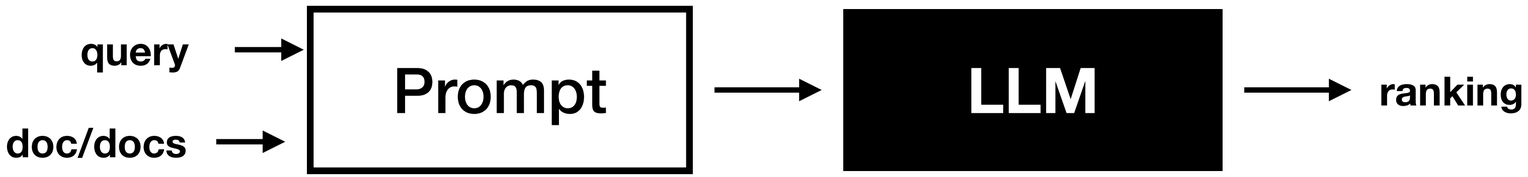
An Investigation of Prompt Variations for Zero-shot LLM-based Rankers

Shuoqi Sun[1*], Shengyao Zhuang[2], Shuai Wang[3], **Guido Zuccon**[3]

[1] RMIT University, Australia
[2] CSIRO, Australia
[3] The University of Queensland, Australia

[*] This work was conducted while Shuoqi Sun was a student at The University of Queensland.

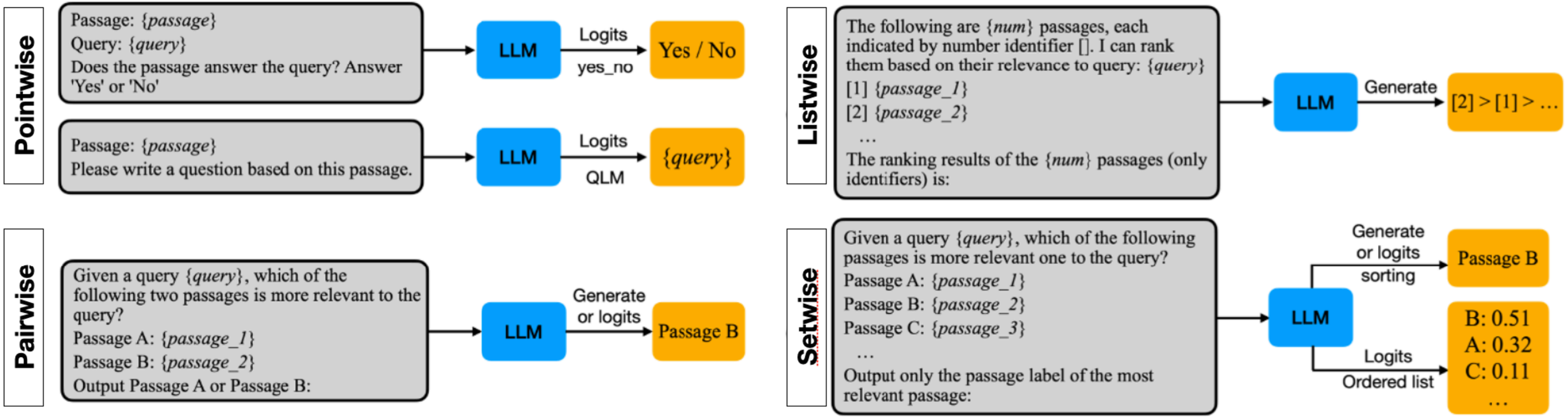
LLM Rankers: Prompting LLMs to Rank Documents



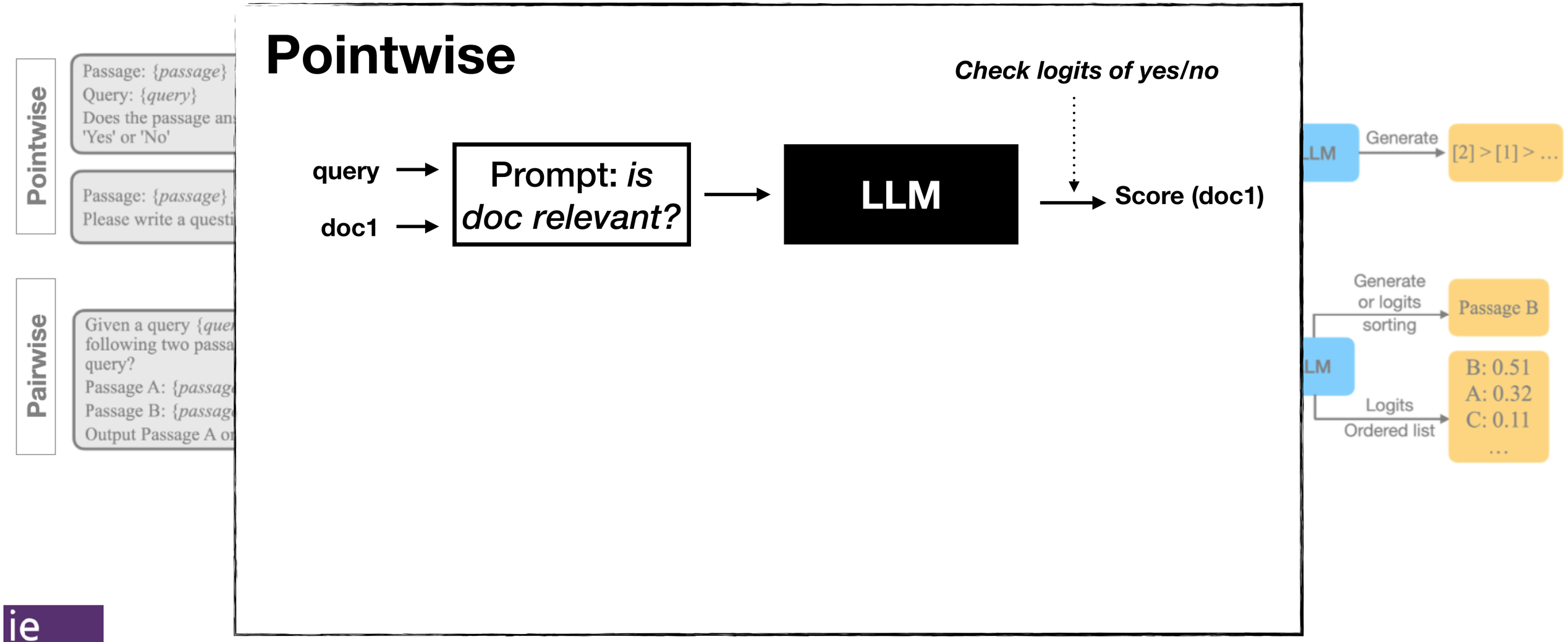
- All are “zero-shot”: i.e. once you obtained the pre-trained, instruction tuned LLM, no need to do SFT/contrastive training or RL
  - (Thought training is possible)

ie lab

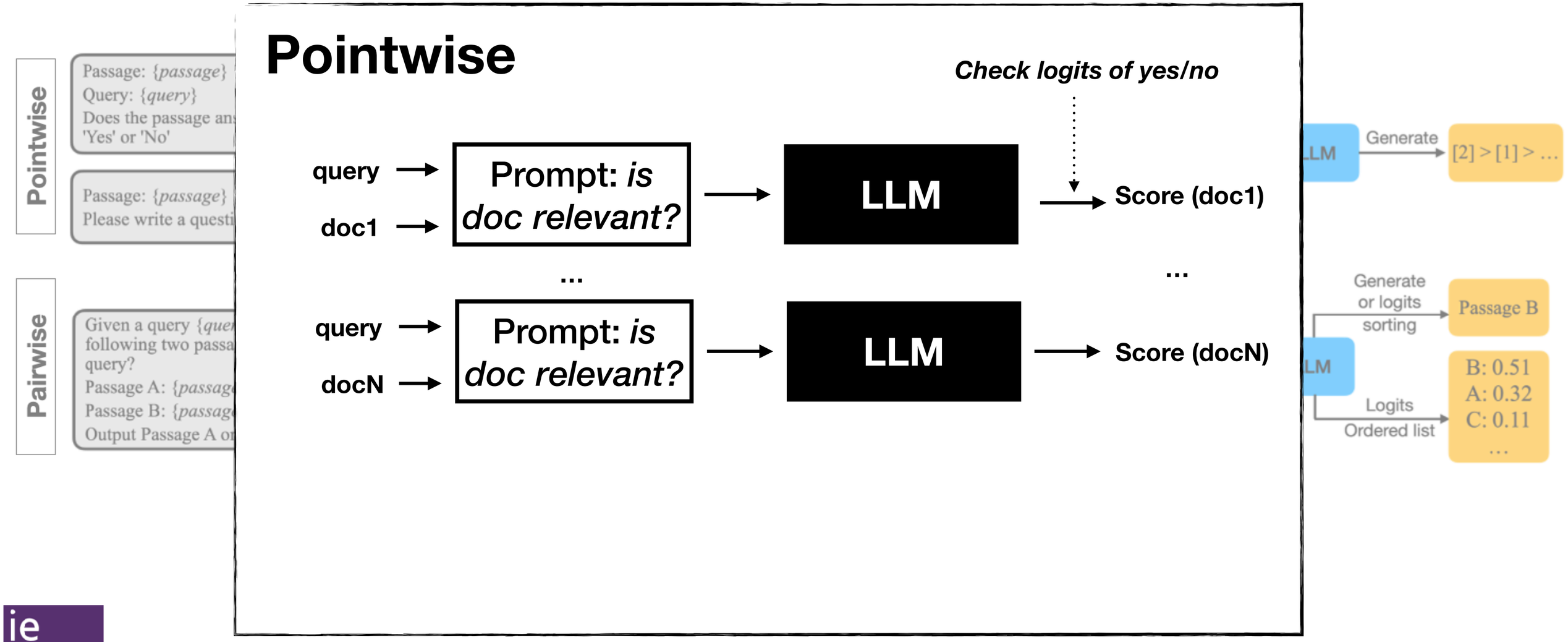
Families LLM-based Rankers



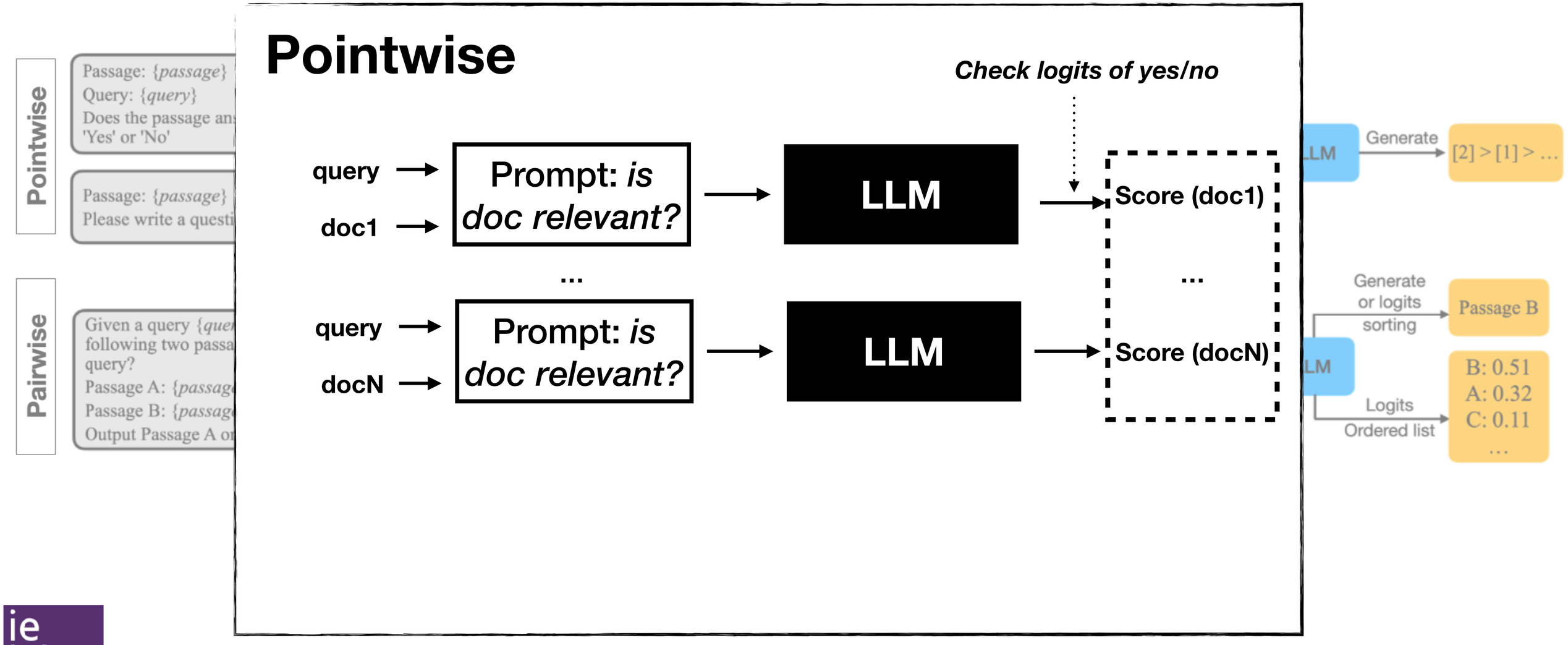
Families LLM-based Rankers



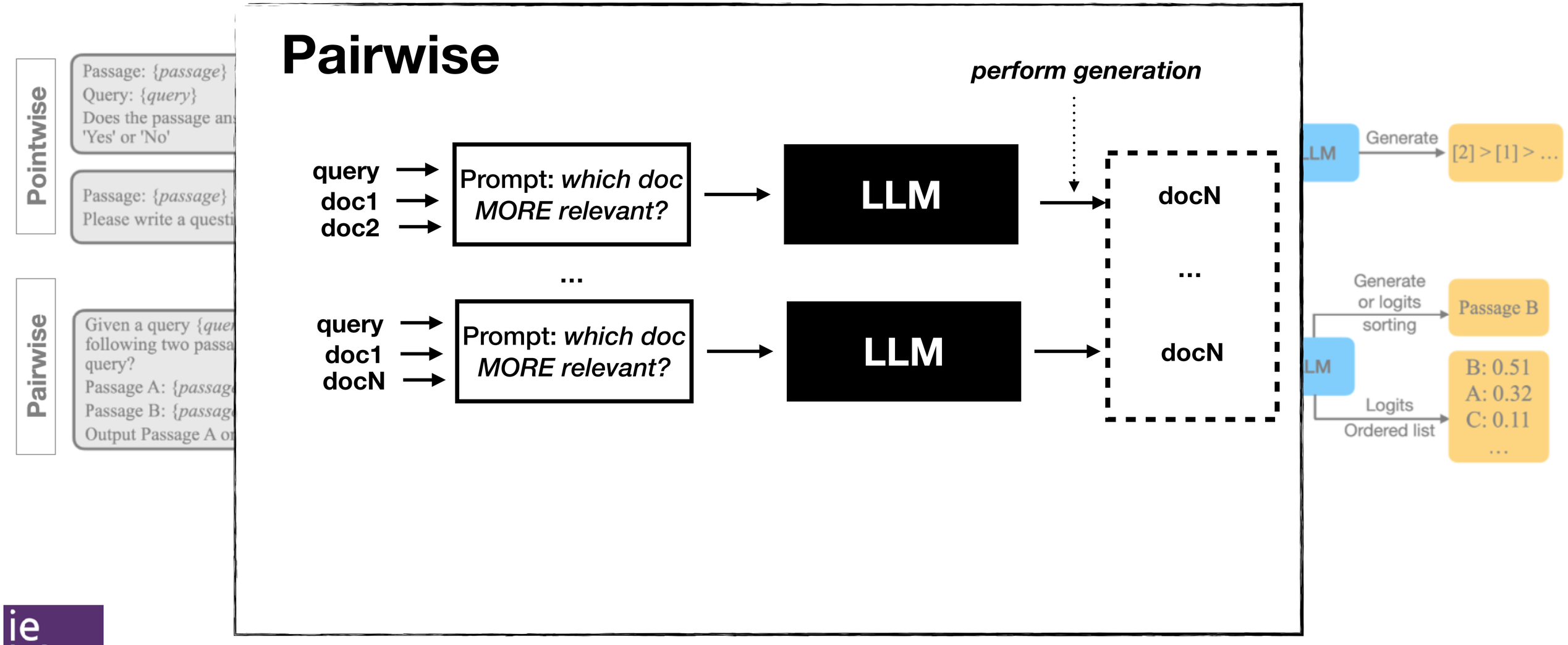
Families LLM-based Rankers



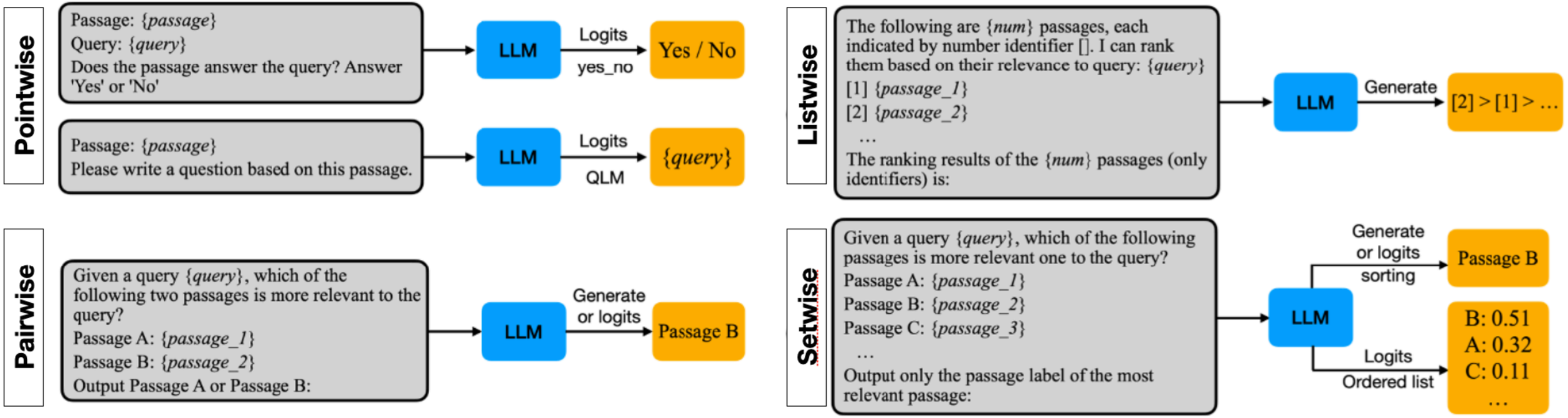
Families LLM-based Rankers



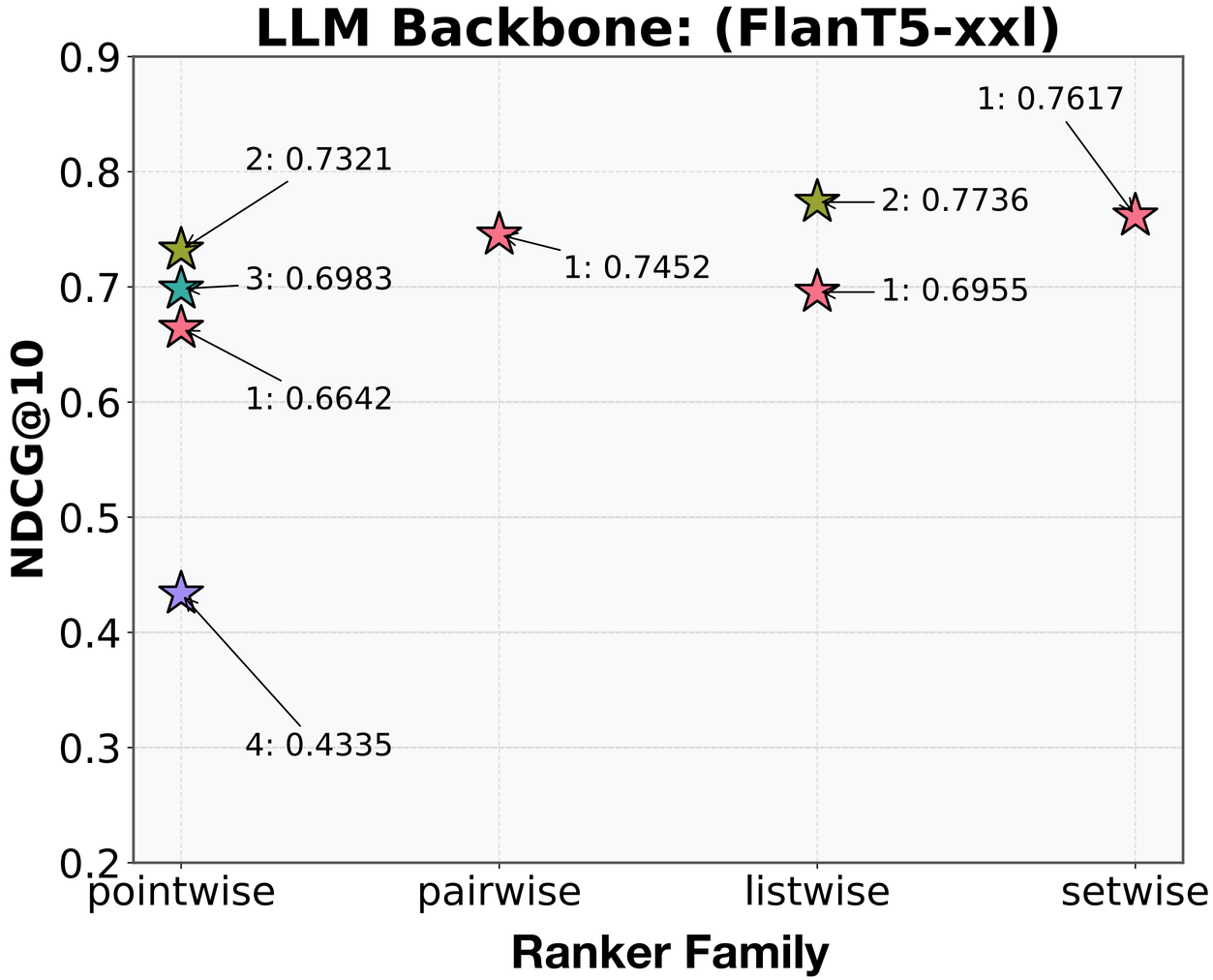
Families LLM-based Rankers



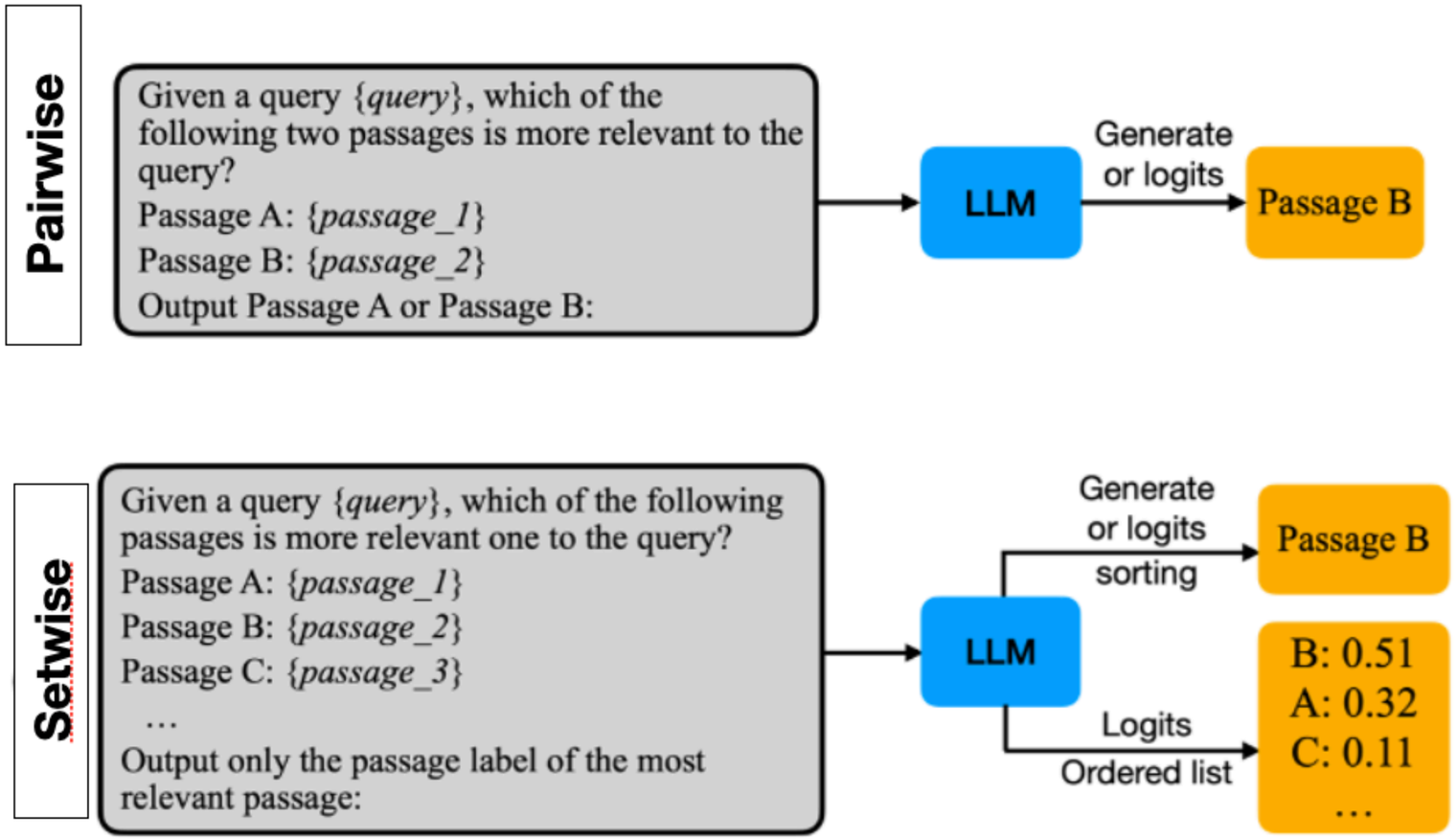
Families LLM-based Rankers



Different ranking mechanisms lead to different effectiveness



Prompts don’t differ just because of ranking mechanism implemented



Prompts don’t differ just because of ranking mechanism implemented

The PRP Prompt

Passage: {text} Query: {query} Does the passage answer the query?

The RankGPT Prompt

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.

ie lab

Prompts don’t differ just because of ranking mechanism implemented

Role Playing

Task Instructions

Evidence Ordering (wrt query) &
Position of Evidence (wrt instructions)

Output Type

Tone Words

The RankGPT Prompt

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.

ie lab

Prompts don’t differ just because of ranking mechanism implemented

What is the effect of differences in wording of these prompt components?

Evidence Ordering (wrt query) &
Position of Evidence (wrt instructions)

Output Type

Tone Words

The RankGPT Prompt

You are RankGPT, an intelligent assistant
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.

ie lab

Prompts don’t differ just because of ranking mechanism implemented

The RankGPT Prompt

You are RankGPT, an intelligent assistant

Tone Words

any word or explain.

What is the effect of differences in wording of these prompt components?

Is effectiveness differences b/w rankers due to:

- RQ1: the actual ranking mechanism, or the choice of words?
- RQ2: LLM characteristics such as backbone and size?

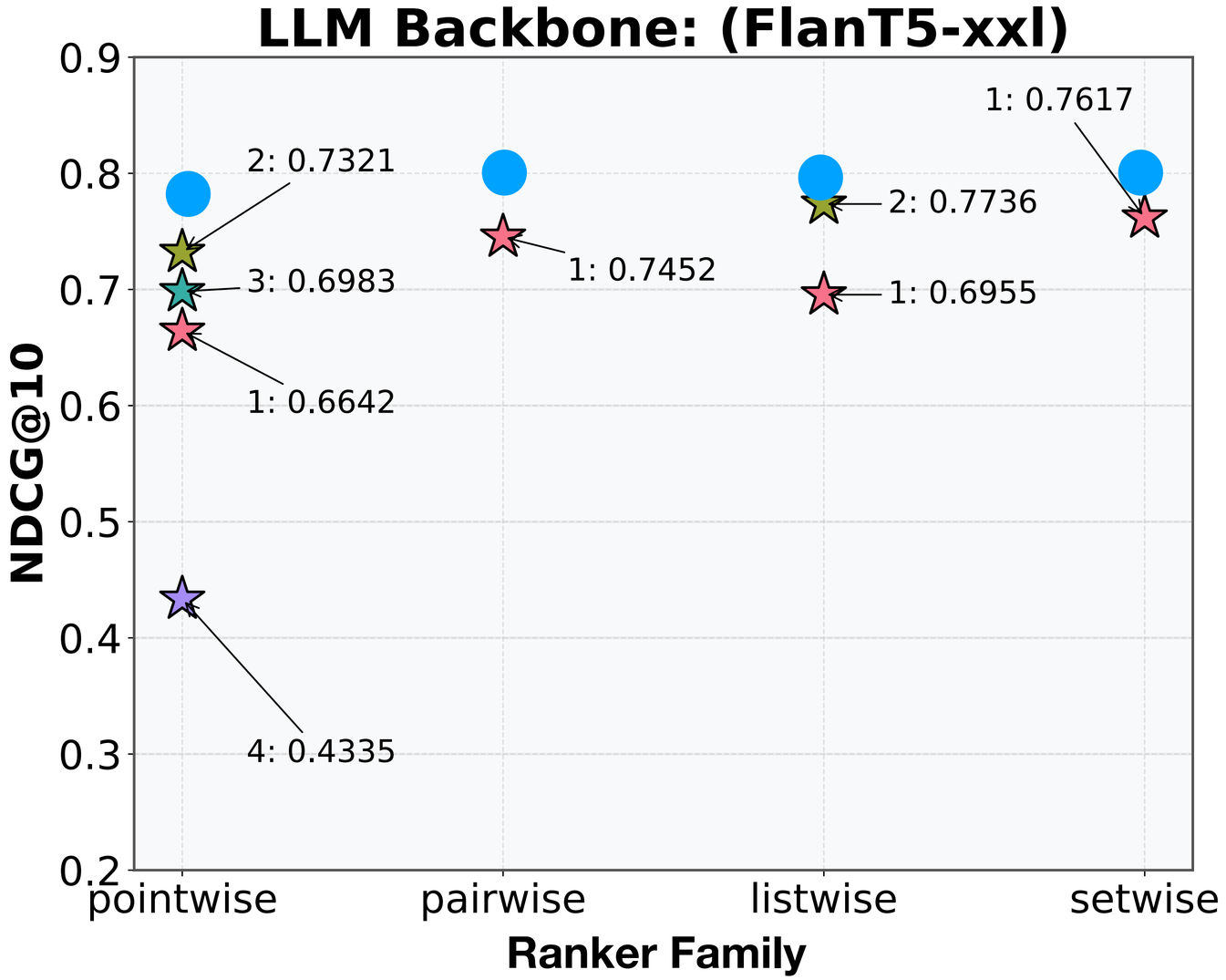
We explore variations in prompt components wordings and ordering

- 1,248 prompt variations
  - e.g. Tone Words: “Please”, “You better get this right or you will be punished”, ….
- 12,400+ GPU-hours, 12,000+ results analysed
- 3 LLM backbone families: FlanT5 (L, XL, XXL), Mistral-7B, Llama3-8B
- Experimented across DL 19, DL 20, COVID (BEIR)

ie lab

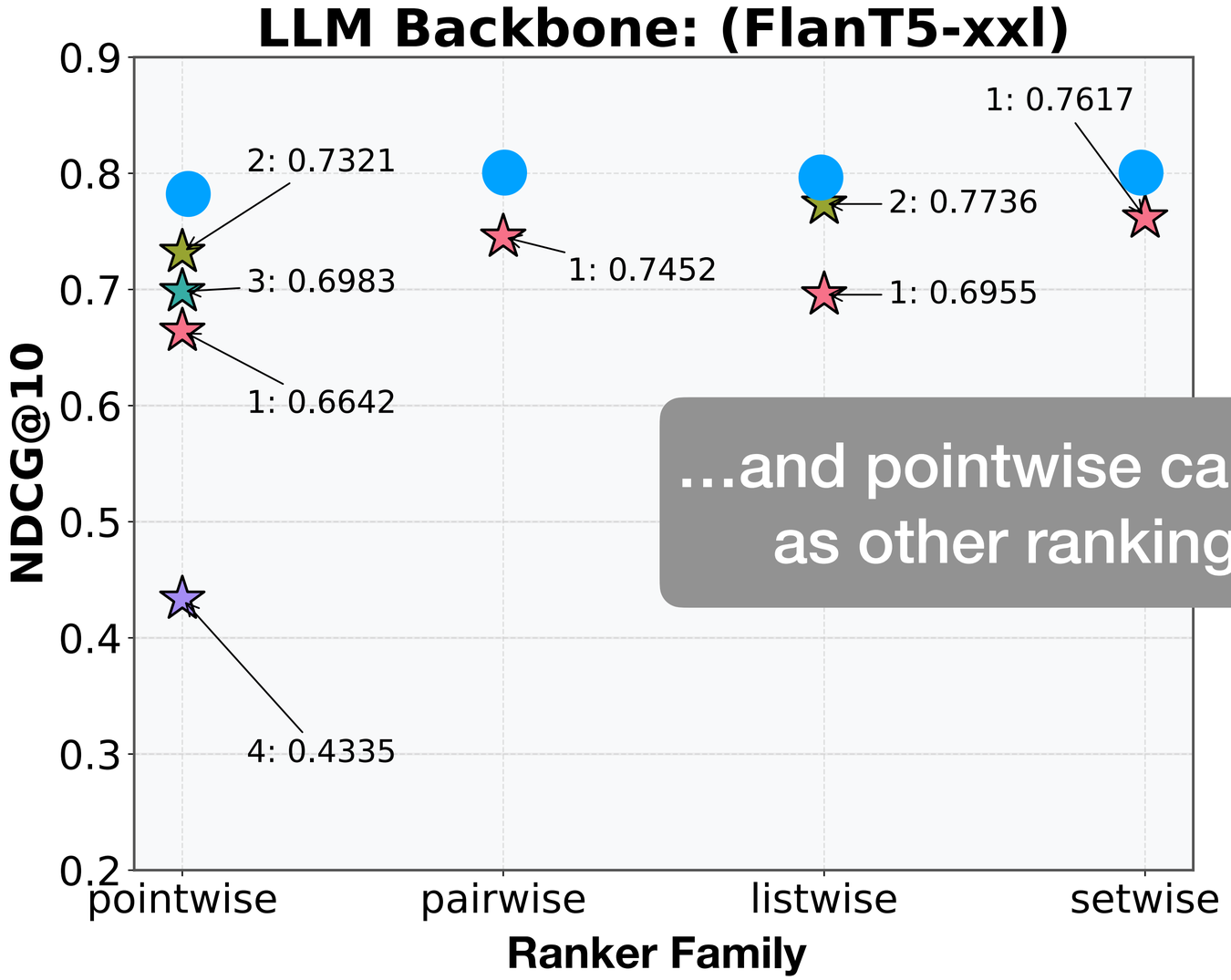
Prompts Better Than Original Ones Do Exist

● = best prompt variation per ranker family



Prompts Better Than Original Ones Do Exist

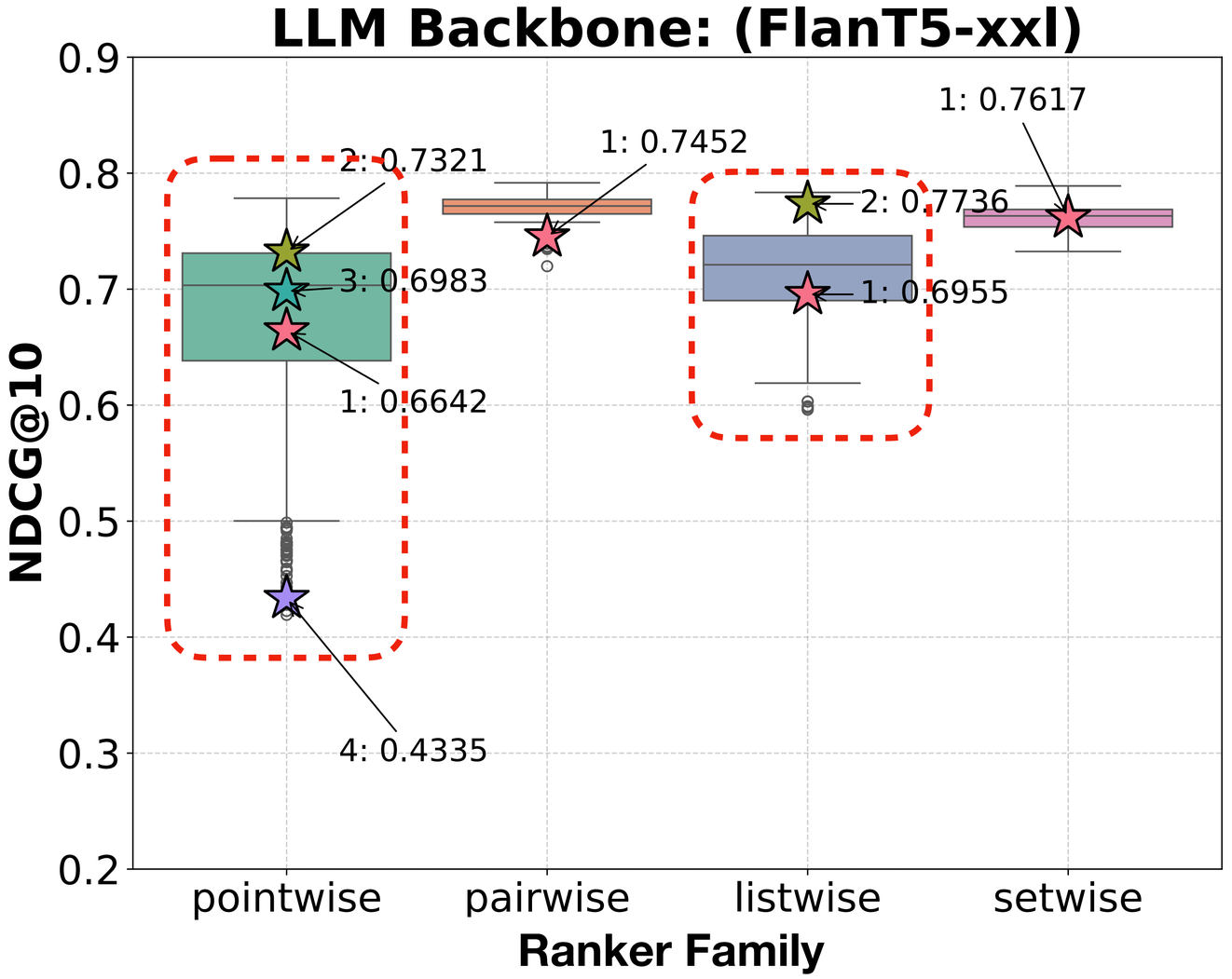
● = best prompt variation per ranker family



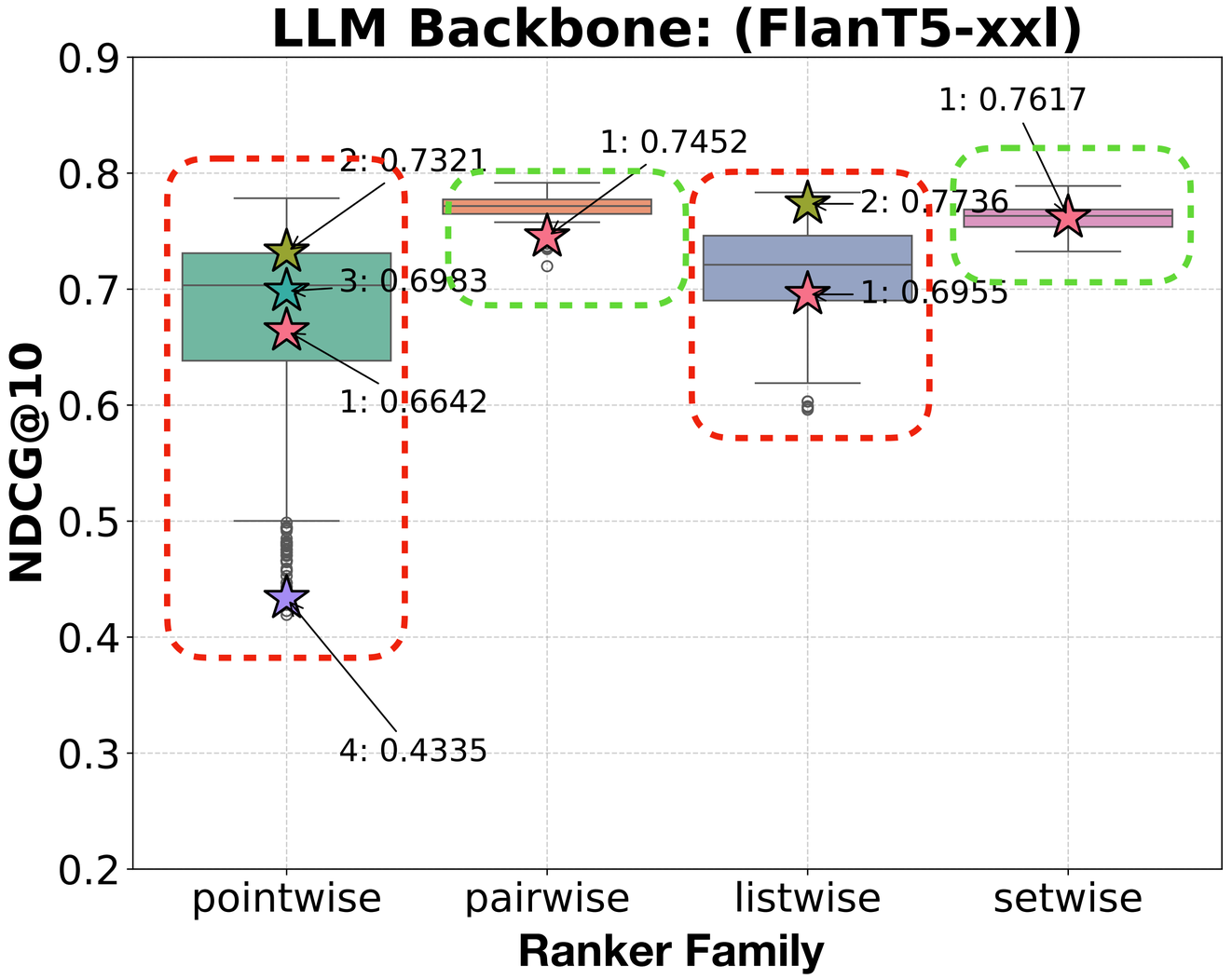
…and pointwise can be as good as other ranking methods

ie lab

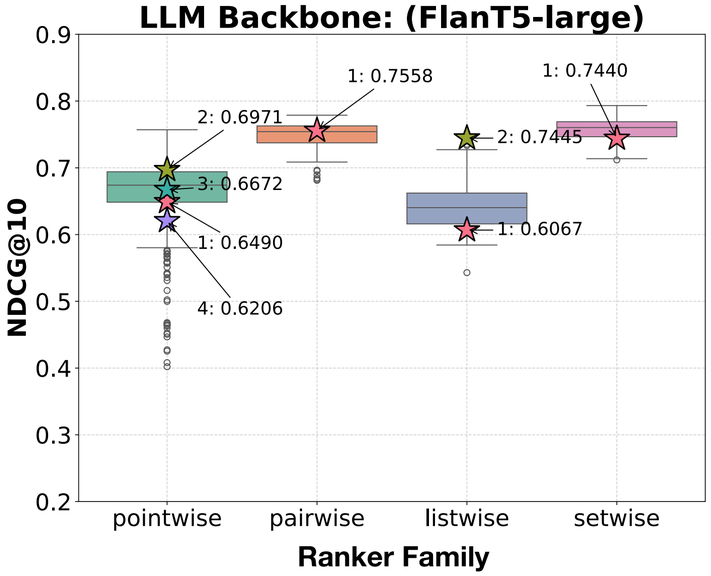
LLM Rankers Can Be (highly) Sensitive to Prompt Variations

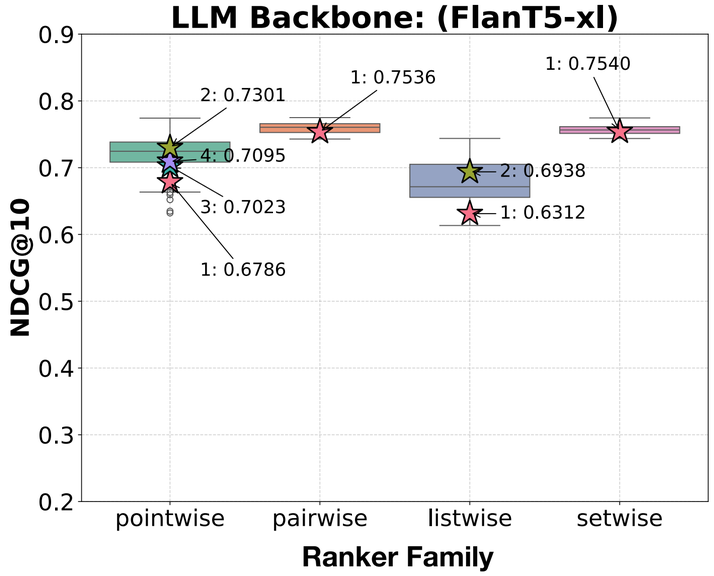


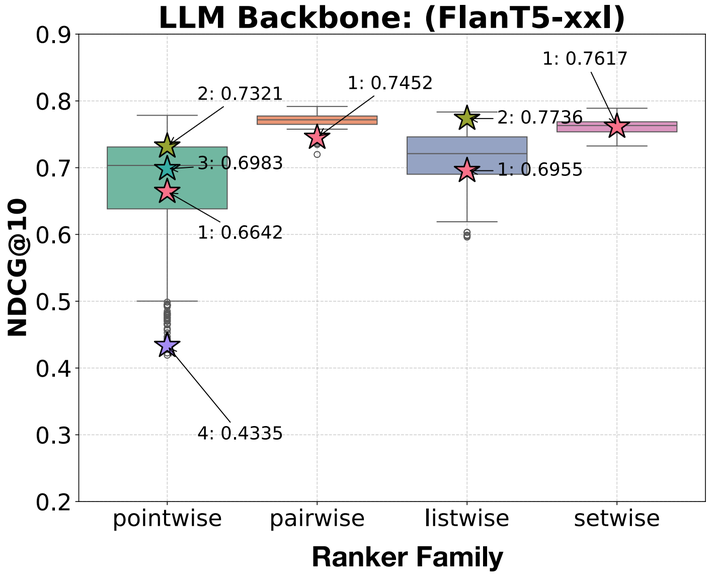
Different LLM Rankers Exhibit Different Variability

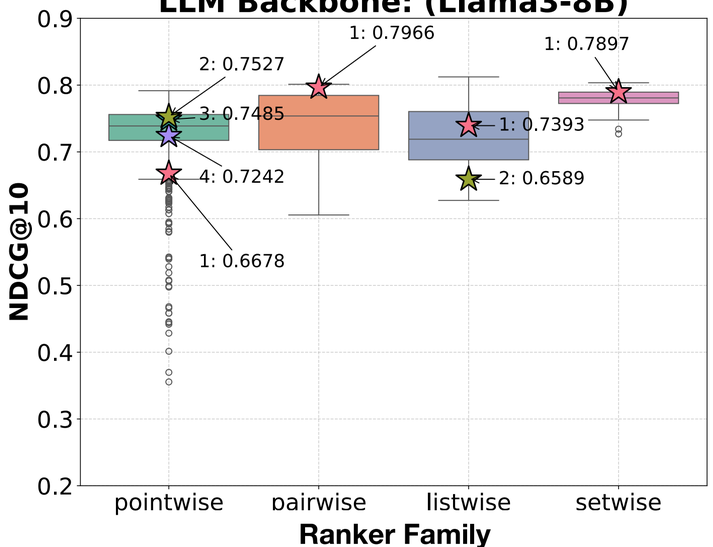


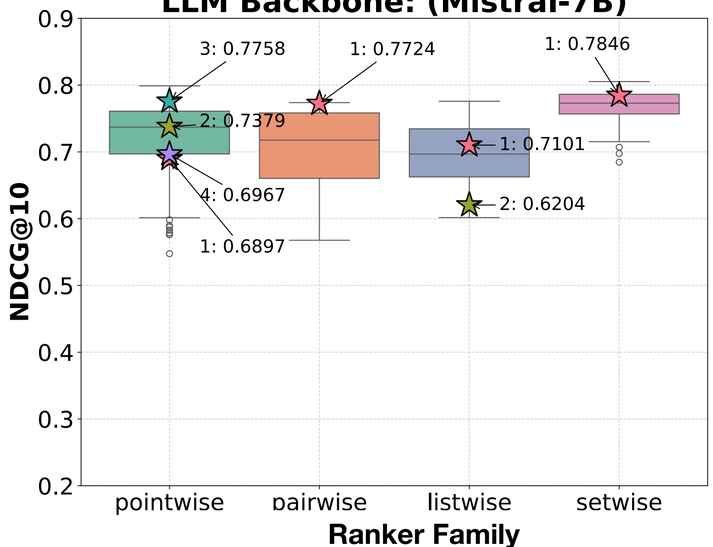
LLM Backbones Influence Effectiveness & Variations Differently Across LLM Rankers











In the paper we also…

- show similar findings across datasets
- analyse role of each prompt component type, and instance within prompt component type

ie lab

Key Takeaways

Prompt components beyond ranking method significantly impact effectiveness

Each ranking method has distinct component preferences

No universal "best prompt" exists: depends on ranking method, dataset, and LLM

Key Takeaways

Prompt components beyond ranking method significantly impact effectiveness

Each ranking method has distinct component preferences

No universal "best prompt" exists: depends on ranking method, dataset, and LLM

Future work: *automatic prompt optimisation & prompt performance prediction*