R²LLMs: Retrieval and Ranking with LLMs

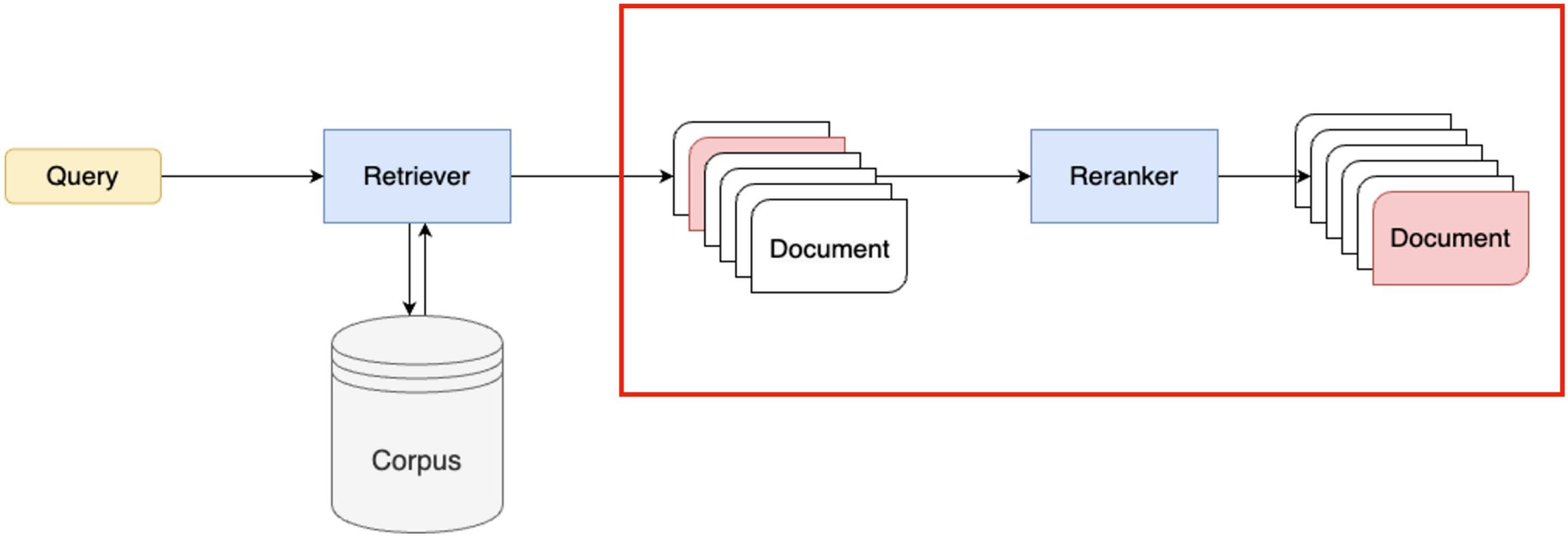
Guido Zuccon[1], **Shengyao** Zhuang[2], **Xueguang** Ma[3], **Bevan** Koopman[1,2]

[1] ielab, The University of Queensland, Australia & Google Research Australia
[2] ielab, CSIRO, Australia
[3] The University of Waterloo

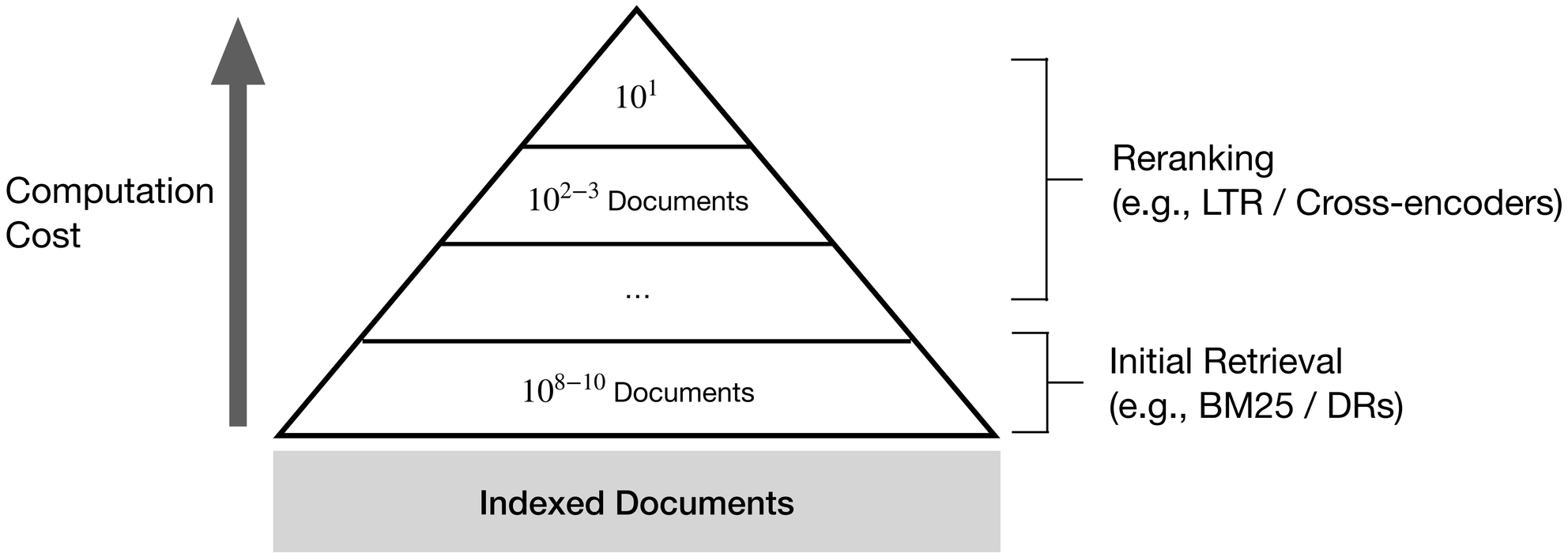
https://ielab.io/tutorials/r2llms.html

Part 3: LLM-based Rankers

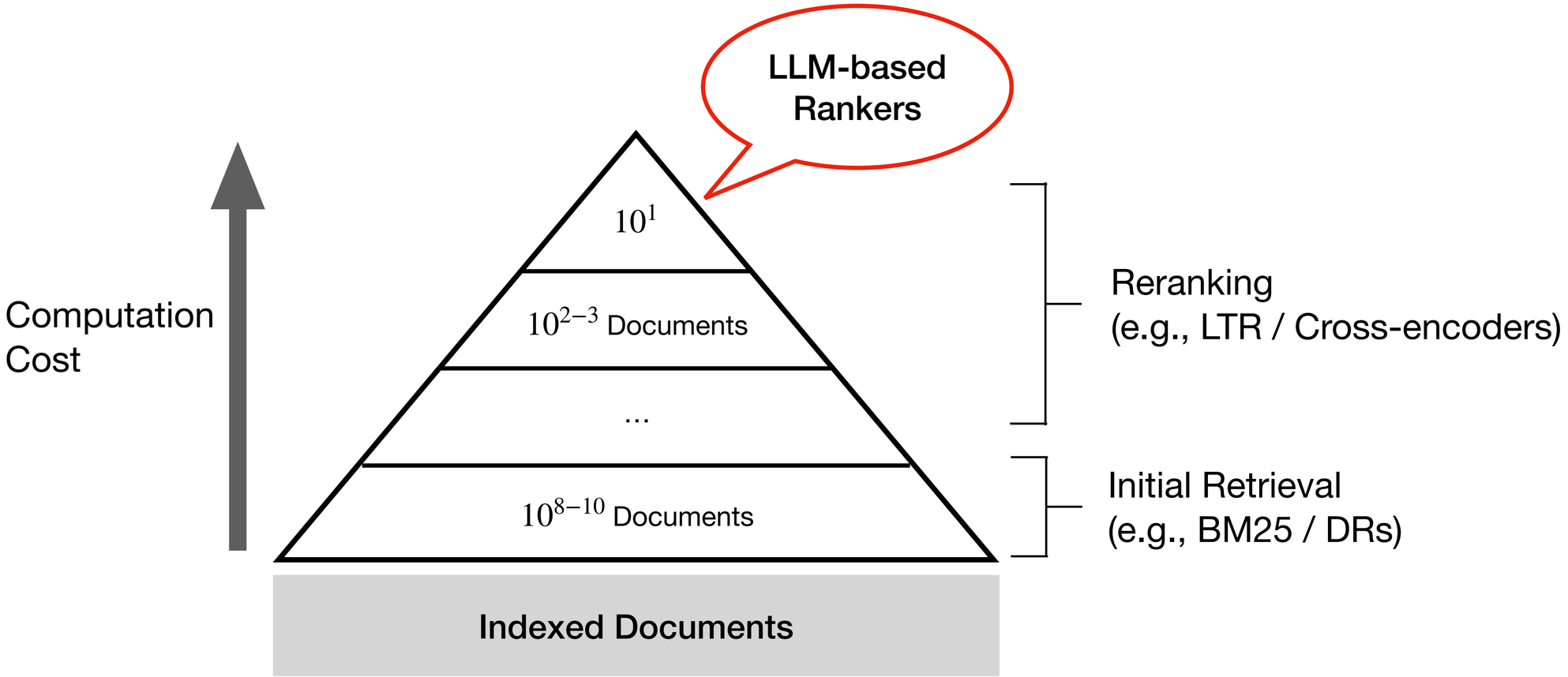
PART 3: LLM-based Rankers



PART 3: LLM-based Rankers

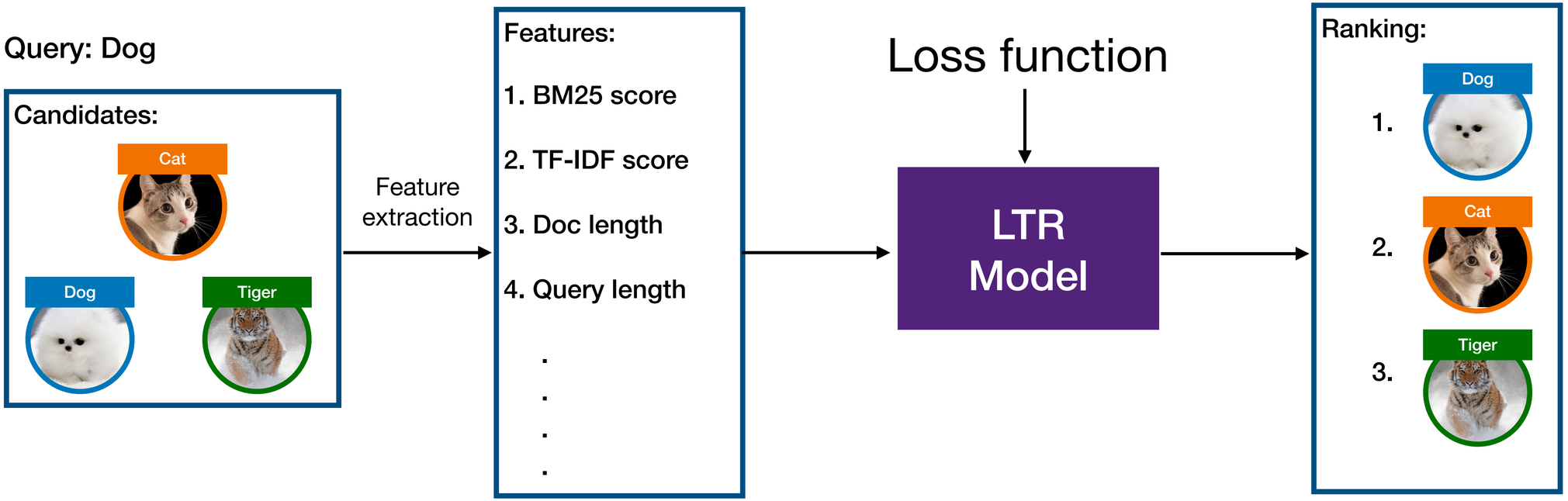


PART 3: LLM-based Rankers



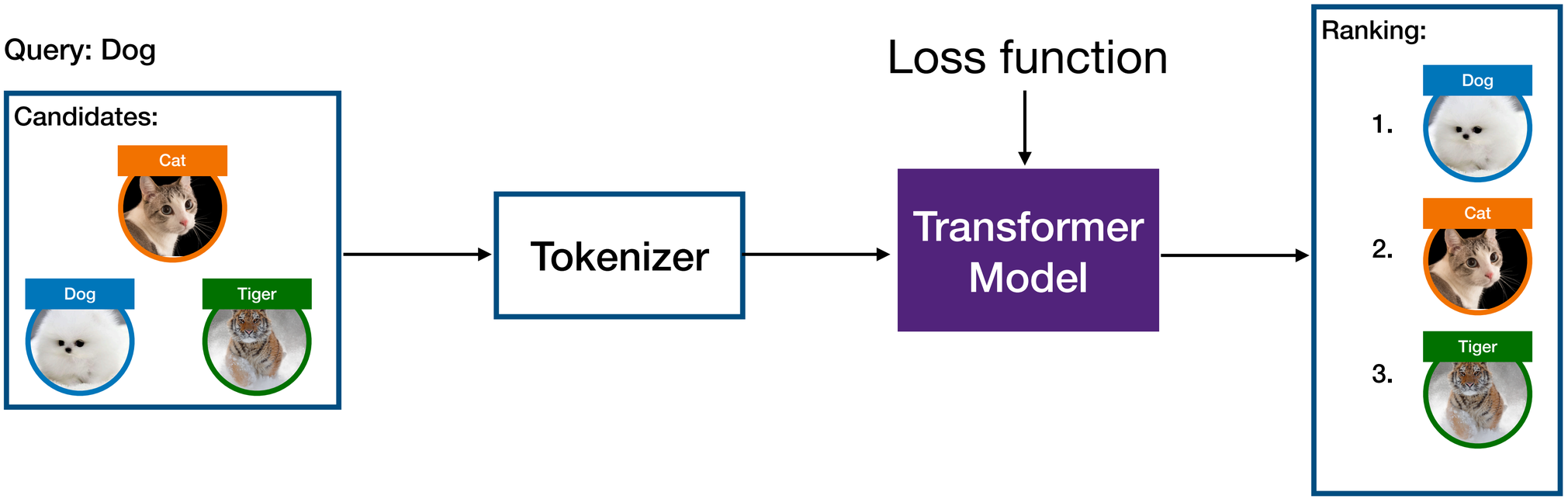
Recap of LTR and Cross-encoders

- Learning to Rank (LTR):



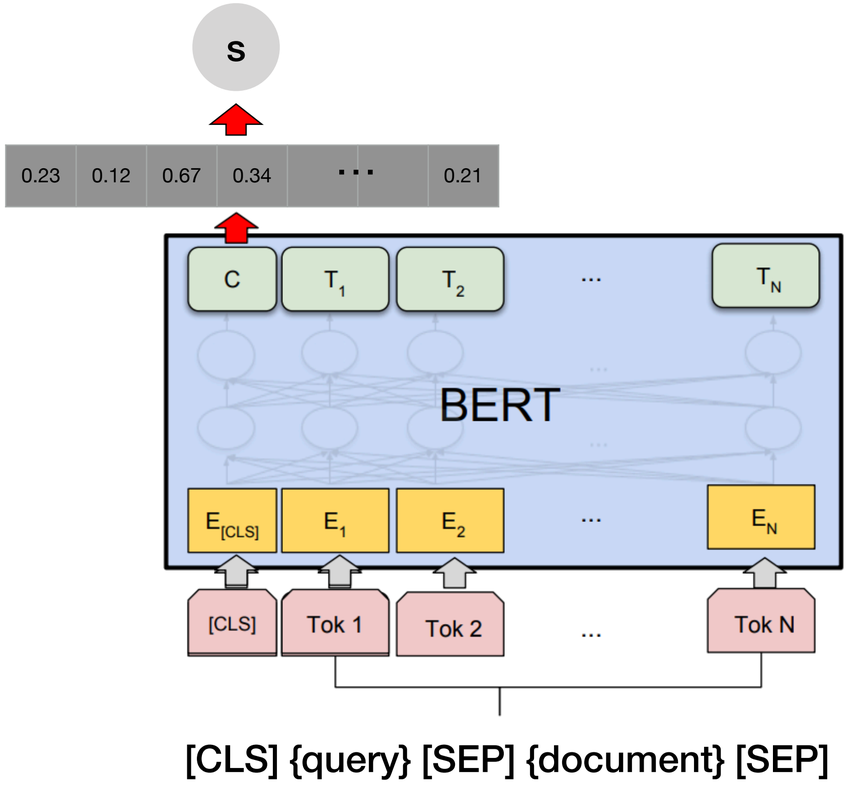
Recap of LTR and Cross-encoders

- Transformer ranker



Recap of LTR and Cross-encoders

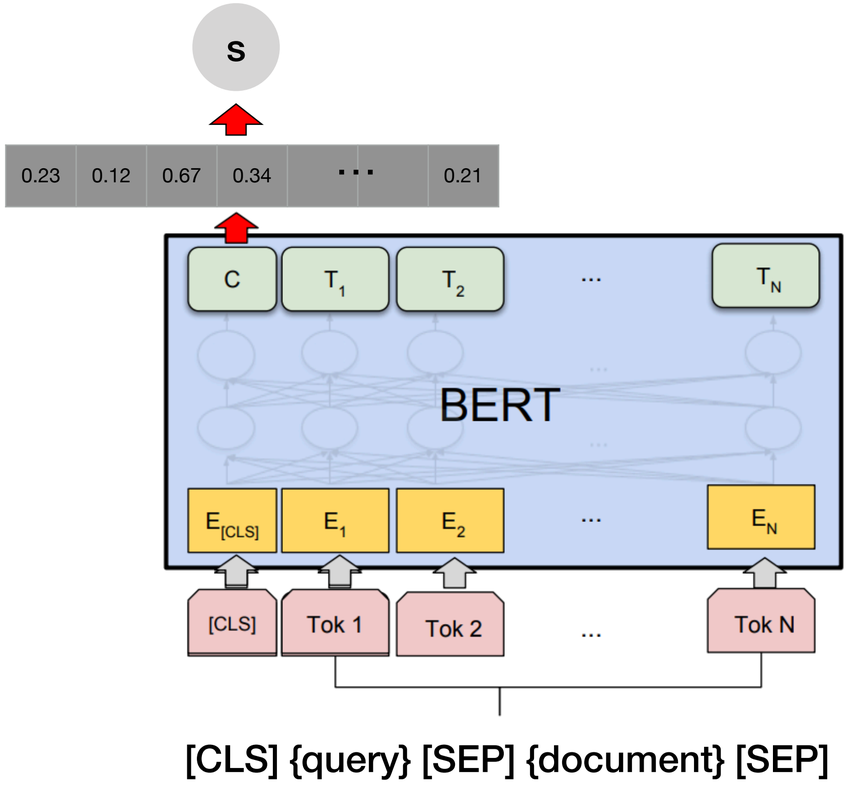
- monoBERT: cross-encoder architecture



Nogueira, R., Yang, W., Cho, K., Lin, J., 2019. Multi-Stage Document Ranking with BERT. *arXiv preprint*

Recap of LTR and Cross-encoders

- monoBERT: cross-encoder architecture

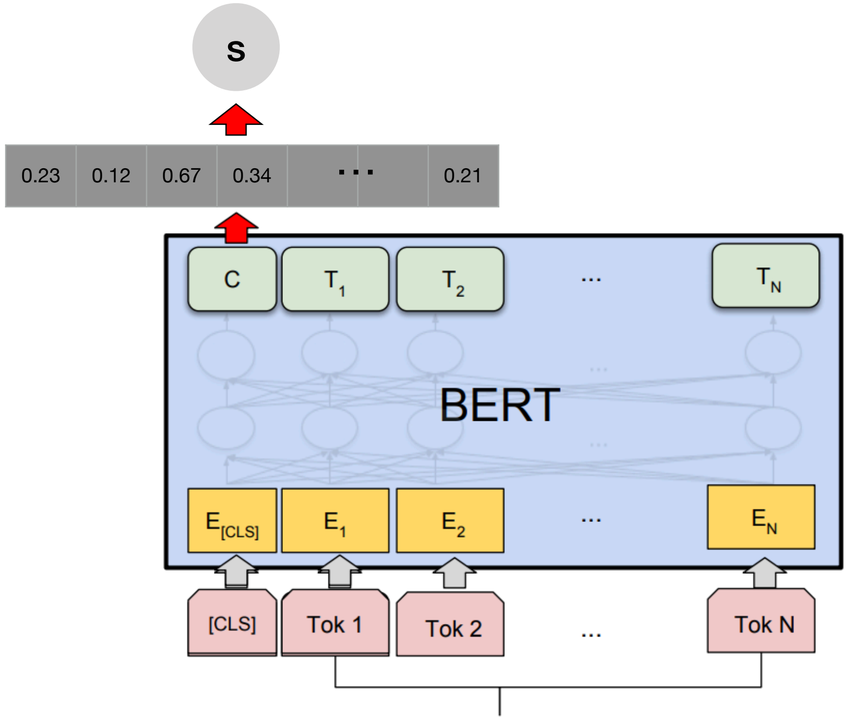


LLM-based cross-encoder?

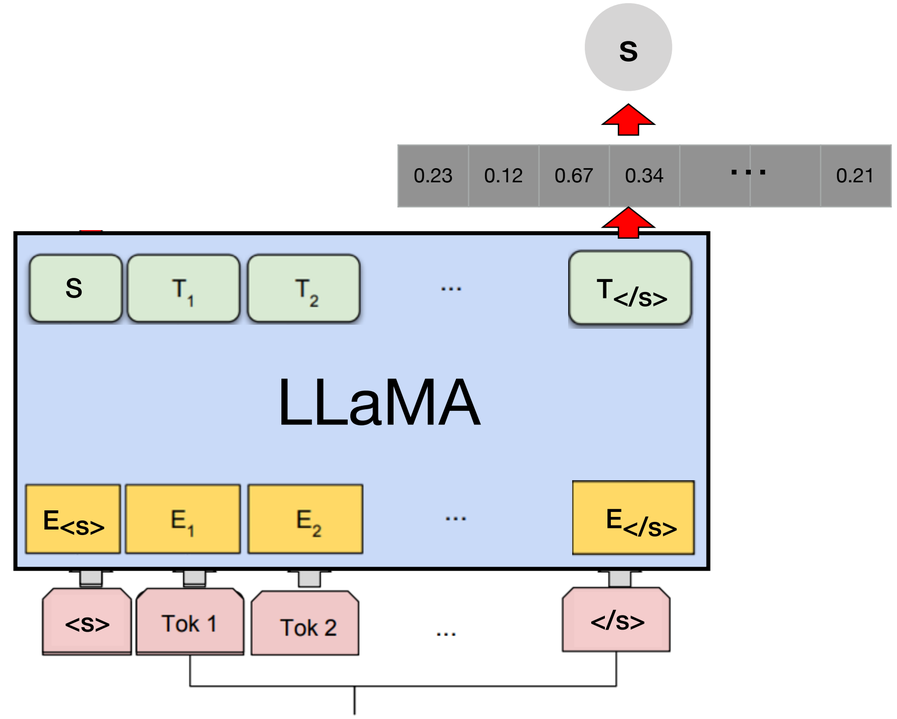
Nogueira, R., Yang, W., Cho, K., Lin, J., 2019. Multi-Stage Document Ranking with BERT. *arXiv preprint*

LLM-based Cross-encoder

- RankLLaMA



[CLS] {query} [SEP] {document} [SEP]



query: {query} document: {document} </s>

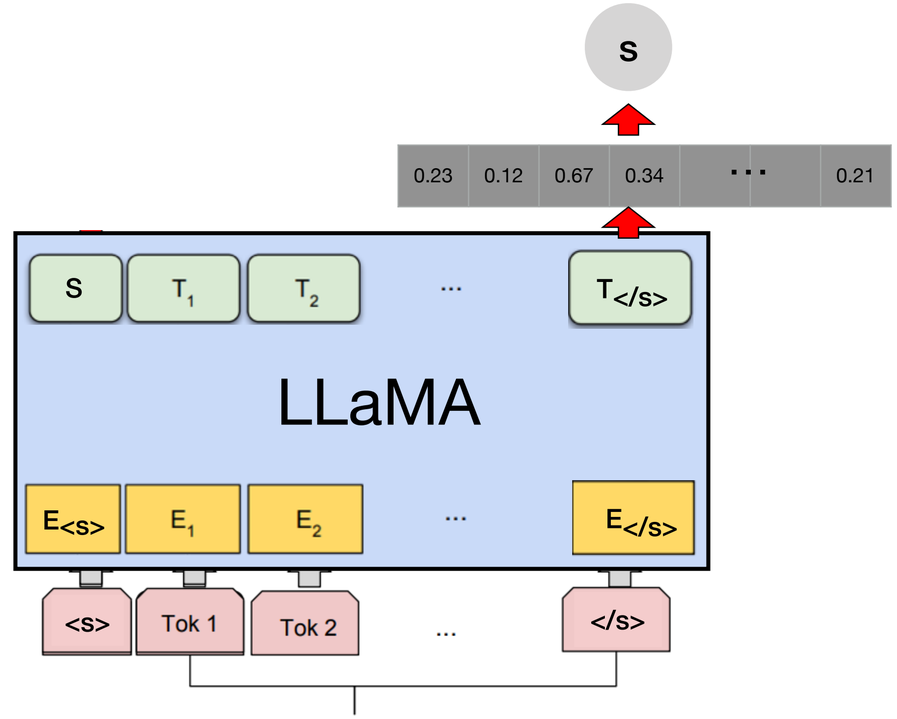
Ma, X., Wang, L., Yang, N., Lin, J., 2023. Fine-Tuning LLaMA for Multi-Stage Text Retrieval. *SIGIR*

LLM-based Cross-encoder

- RankLLaMA

$$\text{input} = \text{‘query: } \{Q\} \text{ document: } \{D\}\text{</s>’}$$
$$\text{Sim}(Q, D) = \text{Linear}(\text{Decoder}(\text{input})[-1])$$

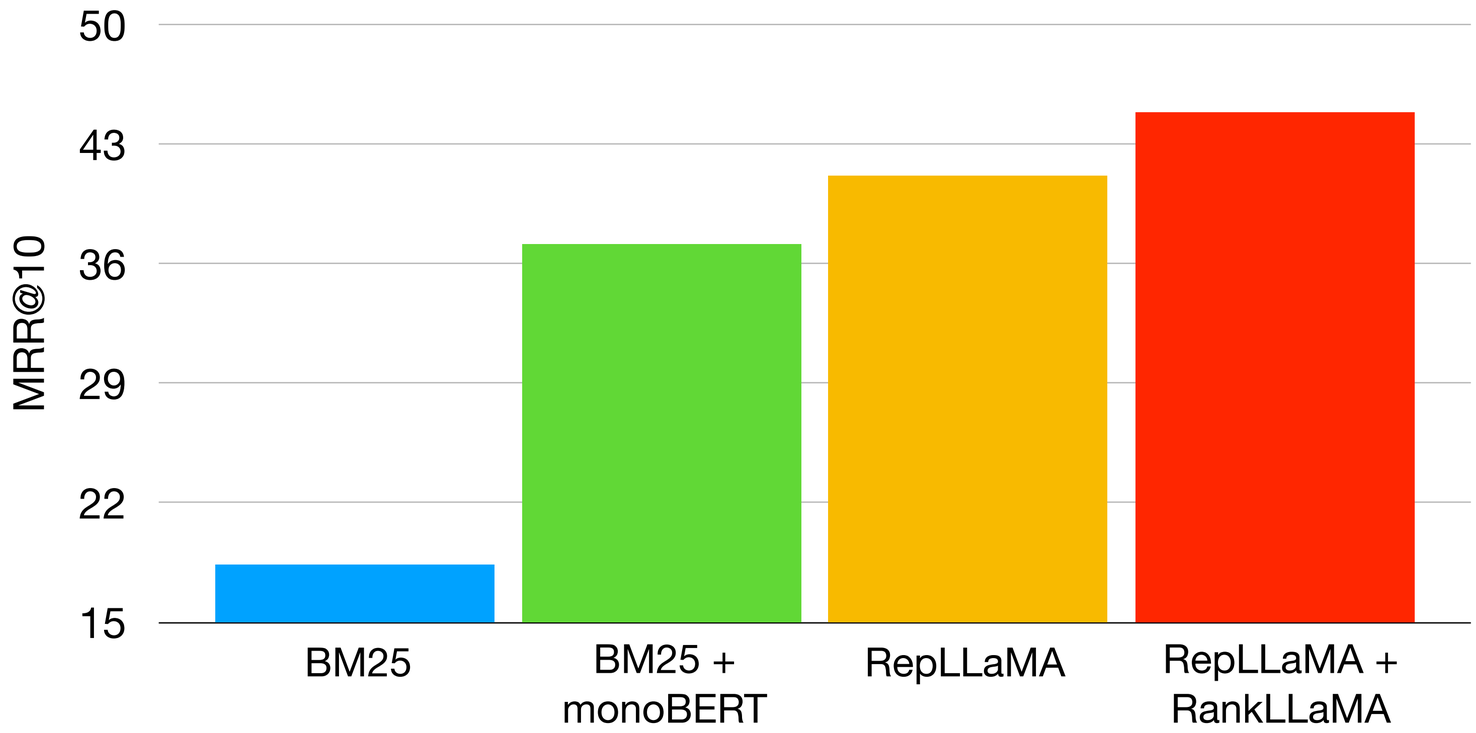
$$\mathcal{L}(Q, D^+, \{D_\text{N}\}) = -\log p(D = D^+ \mid Q) =$$
$$-\log \frac{\exp(\text{Sim}(Q, D^+))}{\exp(\text{Sim}(Q, D^+)) + \sum_{D_i^- \in \{D_\text{N}\}} \exp(\text{Sim}(Q, D_i^-))}$$



Ma, X., Wang, L., Yang, N., Lin, J., 2023. Fine-Tuning LLaMA for Multi-Stage Text Retrieval. *SIGIR*

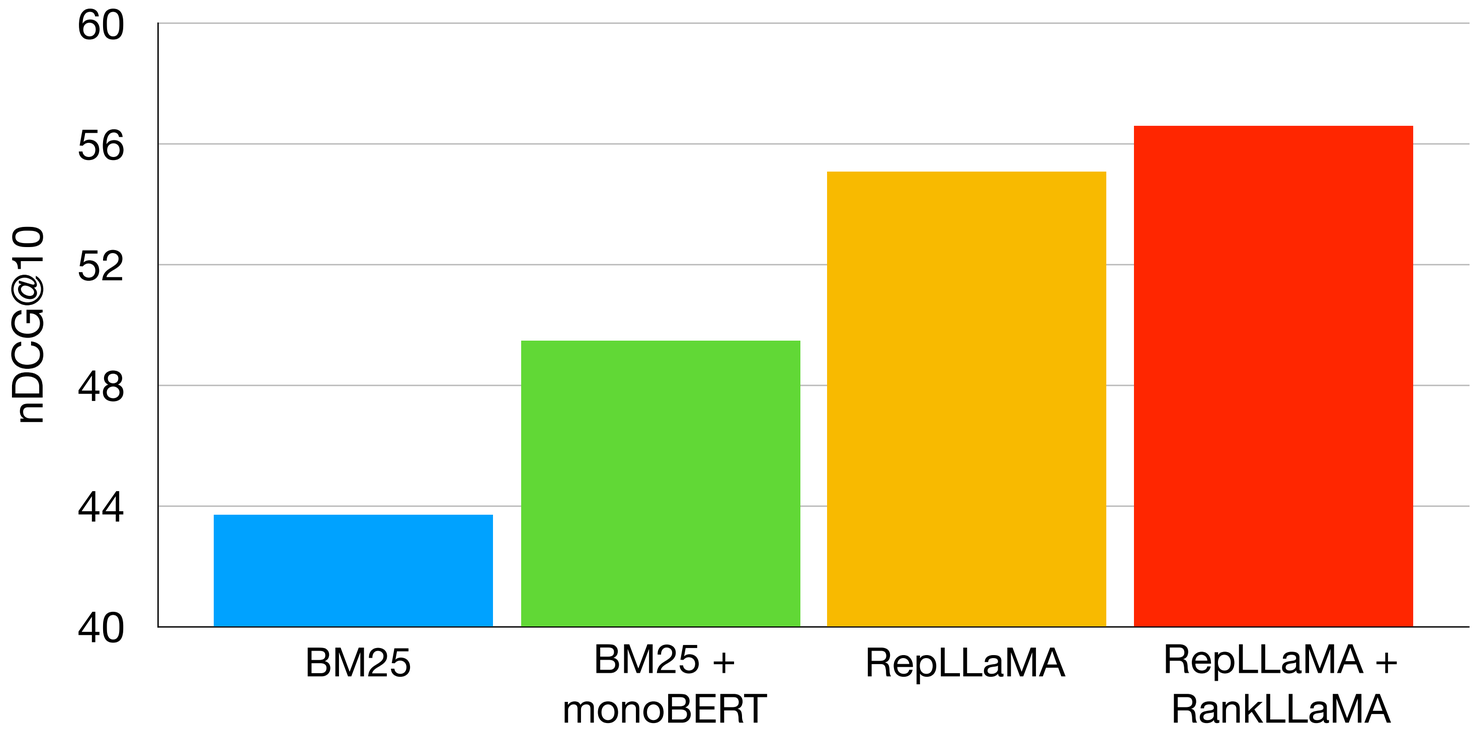
RankLLaMA Results

- In-domain results: MSMARCO



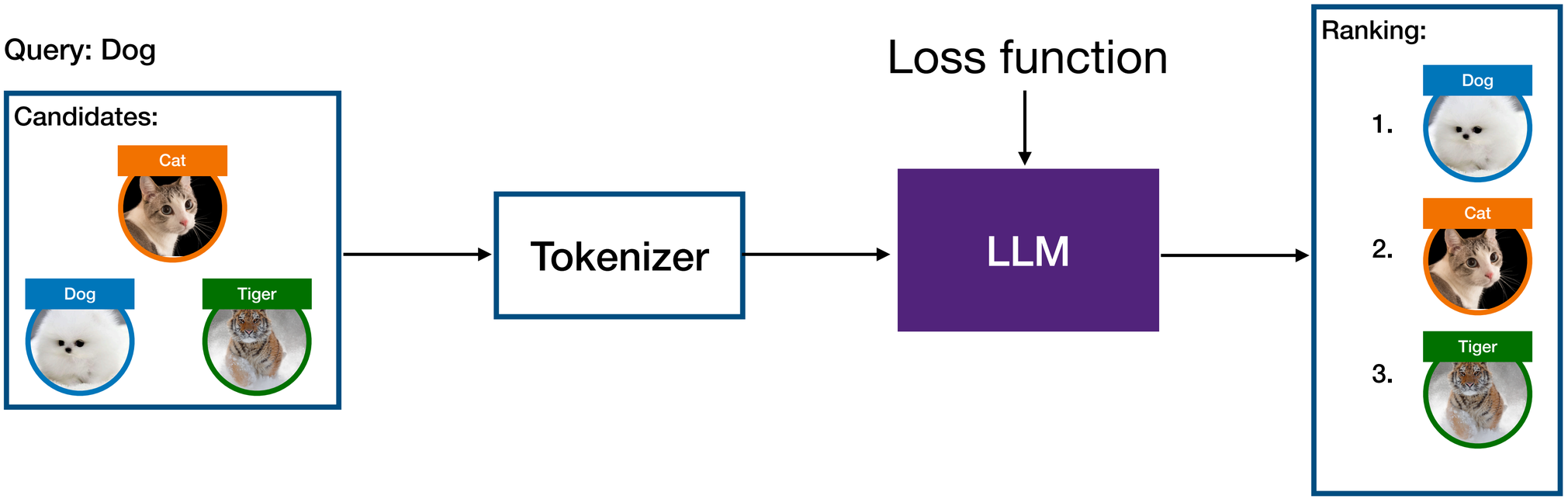
RankLLaMA Results

- Out-of-domain results: BEIR



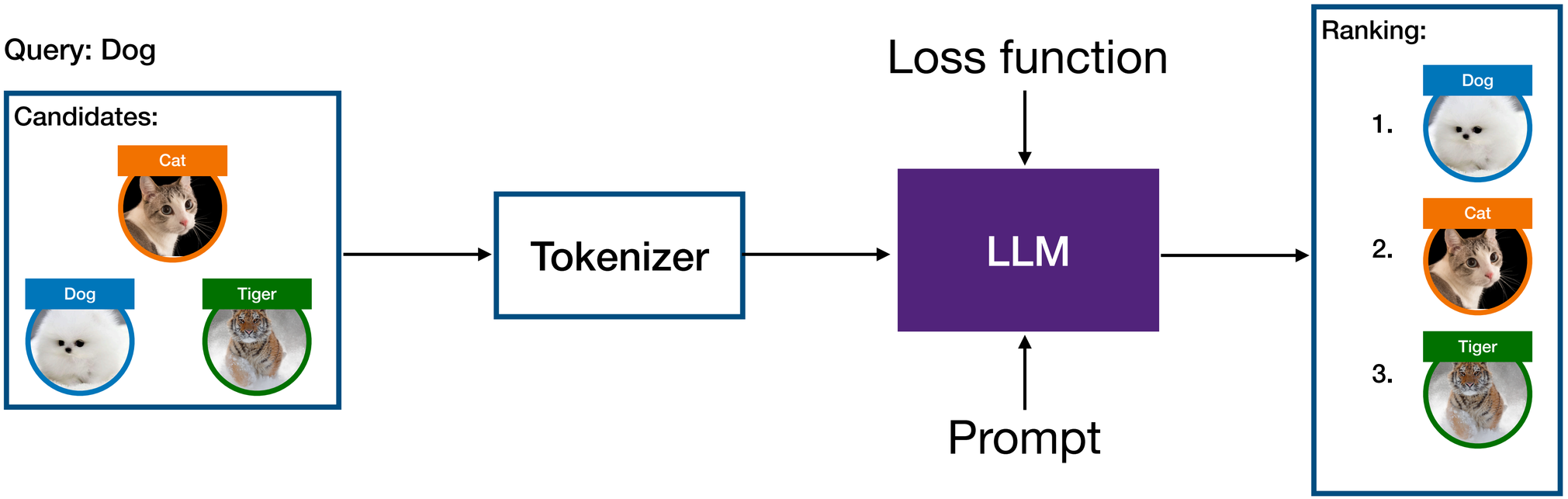
LLM-based Cross-encoder

- LLM-based ranker

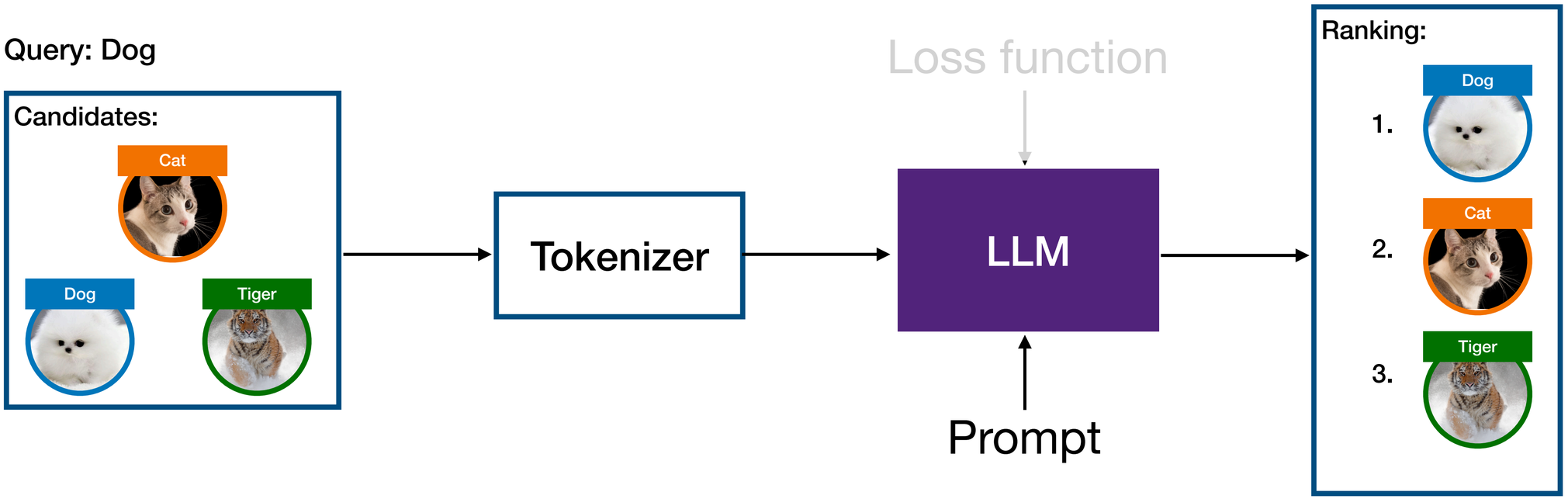


LLM-based Cross-encoder

- LLM-based ranker



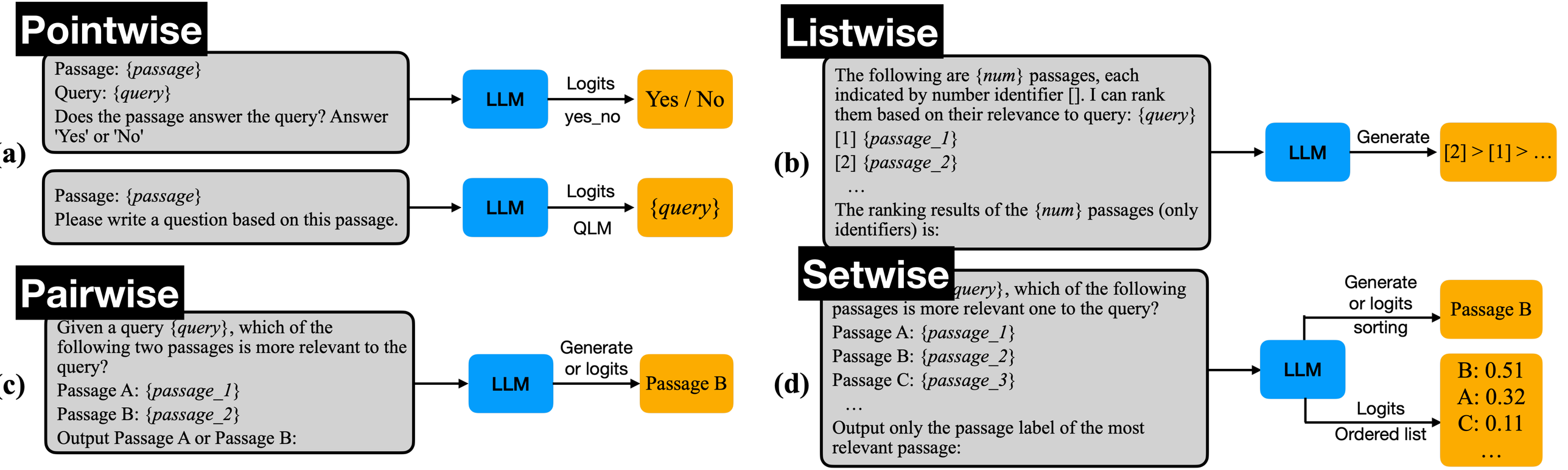
Zeroshot LLM-based Rankers (LLM as Rankers)



- All are “zero-shot”: i.e. once you obtained the pre-trained, instruction tuned LLM, no need to do further post-training.

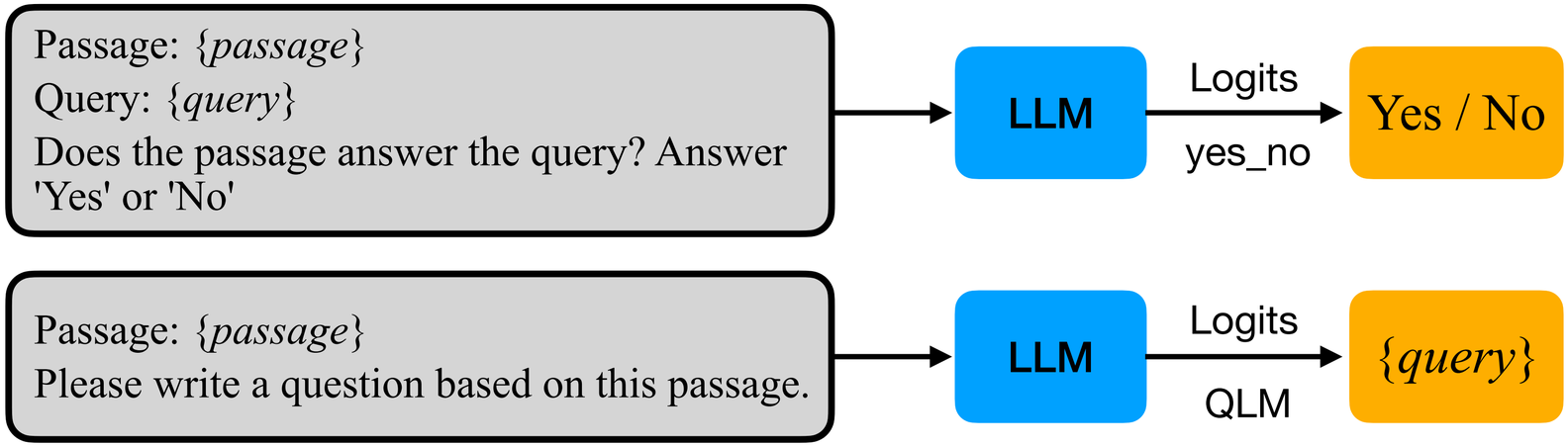
- Input could be one or multiple documents, depending on the prompt.

Zeroshot LLM-based Rankers: 3.5 types



Zhuang, S., Zhuang, H., Koopman, B. and Zuccon, G., 2023. A Setwise Approach for Effective and Highly Efficient Zero-shot Ranking with Large Language Models. *SIGIR*

Pointwise



- Yes / No label generation.
- Query generation likelihood.

Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., Kumar, A. and Newman, B., 2022. Holistic evaluation of language models. *arXiv.*
Sachan, D.S., Lewis, M., Joshi, M., Aghajanyan, A., Yih, W.T., Pineau, J. and Zettlemoyer, L., 2022. Improving passage retrieval with zero-shot question generation. *EMNLP.*
Zhuang, S., Liu, B., Koopman, B., Zuccon, G., 2023. Open-source Large Language Models are Strong Zero-shot Query Likelihood Models for Document Ranking. *EMNLP*.

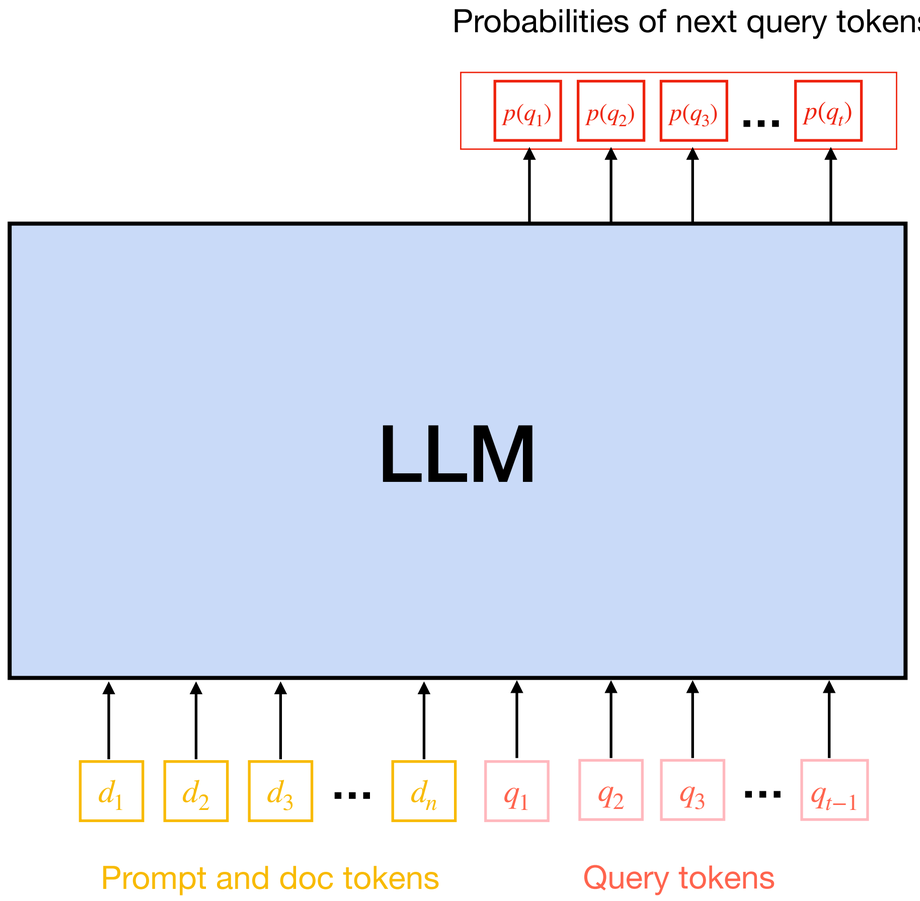
Pointwise

- Query generation likelihood.

Passage: {*passage*}
Please write a question based on this passage.

Rank by query likelihood:

$$P(Q \mid D) = \sum_{i}^{t} \log p(q_i)$$



Sachan, D.S., Lewis, M., Joshi, M., Aghajanyan, A., Yih, W.T., Pineau, J. and Zettlemoyer, L., 2022. Improving passage retrieval with zero-shot question generation. *EMNLP.*
Zhuang, S., Liu, B., Koopman, B., Zuccon, G., 2023. Open-source Large Language Models are Strong Zero-shot Query Likelihood Models for Document Ranking. *EMNLP*.

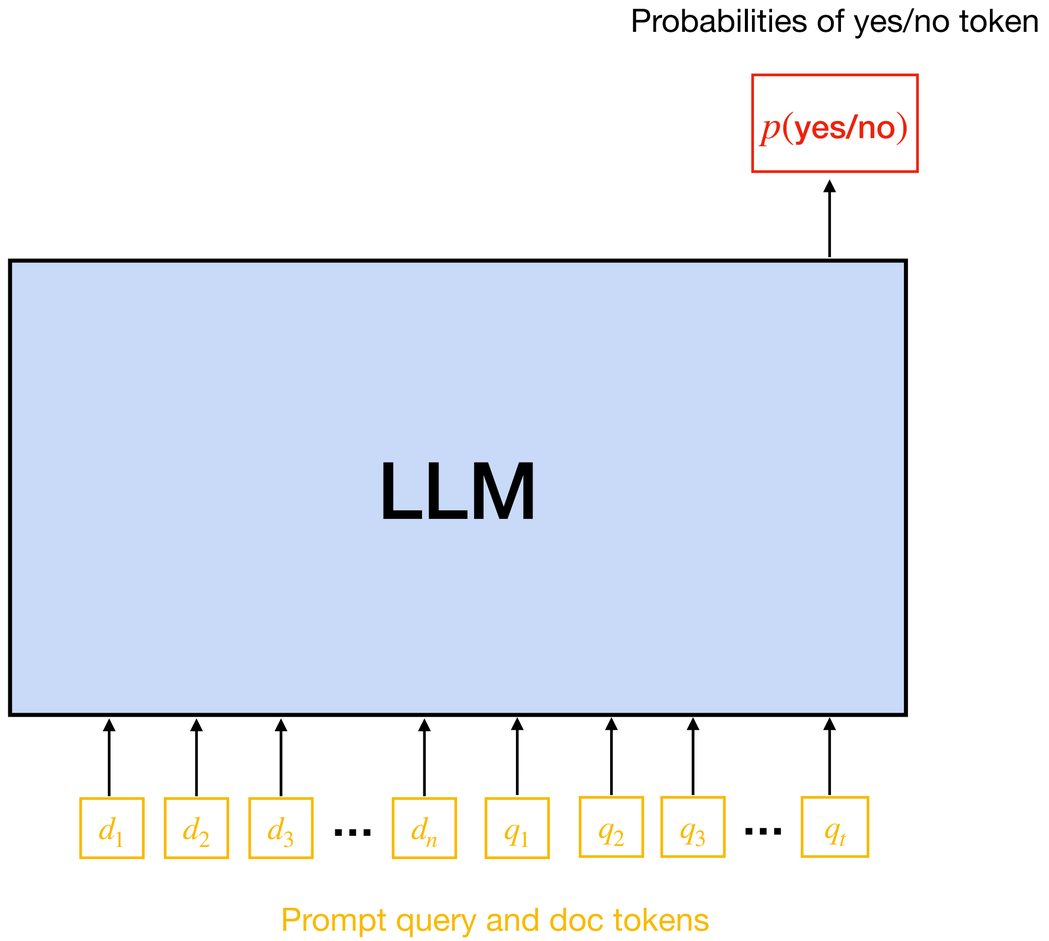
Pointwise

- Yes / No label generation.

Passage: {*passage*}
Query: {*query*}
Does the passage answer the query? Answer 'Yes' or 'No'

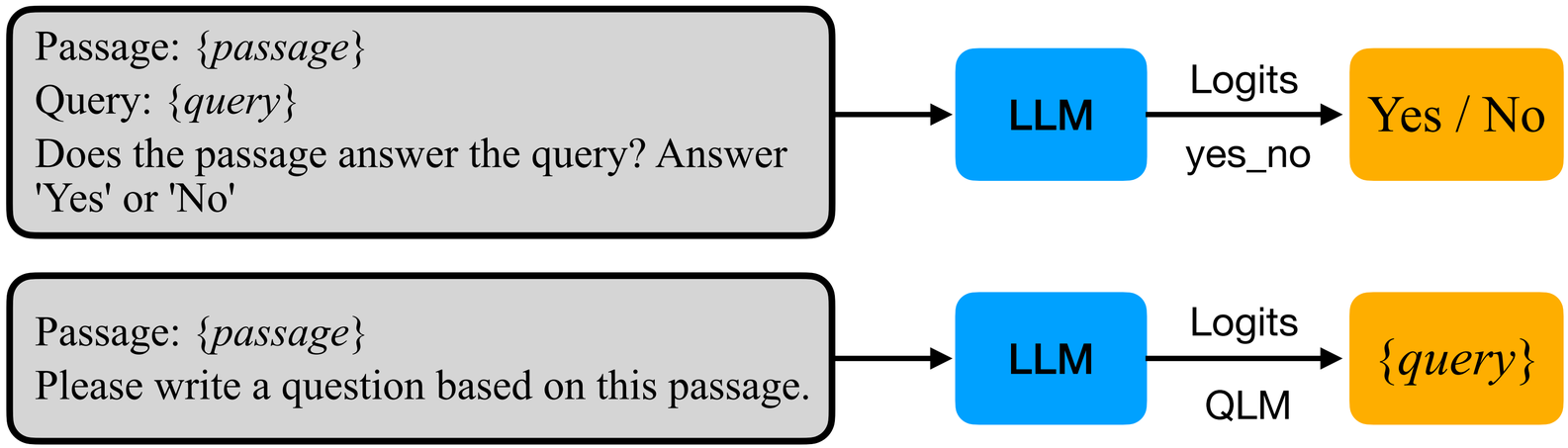
Rank by normalised yes probability:

$$P(\textbf{yes} \mid Q, D) = \frac{\log p(\textbf{yes})}{\log p(\textbf{yes}) + \log p(\textbf{no})}$$



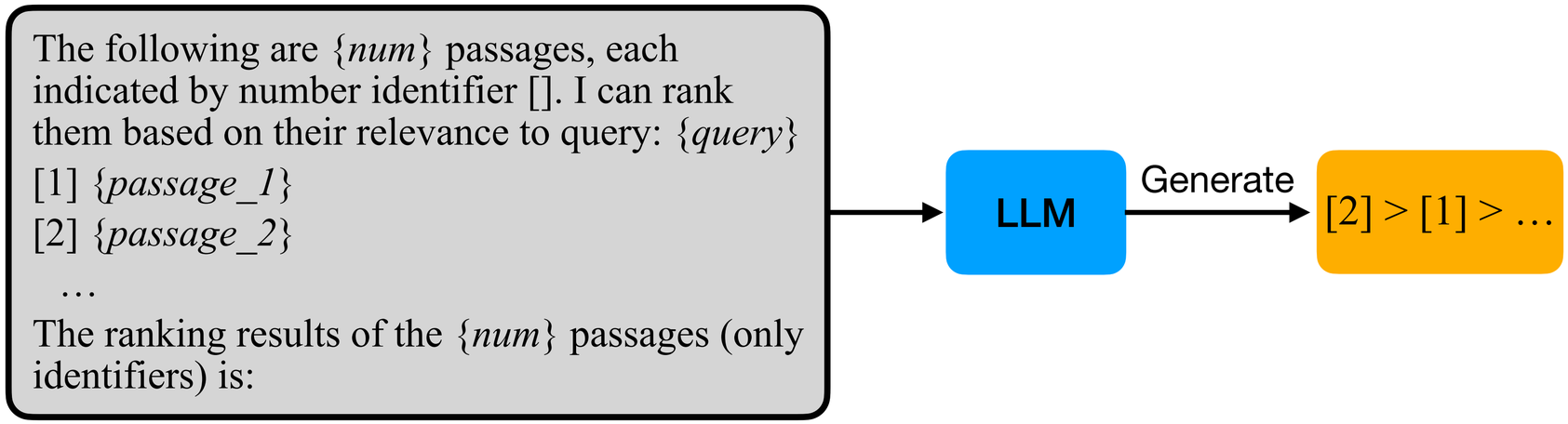
Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., Kumar, A. and Newman, B., 2022. Holistic evaluation of language models. *arXiv.*

Pointwise



- Each query-doc pair can be processed in parallel.
- It needs access to the model weights (need token logits).

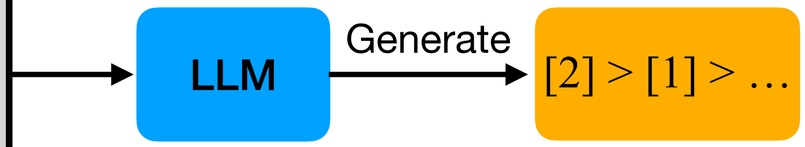
Listwise



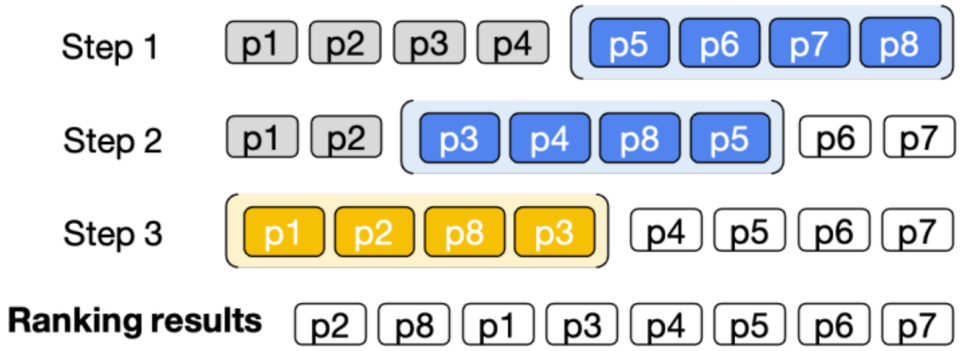
Ma, X., Zhang, X., Pradeep, R. and Lin, J., 2023. Zero-Shot Listwise Document Reranking with a Large Language Model. *arXiv*.
Pradeep, R., Sharifymoghaddam, S. and Lin, J., 2023. RankVicuna: Zero-Shot Listwise Document Reranking with Open-Source Large Language Models. *arXiv*.
Sun, W., Yan, L., Ma, X., Ren, P., Yin, D. and Ren, Z., 2023. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agent. *EMNLP.*

Listwise

The following are {*num*} passages, each indicated by number identifier []. I can rank them based on their relevance to query: {*query*}
[1] {*passage_1*}
[2] {*passage_2*}
…
The ranking results of the {*num*} passages (only identifiers) is:

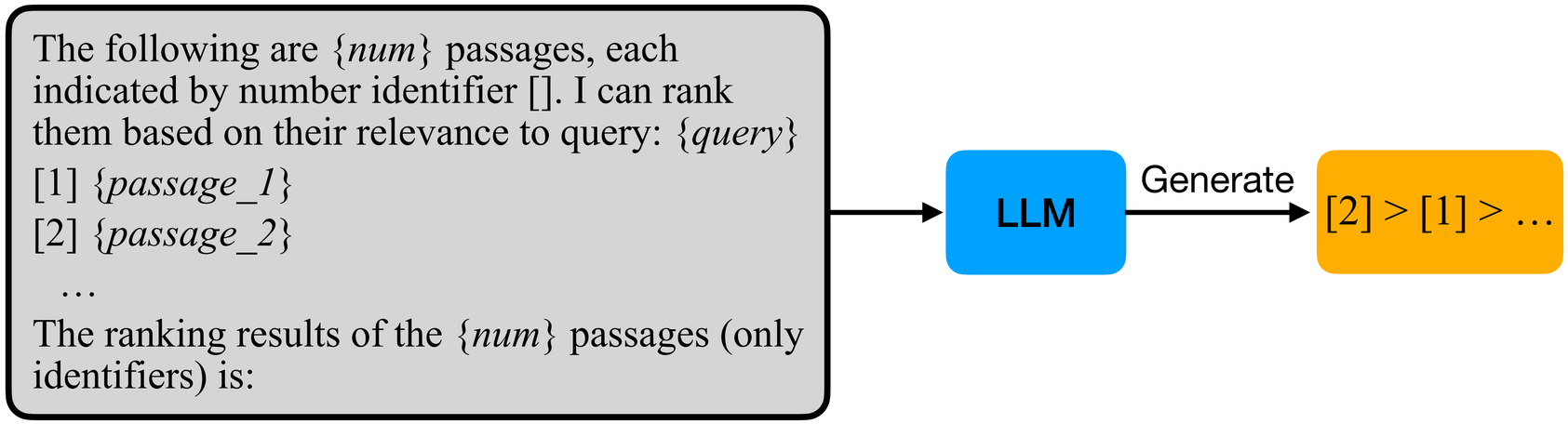


- Sliding window



Ma, X., Zhang, X., Pradeep, R. and Lin, J., 2023. Zero-Shot Listwise Document Reranking with a Large Language Model. *arXiv*.
Pradeep, R., Sharifymoghaddam, S. and Lin, J., 2023. RankVicuna: Zero-Shot Listwise Document Reranking with Open-Source Large Language Models. *arXiv*.
Sun, W., Yan, L., Ma, X., Ren, P., Yin, D. and Ren, Z., 2023. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agent. *EMNLP.*

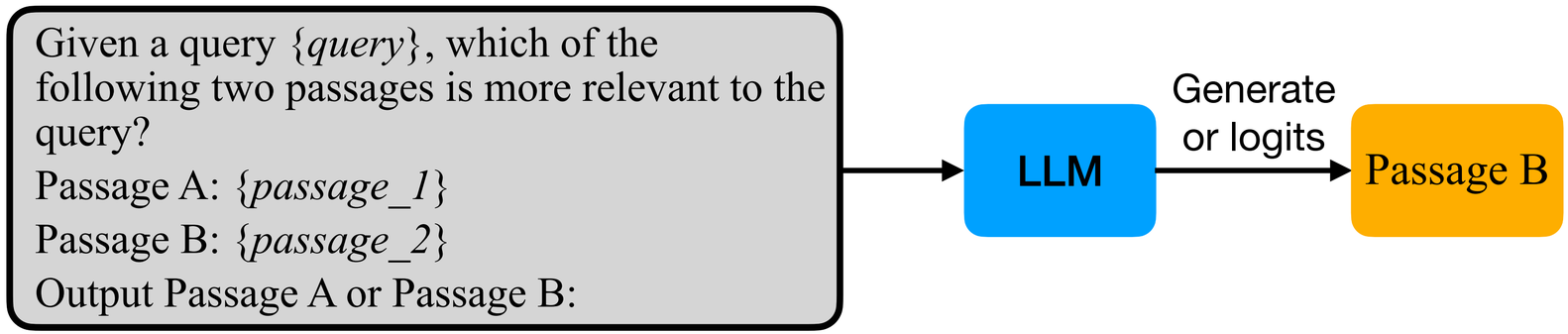
Listwise



- Cannot parallel.
- No need access to the model weights, pure generation.

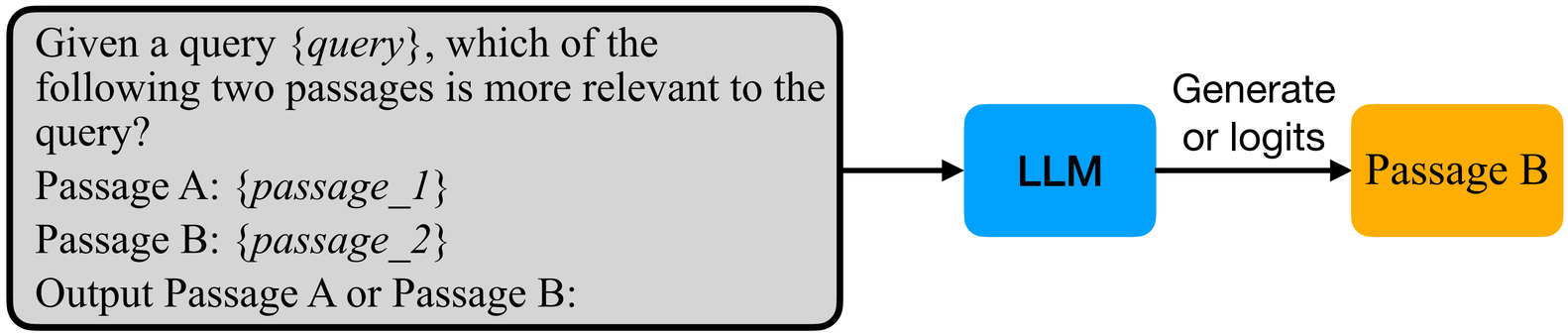
Ma, X., Zhang, X., Pradeep, R. and Lin, J., 2023. Zero-Shot Listwise Document Reranking with a Large Language Model. *arXiv*.
Pradeep, R., Sharifymoghaddam, S. and Lin, J., 2023. RankVicuna: Zero-Shot Listwise Document Reranking with Open-Source Large Language Models. *arXiv*.
Sun, W., Yan, L., Ma, X., Ren, P., Yin, D. and Ren, Z., 2023. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agent. *EMNLP.*

Pairwise



Qin, Z., Jagerman, R., Hui, K., Zhuang, H., Wu, J., Shen, J., Liu, T., Liu, J., Metzler, D., Wang, X. and Bendersky, M., 2023. Large language models are effective text rankers with pairwise ranking prompting. *NAACL-findings*.

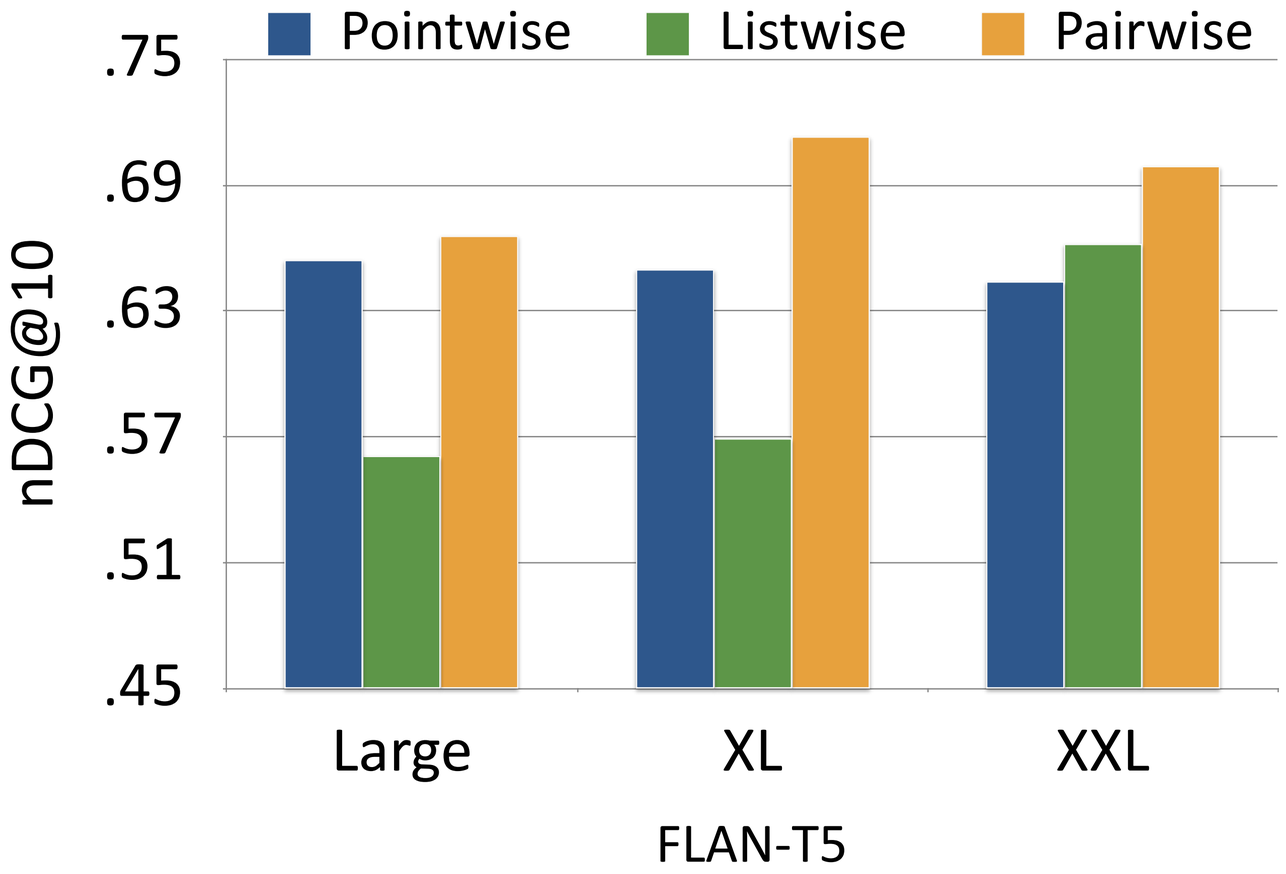
Pairwise



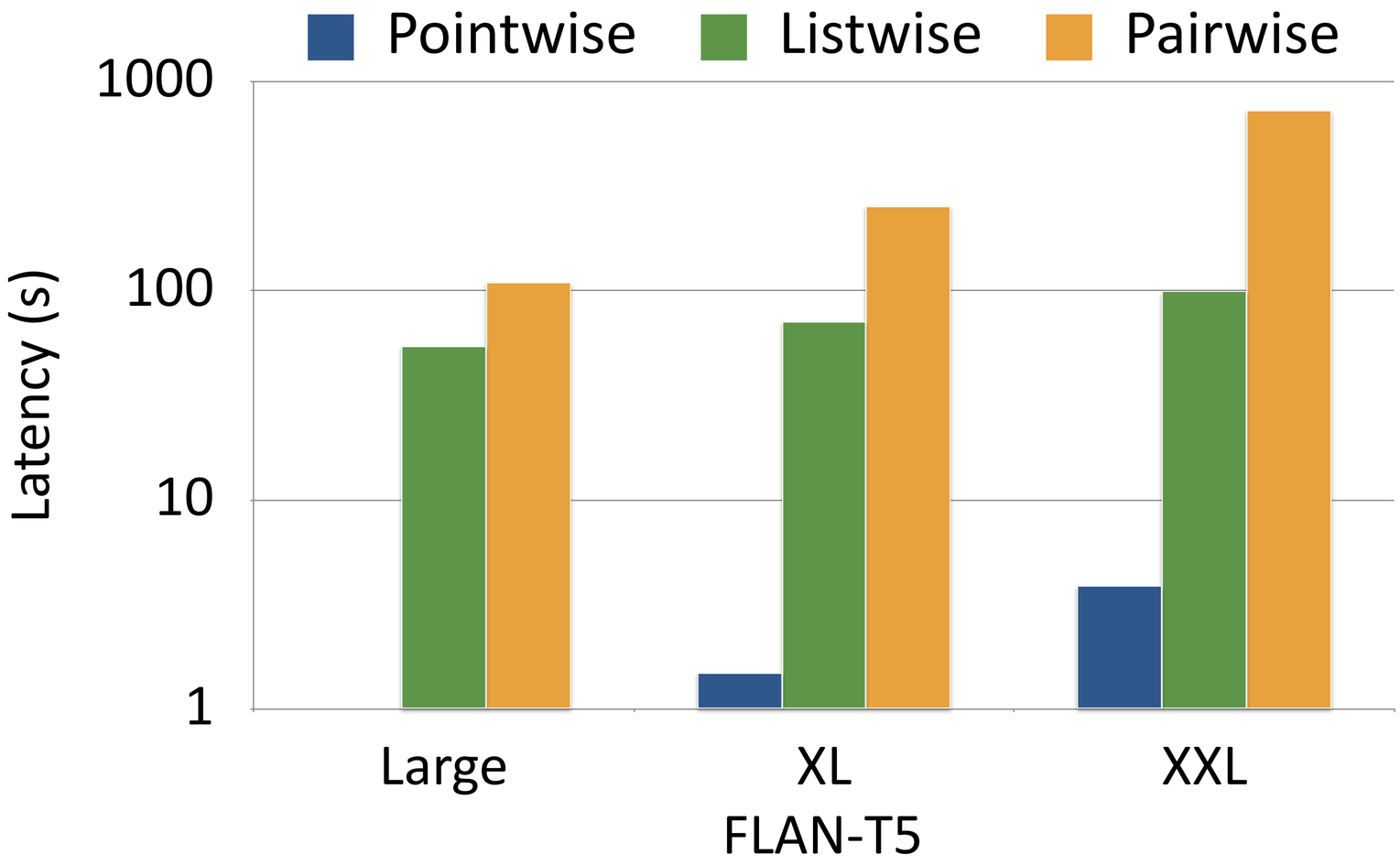
- **Scoring**: Compare all the pairwise preferences. This can be done in parallel.
- Both generation and logits are supported.

Qin, Z., Jagerman, R., Hui, K., Zhuang, H., Wu, J., Shen, J., Liu, T., Liu, J., Metzler, D., Wang, X. and Bendersky, M., 2023. Large language models are effective text rankers with pairwise ranking prompting. *NAACL-findings*.

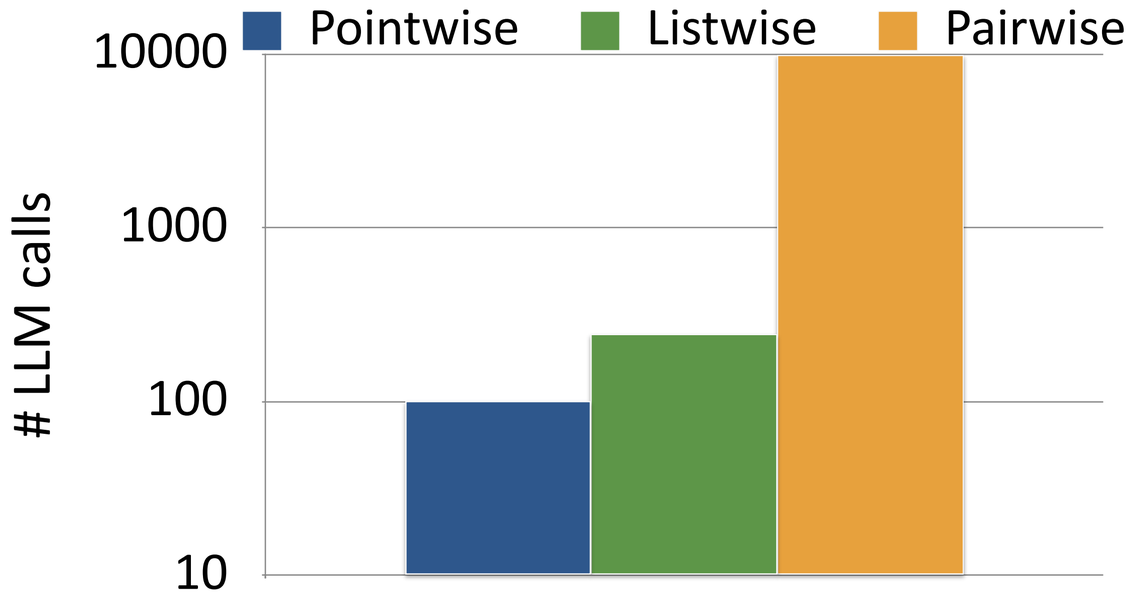
Effectiveness

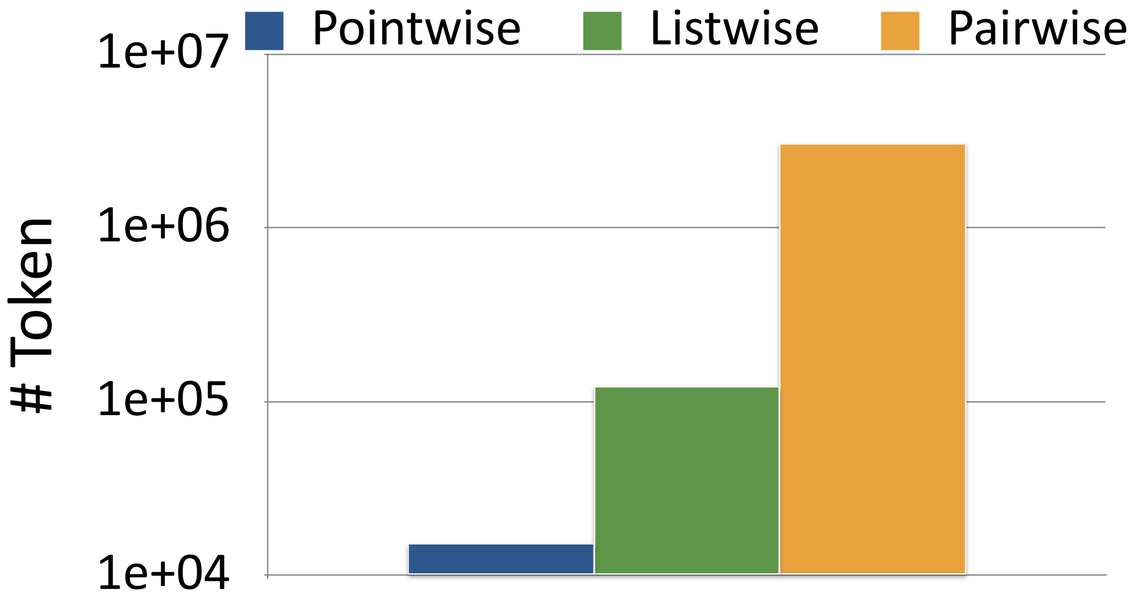


Efficiency


Pointwise
Listwise
Pairwise
1000
100
10
1
Latency (s)
Large
XL
XXL
FLAN-T5

Efficiency


Pointwise
Listwise
Pairwise
10000
1000
100
10
LLM calls

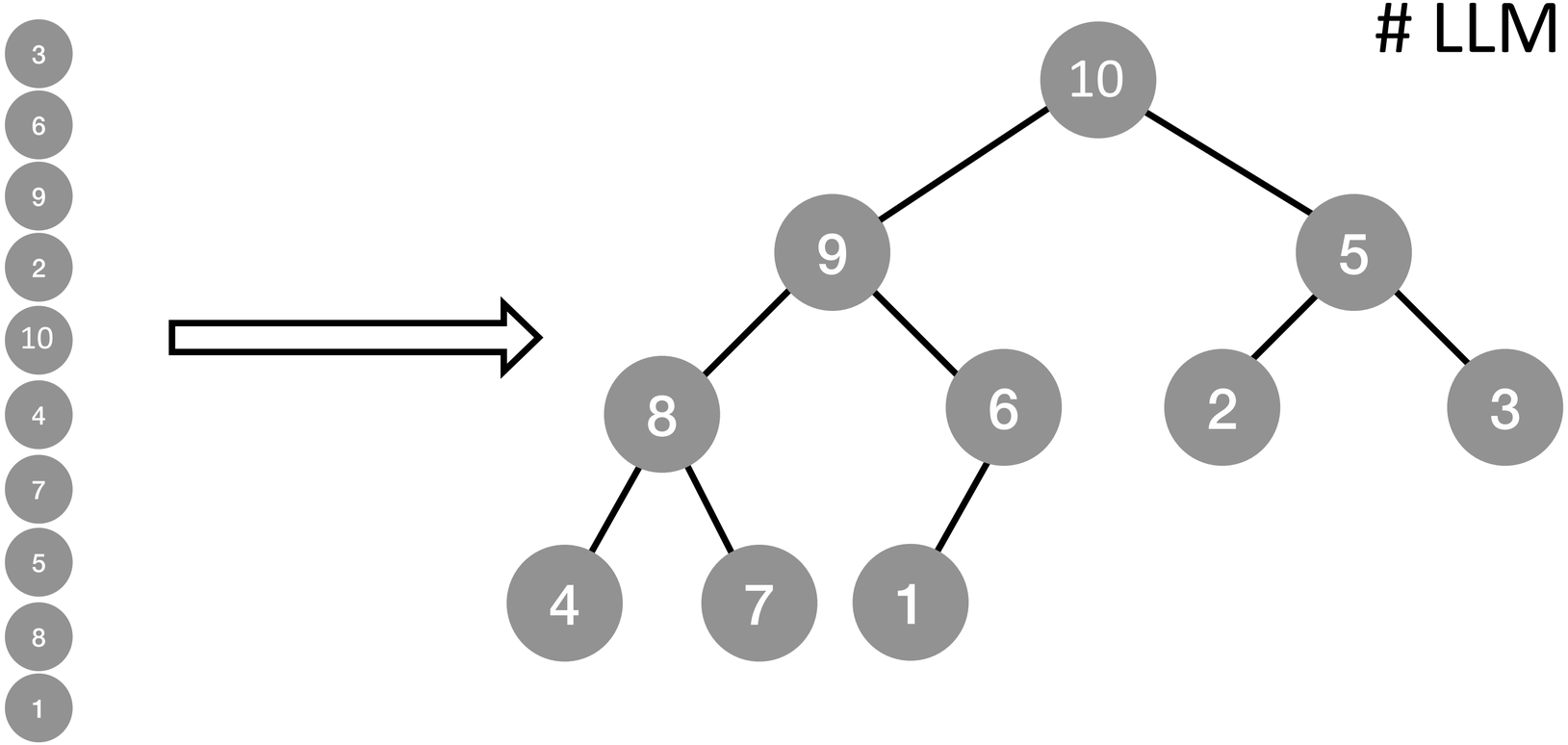

Pointwise
Listwise
Pairwise
1e+07
1e+06
1e+05
1e+04
Token

Pairwise

- **Sorting-based ranking: Heapsort**

  Step 1: Build max heap tree

LLM call: $O(nlogn)$

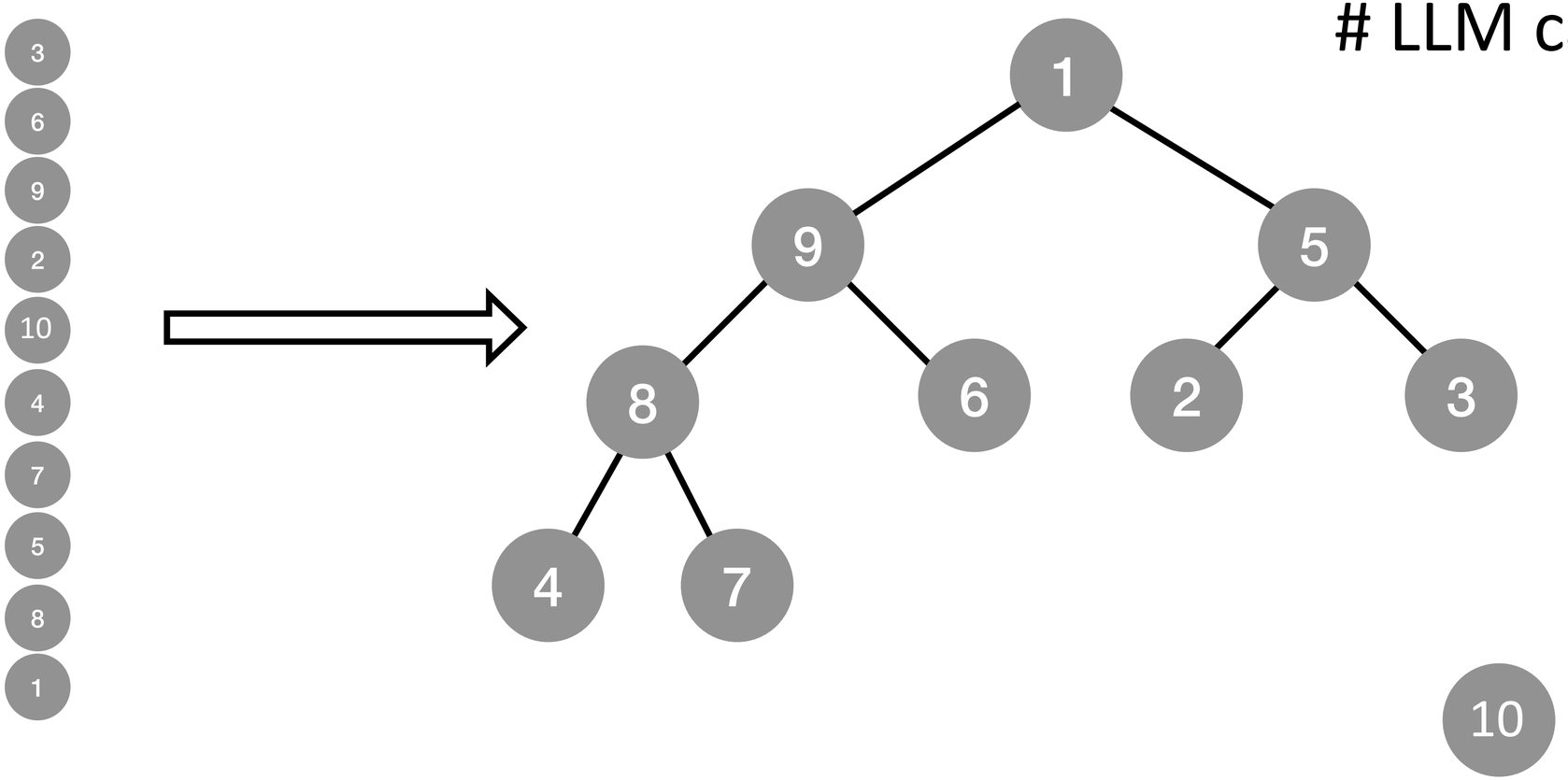


Pairwise

- **Sorting-based ranking: Heapsort**

  Step 1: Build max heap tree

LLM call: $O(nlogn)$

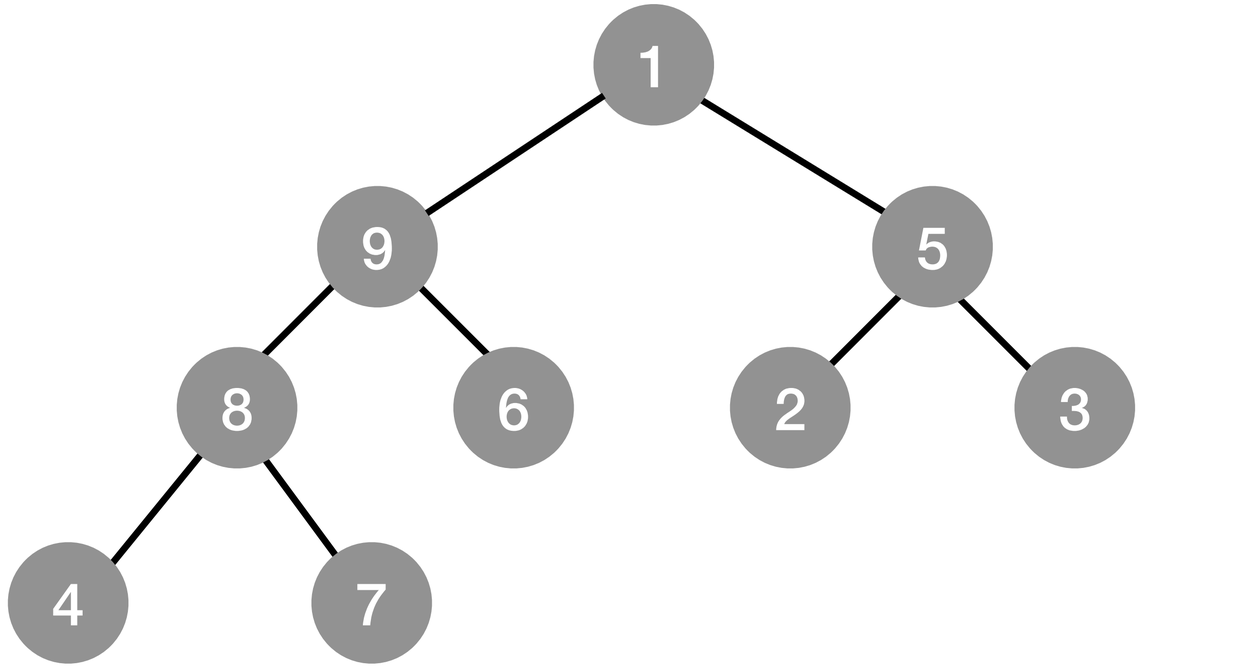


Pairwise

- **Sorting-based ranking: Heapsort**

  Step 2: heapify the tree

LLM call: 0



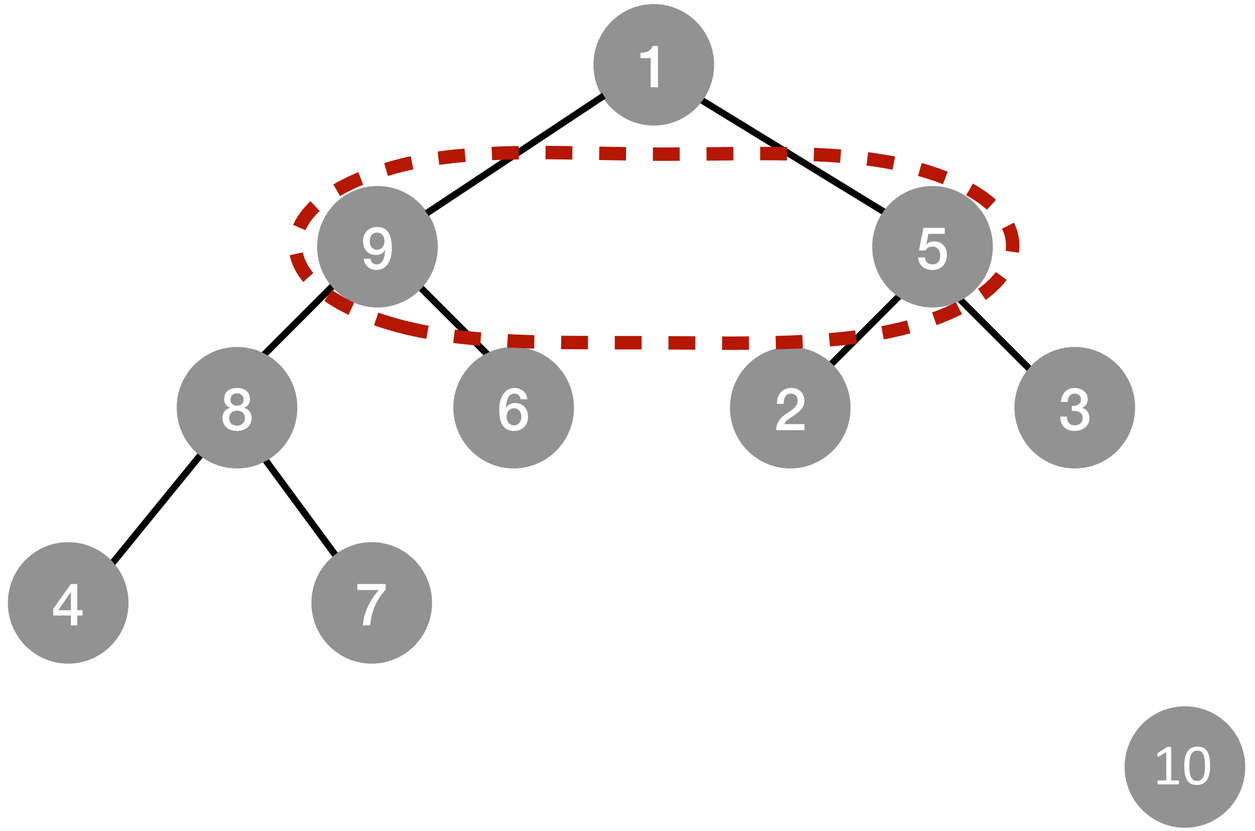
10

Pairwise

- **Sorting-based ranking: Heapsort**

  Step 2: heapify the tree

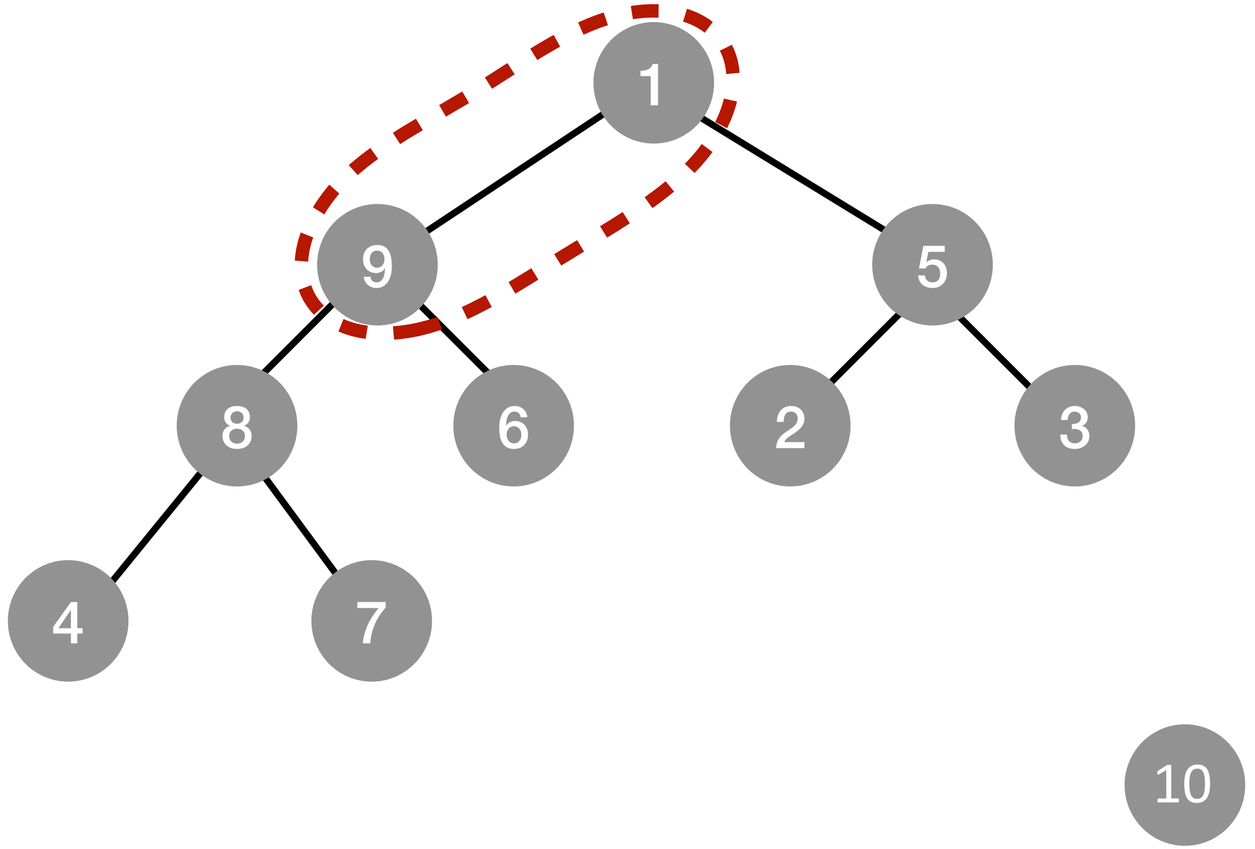
LLM call: 1



Pairwise

- **Sorting-based ranking: Heapsort**

  Step 2: heapify the tree

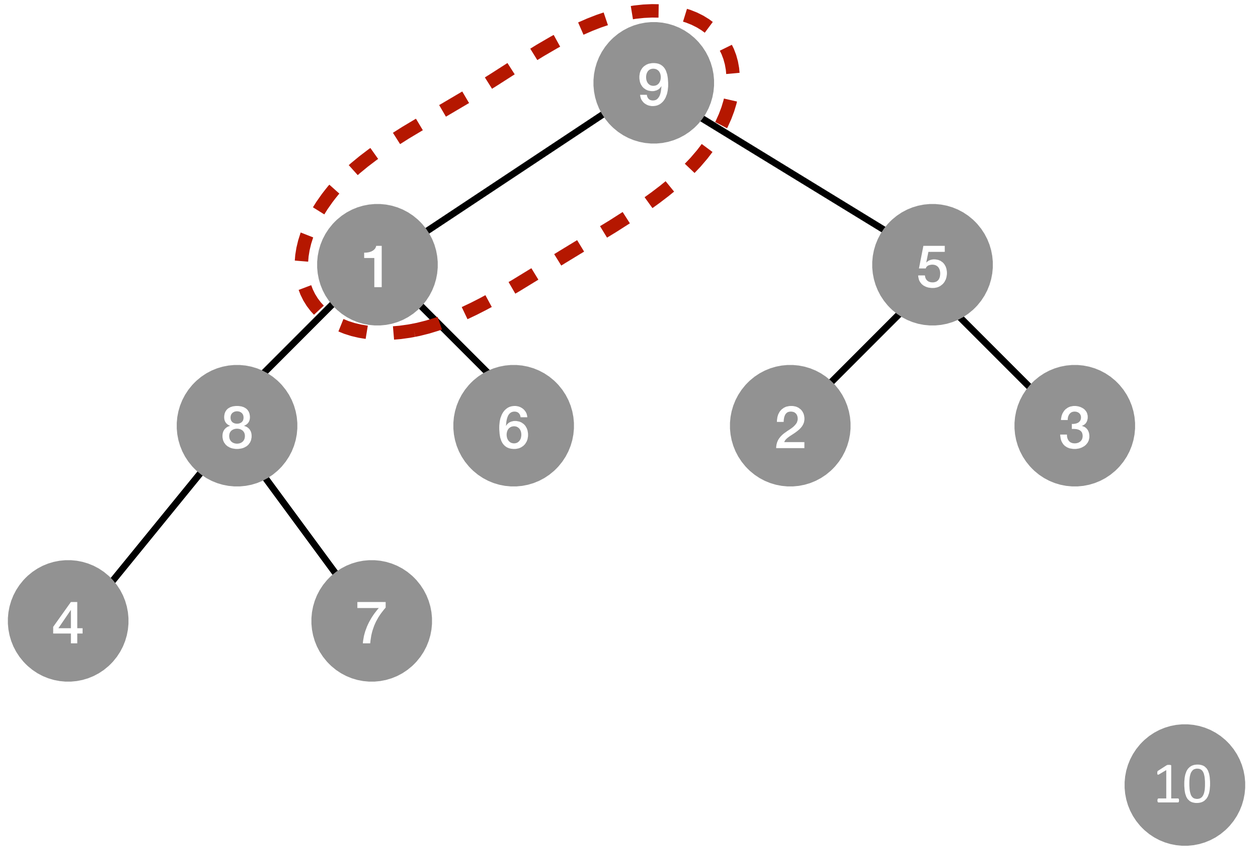


LLM call: 2

Pairwise

- **Sorting-based ranking: Heapsort**

  Step 2: heapify the tree



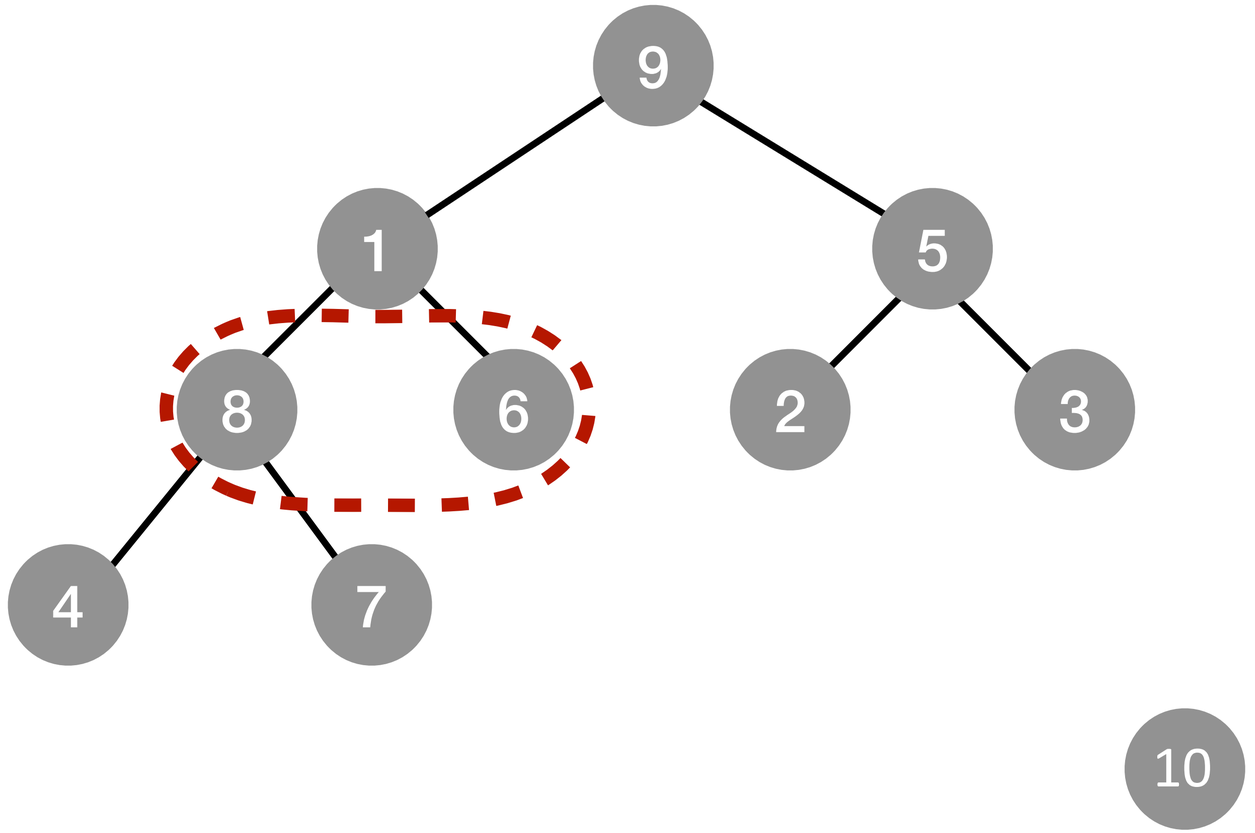
LLM call: 2

Pairwise

- **Sorting-based ranking: Heapsort**

  Step 2: heapify the tree

LLM call: 3

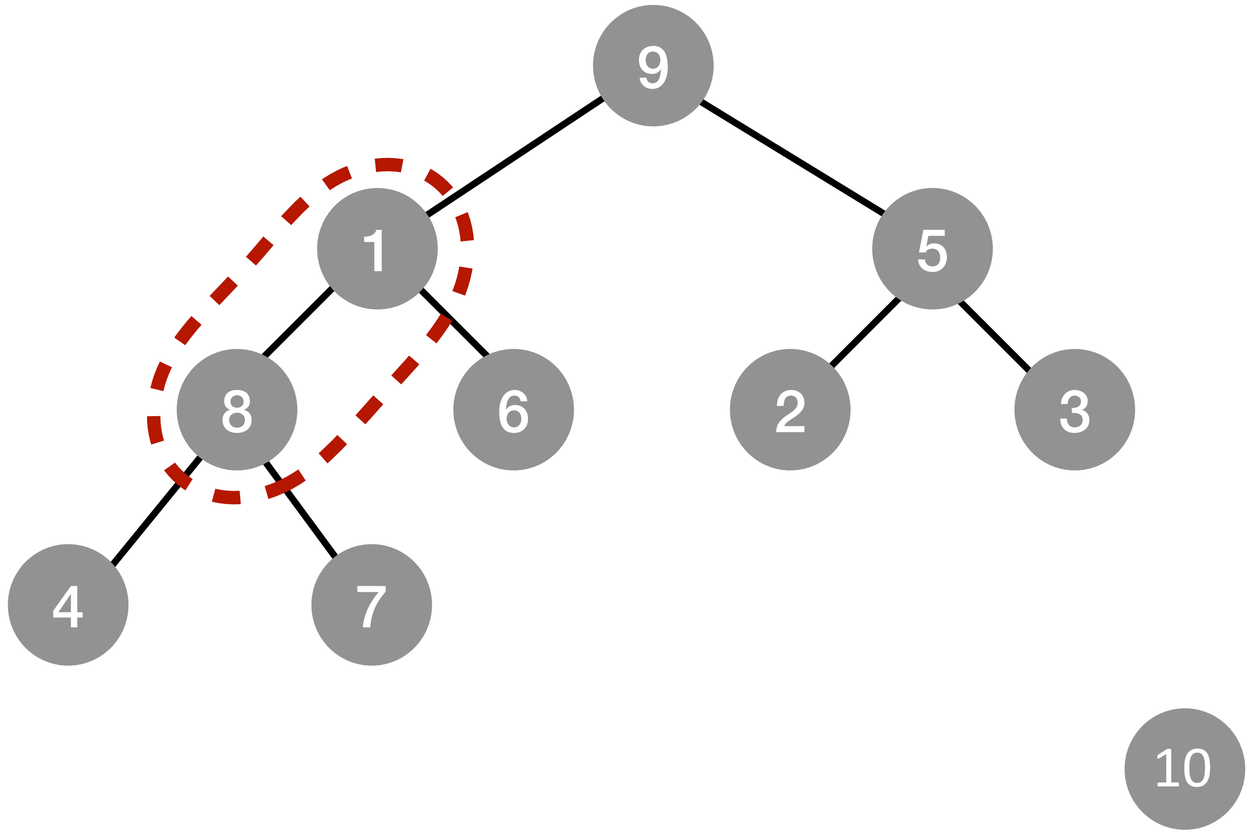


Pairwise

- **Sorting-based ranking: Heapsort**

  Step 2: heapify the tree

LLM call: 4

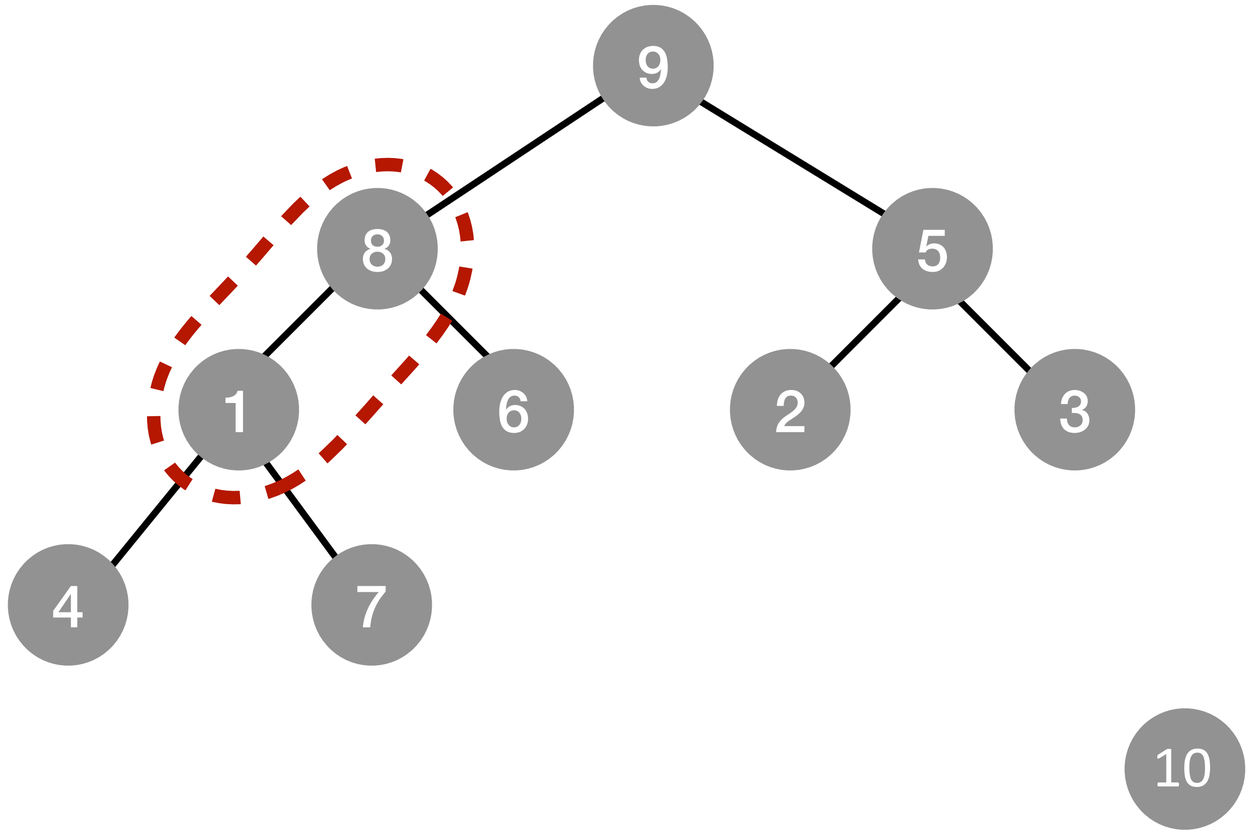


Pairwise

- **Sorting-based ranking: Heapsort**

  Step 2: heapify the tree

LLM call: 4

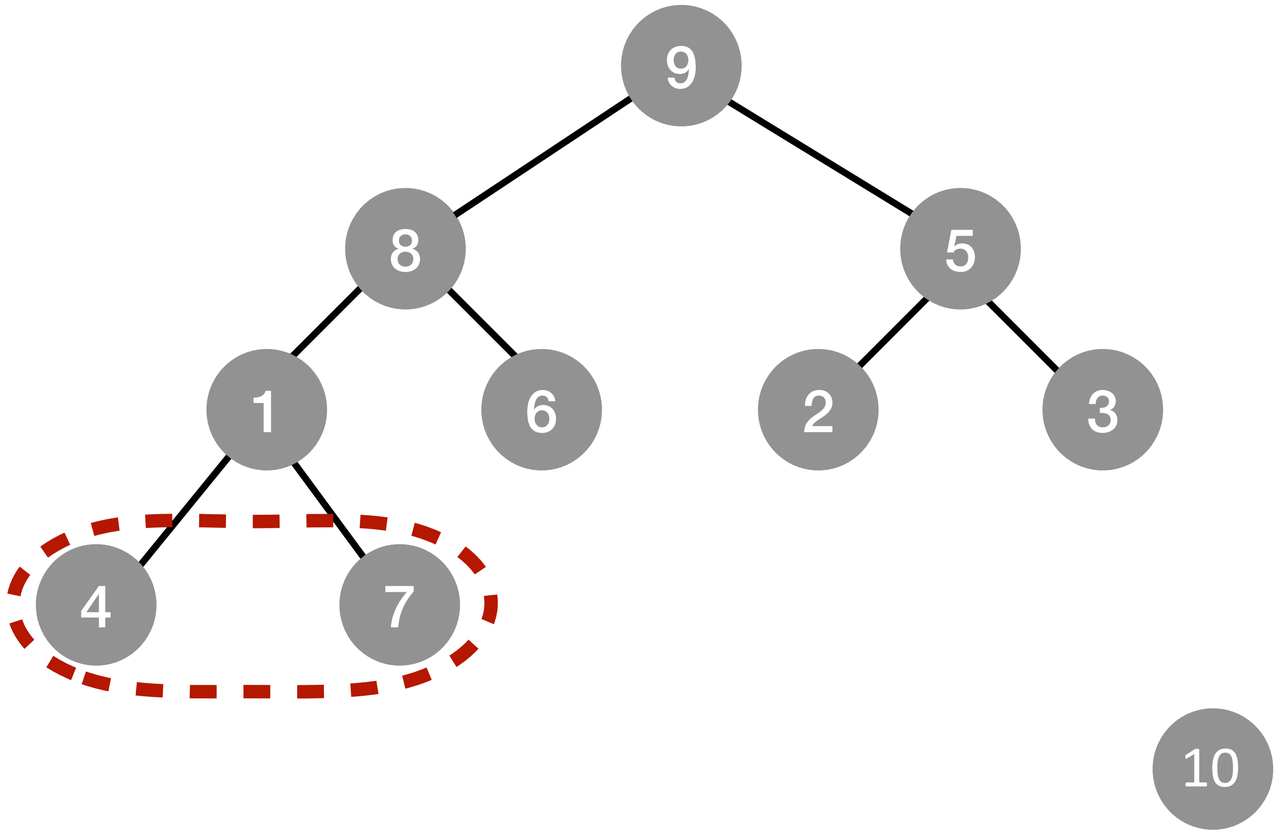


Pairwise

- **Sorting-based ranking: Heapsort**

  Step 2: heapify the tree

LLM call: 5

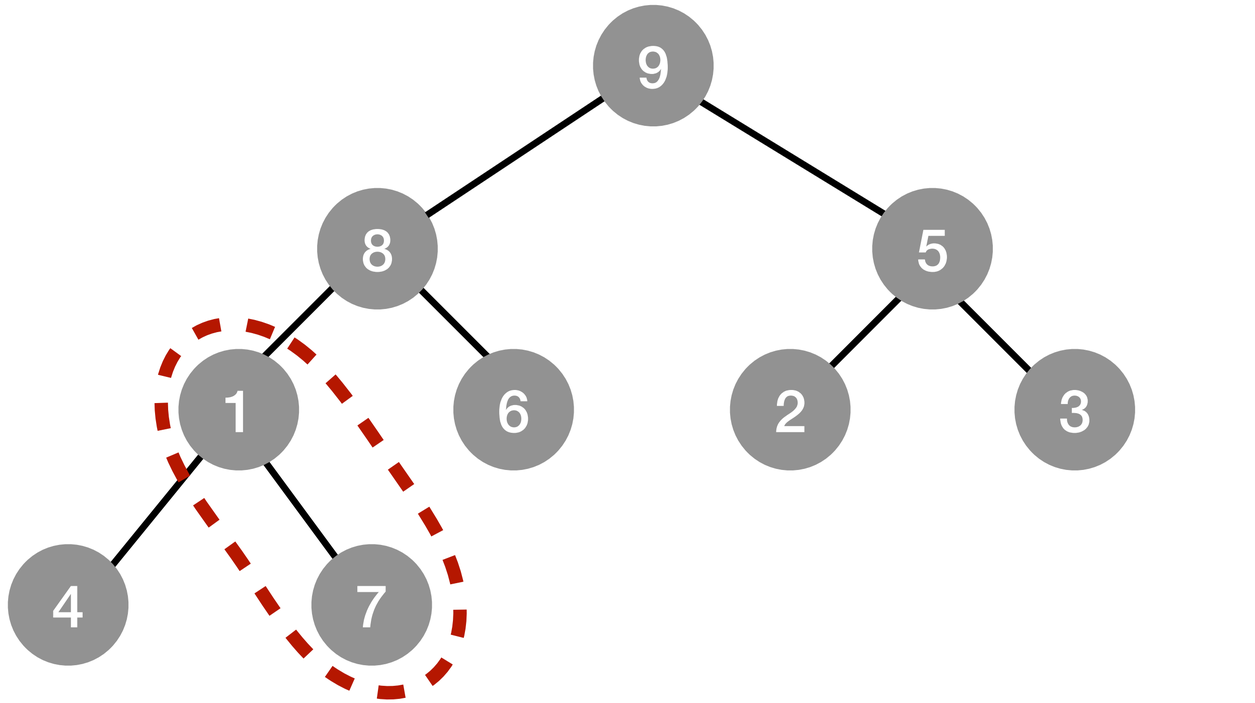


Pairwise

- **Sorting-based ranking: Heapsort**

  Step 2: heapify the tree

LLM call: 6



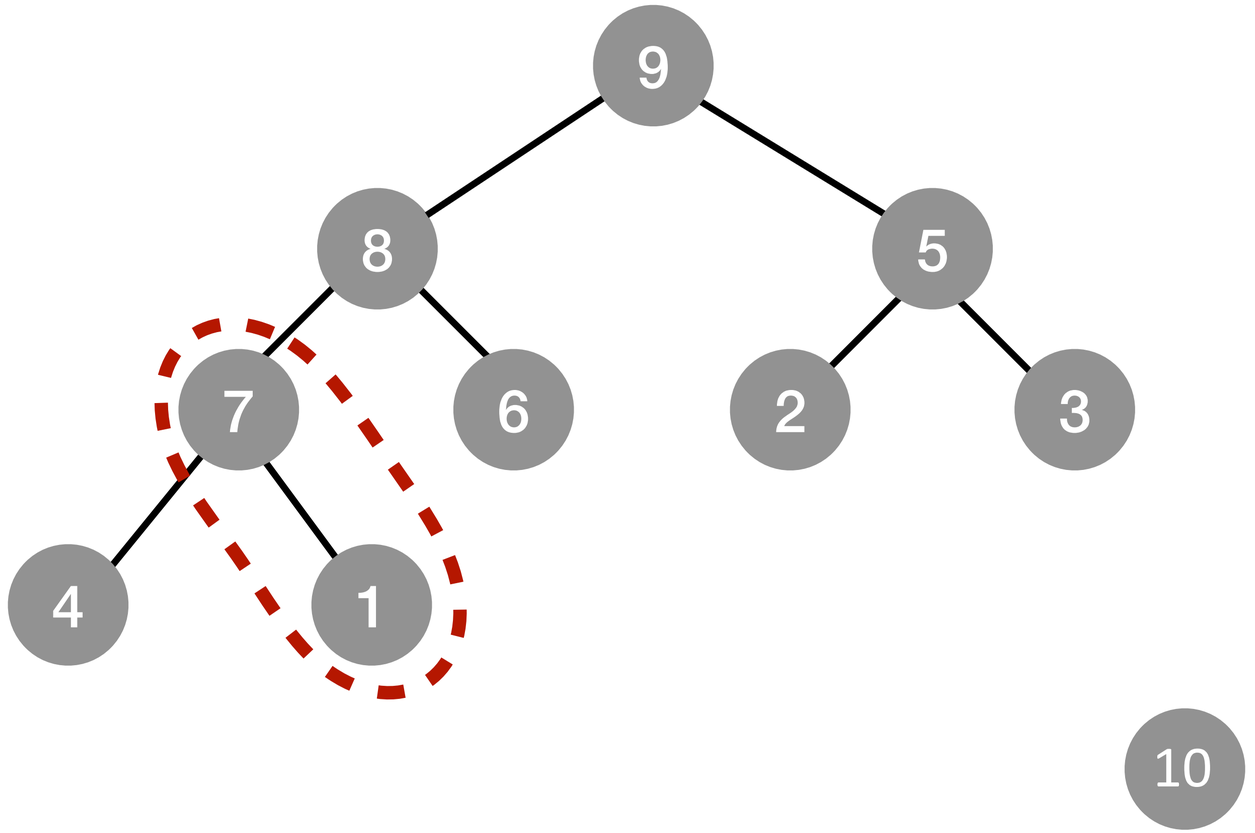
10

Pairwise

- **Sorting-based ranking: Heapsort**

  Step 2: heapify the tree

LLM call: 6

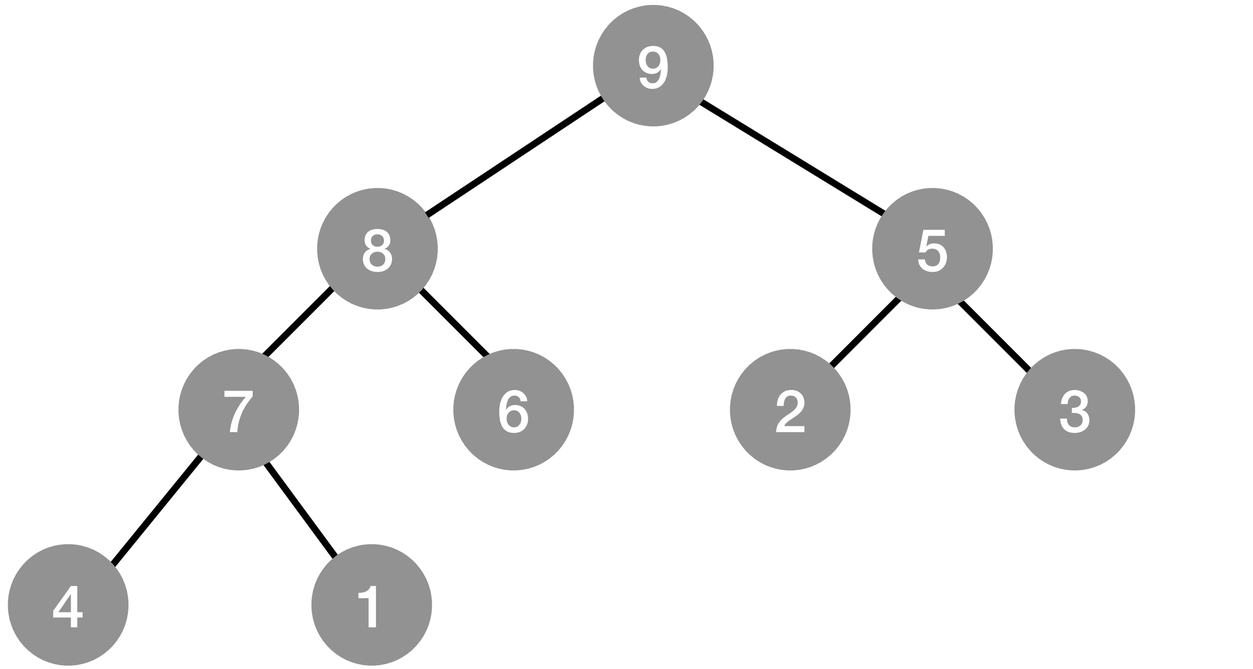


Pairwise

- **Sorting-based ranking: Heapsort**

  Step 2: heapify the tree

LLM call: 6



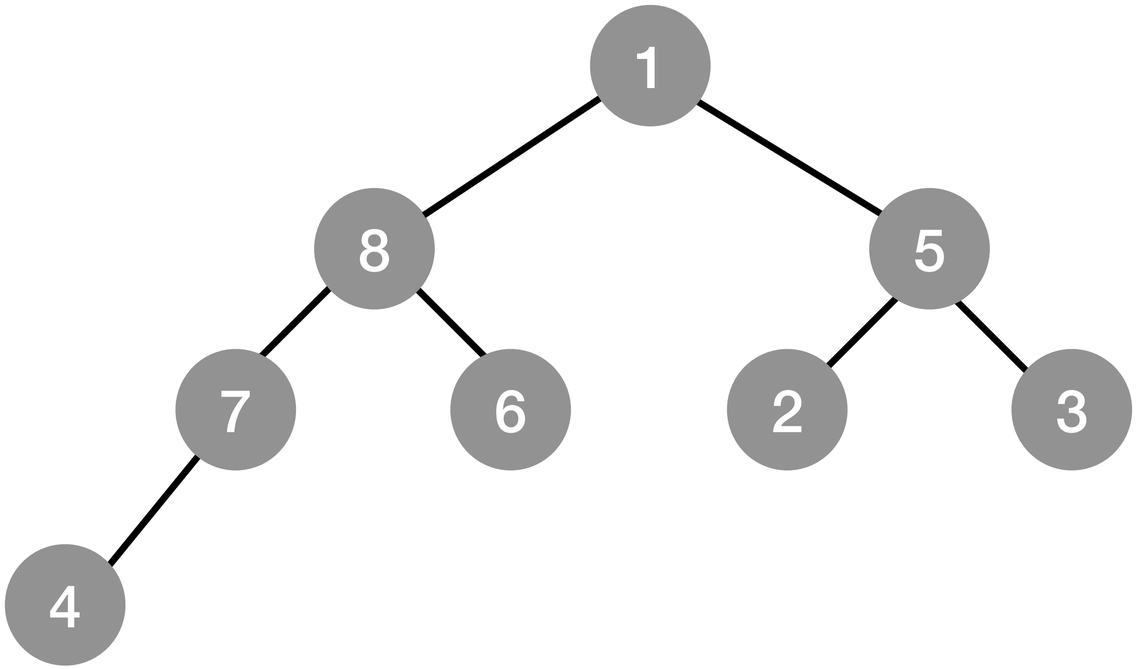
10

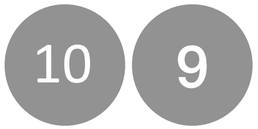
Pairwise

- **Sorting-based ranking: Heapsort**

  Step 2: heapify the tree

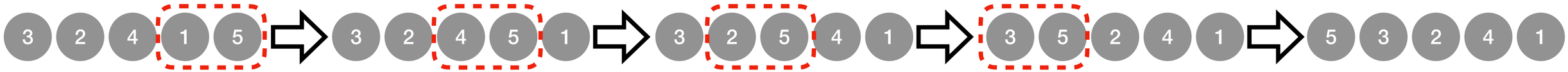
LLM call: 6



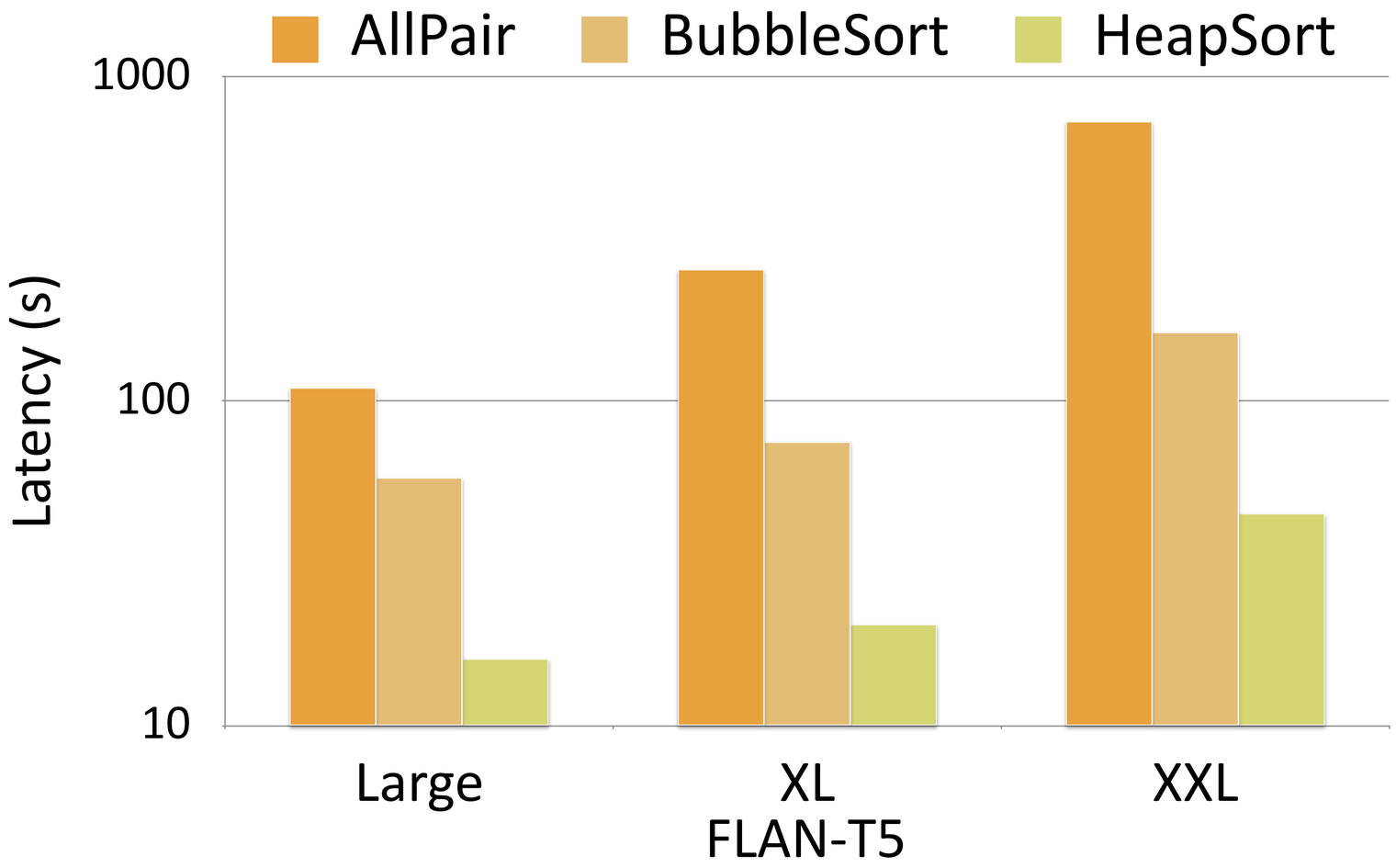


Pairwise

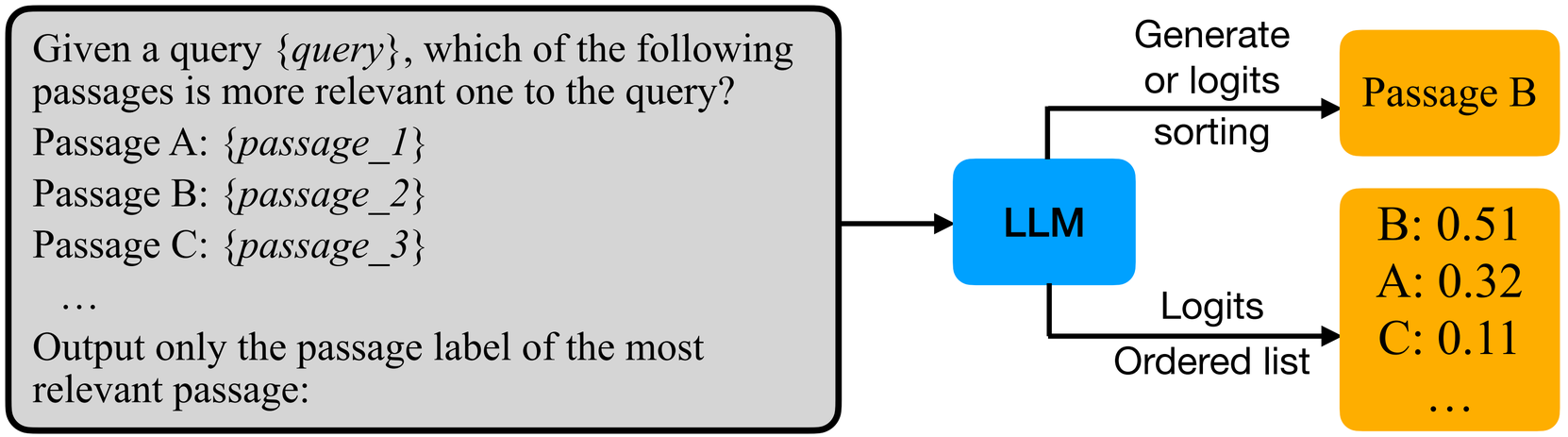
- **Sorting-based ranking: Bubblesort**



Pairwise Efficiency



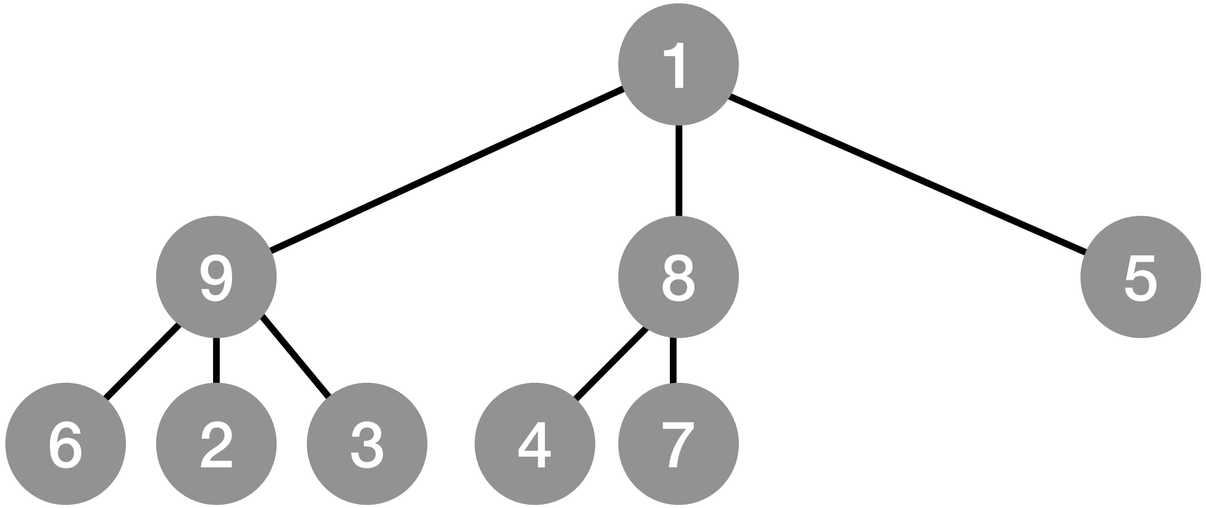
LLMs as Rankers: Setwise



Zhuang, S., Zhuang, H., Koopman, B. and Zuccon, G., 2023. A Setwise Approach for Effective and Highly Efficient Zero-shot Ranking with Large Language Models. *SIGIR*

Setwise

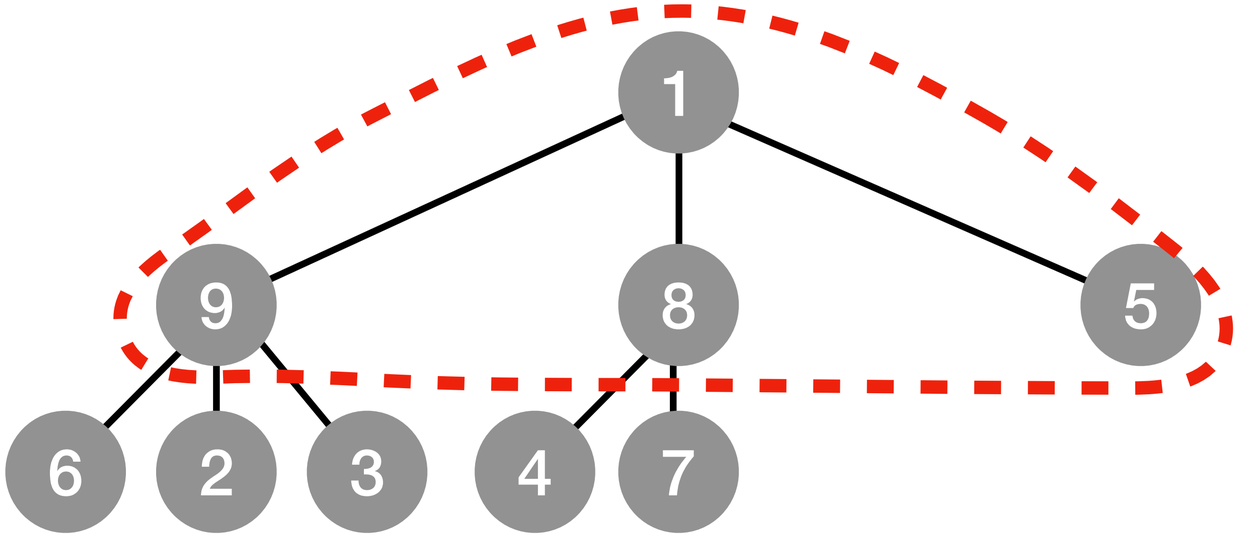
- **Sorting-based ranking: Heapsort**



Setwise

- **Sorting-based ranking: Heapsort**

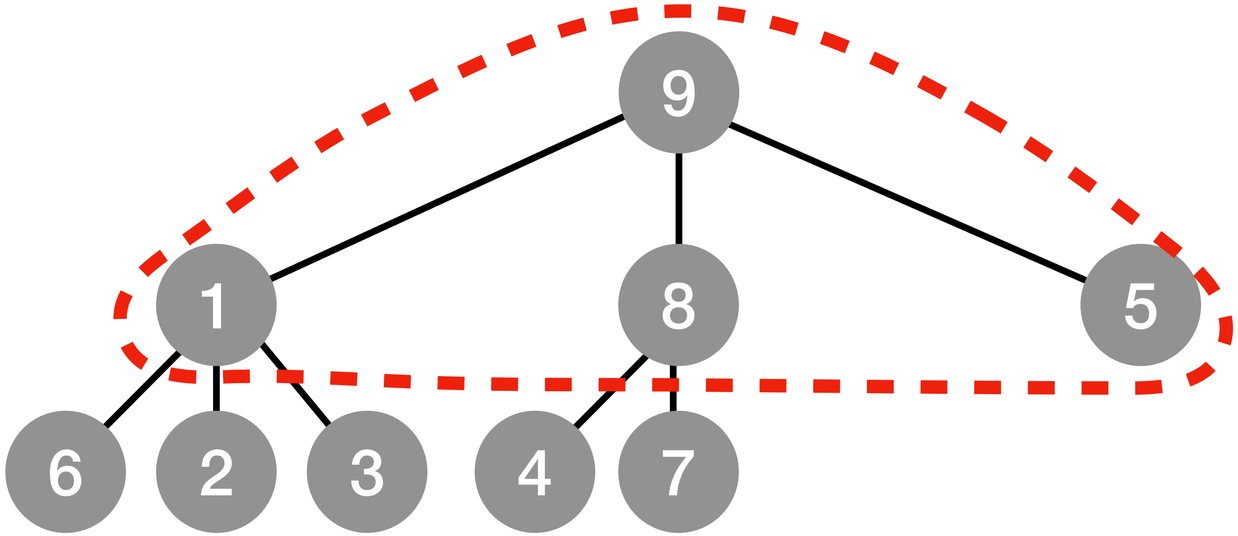
LLM call: 1



Setwise

- **Sorting-based ranking: Heapsort**

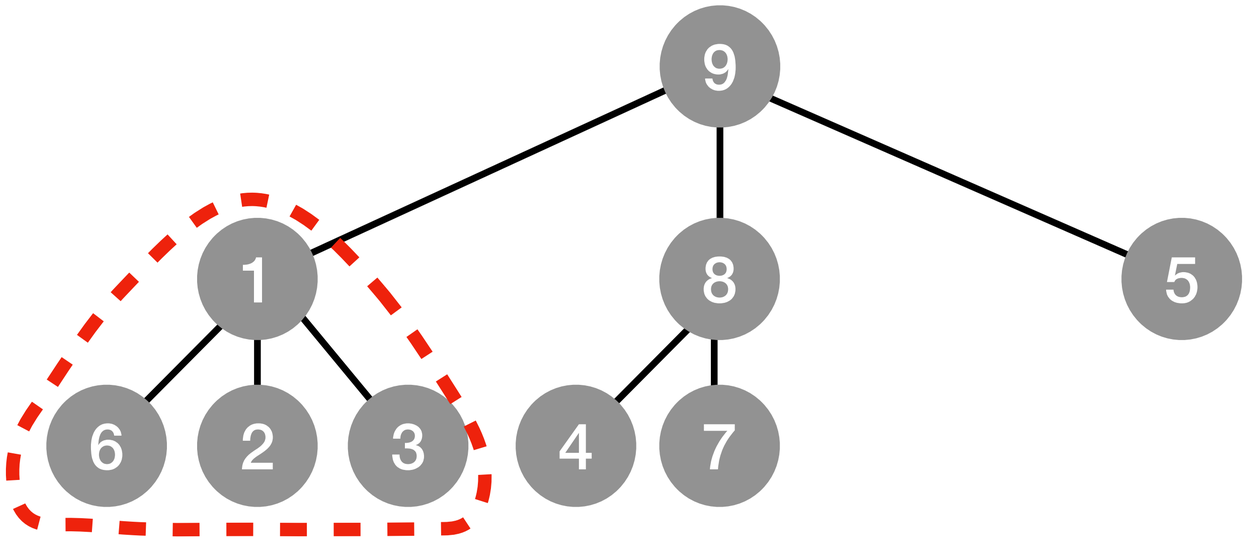
LLM call: 1



Setwise

- **Sorting-based ranking: Heapsort**

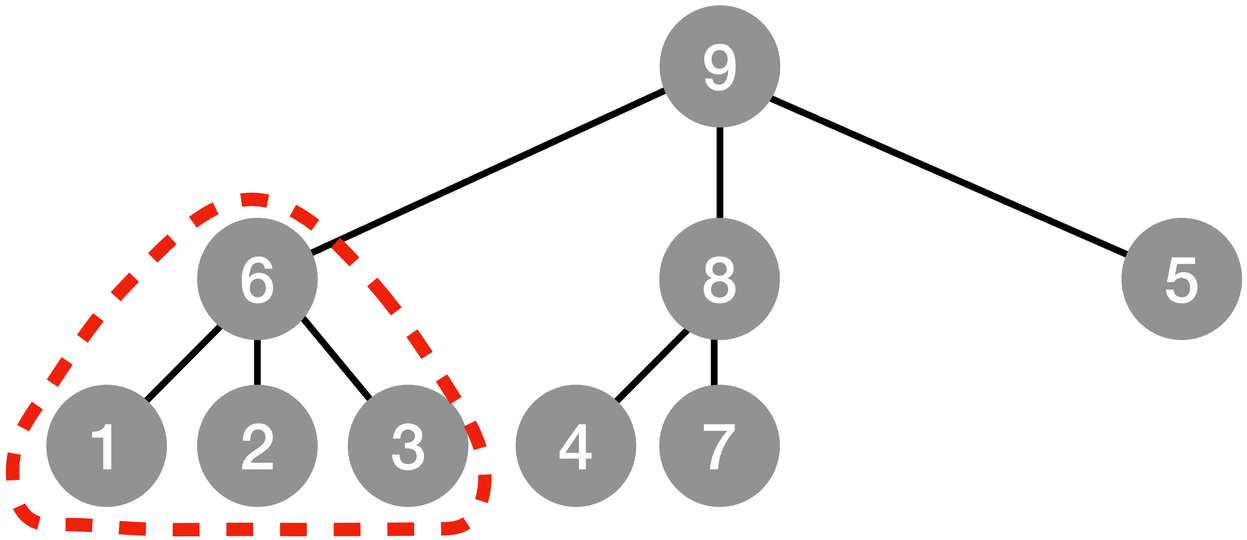
LLM call: 2



Setwise

- **Sorting-based ranking: Heapsort**

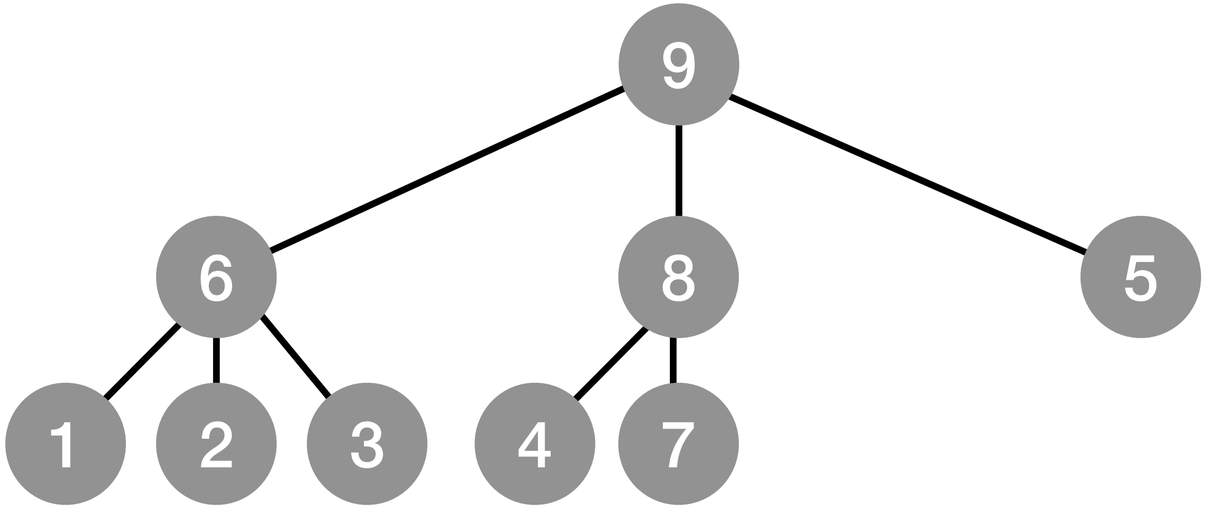
LLM call: 2



Setwise

- **Sorting-based ranking: Heapsort**

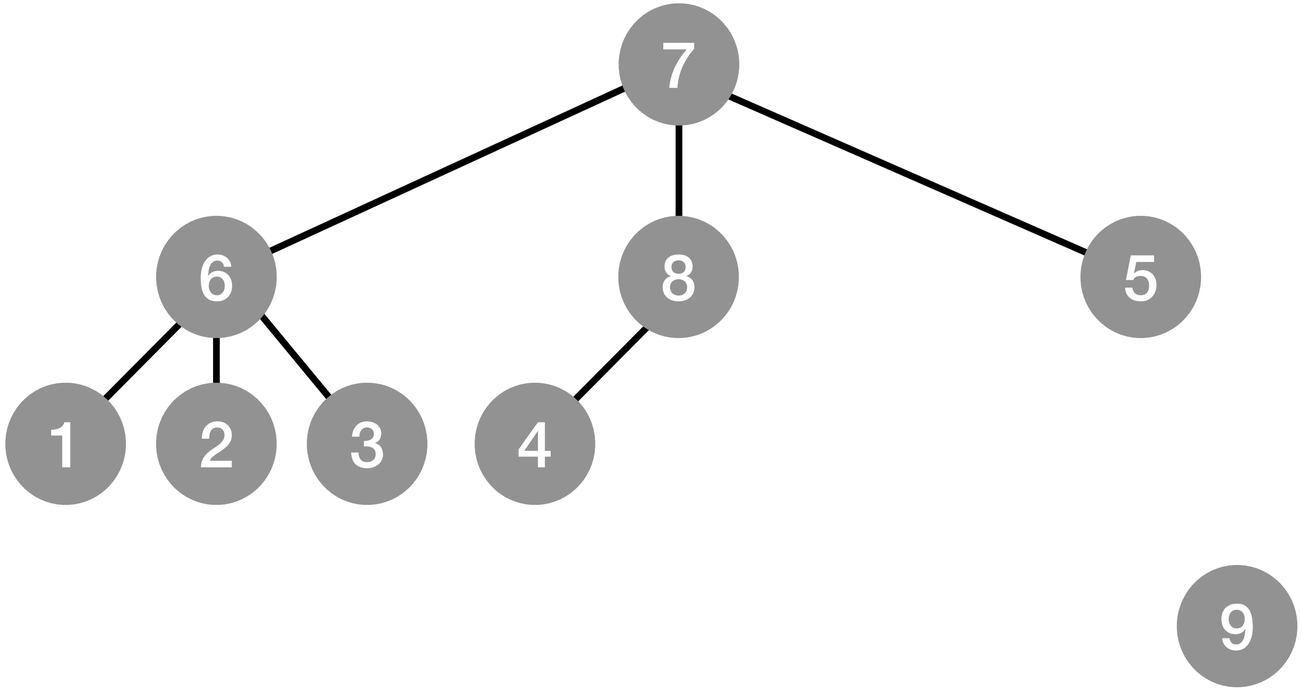
LLM call: 2



Setwise

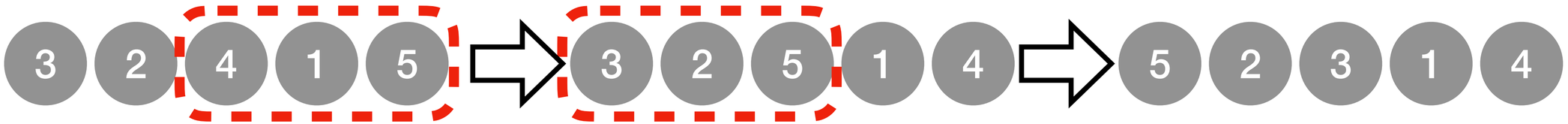
- **Sorting-based ranking: Heapsort**

LLM call: 2

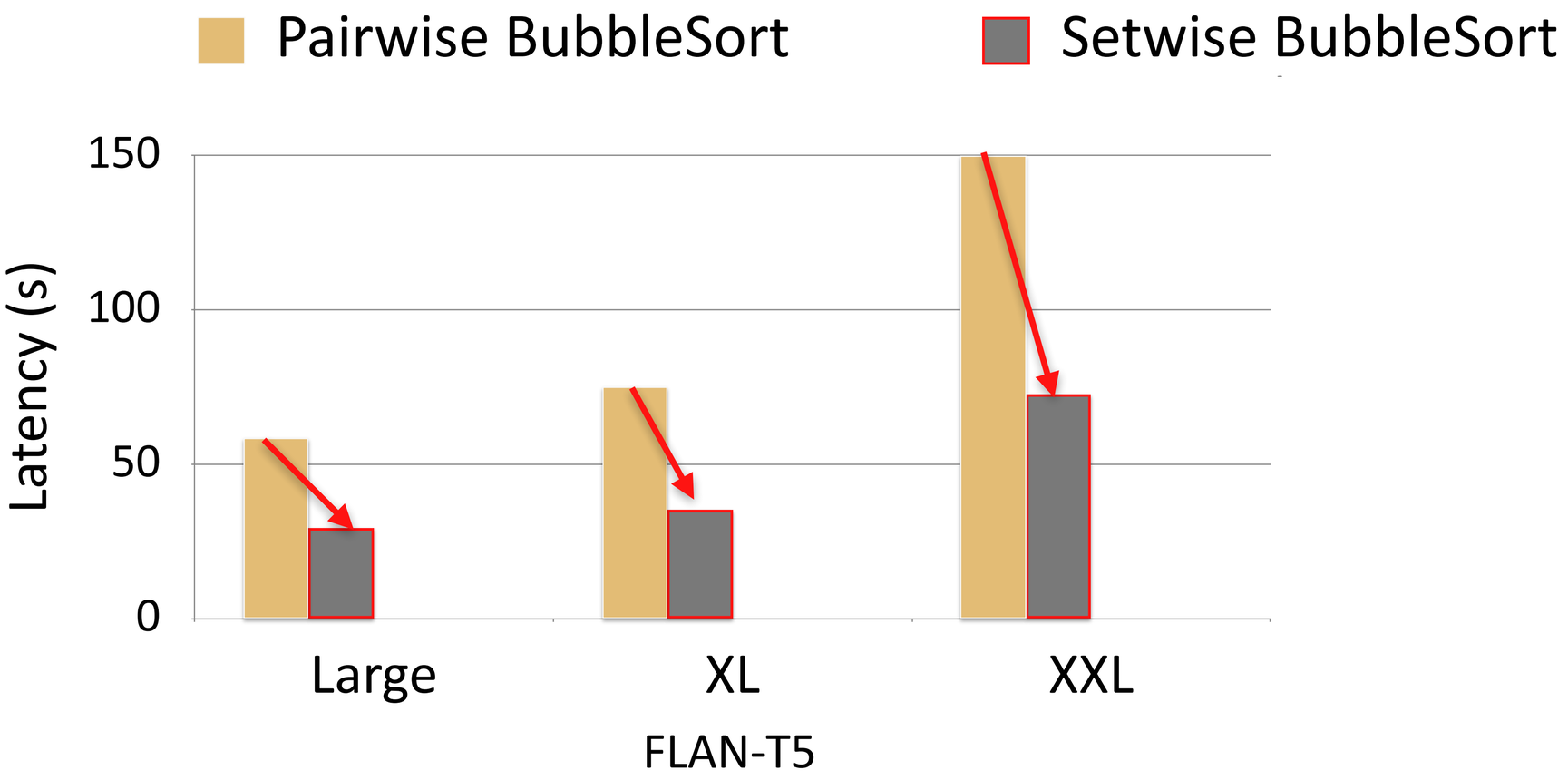


Setwise

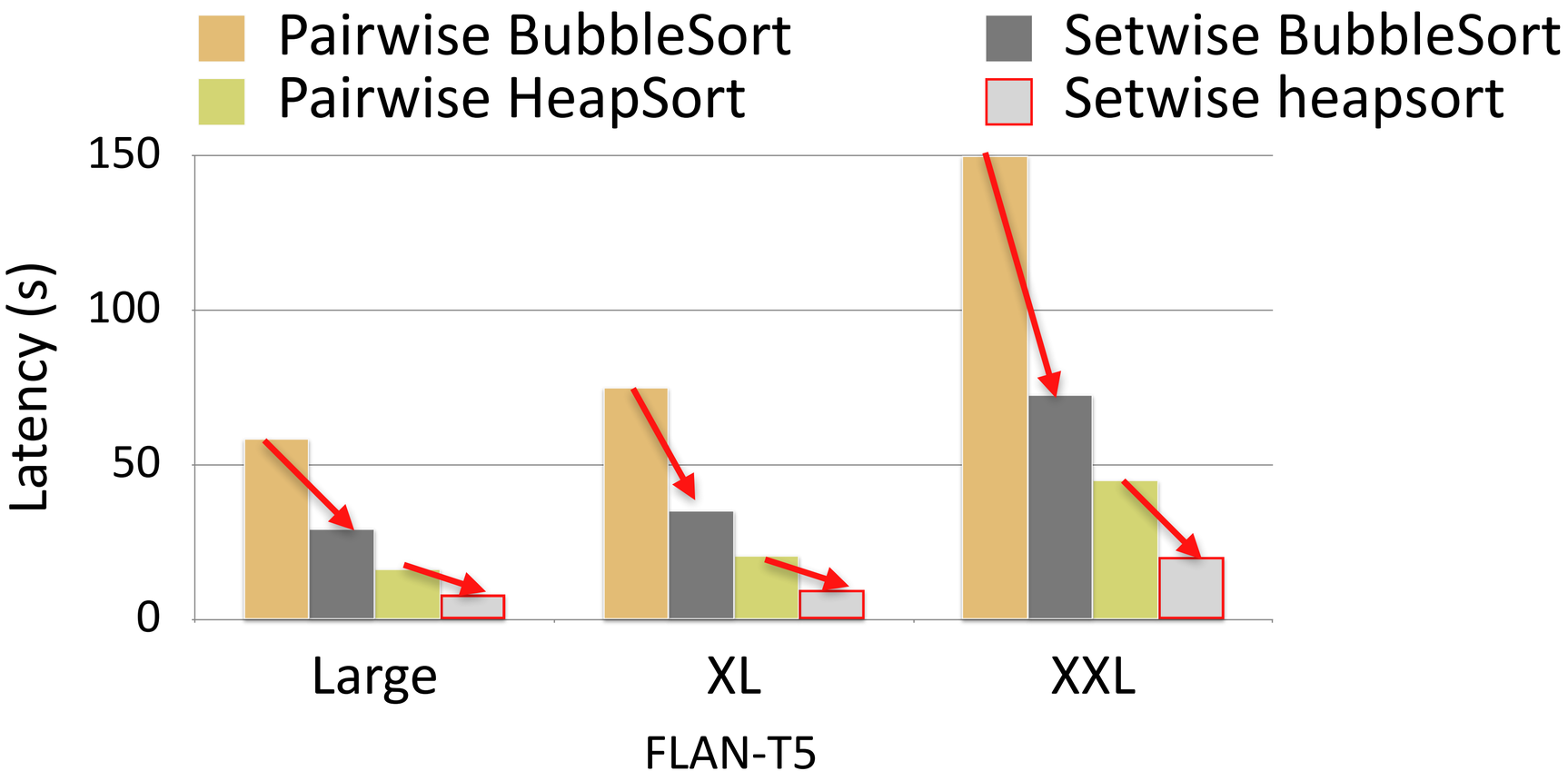
- **Sorting-based ranking: Bubblesort**



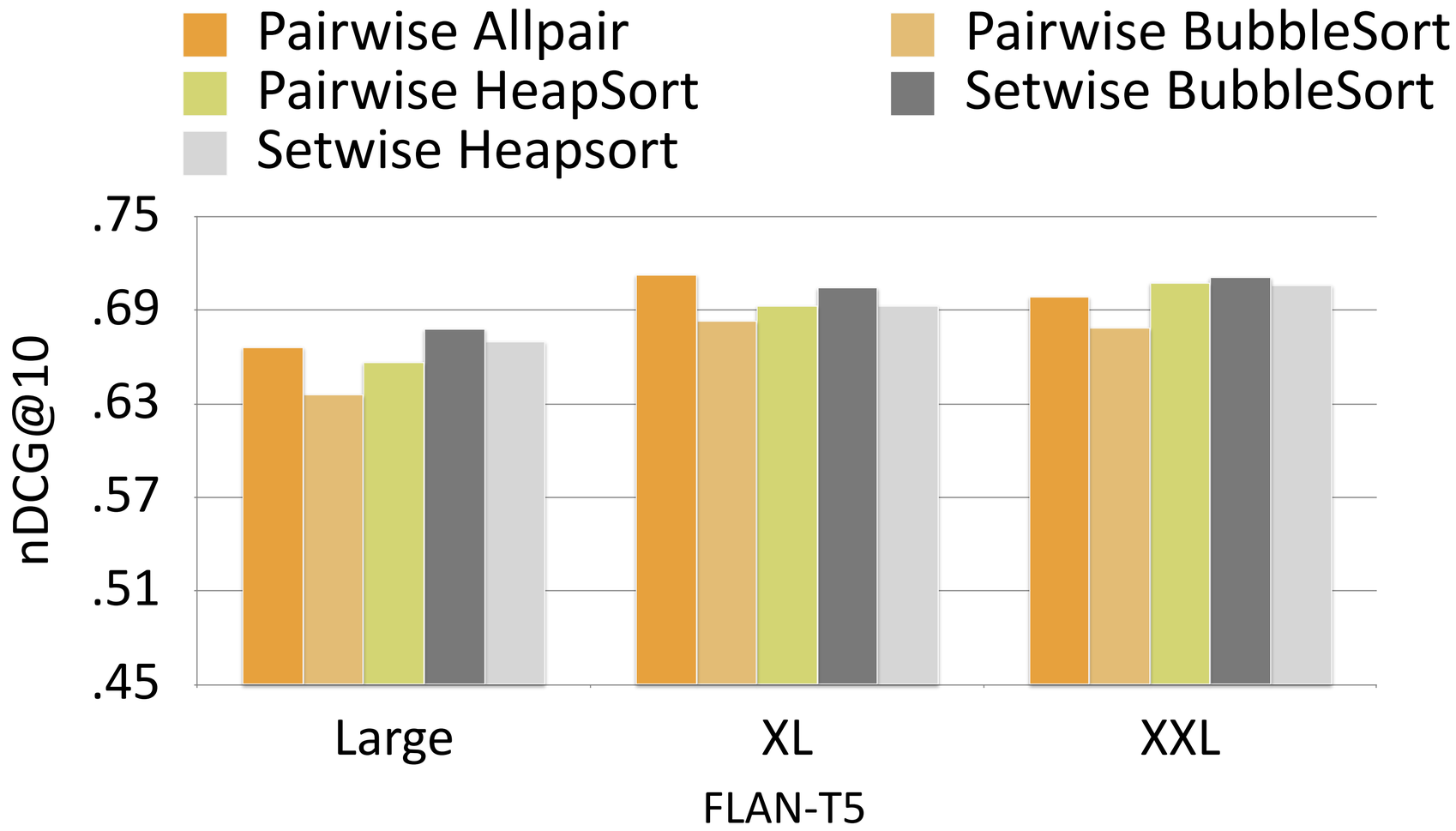
Setwise Efficiency



Pairwise Efficiency

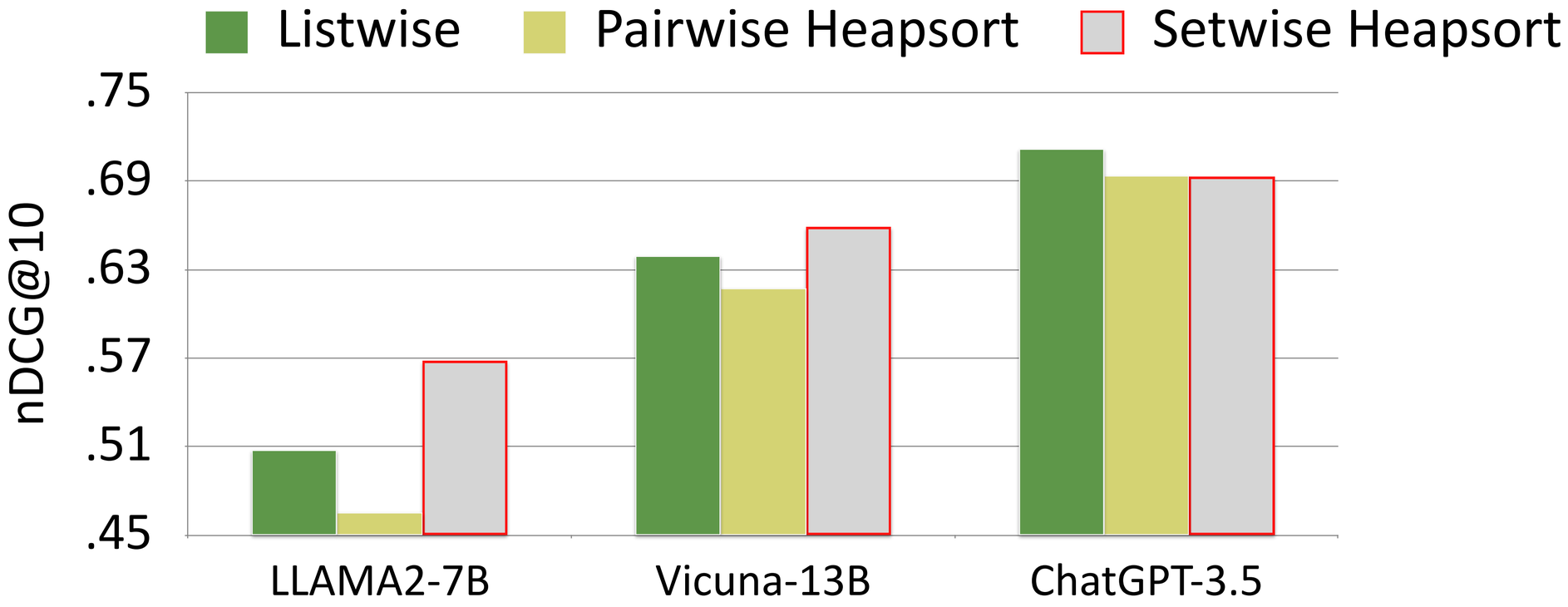


Setwise Effectiveness

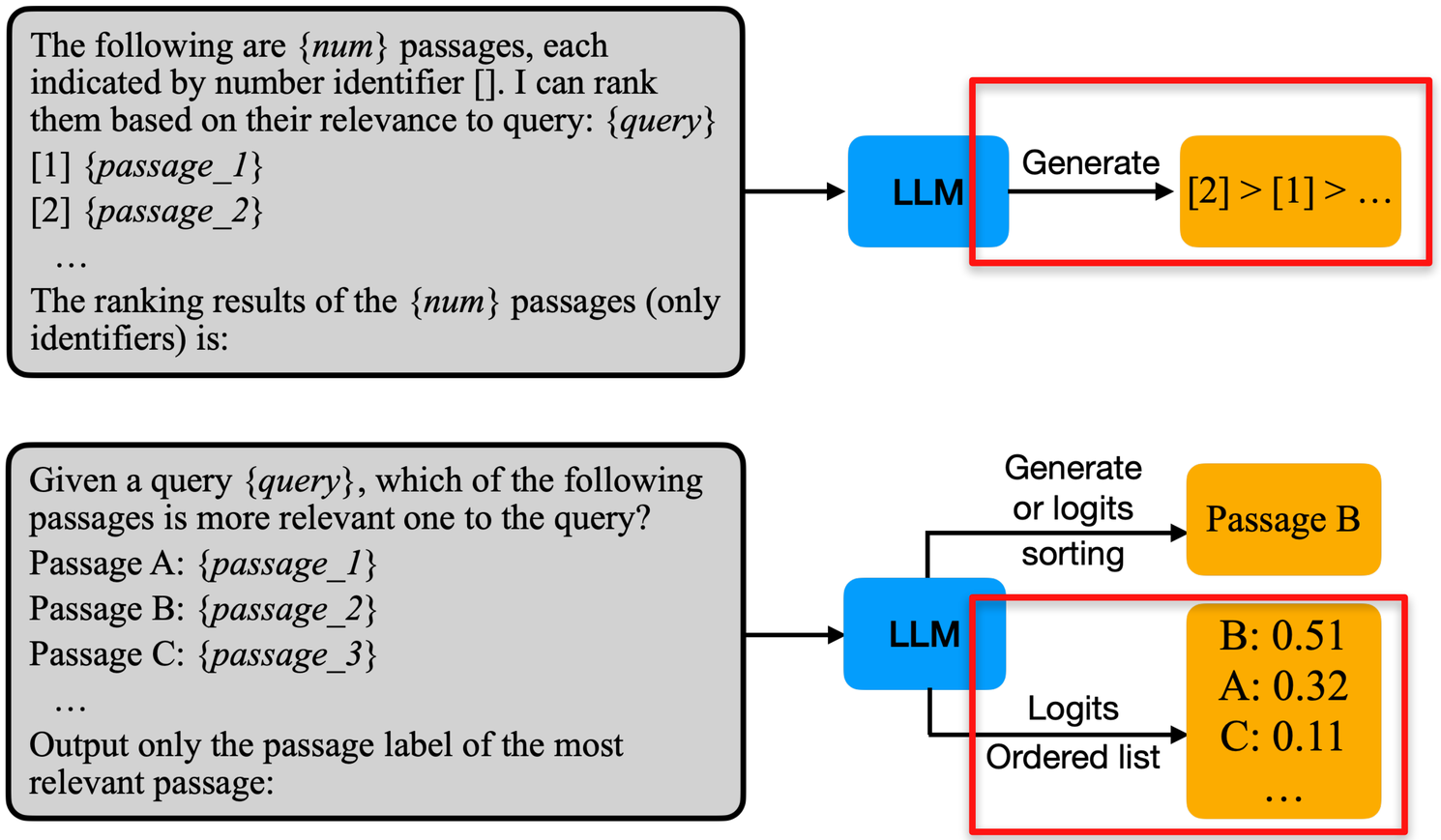


Setwise Effectiveness

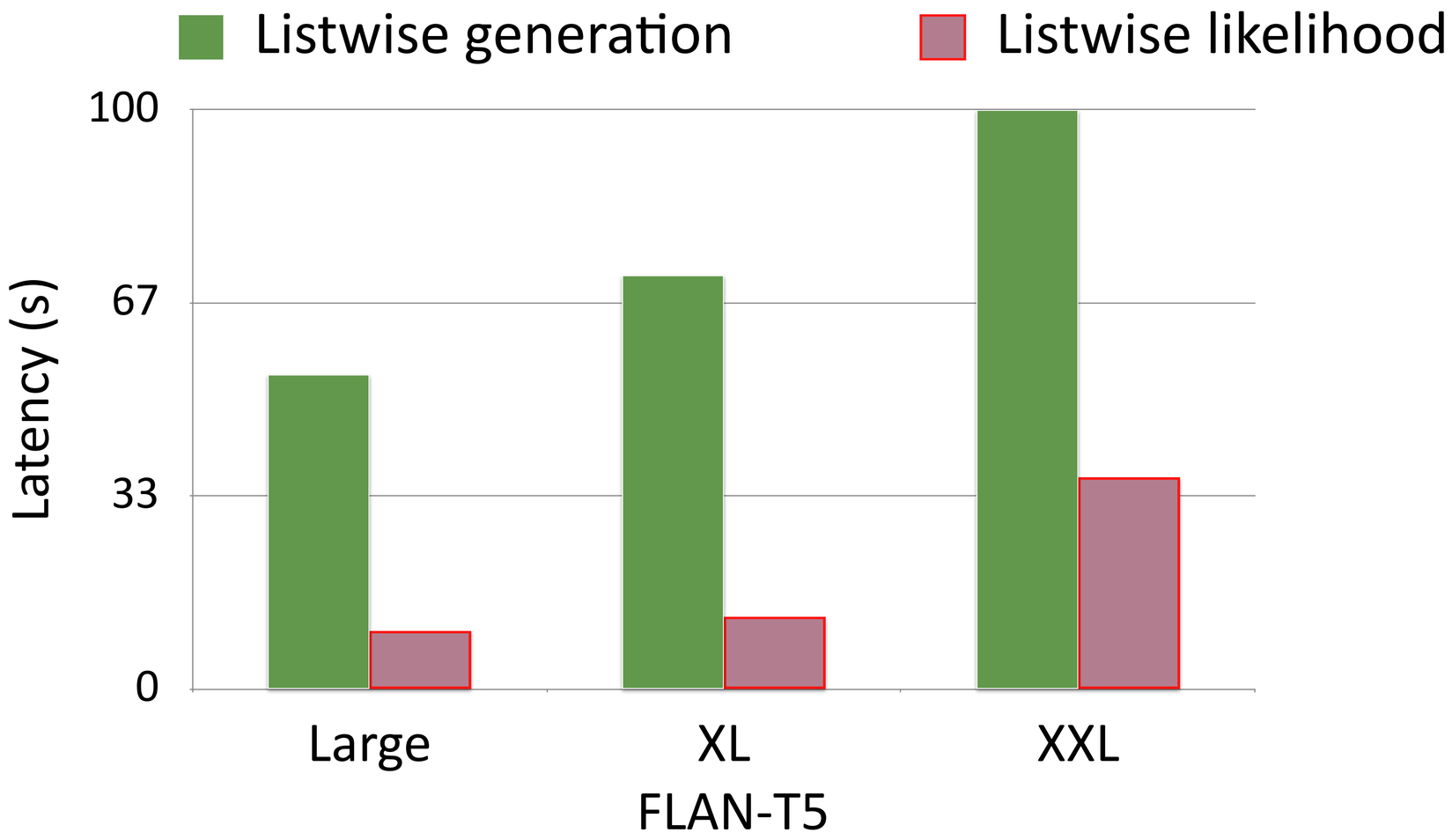
On other backbones



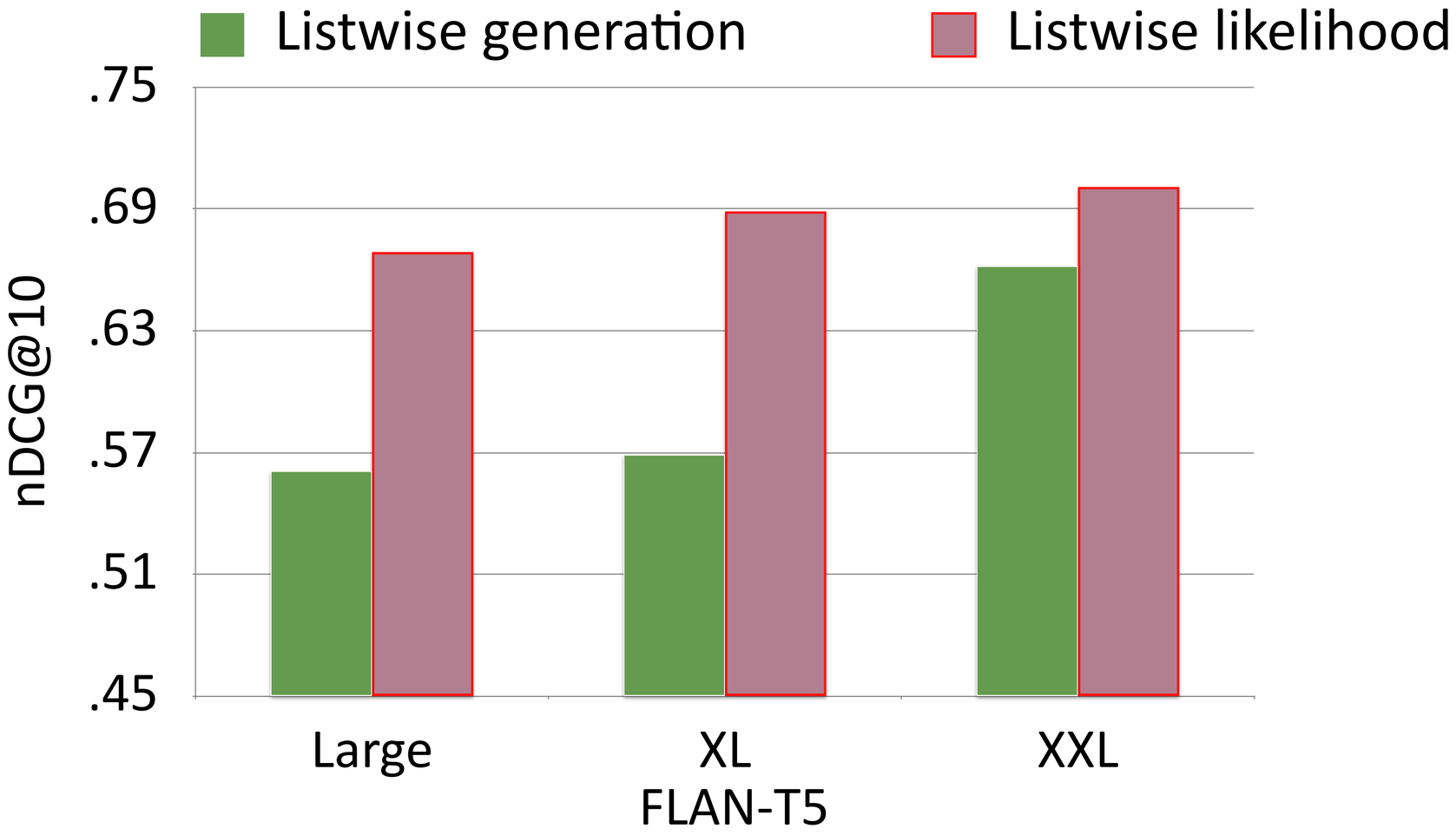
Listwise Ranking with Setwise Prompting



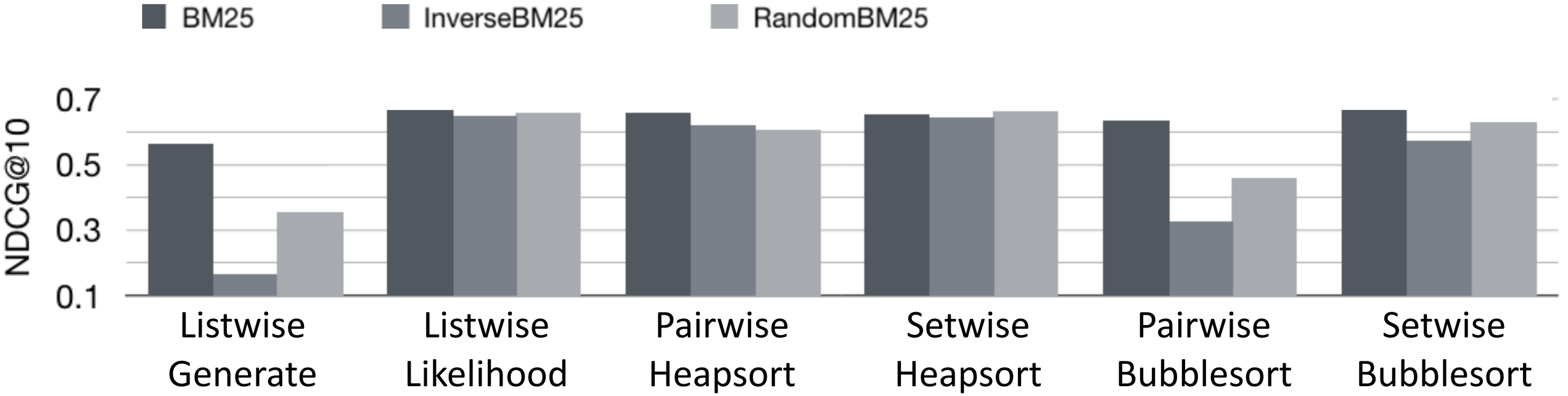
Efficiency


Listwise generation
Listwise likelihood
100
67
33
0
Latency (s)
Large
XL
XXL
FLAN-T5

Effectiveness



Sensitivity to the Initial Ranking



Properties

Methods	Logits	Generate
pointwise.qlm	✓	
pointwise.yes_no	✓	
listwise.generation		✓
listwise.likelihood	✓	
pairwise.allpair	✓	✓
pairwise.heapsort	✓	✓
pairwise.bubblesort	✓	✓
setwise.heapsort	✓	✓
setwise.bubblesort	✓	✓

Properties

Methods	Logits	Generate	Batching
pointwise.qlm	✓		✓
pointwise.yes_no	✓		✓
listwise.generation		✓	
listwise.likelihood	✓		
pairwise.allpair	✓	✓	✓
pairwise.heapsort	✓	✓	
pairwise.bubblesort	✓	✓	
setwise.heapsort	✓	✓	
setwise.bubblesort	✓	✓	

Properties

Methods	Logits	Generate	Batching	Top-k
pointwise.qlm	✓		✓	
pointwise.yes_no	✓		✓	
listwise.generation		✓		✓
listwise.likelihood	✓			✓
pairwise.allpair	✓	✓	✓	
pairwise.heapsort	✓	✓		✓
pairwise.bubblesort	✓	✓		✓
setwise.heapsort	✓	✓		✓
setwise.bubblesort	✓	✓		✓

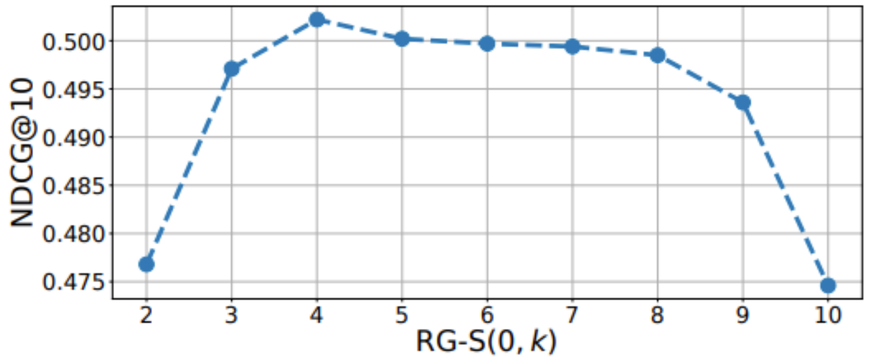
Properties

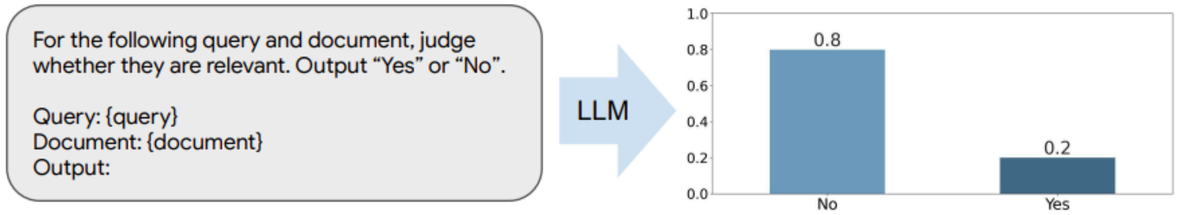
Methods	Logits	Generate	Batching	Top-k	# LLM calls
pointwise.qlm	✓		✓		$O(N)$
pointwise.yes_no	✓		✓		$O(N)$
listwise.generation		✓		✓	$O(r * (N/s))$
listwise.likelihood	✓			✓	$O(r * (N/s))$
pairwise.allpair	✓	✓	✓		$O(N^2 - N)$
pairwise.heapsort	✓	✓		✓	$O(k * \log_2 N)$
pairwise.bubblesort	✓	✓		✓	$O(k * N)$
setwise.heapsort	✓	✓		✓	$O(k * \log_c N)$
setwise.bubblesort	✓	✓		✓	$O(k * (N/(c-1)))$

Zeroshot LLM-based Rankers' Variations

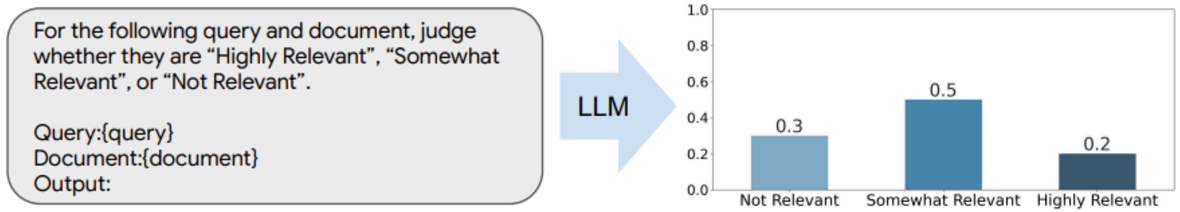
Pointwise Variation

- Beyond yes and no.

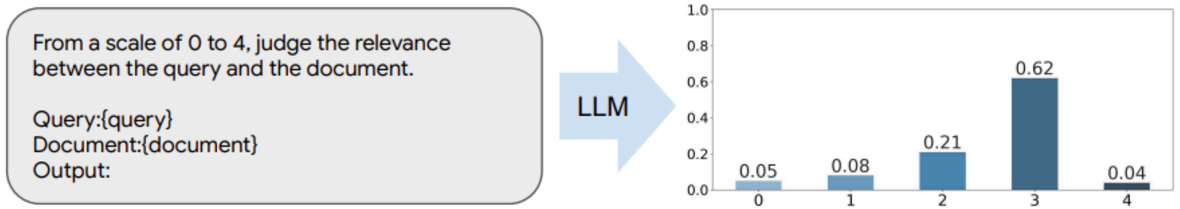




(a) Yes-No relevance generation



(b) Fine-grained relevance label generation

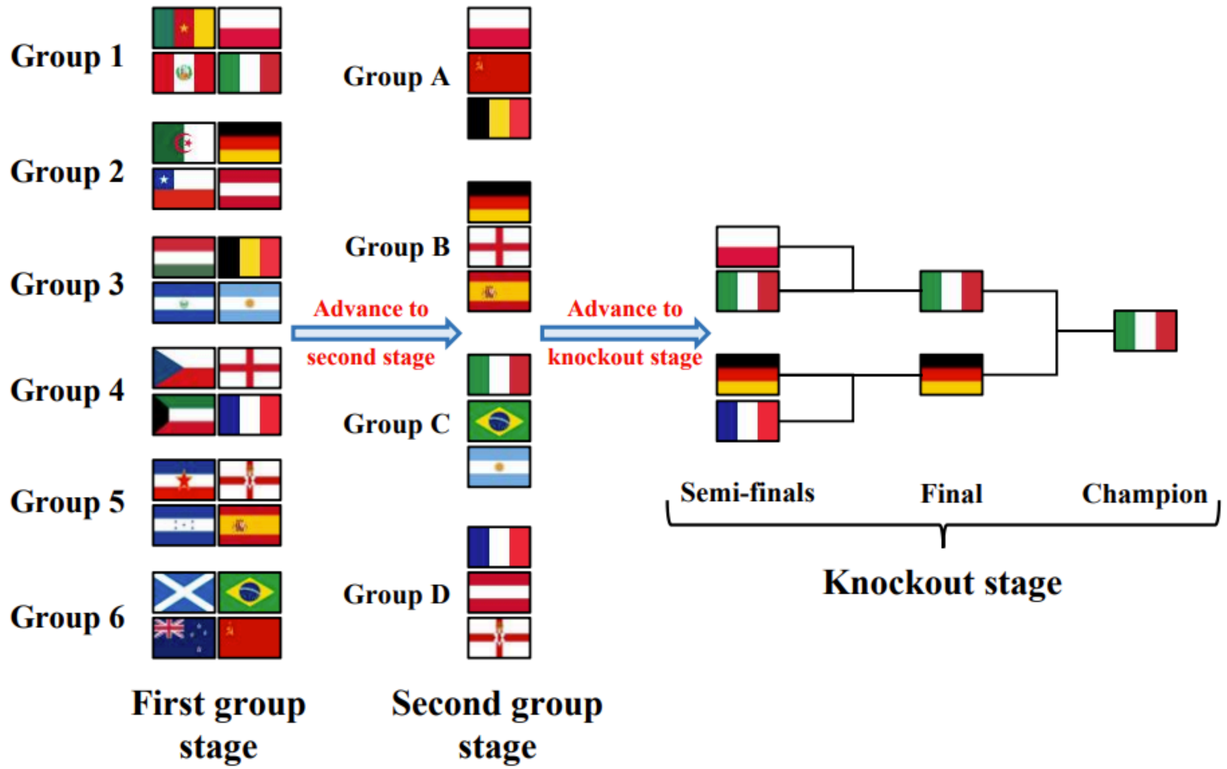


(c) Rating scale relevance generation

Zhuang, H., Qin, Z., Hui, K., Wu, J., Yan, L., Wang, X., Bendersky, M. 2024. Beyond Yes and No: Improving Zero-Shot LLM Rankers via Scoring Fine-Grained Relevance Labels, NAACL

TourRank

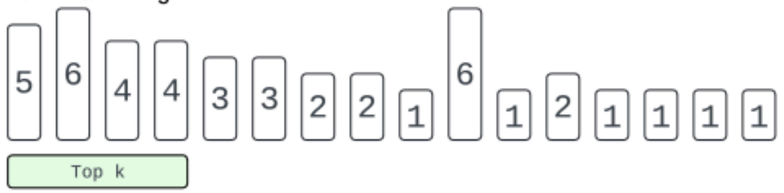
- Sport tournaments scoring



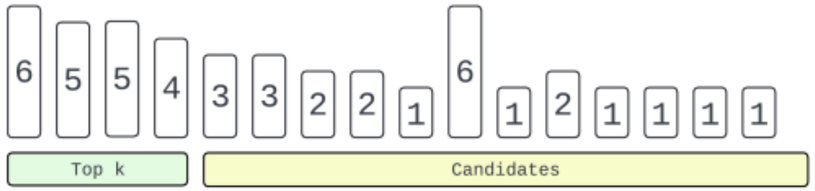
Chen, Y., et al. 2025. TourRank: Utilizing Large Language Models for Documents Ranking with a Tournament-Inspired Strategy, TheWebConf

Setwise Insertion sort

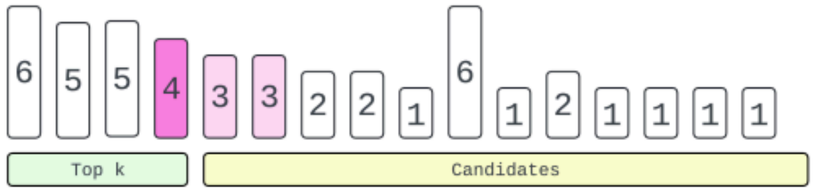
0. Start with a partially sorted list based on an initial ranking.



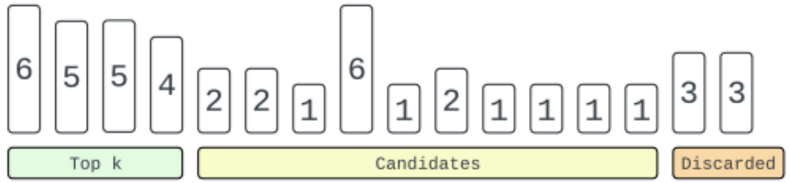
1. Use setwise.heapsort to sort the top-k elements.



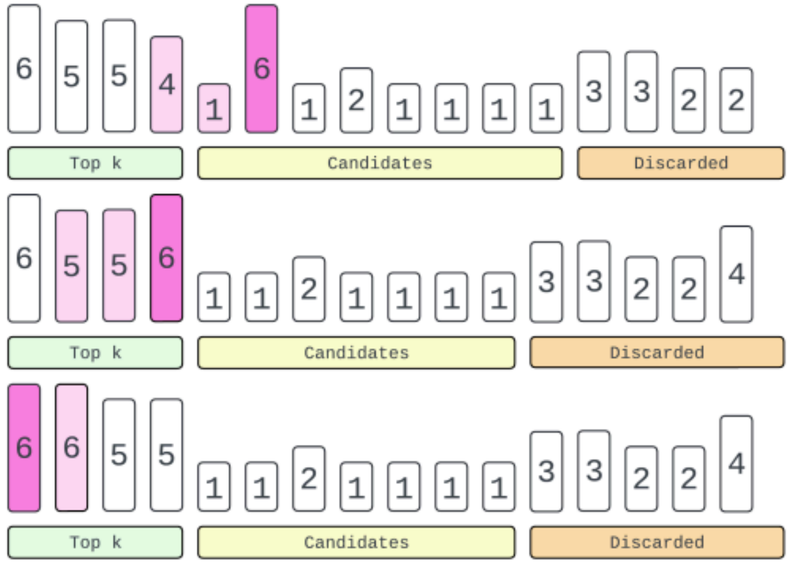
2. Compare small candidate sets with the smallest element in top-k (the "guard").



3. Discard candidates if the guard is the largest.



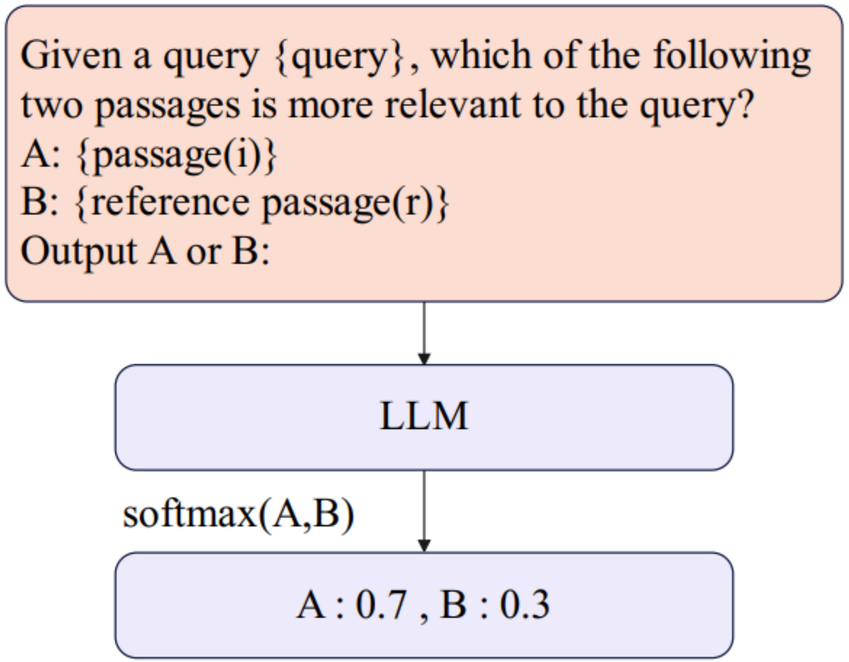
4. If at least one candidate is larger than the guard, discard the guard, insert the candidate into top k, and fix the order.

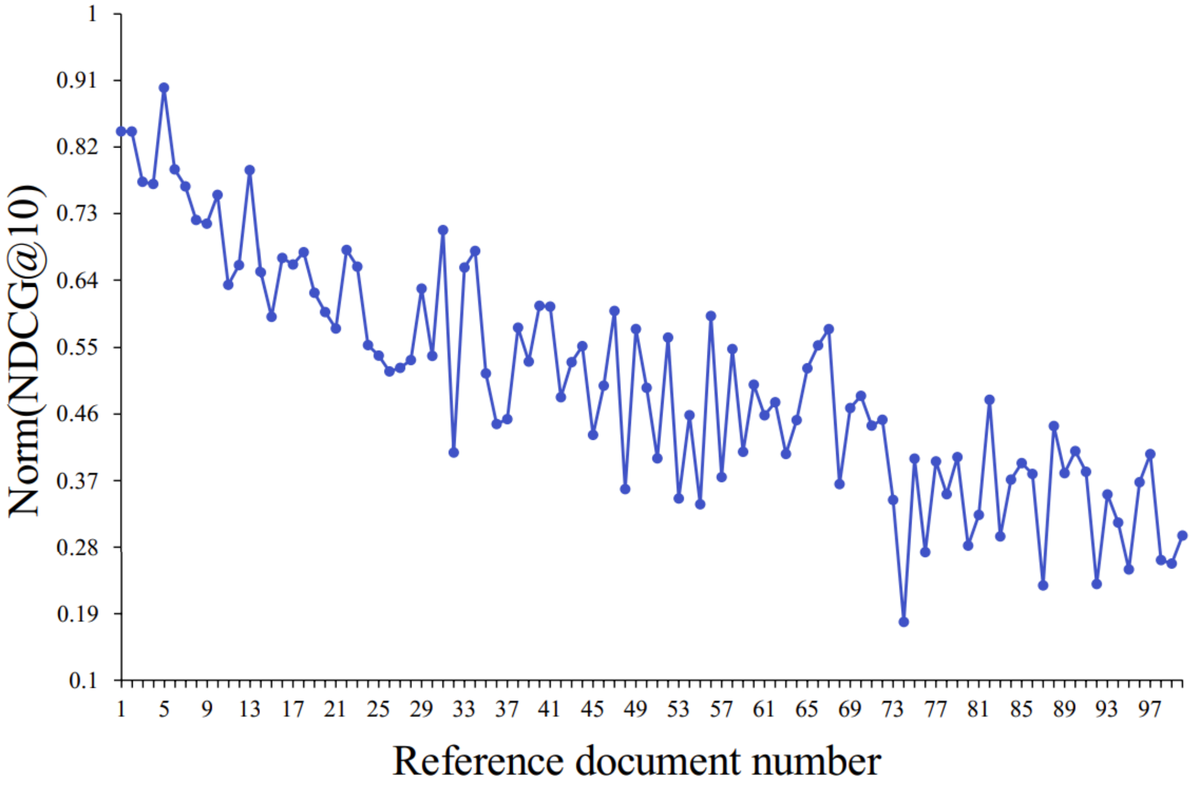


5. Repeat steps 2–4 until the candidate list is empty.

Podolak, J., Peric, L., Janicijevic, M., Petcu, R. 2025. Beyond Reproducibility: Advancing Zero-shot LLM Reranking Efficiency with Setwise Insertion. SIGIR

Combine Pointwise and Pairwise





Li, J., Hu, X., Zhao, Y., Zhuang, S., Zhang, H. 2025. Beyond Reproducibility: Advancing Zero-shot LLM Reranking Efficiency with Setwise Insertion. arXiv
Long, K., Li, S., Xu, C., Tang, J., Wang, T. 2025. Precise Zero-Shot Pointwise Ranking with LLMs through Post-Aggregated Global Context Information. SIGIR
Wisznia, J., Bolanos, C., Tollo, J., Marraffini, G., Gianolini, A., Hsueh, N., Corro, L. 2025, Are Optimal Algorithms Still Optimal? Rethinking Sorting in LLM-Based Pairwise Ranking with Batching and Caching. ACL

Impact of Prompt Variations on LLM Rankers

- Prompts proposed for different LLM ranker methods vary largely
- Not just in terms of instruction for ranking, but also for (unrelated) additional wordings, e.g. role playing, and ordering of components (e.g. passage first, then query — or vice versa?)
- What are the effects of prompt wordings on methods? What makes a good prompt?

Method	Prompt
PRP, Qin et al.	Passage: {text} Query: {query} Does the passage answer the query?
RankGPT, Sun et al.	You are RankGPT, an intelligent assistant that can rank passages based on their relevancy to the query. I will provide you with num passages, each indicated by number identifier []. Rank the passages based on their relevance to query: {query}. {PASSAGES} Search Query: {query}. Rank the num passages above based on their relevance to the search query. The passages should be listed in descending order using identifiers. The most relevant passages should be listed first. The output format should be [] > [], e.g., [1] > [2]. Only response the ranking results, do not say any word or explain.

Role Playing

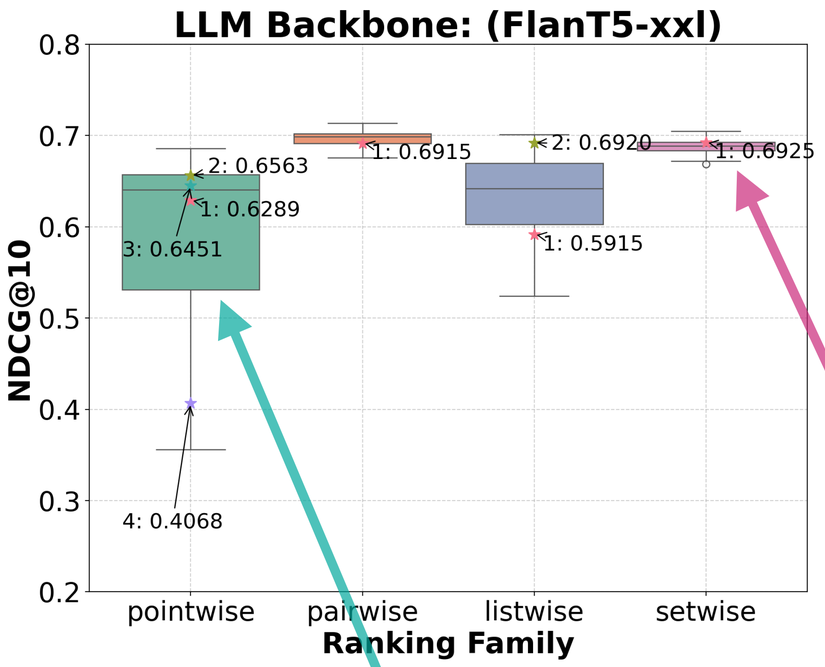
Formatting Instr.

Restriction on Output

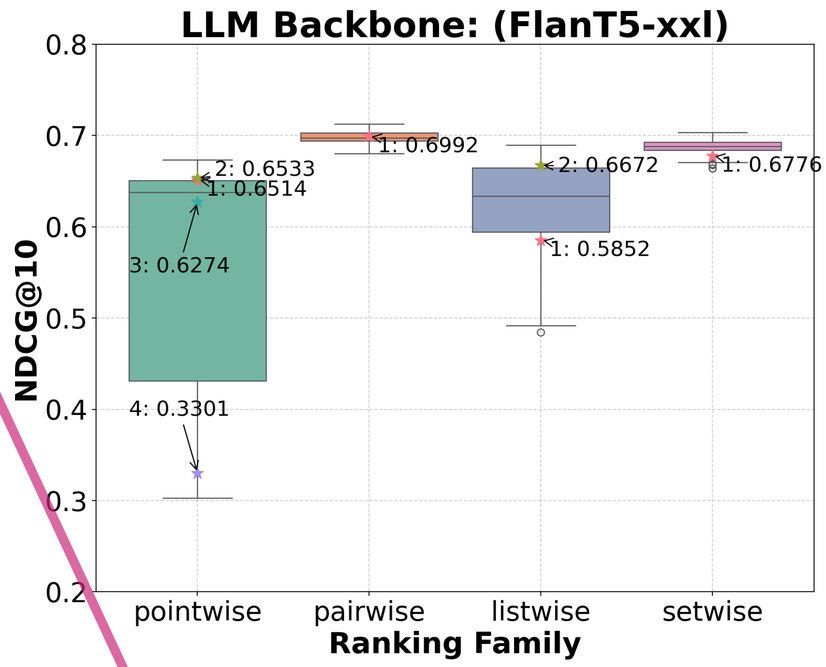
Sun, S., Zhuang, S., Wang, S. and Zuccon, G., 2024. An Investigation of Prompt Variations for Zero-shot LLM-based Rankers. *ECIR*

Impact of Prompt Variations on LLM Rankers

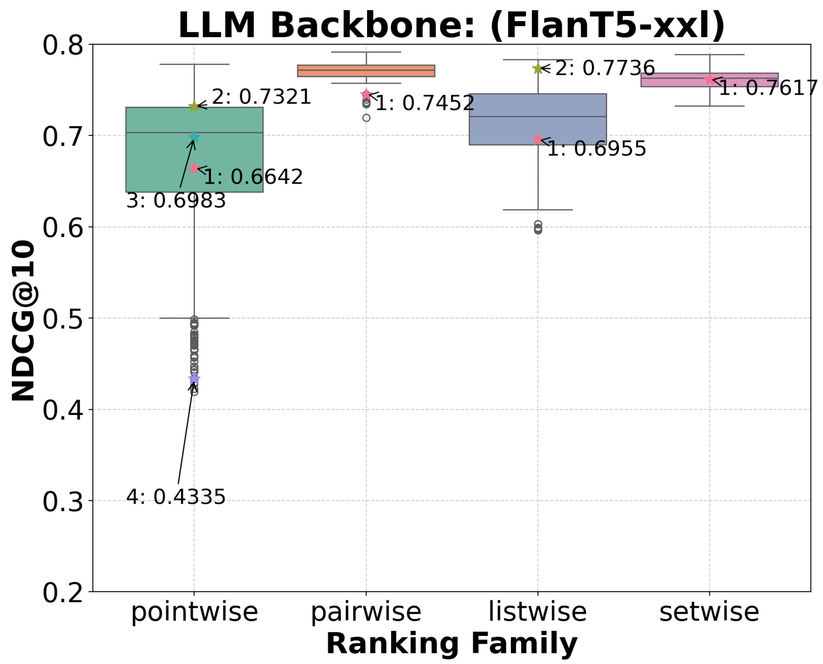
TREC DL 2019



TREC DL 2020



COVID (BEIR)



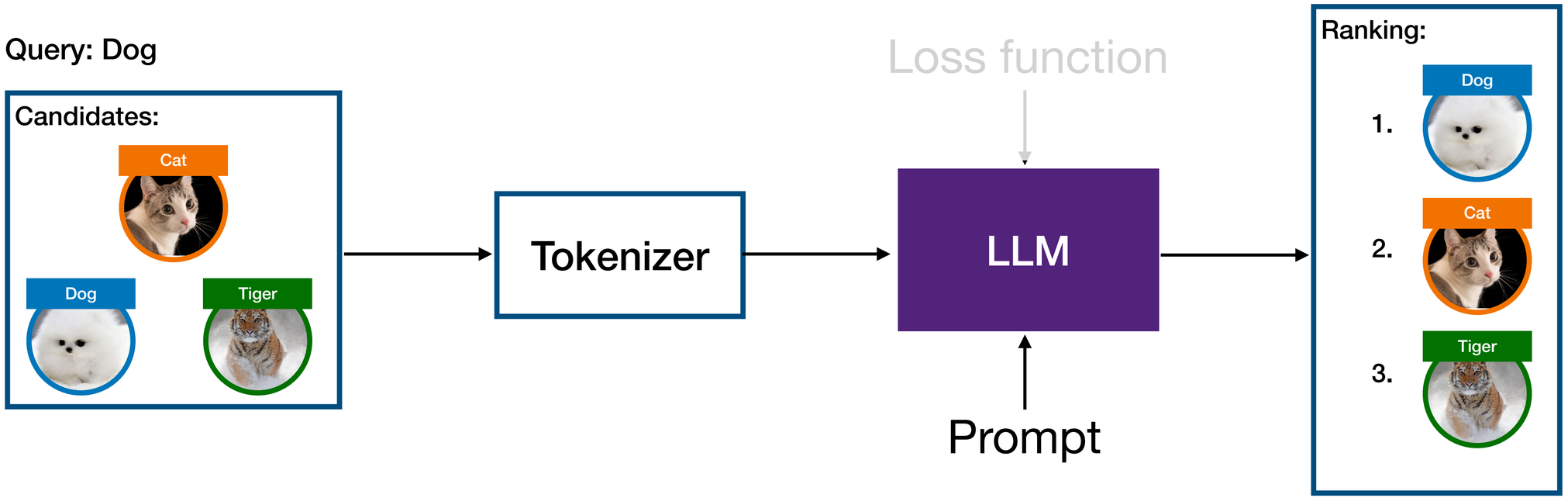
Pointwise, listwise often display large variations of effectiveness across different prompts

Pairwise, set wise very robust across different prompts

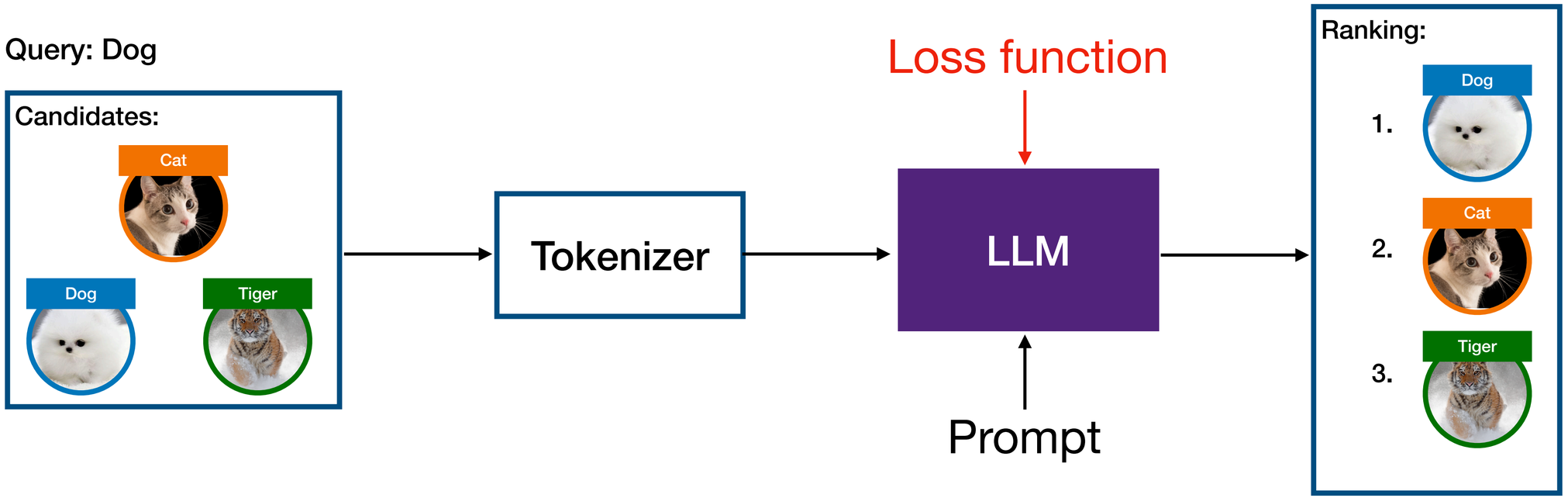
For all approaches, there are better prompts than the one in the original paper

Sun, S., Zhuang, S., Wang, S. and Zuccon, G., 2024. An Investigation of Prompt Variations for Zero-shot LLM-based Rankers. *ECIR*

LLM as Rankers



LLM as Rankers



Distillation

- RankZephyr: 7B listwise ranker distilled from large GPT Models, outperforms GPT Models

Pradeep, R., Sharifymoghaddam, S., Lin, J. 2023. RankZephyr: Effective and Robust Zero-Shot Listwise Reranking is a Breeze!. arXiv

Distillation

- RankZephyr: 7B listwise ranker distilled from large GPT Models, outperforms GPT Models
  - Stage1: 100k training queries from MS MARCO, with rankings provided by ChatGPT 3.5.
    - Excludes malformed generations.
    - Has randomly shuffled input order.

Pradeep, R., Sharifymoghaddam, S., Lin, J. 2023. RankZephyr: Effective and Robust Zero-Shot Listwise Reranking is a Breeze!. arXiv

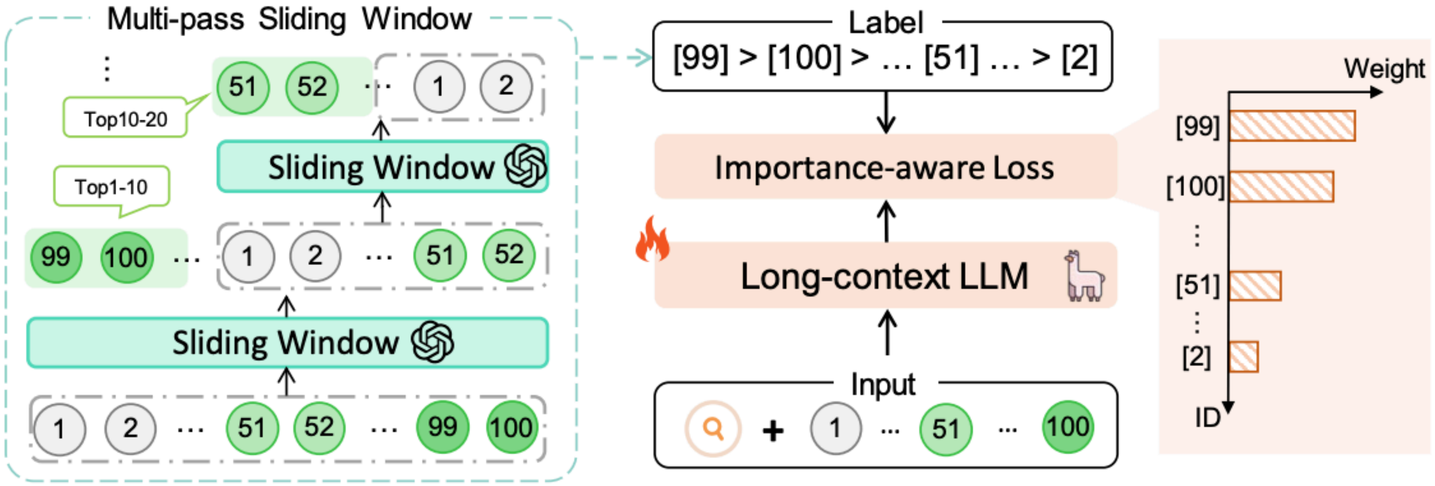
Distillation

- RankZephyr: 7B listwise ranker distilled from large GPT Models, outperforms GPT Models
  - Stage1: 100k training queries from MS MARCO, with rankings provided by ChatGPT 3.5.
    - Excludes malformed generations.
    - Has randomly shuffled input order.
  - Stage2: 5k queries with rankings from GPT-4.5.
    - Diverse queries to maximize query embedding differences.

Pradeep, R., Sharifymoghaddam, S., Lin, J. 2023. RankZephyr: Effective and Robust Zero-Shot Listwise Reranking is a Breeze!. arXiv

Distillation

- Sliding Windows vs Full ranking distillation.

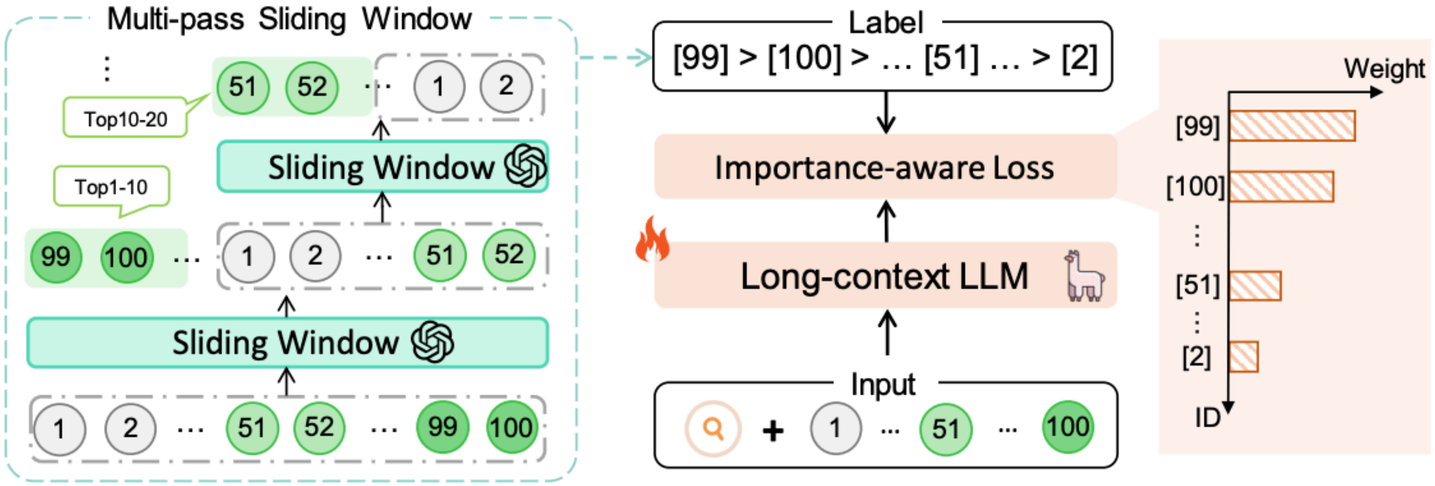


Models	Strategy	DL19	DL20	Covid	DBPedia	SciFact	NFCorpus	Signal	Robust04	Touche	News	Avg.
Zero-shot												
GPT-4o	Sliding	**74.78**	69.52	**83.41**	**45.56**	**77.41**	**39.67**	**34.20**	**60.25**	32.26	**51.92**	**53.09**
	Full	73.94	**70.03**	82.10	43.31	74.85	39.00	32.63	55.95	30.42	47.96	50.78
SFT from GPT-4o												
RankMistral_{20}	Sliding	70.34	69.58	80.86	42.52	75.82	38.38	33.90	54.69	**37.18**	**51.42**	51.85
RankMistral_{100}	Full	**72.55**	**71.29**	**82.24**	**43.54**	**77.04**	**39.14**	**33.99**	**57.91**	34.63	50.59	**52.40**

Liu W, Ma, X., Zhu, Y., Zhao, Z., Wang, S., Yin, D., Dou, Z. 2025. Sliding Windows Are Not the End: Exploring Full Ranking with Long-Context Large Language Models. ACL

Distillation

- Sliding Windows vs Full ranking distillation.

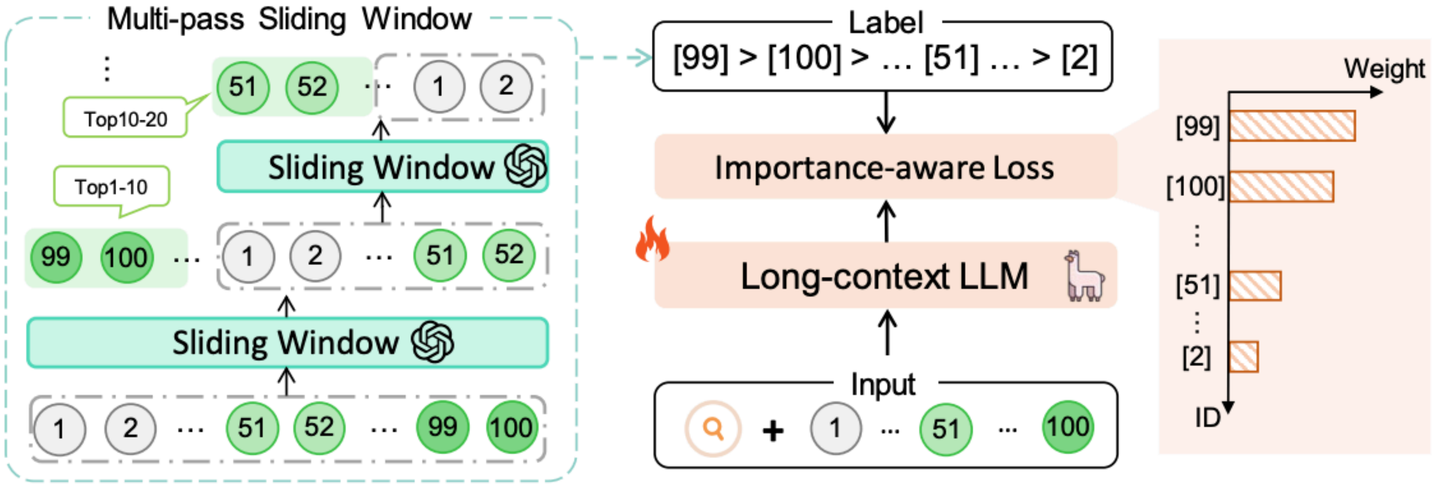


Models	Strategy	DL19	DL20	Covid	DBPedia	SciFact	NFCorpus	Signal	Robust04	Touche	News	Avg.
Zero-shot												
GPT-4o	Sliding	**74.78**	69.52	**83.41**	**45.56**	**77.41**	**39.67**	**34.20**	**60.25**	32.26	**51.92**	**53.09**
	Full	73.94	**70.03**	82.10	43.31	74.85	39.00	32.63	55.95	30.42	47.96	50.78
SFT from GPT-4o												
RankMistral_{20}	Sliding	70.34	69.58	80.86	42.52	75.82	38.38	33.90	54.69	**37.18**	**51.42**	51.85
RankMistral_{100}	Full	**72.55**	**71.29**	**82.24**	**43.54**	**77.04**	**39.14**	**33.99**	**57.91**	34.63	50.59	**52.40**

Liu W, Ma, X., Zhu, Y., Zhao, Z., Wang, S., Yin, D., Dou, Z. 2025. Sliding Windows Are Not the End: Exploring Full Ranking with Long-Context Large Language Models. ACL

Distillation

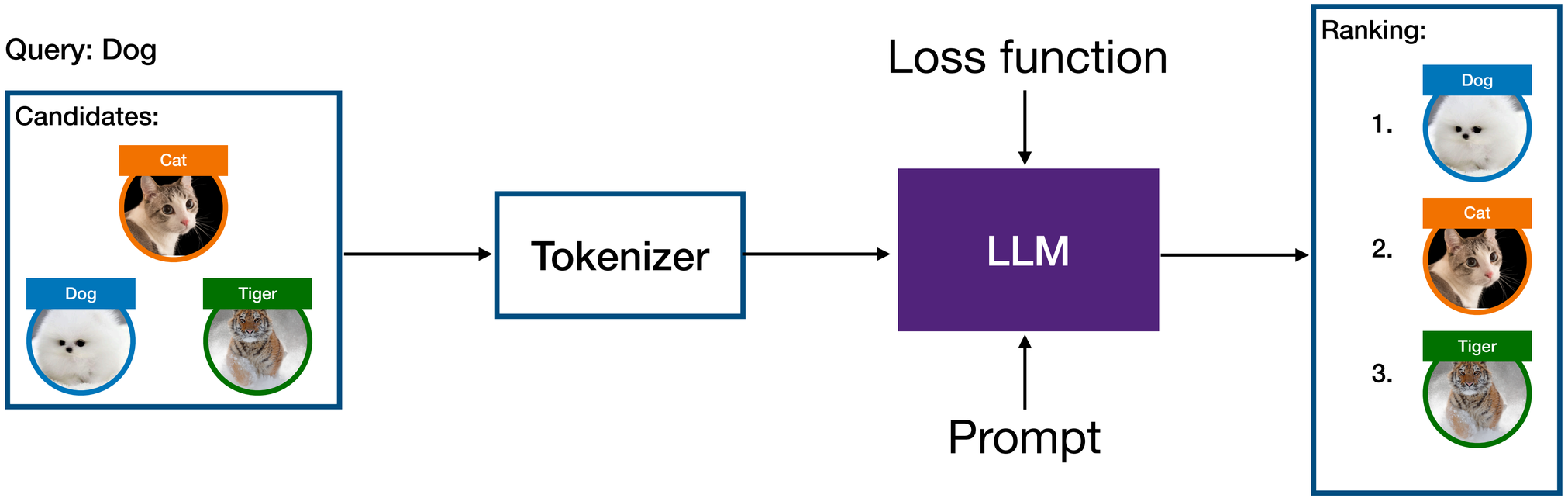
- Sliding Windows vs Full ranking distillation.



Models	Strategy	DL19	DL20	Covid	DBPedia	SciFact	NFCorpus	Signal	Robust04	Touche	News	Avg.
Zero-shot												
GPT-4o	Sliding	**74.78**	69.52	**83.41**	**45.56**	**77.41**	**39.67**	**34.20**	**60.25**	32.26	**51.92**	**53.09**
	Full	73.94	**70.03**	82.10	43.31	74.85	39.00	32.63	55.95	30.42	47.96	50.78
SFT from GPT-4o												
RankMistral_{20}	Sliding	70.34	69.58	80.86	42.52	75.82	38.38	33.90	54.69	**37.18**	**51.42**	51.85
RankMistral_{100}	Full	**72.55**	**71.29**	**82.24**	**43.54**	**77.04**	**39.14**	**33.99**	**57.91**	34.63	50.59	**52.40**

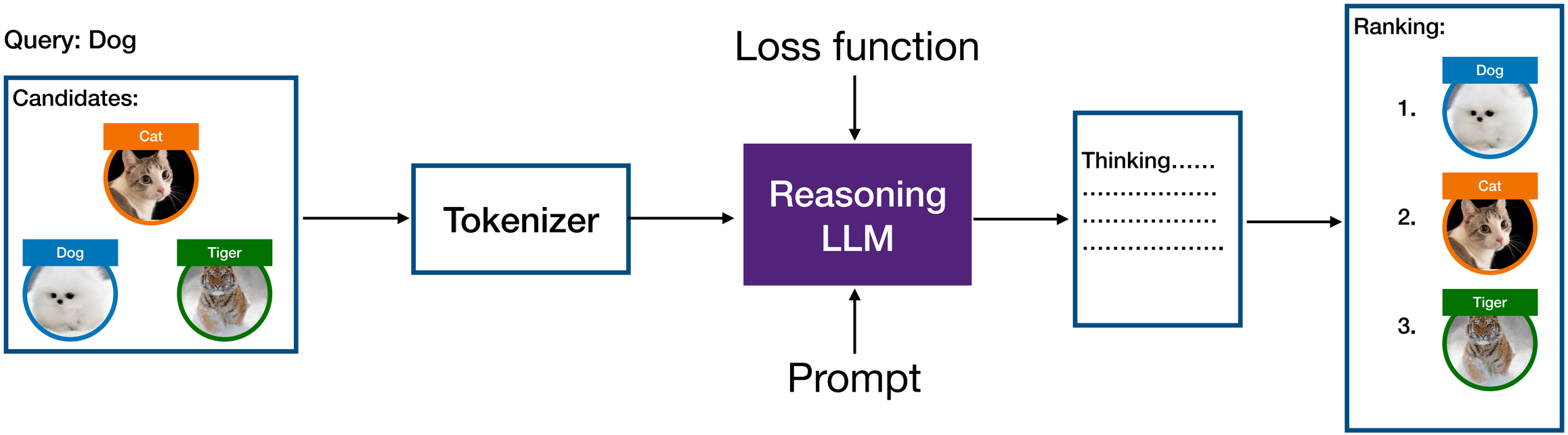
Liu W, Ma, X., Zhu, Y., Zhao, Z., Wang, S., Yin, D., Dou, Z. 2025. Sliding Windows Are Not the End: Exploring Full Ranking with Long-Context Large Language Models. ACL

LLM as Rankers



Reasoning LLMs

Reasoning Rankers



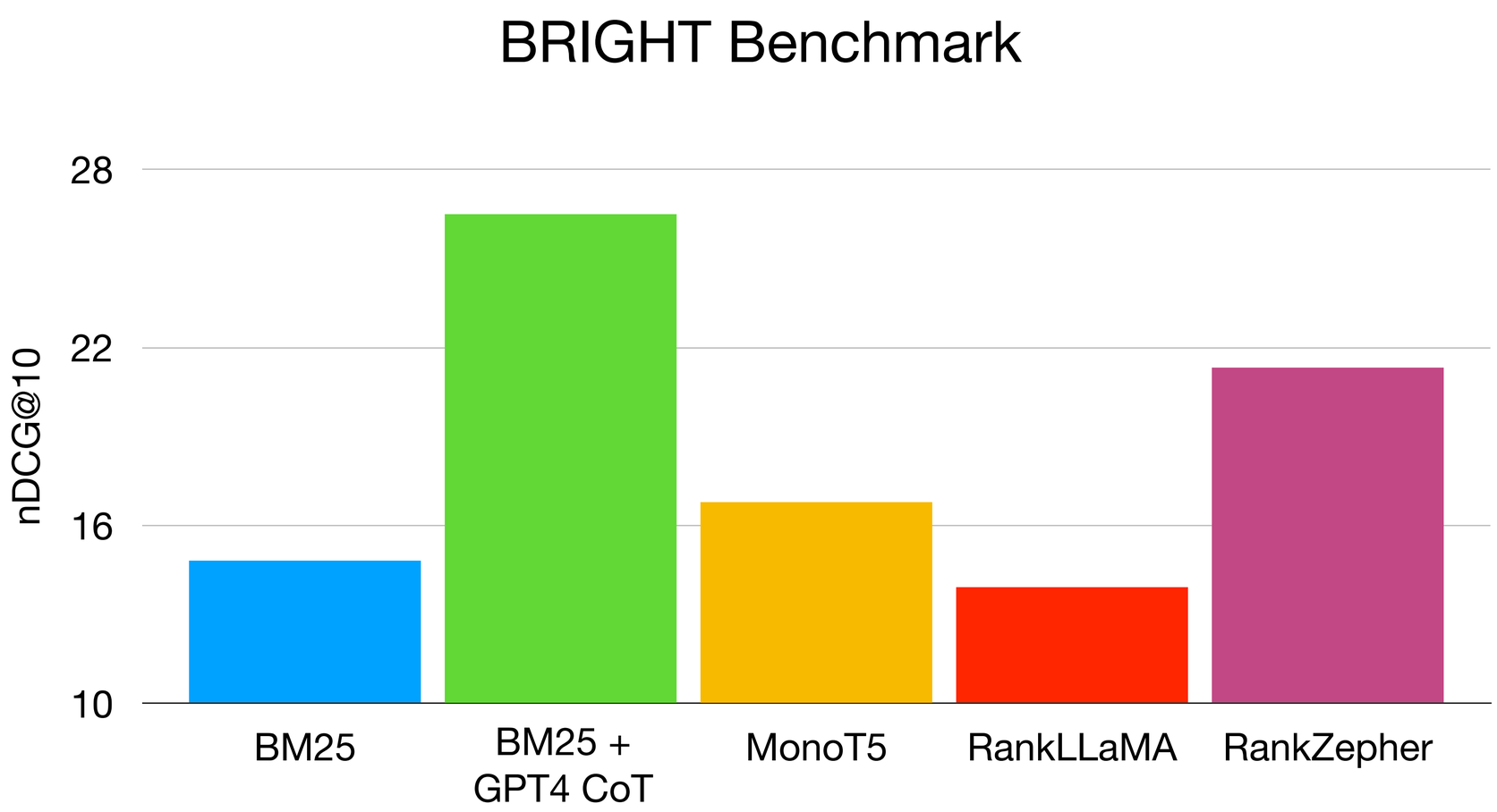
Why Reasoning?

BRIGHT Benchmark

Dataset	Total Number			Avg. Length		Source		Examples
	Q	$\mathcal{D}$	$\mathcal{D}^+$	Q	$\mathcal{D}$	Q	$\mathcal{D}$	
StackExchange								
Biology	103	57,359	3.6	115.2	83.6	StackExchange post	Web pages: article, tutorial, news, blog, report ...	Tab. 20
Earth Science	116	121,249	5.3	109.5	132.6			Tab. 21
Economics	103	50,220	8.0	181.5	120.2			Tab. 22
Psychology	101	52,835	7.3	149.6	118.2			Tab. 23
Robotics	101	61,961	5.5	818.9	121.0			Tab. 24
Stack Overflow	117	107,081	7.0	478.3	704.7			Tab. 25
Sustainable Living	108	60,792	5.6	148.5	107.9			Tab. 26
Coding								
LeetCode	142	413,932	1.8	497.5	482.6	Coding question	Coding Q&Sol	Tab. 27
Pony	112	7,894	22.5	102.6	98.3	Coding question	Syntax Doc	Tab. 28
Theorems								
AoPS	111	188,002	4.7	117.1	250.5	Math Olympiad Q	STEM Q&Sol	Tab. 29
TheoremQA-Q	194	188,002	3.2	93.4	250.5	Theorem-based Q	STEM Q&Sol	Tab. 30
TheoremQA-T	76	23,839	2.0	91.7	354.8	Theorem-based Q	Theorems	Tab. 31

Su, H., et al., 2025. BRIGHT: A REALISTIC AND CHALLENGING BENCHMARK FOR REASONING-INTENSIVE RETRIEVAL. ICLR

Why Reasoning?



Reasoning Rankers

- JudgeRank: Zeroshot pointwise reasoning ranker.

Query Analysis Prompt

You will be presented with a/an *{query name}*.

Your task consists of the following step:

1. Analyze the *{query name}*:
- Carefully read each sentence of the *{query name}*.
- Identify the core problem or question being asked.

Here is the *{query name}*:
{query}

(a). Query Analysis Prompt of JudgeRank

Document Analysis Prompt

You will be presented with a/an *{query name}*, an analysis of the query, and a/an *{doc name}*.

Your task consists of the following steps:
1. Analyze the *{doc name}*:
- Thoroughly examine each sentence of the *{doc name}*.
- List all sentences from the *{doc name}* that *{definition of relevance}* the *{query name}*.
- Briefly explain how each sentence listed *{definition of relevance}* the *{query name}*.

2. Assess overall relevance:
- If the *{doc name}*, particularly the relevant sentences (if applicable), *{definition of relevance}* the *{query name}*, briefly explain why.
- Otherwise, briefly explain why not.

Here is the *{query name}:*
{query}

Here is the analysis of the *{query name}:*
{query analysis}

Here is the *{doc name}:*
{doc}

(b). Document Analysis Prompt of JudgeRank

Judgment Prompt

You will be presented with a/an *{query name}*, an analysis of the *{query name}*, a/an *{doc name}*, and an analysis of the *{doc name}*.

Your task is to assess if the *{doc name}* *{definition of relevance}* the *{query name}* in one word:
- Yes: If the *{doc name}* *{definition of relevance}* the *{query name}*.
- No: Otherwise.

Important: Respond using only one of the following two words without quotation marks: Yes or No.

Here is the *{query name}:*
{query}

Here is the analysis of the *{query name}:*
{query analysis}

Here is the *{doc name}:*
{doc}

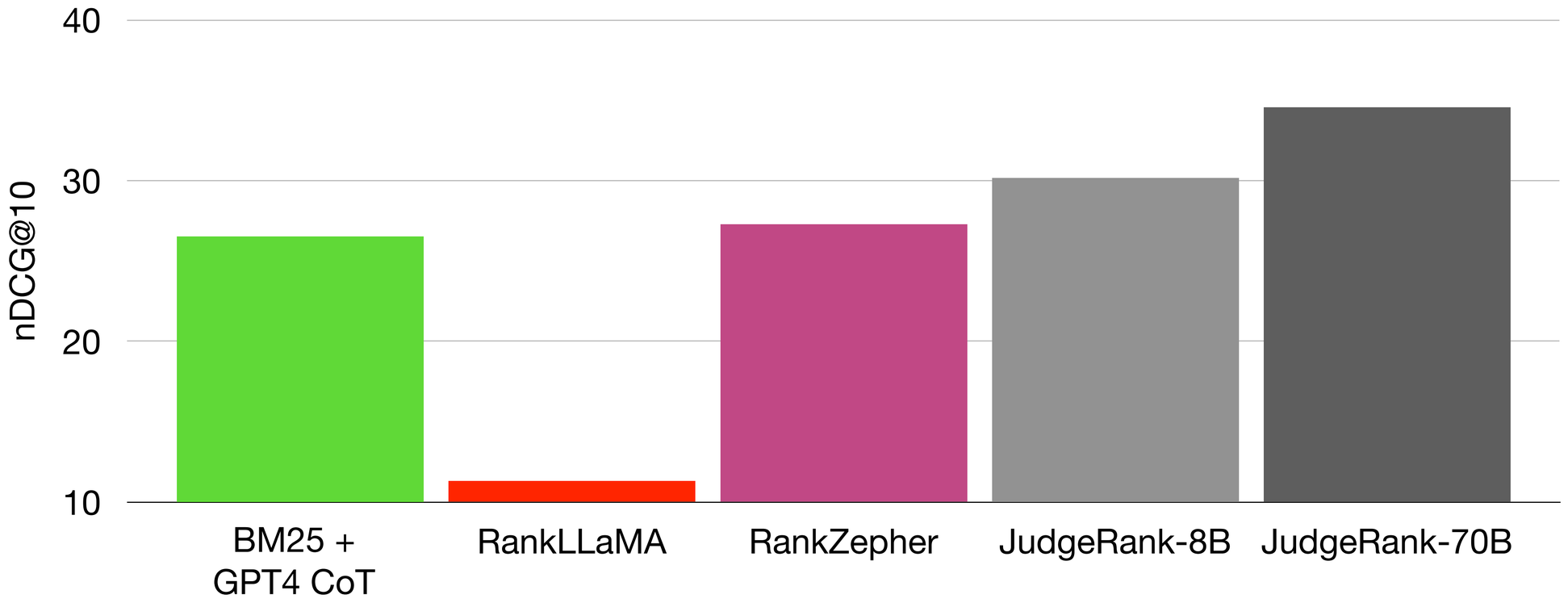
Here is the analysis of the *{doc name}:*
{doc analysis}

(c). Judgment Prompt of JudgeRank

Niu, T., Joty, S., Liu, Y., Xiong, C., Zhou, Y., Yavuz, S. 2024. JudgeRank: Leveraging Large Language Models for Reasoning-Intensive Reranking. arXiv

Reasoning Rankers

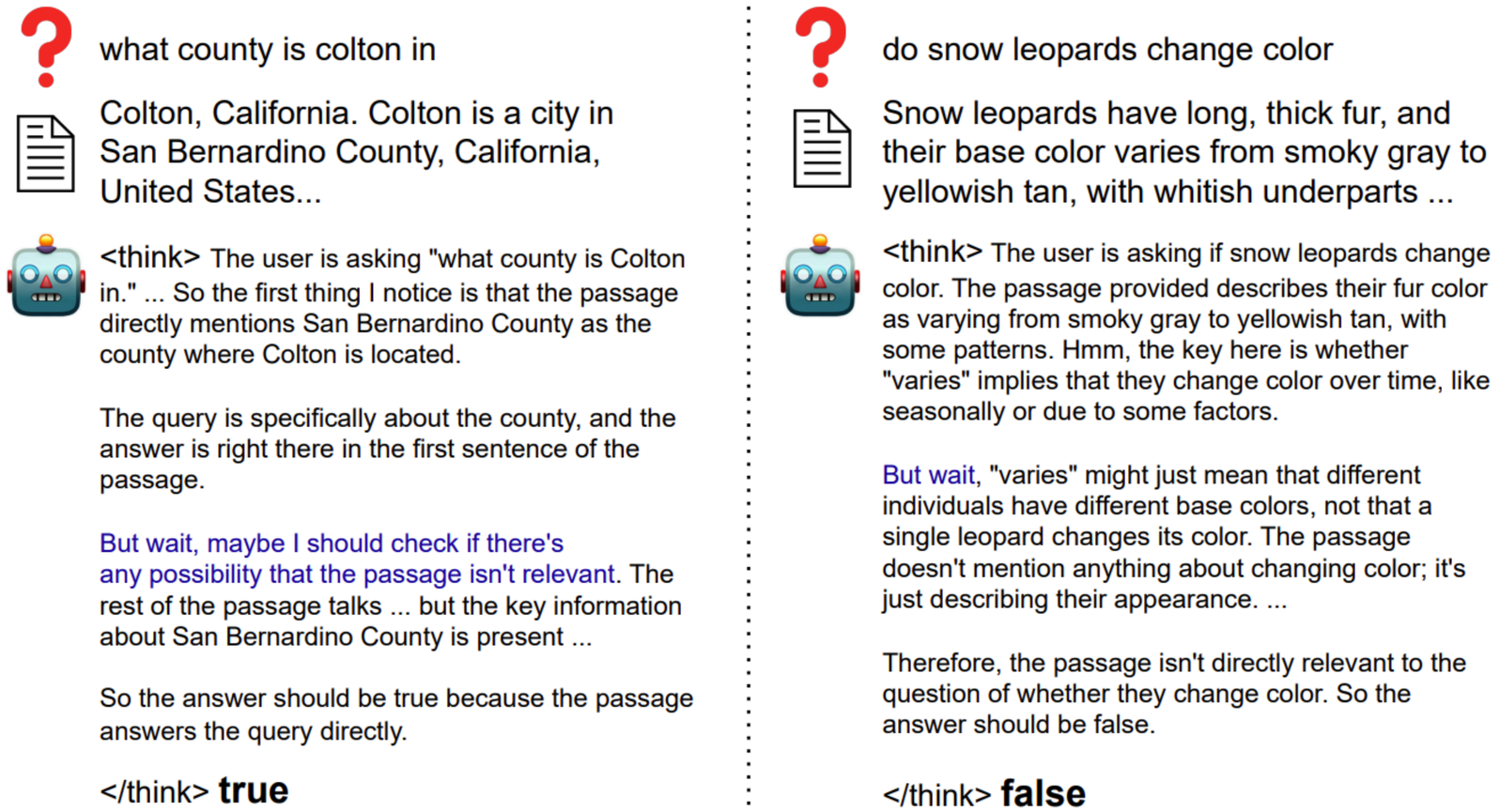
- JudgeRank: BRIGHT results



Niu, T., Joty, S., Liu, Y., Xiong, C., Zhou, Y., Yavuz, S. 2024. JudgeRank: Leveraging Large Language Models for Reasoning-Intensive Reranking. arXiv

Reasoning Rankers

- Rank1: Pointwise reasoning ranker distilled from DeepSeek-R1



Weller, O., Ricci, K., Yang, E., Yates, A., Lawrie, D., Durme, B. 2025, Rank1: Test-Time Compute for Reranking in Information Retrieval. arXiv

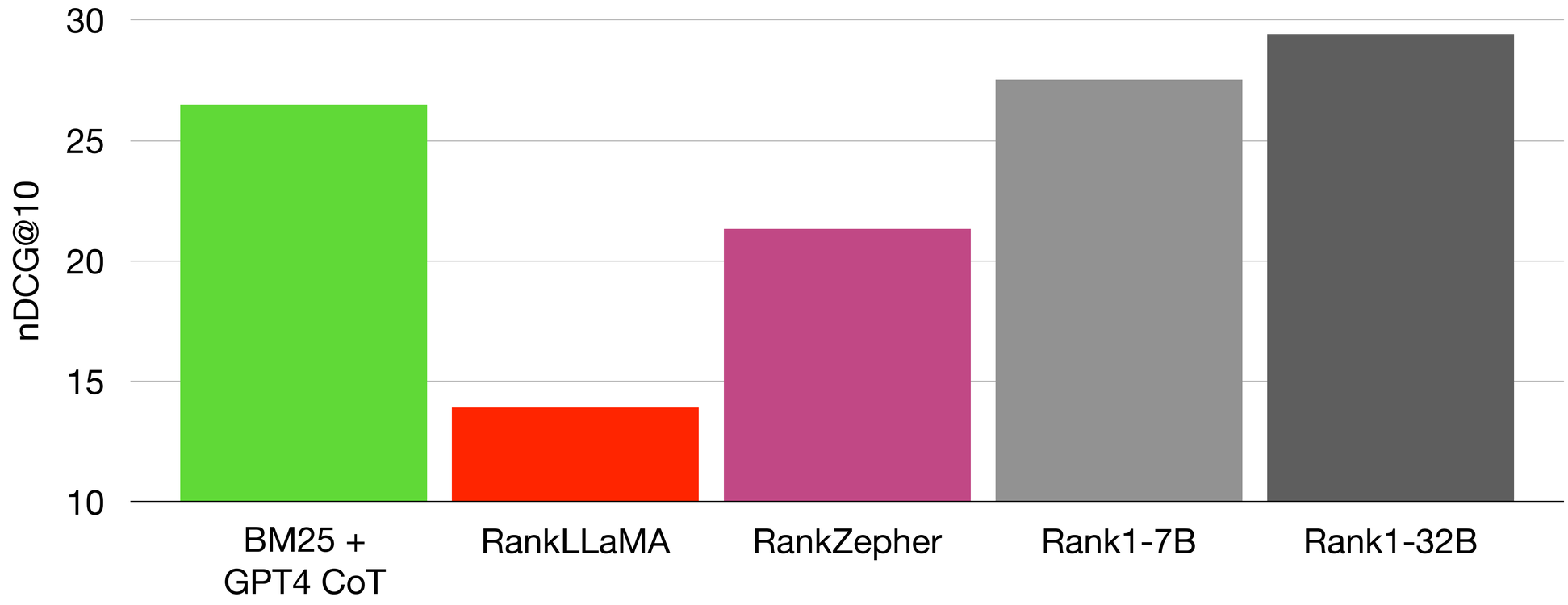
Reasoning Rankers

- Rank1: Pointwise reasoning ranker distilled from DeepSeek-R1
  - Positive and negative data points from MSMARCO.
    - Labelled postive docs, hard/easy negatives from retrievers and rerankers
  - Extensively filtered.
    - R1 disagreed on positives and easy negatives.
    - A model trained on first round to filter out disagreed hard negatives.

Weller, O., Ricci, K., Yang, E., Yates, A., Lawrie, D., Durme, B. 2025, Rank1: Test-Time Compute for Reranking in Information Retrieval. arXiv

Reasoning Rankers

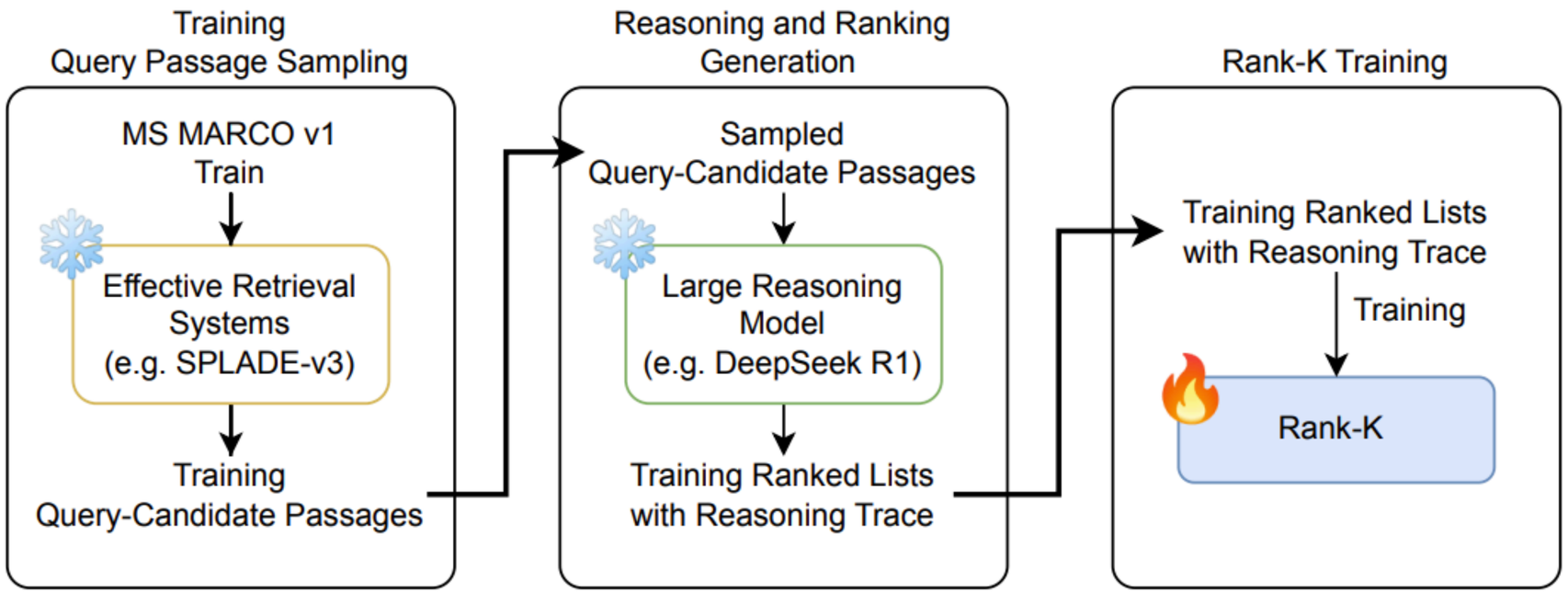
- Rank1: BRIGHT results



Weller, O., Ricci, K., Yang, E., Yates, A., Lawrie, D., Durme, B. 2025, Rank1: Test-Time Compute for Reranking in Information Retrieval. arXiv

Reasoning Rankers

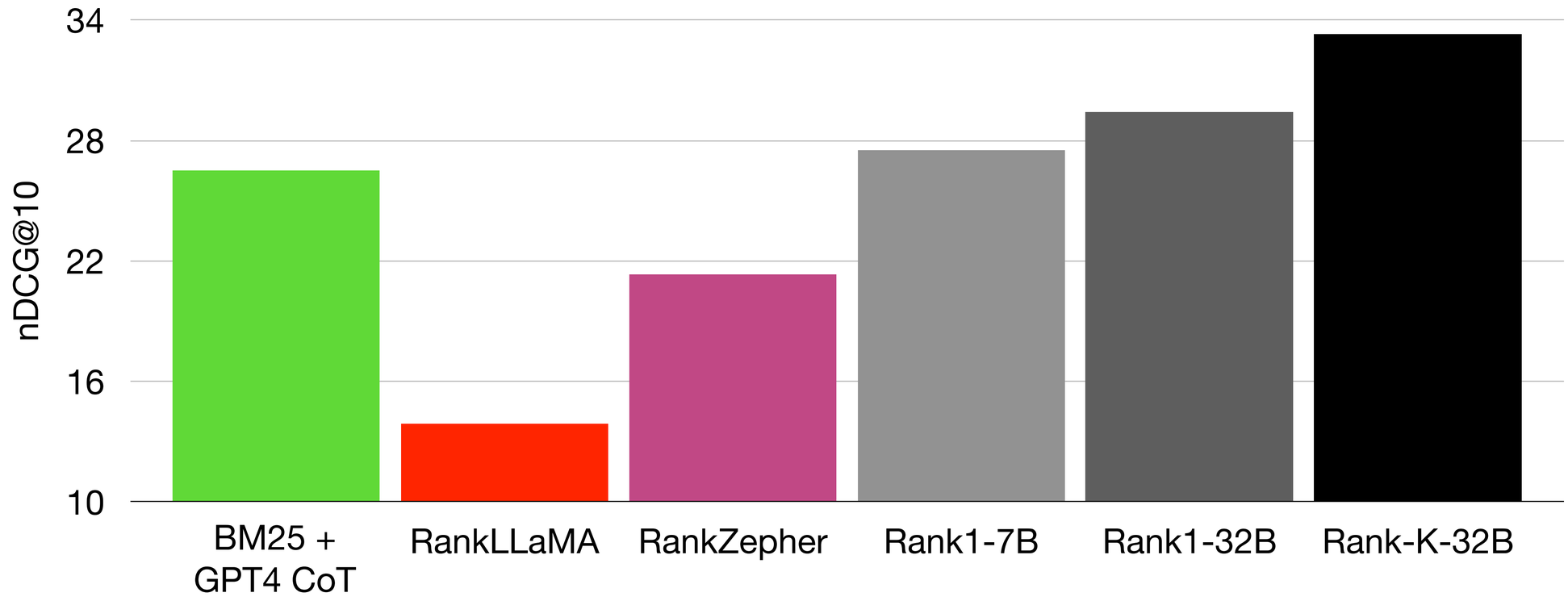
- Rank-K: Listwise reasoning ranker distilled from DeepSeek-R1



Yang, E., Yates, A., Ricci, K., Weller, O., Chari, V., Durme, B., Lawrie, D. 2025, Rank-K: Test-Time Reasoning for Listwise Reranking. arXiv

Reasoning Rankers

- Rank-K: BRIGHT results



Yang, E., Yates, A., Ricci, K., Weller, O., Chari, V., Durme, B., Lawrie, D. 2025, Rank-K: Test-Time Reasoning for Listwise Reranking. arXiv

Reasoning Rankers

- Rank-R1: Reinforcement Learning trained setwise reasoning ranker

Given a query {*query*}, which of the following passages is more relevant one to the query?
Passage A: {*passage_1*}
Passage B: {*passage_2*}
Passage C: {*passage_3*}
…
Output only the passage label of the most relevant passage:

SYSTEM:
A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer. The reasoning process and answer are enclosed within <think> </think> and <answer> </answer> tags, respectively, i.e., <think> reasoning process here </think> <answer> answer here </answer>.

USER:
Given the query: "{query}", which of the following documents is most relevant?
[1] {document1}
[2] {document2}
....
[20] {document20}
After completing the reasoning process, please provide only the label of the most relevant document to the query, enclosed in square brackets, within the answer tags. For example, if the third document is the most relevant, the answer should be: <think> reasoning process here </think> <answer>[3]</answer>.

Reasoning Rankers

- Rank-R1: Reinforcement Learning trained setwise reasoning ranker

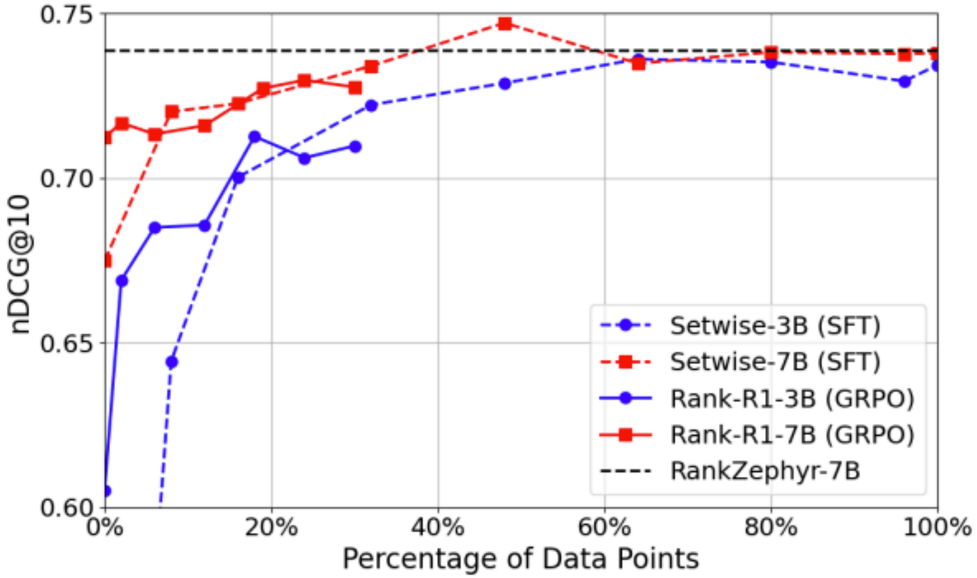
  - GRPO RL algorithm.
  - MSMARCO training data.
  - Reward: 1 if the labelled relevant document is selected and the format matches <think> </think> <answer> </answer>, otherwise 0

SYSTEM:
A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer. The reasoning process and answer are enclosed within <think> </think> and <answer> </answer> tags, respectively, i.e., <think> reasoning process here </think> <answer> answer here </answer>.

USER:
Given the query: "{query}", which of the following documents is most relevant?
[1] {document1}
[2] {document2}
....
[20] {document20}
After completing the reasoning process, please provide only the label of the most relevant document to the query, enclosed in square brackets, within the answer tags. For example, if the third document is the most relevant, the answer should be: <think> reasoning process here </think> <answer>[3]</answer>.

Reasoning Rankers

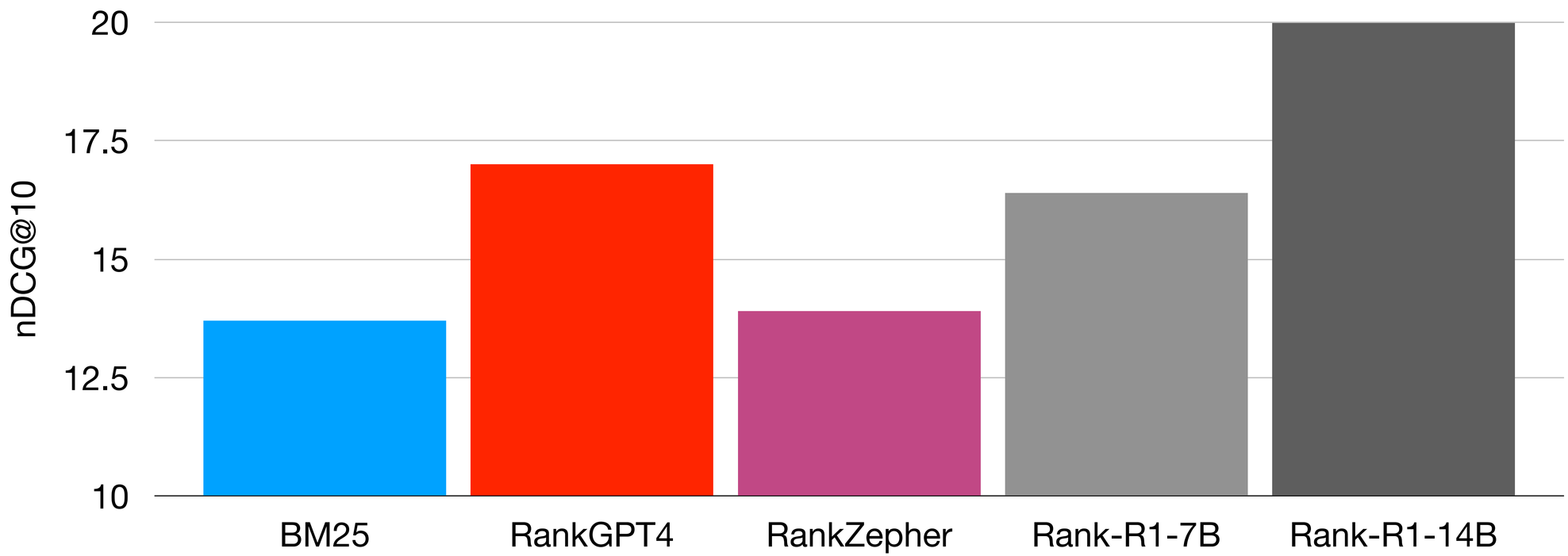
- Rank-R1: Reinforcement Learning trained setwise reasoning ranker



Zhuang, S., Ma, X., Koopman, B., Lin, J., Zuccon, G. 2025. Rank-R1: Enhancing Reasoning in LLM-based Document Rerankers via Reinforcement Learning. arXiv

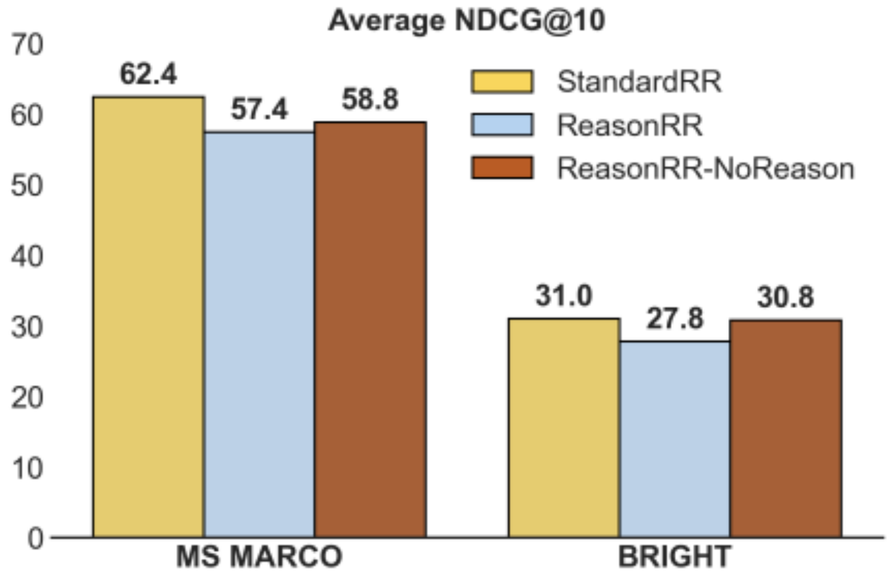
Reasoning Rankers

- Rank-R1: BRIGHT results



Zhuang, S., Ma, X., Koopman, B., Lin, J., Zuccon, G. 2025. Rank-R1: Enhancing Reasoning in LLM-based Document Rerankers via Reinforcement Learning. arXiv

Is Reasoning Truly Necessary for Ranking?



	StackExchange							Coding		Theorem-based			Avg.
	Bio.	Earth.	Econ.	Psy.	Rob.	Stack.	Sus.	Leet.	Pony	AoPS	TheoQ.	TheoT.	
BM25 + GPT-4 CoT	53.6	54.1	24.3	38.7	18.9	27.7	26.3	19.3	17.6	3.9	19.2	20.8	27.0
+ Qwen2.5-1.5B													
StandardRR	**37.0**	**21.7**	**16.8**	23.1	**16.1**	10.0	**26.3**	2.6	**30.6**	1.8	**16.1**	**26.1**	**19.0**
ReasonRR	32.5	20.3	12.3	**25.5**	11.1	**15.3**	23.5	**6.6**	12.3	**3.4**	10.6	13.7	15.6
+ Qwen2.5-3B													
StandardRR	**41.6**	27.1	**20.9**	31.9	**22.2**	16.9	**30.3**	**13.2**	**42.0**	2.7	16.2	30.6	**24.6**
ReasonRR	37.3	**27.8**	20.7	**33.1**	18.3	**24.3**	25.2	11.3	26.2	**4.7**	**20.7**	**34.0**	23.6
+ Qwen2.5-7B													
StandardRR	**47.1**	**38.0**	**28.1**	**44.1**	**26.1**	**29.5**	**36.5**	**19.3**	**37.5**	4.6	**22.4**	**39.4**	**31.0**
ReasonRR	47.0	35.4	24.0	35.2	20.0	25.2	31.0	15.1	36.0	**5.9**	22.2	36.6	27.8

Jedidi, N., Chuang, Y., Glass, J., Lin, 2025. Jimmy. Don't "Overthink" Passage Reranking: Is Reasoning Truly Necessary?. arXiv

Is Reasoning Truly Necessary for Ranking?

- Seems not necessary for simple queries. (Results on MS MARCO and BEIR).
- Ineffective when base model is “small”.
- What about efficiency in reasoning models?!

In this part we covered

- SFT: From monoBERT to RankLLaMA.
- Zero-shot LLM rankers: Pointwise, Pairwise, Listwise, Setwise.
- Reasoning rankers: distillation and RL

End of Part 3

...