40.0
BlockRank	**Attn**	**29.1**	**42.0**

Agenda

01 IR in the age of LLMs

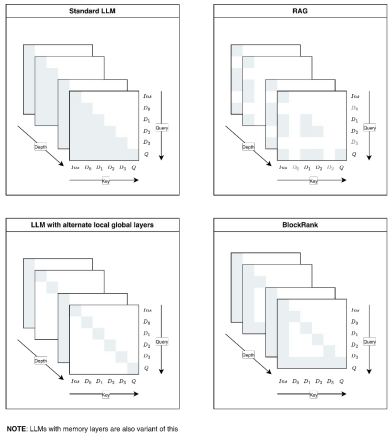
02 ICR Attention Analysis

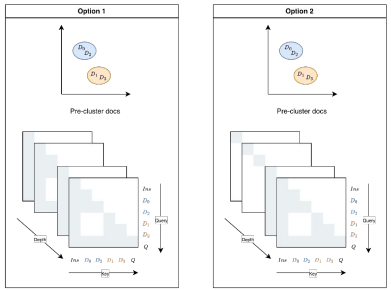
03 BlockRank - scalable ICR

04 **Future Directions**

Better Modeling?

- Clustered Block Attention
- Memory Layers
- Landmark Attention





Pre/mid-training tasks for ICR?

Can we structure web pre-training data in a way that improve ICR quality?

A progressive end-to-end approach?

Use initial few layers to do retrieval, next few layers to do ranking and response generation while reusing the intermediate representations of the retrieval layers.

Thank You!

Arxiv - 2510.0596,

Github - nilesh2797/BlockRank

Reach out - nilesh@cs.utexas.edu