RAG 성능 고도화 제조, 금융, Lifelog

Domain별 High-Performance RAG 노하우, 대방출

Who is Speaker?

Choi Jaehun

AI Research Engineer

Braincrew Inc. RAG 팀 Leader

Multi-Domain의 RAG Project 수행

- Ship-Building
- Injection-Molding
- Construction
- Financial
- Lifelog

Linked-In

Github

CONTENTS

PART 01

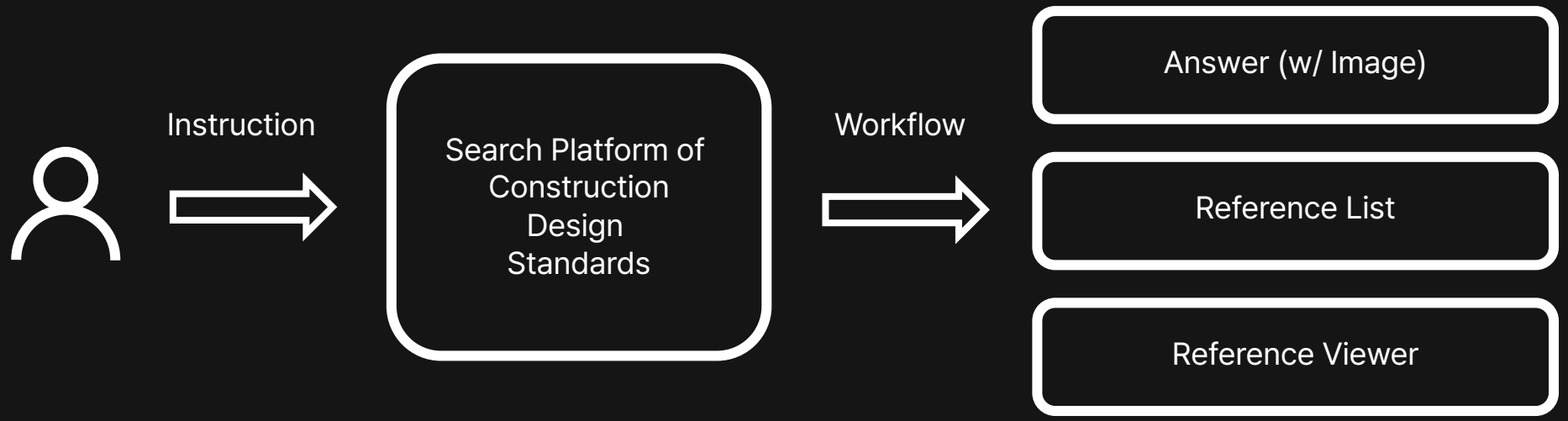
Manufacturing Domain

High-Performance RAG Recipe in Manufacturing Domain

01

Our Challenges

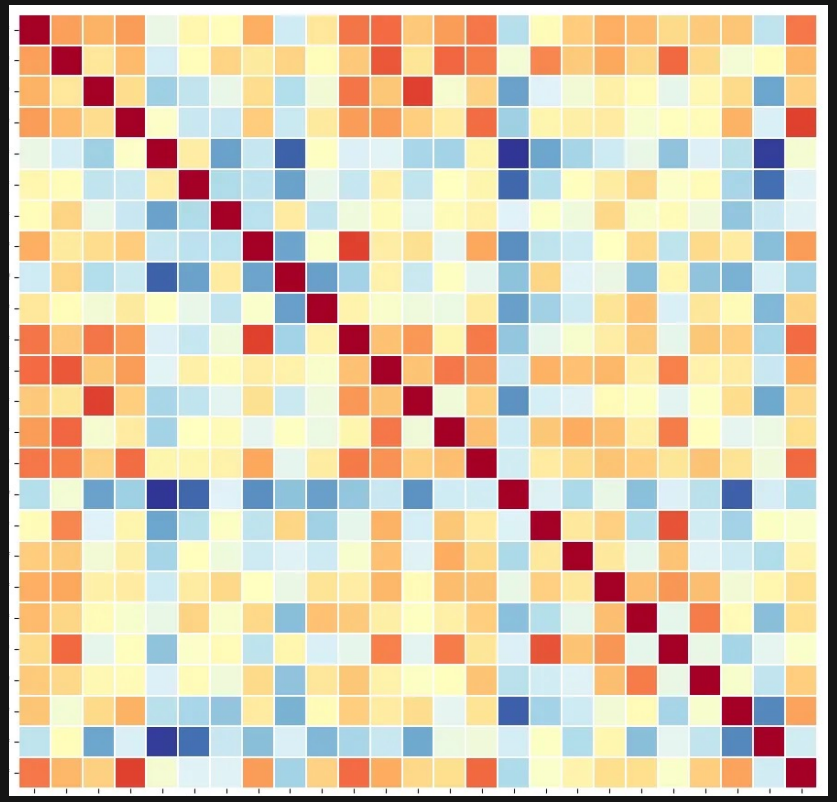
** Integration Search platform for construction design standards*



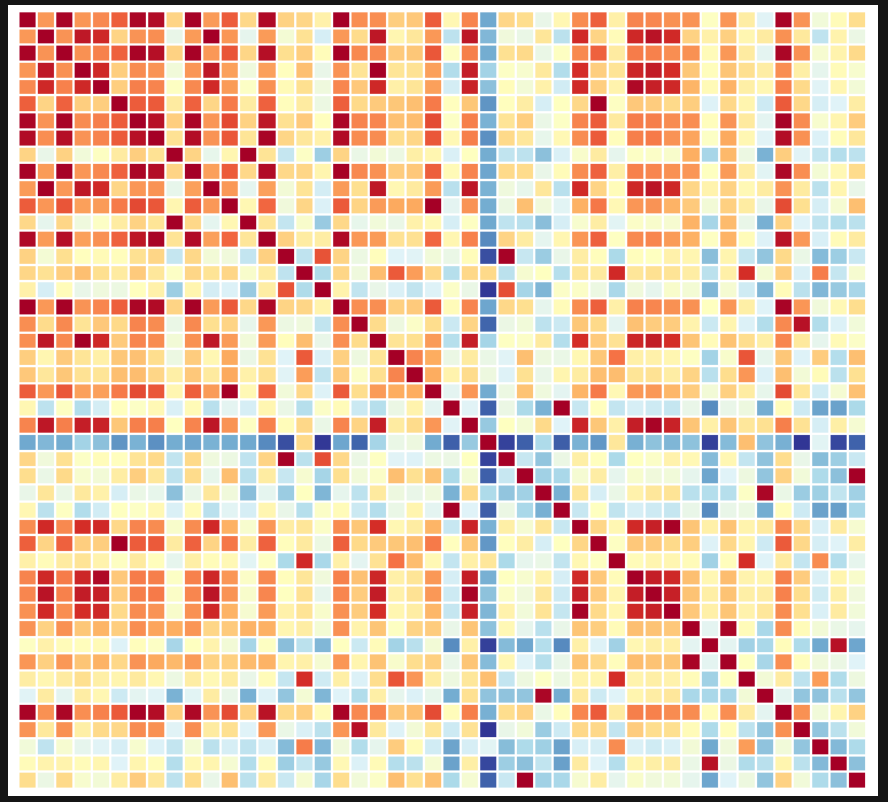
Latency 10s, Recall 85% in each sub-domain

Insights

** Every sub-domain documents is relevancy?*



[Figure 01] Document Relevance 84% about sub-domain 1

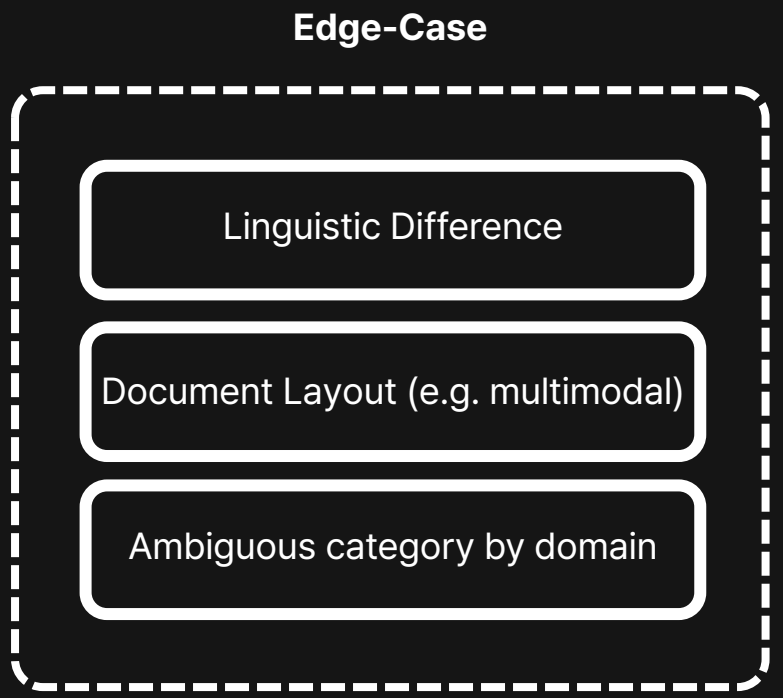


[Figure 02] Document Relevance 96% about sub-domain 2

EDA First, and analyze the document through domain sight

Insights

** Every sub-domain documents is relevancy?*



- Document Layout (for Retriever)
- Domain Specific Representation (for Retriever & Generation)
- Linguistic Difference (for Retriever, Embedding)
- Document Contents Characteristic (for Retriever)
- Prompt Engineering (for Generation)

Insights

** Metadata structure in Multi-Modal object*

```
{
    ...
    'uuid' : str,
    'file' : str,
    'image_path' : str,
    'image_title' : Optional[str],
    'image_context' : Optional[str],
    'image_type' : Optional[str],
    'image_keywords' : List[str],
    'potential_questions' : List[str]
    ...
}
```

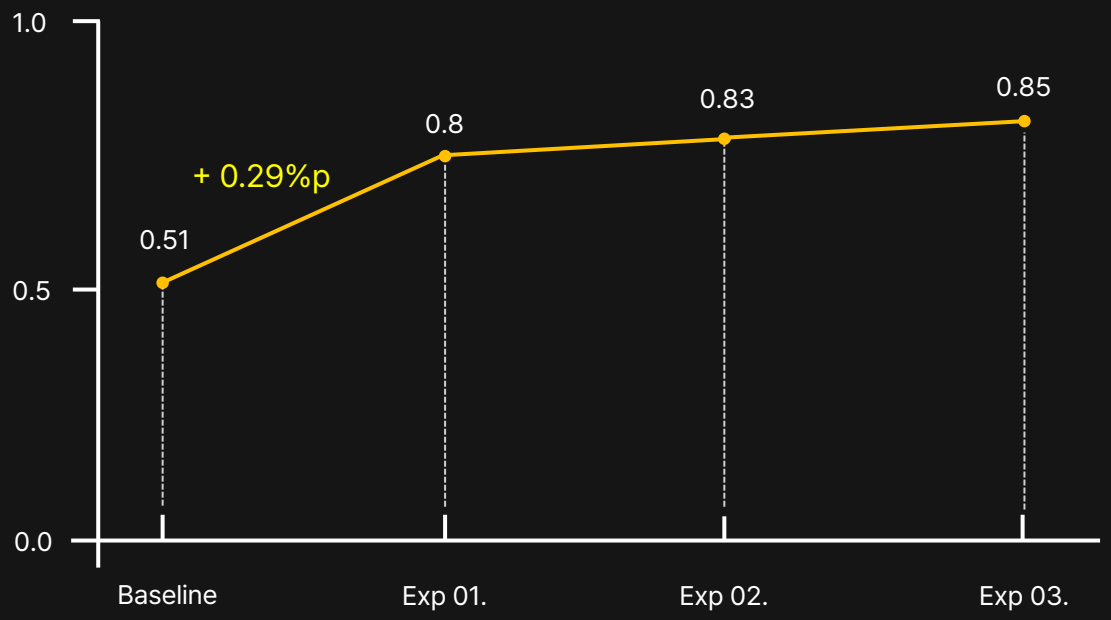
[Image Metadata Schema]

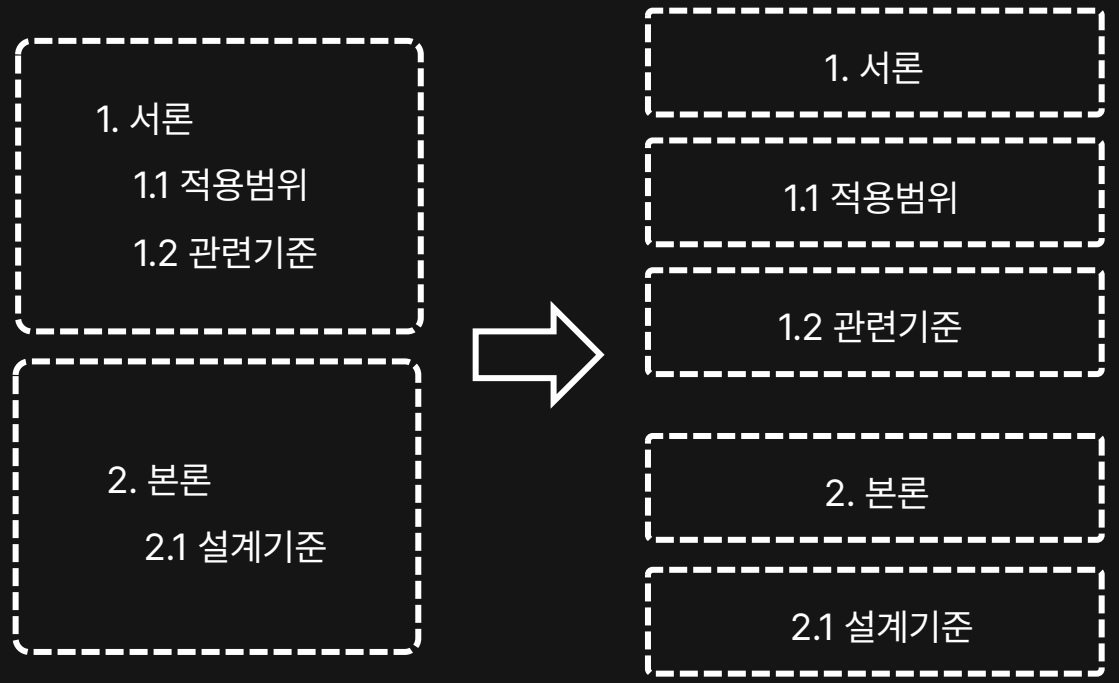
```
{
    ...
    'uuid' : "ID-1",
    'file' : "Page_1_Index_1.png",
    'image_path' : "images/test/Page_1_Index_1.png",
    'image_title' : "OO계획서 도표",
    'image_context' : "OO계획서 개정 이력 표",
    'image_type' : "표/도면",
    'image_keywords' : ["OO", "□ □"],
    'potential_questions' : ["OO 계획서의 ~~~"]
    ...
}
```

[Example of Image Metadata Schema]

Insights

** Improving chunking strategy & chunk structure*

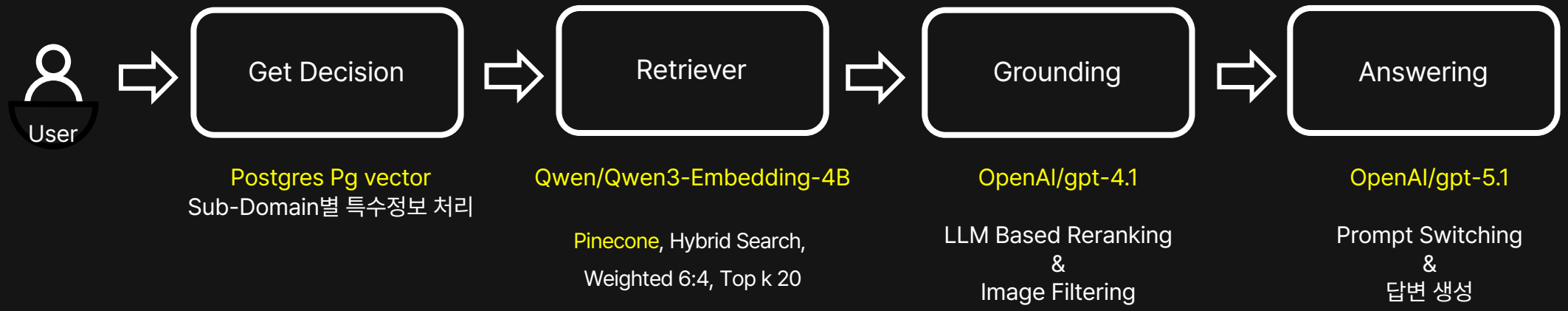




Structure the chunks by independent regularizations.

Workflow

** Final workflow for requirements*



Cases of Failure

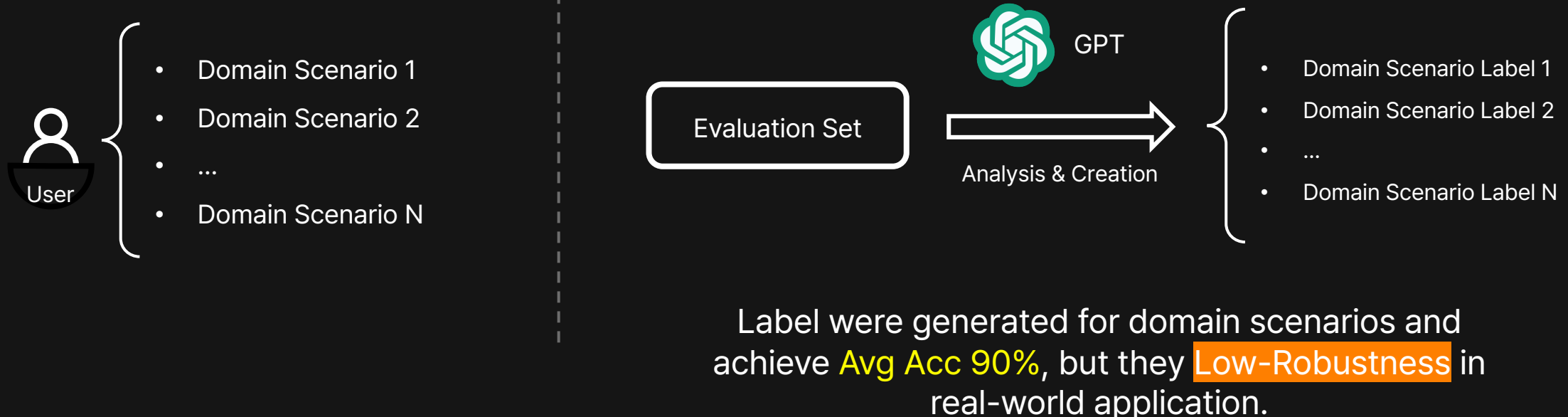
** Necessity of the "Get Vocab" Module*

	Use to Get Vocab	Not use to Get Vocab
Latency	4.7s	0s
Recall	0.818	0.819

We discovered that the module's can have meaningless effects

Cases of Failure

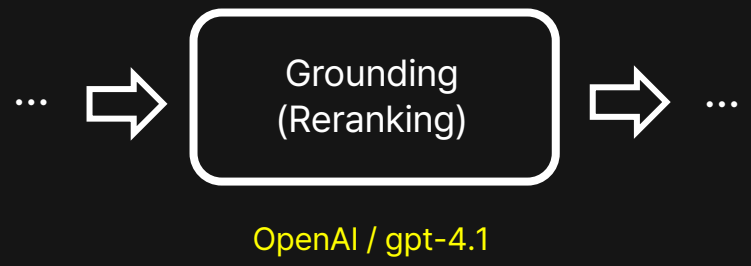
** Classification of Domain Scenario*

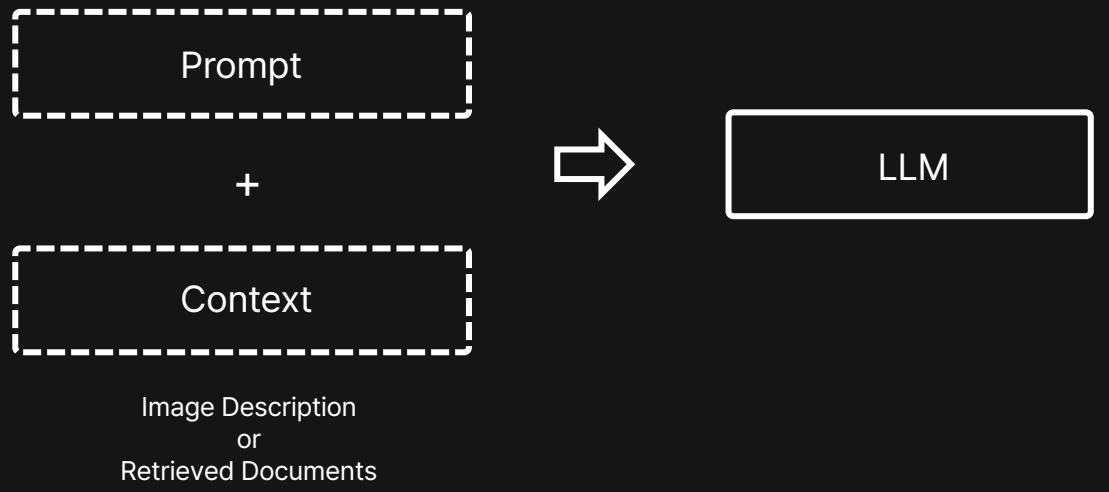


Label were generated for domain scenarios and achieve Avg Acc 90%, but they Low-Robustness in real-world application.

Cases of Failure

** Is effective method about LLM-based reranking?*





LLM based Reranking is still alive?

Ref)
* How Good are LLM-based Rerankers? An Empirical Analysis of State-of-the-Art Reranking Models [2025. EMNLP]
* Efficiency-Effectiveness Reranking FLOPs for LLM-based Rerankers [2025, EMNLP-Industry]

PART 02

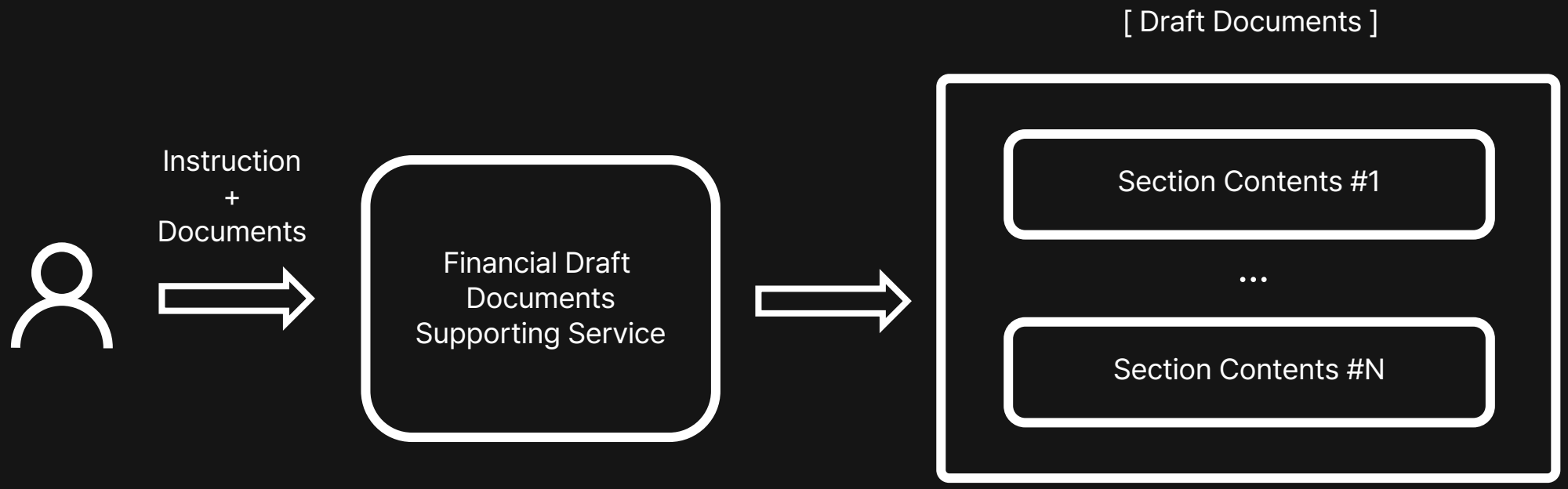
Financial Domain

High-Performance RAG Recipe in Financial Domain

02

Our Challenges

** Approval application drafting supporting service*



Qualitative Evaluation 80% in this project

Insights

** Analysis the Target Document Structure and Set-up Strategy*

Target Document

Section 01
Section 02
Section 03
Section 04
Section 05
Section 06
Section 07
Section 08
Section 09

Section 01 : Type A, Type B

Section 02 : Type A

Section 03 : Type B

Section 04 : Type A, Type C

Section 05 : Type A, Type B, Type C

Section 06 : Type B

Section 07 : Type A

Section 08 : Type A, Type B

Section 09 : Type B, Type C

Each section refer to different sources.
Identified through expert interview and domain analysis.

Insights

** Which tabular format LLM understands very well?*

Format	Accuracy	95% Confidence Interval	Tokens
Markdown-KV	60.7%	57.6% – 63.7%	52,104
XML	56.0%	52.9% – 59.0%	76,114
INI	55.7%	52.6% – 58.8%	48,100
YAML	54.7%	51.6% – 57.8%	55,395
HTML	53.6%	50.5% – 56.7%	75,204
JSON	52.3%	49.2% – 55.4%	66,396
Markdown-Table	51.9%	48.8% – 55.0%	25,140
Natural-Language	49.6%	46.5% – 52.7%	43,411
JSONL	45.0%	41.9% – 48.1%	54,407
CSV	44.3%	41.2% – 47.4%	19,524
Pipe-Delimited	41.1%	38.1% – 44.2%	43,098

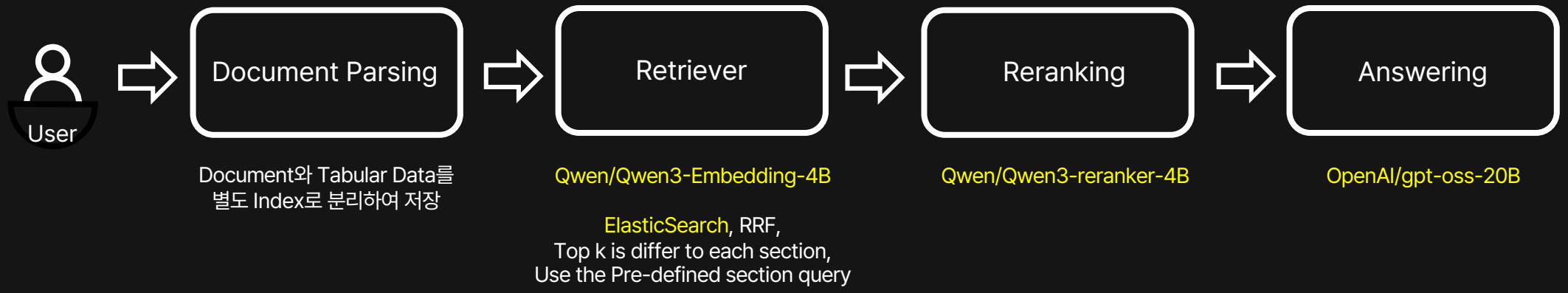
```
# Employee Database

## Record 1

```
id: 1
name: Charlie A0
age: 56
city: New York
department: Operations
salary: 67896
years_experience: 7
project_count: 1
```
```

[Figure 03] Which table format do llms understands best?

Figure 03 출처 : https://www.improvingagents.com/blog/best-input-data-format-for-llms?cmid=34b34591-5868-43e7-8016-bbb6fbea4bf1

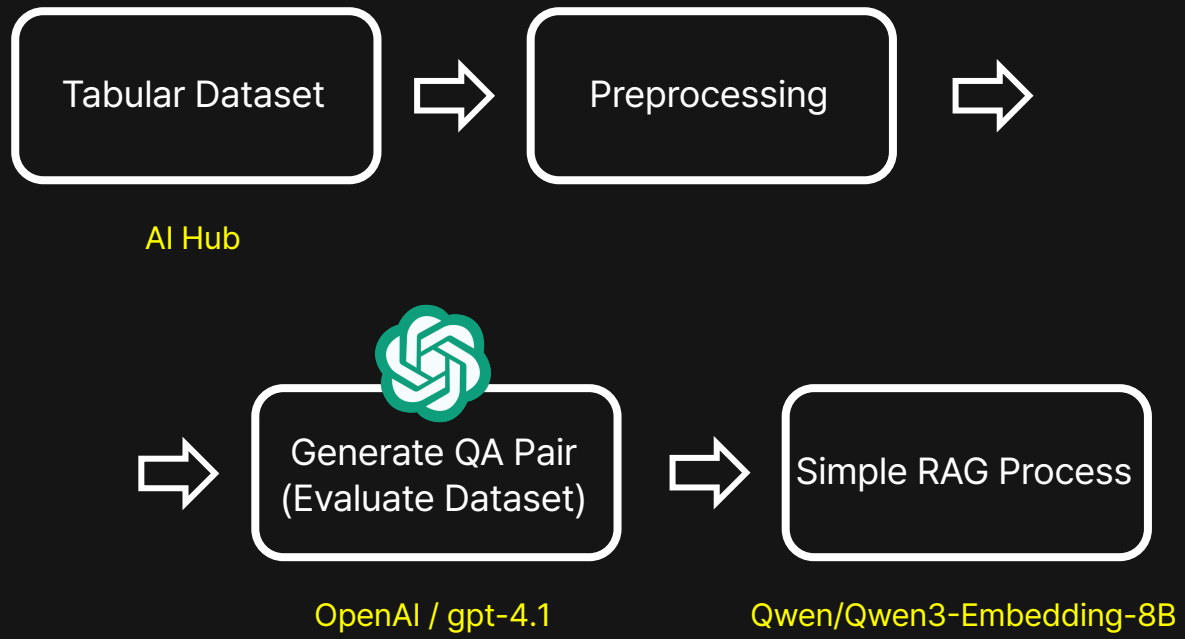
Workflow

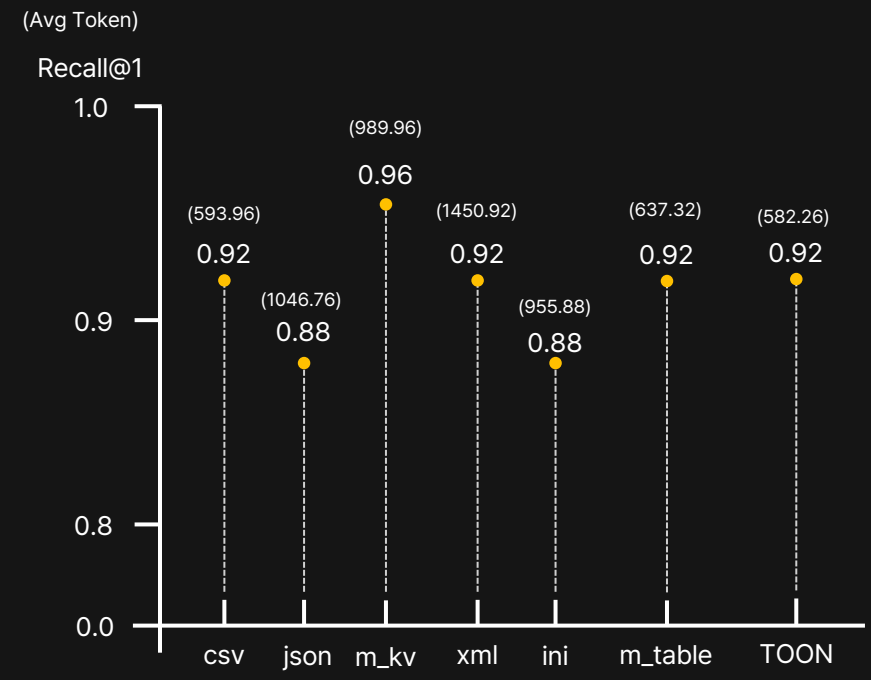
** Final workflow for requirements*



Cases of Failure

** Which tabular format is best understands in Retriever*





* Best Performance Tabular Format : Markdown KV

* Best Efficiency Tabular Format : TOON

Ref)

* https://www.ncloud-forums.com/topic/594/ [N Cloud]

PART 03

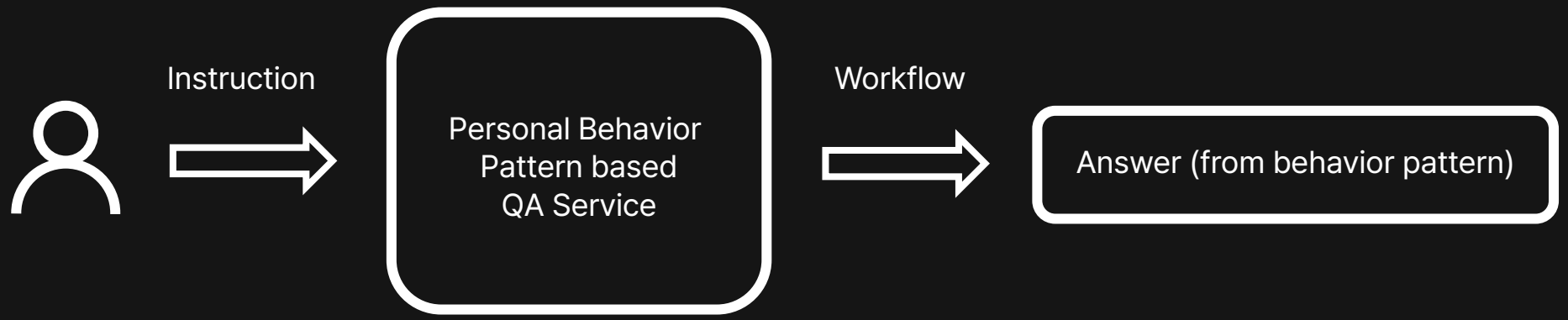
Lifelog Domain

High-Performance RAG Recipe in Lifelog Domain

03

Our Challenges

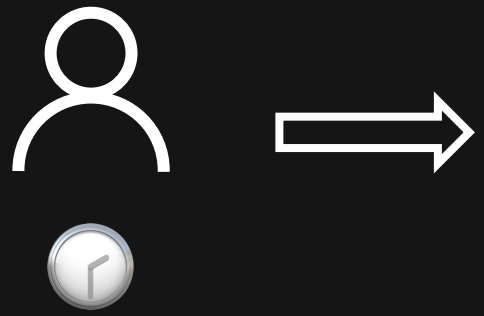
** Personal Behavior-Pattern-based QA Service*



Latency 4s, Recall 90%, No Cost in this project

Our Challenges

** What is Behavior-Pattern Data?*



Daily Information : YYYY / MM / DD

Time Duration : 06:00 ~ 06:20
Location : 침실
Acting : 기상 후 침대자리를 정리하고 거실로 이동하였다.

...

Variable captured about personal acting data at that time

Insights

** Refine the User Scenario*

General Query : queries that can be answered based on a single condition.
Cases where direct access to information is possible using an 1-dimensional condition for an simple target.

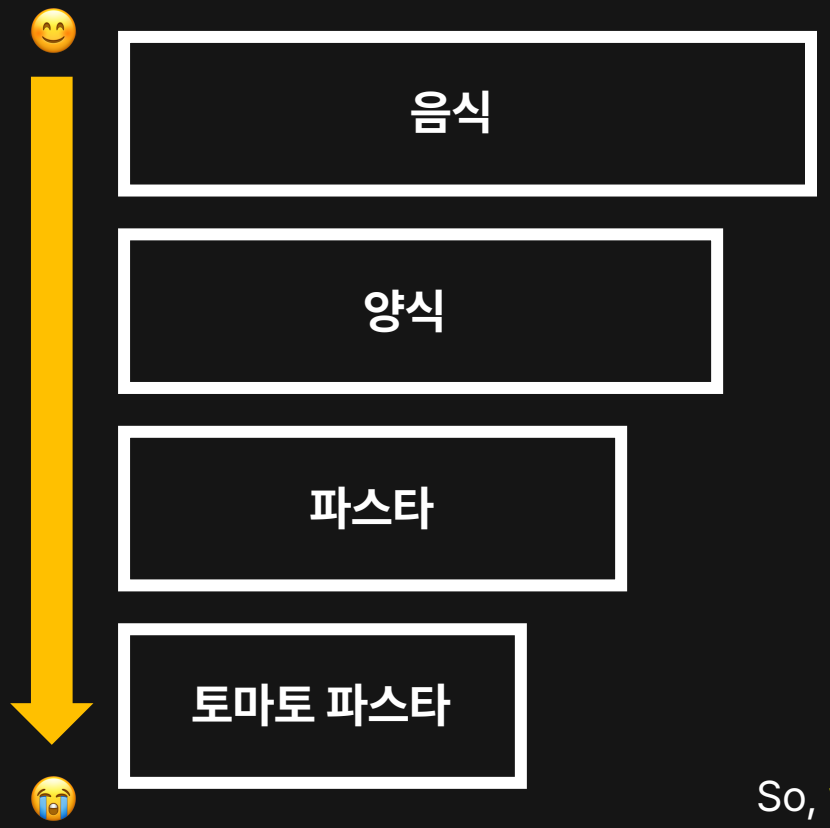
E.g. 8월 15일에 먹은 맥주는 무엇인가요?

Statistic Query : queries that can be answered only by considering multiple conditions.
Cases where access to information is possible when multi-dimensional condition for an target are applied and used appropriately.

E.g. 양식은 몇 번 먹었어?

Insights

** Data Regularization : Hierarchical category*



양식은 몇 번 먹었어?

- 파스타
- 바질 파스타
- 로제 파스타
- 토마토 파스타
- 로제 토마토 파스타
- 스테이크
- 치킨 스테이크
- 감바스

In Retriever side, End-User never use unified representation.
So, we must transform the user query or reflect linguistic diversity into the metadata.

(❄️)　　(🔥)

Insights

** Data Regularization : Reformatting Raw Data*

One Day (24h)

Place Info

Group Info

Transfer Info

Object Info

Act Info

...

One Day (24h)

Act Scope

...

Act Scope

Act Scope

Time Info

Place Info

Group Info

Transfer Info

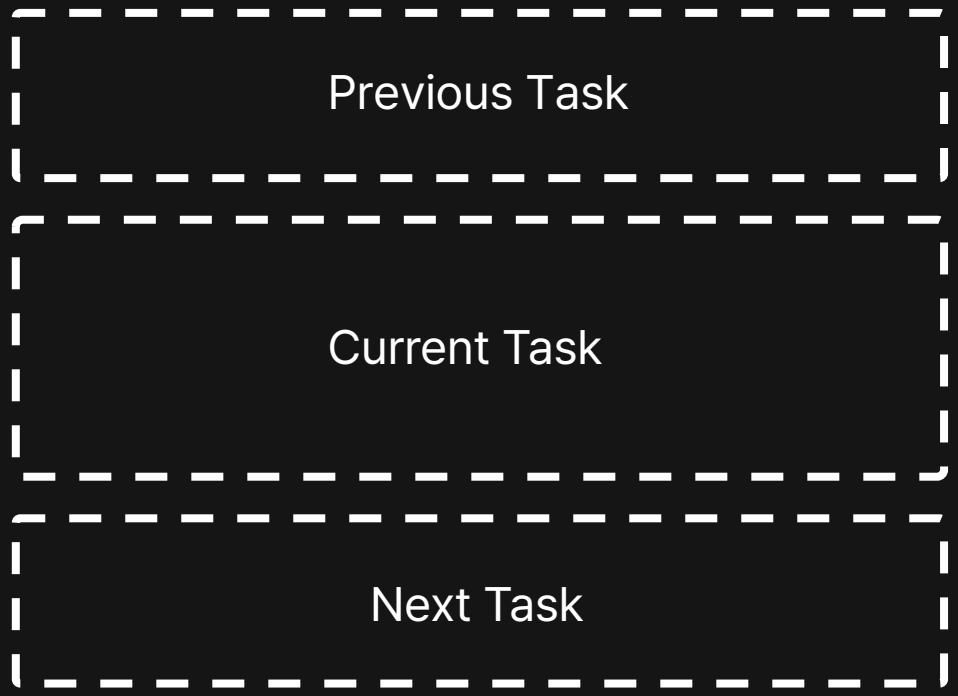
Object Info

...

Our goal is retrieved the behavior pattern for answering.
We found that metadata should be established based on behavior at specific points in time

Insights

** Data Regularization : Chunk Structure*



	Our chunk structure	Client chunk structure
Acc	83%	73%
TTFT(s)	4s 이내	4s 이내

When we construct the chunk structure, retrieval performance varied depending on the scope of the task, while an atomic decomposition show effective results.

Insights

** Try to various retriever strategies*

Graph RAG

- Recall 95%
- Latency 2s
- Cost Yes

Dense Retriever

- Recall 67%
- Latency 3s
- Cost No

Graph-Traverse

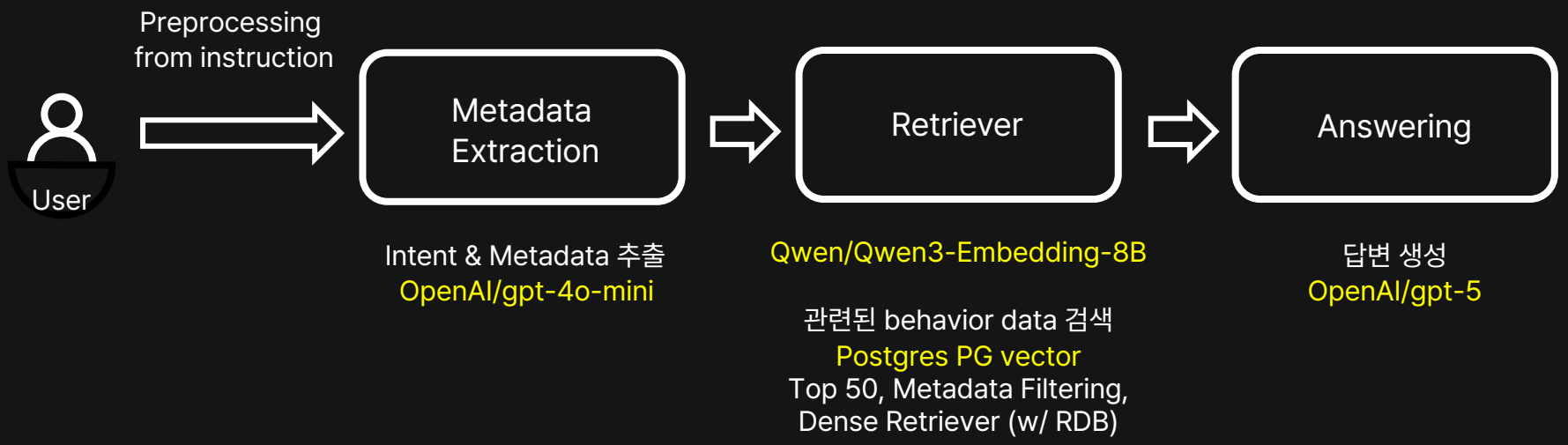
- Recall 87.5%
- Latency 10s
- Cost No

* Scenario based Retriever

- Recall 90%
- Latency 3.8s
- Cost No

Workflow

** Final workflow for requirements*



Cases of Failure

** Metadata Extraction*

8/4 ~ 8/8중 내가 파스타를 먹은 날은 몇일이야?

Metadata Extraction

OpenAI/gpt-4o-mini

```
{
  "intent": "statistic_query",
  "date": ["2025-08-04", ... ,"2025-08-08"],
  "region": None,
  "activity_info": None,
  "place_info": None,
  "place_info_name": None,
  "transfer_info": None,
  "food_type": "양식",
  "food_name": ”파스타"
  ...
}
```

Core Metadata is Date, Place, Act

PART 04

FAQ

04

Q1. RAG를 할 때 가장 중요한 포인트가 무엇인가요?

가장 중요하다고 생각하는 포인트는 바로 '평가' 입니다. RAG는 Retriever & Generation의 기술을 적절히 사용하는 것이 아닙니다. Domain에 가장 밀접하게 맞닿아 있음과 동시에 구현 시 정말 많은 모듈을 함께 고려하게 됩니다.

User Query의 변형, 데이터의 구성과 특징, Pre/Post Retriever, Retriever 방식과 Scoring, Embedding Model의 의존성, Prompt Engineering, Workflow / Agent Architecture 설계 등 일관된 성능을 뽑아내기 위해 고려해야 할 모듈도, 변인이 되는 부분이 너무 다양합니다. 이것들이 한데모여 하나의 Flow를 형성하게 되고 강건하고 높은 성능을 뽑기 위해서는 수행과정에서부터 올바른 성능평가를 이정표삼아 순차적으로 구축해야 합니다.

결국, 수행과정속에서의 신뢰성 있는 평가기준 수립과 평가 작업을 병행되지 않으면 인정할 수 없는 결과물이 만들어진다고 생각합니다.

그럼에도 불구하고, 여전히 Evaluation에 완전무결한 정답은 없기에 현재로서는 평가자체에 대한 고도화 및 체계화가 가장 신뢰성 있는 수단이 되지 않을까 싶습니다. 자세한 내용은 평가(Evaluation) Session 참고바랍니다.

Q2. 요즘은 MCP/Agent가 대세 아닌가요?

RAG안에도 무궁무진한 기술폭이 존재하는데 이것들이 무르익지 않고 MCP/Agent의 시대로 넘어가지 않았나 하는 생각을 크게 하고 있습니다.

따라서, 산업계에 정말 유효하게 Agent를 적용하고 싶다면, MCP/Agent에 대한 고찰과 함께 그것의 기반이 되는 RAG 역시 지속적으로 연구해야 한다는 생각을 강력하게 하고 있습니다.

더 나아가서 대세라서 MCP/Agent를 적용하는 것이 아닌 적용하고자 하는 환경과 목표 그리고 특성을 고려하여 적용하길 권해드립니다. MCP나 Agent의 유연함이 때로는 독이 되는 경우도 많이 존재하고 제가 경험한 바로는 그러한 경우가 더 많았습니다.

Thank You