Weekly Report (260115)

[Recap] Case study (ReasonIR + GPT4 query)

Query: How best to count bees entering and leaving a hive to measure hive activity?

- **Retrieved documents** ← **'bees' match**
  - Monitor the hive environment, …
  - Caring for a colony isn't a set-and-forget task; …
- **Gold document:** Poisson-Distribution ← **theory/principle**

Query: 판 하나가 통째로 섭입된 적이 있는가?

- **Retrieved documents:** ← **섭입의 원리, 이론적 가능성, …**
- **Gold document:** Intermontane Plate ← **entity name**

Query: Does taking a shower have the same effect on muscles as warming up?

- **Retrieve documents** ← **'post workout shower' match**
  - Post-workout showers can be your secret weapon when …
  - A post-workout shower, especially one that alternates between warm and cold water, triggers a process known as vasoconstriction and vasodilation …
- **Gold document:** Vasodilation, also known as vasorelaxation, is the widening of blood vessels. ← **theory/principle**

Summary

- (Recap.) Hypothesis: Abstraction Gap between query & document
  - Validation: prompt LLM to write possible answer documents in multiple level (theory / entity / example)

Method	Bio	Earth	Econ	Psy	Rob	Stack	Sus	Pony	Leet	AoPS	TheoQ	TheoT	Avg
baseline [w/o feedback]	42.7	43.5	29.8	37.2	27	30.4	28.3	27.5	35.9	5.7	41.9	40.8	32.6
baseline [given top5 feedback, rewrite query]	40.7	40.9	**30.5**	37.8	28.0	28.7	28.0	14.6	35.9	6.3	40.8	36.6	30.7
Ours	**46.2**	**44.9**	29.8	**38.7**	**29.8**	**29.2**	**33.6**	**21.4**	**37.3**	**8.5**	**42.9**	**40.7**	**33.6**
Ablation													
Ours w/o prompt 2 (hierarchy)	44.0	42.0	29.4	39.0	28.9	28.8	33.0	19.4	37.3	8.2	42.0	37.8	32.5

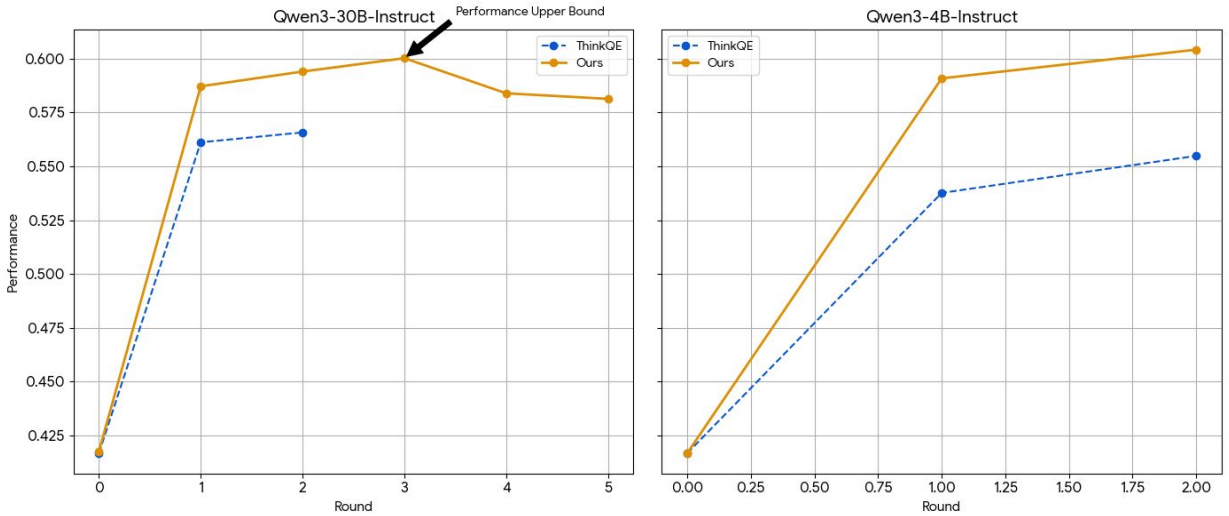
- Generalizable to Open-source Models(Qwen3-4b, 30b)

- Agentic search
  - LLM finds missing abstraction level
  - Some gains
  - limitation: 1) predefined levels 2) affected by initial retrieval performance

- Experiment on Corpus tree
  - LLM-guided hierarchical retrieval (LATTICE)

Hierarchical retrieval: Models

- Models
  - Expanded to Qwen3-{30b, 4b}
  - Static policy (prompt) is used



Agentic search

- Agentic search
  - State: 각 level별 확보된 정보
  - Action: 비어있는 level에 대해 query rewrite (exploration) & 정보량 많은 level에 대해서 query refine (exploitation)
- Prompts

1. Router

```
"Inputs:\n"
"1. **Possible Answers (Previous Query Parts):**\n"
f"- Theory Level: {prev_possible_docs.get('Theory', 'N/A')}\n"
f"- Entity Level: {prev_possible_docs.get('Entity', 'N/A')}\n"
f"- Example Level: {prev_possible_docs.get('Example', 'N/A')}\n\n"
f"- Other Level: {prev_possible_docs.get('Other', 'N/A')}\n\n"

"2. Actual Search Results:\n"
f"{retrieved_docs}\n\n"

"```json\n"
"{\n"
"  \"Reasoning\": \"e.g. Theory 'Poisson' was found in Doc 1. However, Entity 'Submarine' was NOT found; docs discuss 'Bees'.\",\n"
"  \"Actions\": {\n"
"    \"Theory\": \"EXPLOIT\",\n"
"    \"Entity\": \"EXPLORE\",\n"
"    \"Example\": \"EXPLORE\"\n"
"    \"Other\": \"EXPLOIT\"\n"
"  }\n"
"}\n"
"```\n\n"
```

Rewritten query based on predefined levels

Retrieved documents

level마다 어떤 action할지 결정
LLM decide EXPLOIT vs EXPLORE for each level

Agentic search

- Agentic search
  - LLM finds missing abstraction level
- Prompts

  2. Executor

```
if action == 'EXPLOIT':
    instructions_block += (                                          정보량이 많은 level에 대해 refine
        f"### {level} Level: ACTION = EXPLOIT (Refine & Deepen)\n"
        f"- **Context:** The hypothesis answer:\n{prev_content}\n was verified in the retrieved docs.\n"
        "- **Instruction:** Refine this content to be more precise. Use key terms from 'Retrieved Documents'.\n"
        …
    )
elif action == 'EXPLORE':
    instructions_block += (                                          비어있는 level에 대해 rewrite
        f"### {level} Level: ACTION = EXPLORE (Pivot & New Hypothesis)\n"
        f"- **Context:** The previous hypothesis \n{prev_content}\n FAILED (not found or irrelevant).\n"
        "- **Instruction:** Identify the common 'wrong direction' or 'missing gap' in the retrieved docs regarding this level. Treat the previous hypothesis as a **Negative Constraint**.\n"
        …
    )
elif action == 'PRUNE':
    instructions_block += f"### {level} Level: ACTION = PRUNE (Remove this level)\n\n"
```

Agentic search

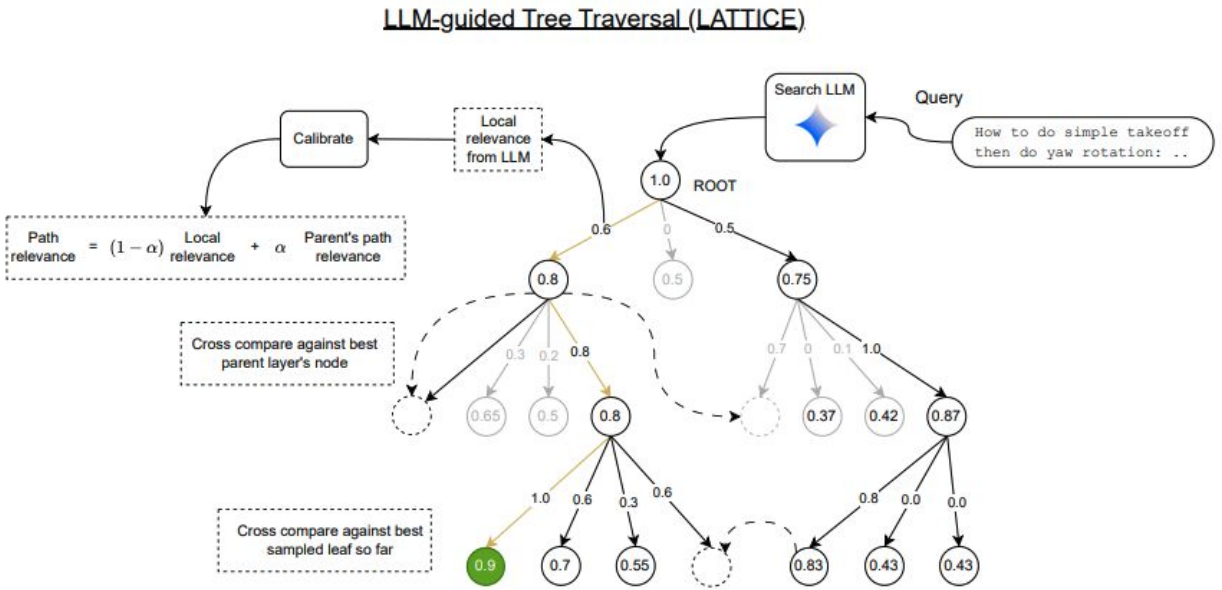
- Results
  - nDCG@10 reported, iteration = 2

	biology	psychology
thinkqe	55.5	**44.3**
static policy (ours)	57.8	43.7
agentic (ours)	**59.2**	41.9

- Limitation
  - (i) The effectiveness of fixed abstraction level(theory/entity/example) differs for domains/queries
    - → If the corresponding anchor document does not exist, the retrieval fails
  - (i) When query rewriting, we use “top-k corpus feedback” to check whether such documents exist
    - → rewriter stuck in locality
  - → Should scan entire corpus to use global information, then leverage the signal to build abstraction levels

Leveraging corpus abstraction tree

- Experiment on [LLM-guided hierarchical retrieval] (LATTICE)
  - Corpus is converted to abstraction tree
- leaf node: 문서들
- branch: 문서 cluster 의 요약
- Traversal: LLM calculates local relevance (=listwise reranking) from root node & beam search
- We view branch nodes = abstractions

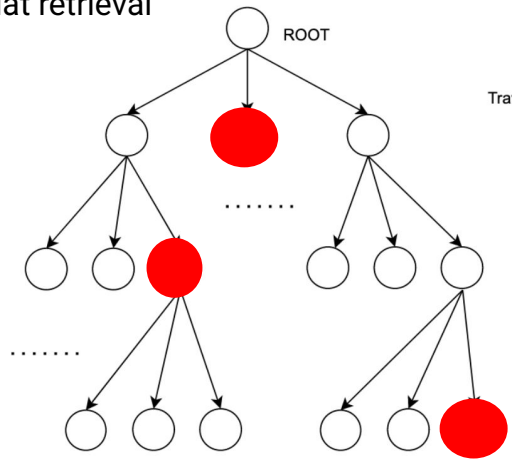


Leveraging corpus abstraction tree

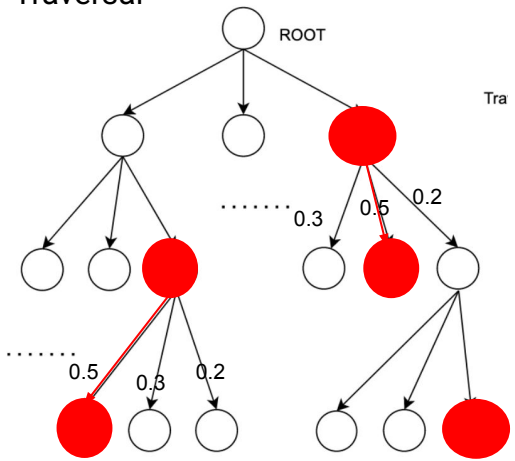
- We view branch nodes = abstractions
- Method (Changes from LATTICE)
  - Flat retrieval → Lattice traversal
  - retrieval pool: flattened tree
  - query: original query or rewritten query
  - document score : rank fusion of [retrieval rank + LATTICE rank]

Rewritten query →

Flat retrieval



Traversal

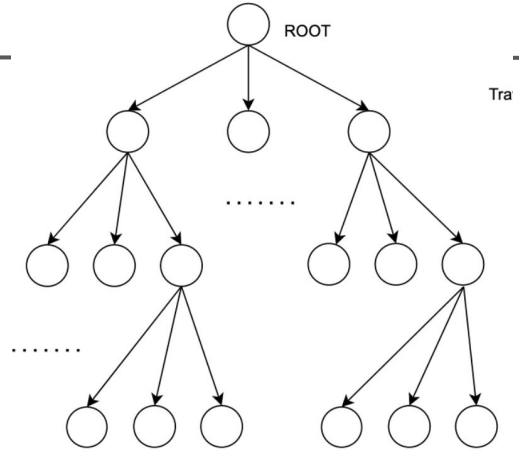


Leveraging corpus abstraction tree

- Results
  - max ndcg across the iteration is reported
  - Qwen-3-4B

	biology	psychology
LATTICE	46.36 (iter=20)	36.86 (iter=20)
flat → lattice	55.28 (iter=3)	39.86 (iter=3)
thinkqe + flat → lattice	59.54 (iter=3)	46.40 (iter=3)
our QR + flat → lattice	**60.67** (iter=3)	**48.36** (iter=3)

Leveraging corpus abstraction tree



- Analysis: Flat retrieval mitigates locality issue
  - biology subset
  - measure whether the gold leaf is under the path

LATTICE

Depth	Hits
1	0.8738
2	0.7767
3	0.699
4	0.6796
5	0.6552

vs flat retrieval: 0.9223

Appendix

Summary

- (Recap.) Hypothesis: Abstraction Gap between query & document
  - Validation: prompt LLM to write possible answer documents in multiple level (theory / entity / example)
  - Generalizable to Open-source Models(Qwen3-4b, 30b)

- Agentic search
  - LLM finds missing abstraction level
  - Some gains
  - limitation: predefined levels are not effective for other subcategories

Idea: leverage corpus to search possible levels

- Experiment on LLM-guided hierarchical retrieval (LATTICE)

TODOs

- Interaction between corpus side abstraction - query side abstraction
  - 현재 구현: top 5문서 → fixed (theory/entity/example) level query generation for once, use same query to traverse the tree
- LATTICE의 hierarchy == our abstraction level? → case study
- Iteration = 3 이후로 성능 하락 문제

Hierarchical retrieval: Terms

Ideally, you would gather preliminary statistics to design the experiment. If you have a high variability, and your observation window is short, then small effects will be swamped by the variability. However, if what you are looking for has a big effect in the number of bees, then counting for a shorter time is going to be fine.

A rule of thumb would be to try recording 5 to 10 segments of 1 minute analysis and see what the variability is. If the standard deviation is small compared to the effects you are seeing, then 1 minute is fine.

If you want a more theoretical justification, Poisson distribution could be used as a first order approximation of the distribution. I don't know much about statistics specific for this kind of insect behaviors though.

- Level (Theory/entity/example)
  - Level: “abstraction” **hops** to reach document’s answer type
  - Mismatch between query’s intent type vs document’s answer type

Query: How best to count bees entering and leaving a hive to measure hive activity?

- **Retrieved documents** ← **‘bees’ match**
  - Monitor the hive environment, ...
  - Caring for a colony isn't a set-and-forget task; ...
- **Gold document:** Poisson-Distribution ← **theory/principle**

Level 0	Entity - 여기 level이 맞는지 모르겠음
Level 1	Lexical matching (Original query)
Level 2	Theory/Principle
Level 3	Application/Example - 얘는 뭔지 모르겠네

Hierarchical retrieval

TODO

- RQ 1: 언제 hop 뛸지? [THIS WEEK]
  - Following R2U, similarity between [generated subquery, retrieved document] as query expansion policy
    - low similarity → explore (think of alternatives)
    - high similarity → exploit (refine with terms from given docs)
- RQ2: 어떤 level이 가능할지?
  - Can we predefine possible levels with the corpus?

taxonomy가 필요하다 했나 taxonomy 라는 용어가 필요하다 했나..?
grounding: ground된 entity 끼리의 관계 = theory?

HOP과는 얘기가 다름 (여기는 너무 explore된 분야)

TODOs

- Interaction between corpus side abstraction - query side abstraction
  - 현재 구현: top 5문서 → fixed (theory/entity/example) level query generation for once, use same query to traverse the tree

Engineering

- Iteration = 3 이후로 성능 하락 문제