NLP Term Project Presentation

TEAM 11

- 2391004 김남우
- 2391013 위다현
- 2391045 조유민

CONTENTS

01 DATA

02 DATA PREPROCESSING

03 STRATEGIES

04 FINAL MODEL

05 RESULTS

DATA

Business

External Documents

Harvard Business Review
Manager’s handbook

Law

External Documents

Barexam qa

Philosophy

External Documents

Stanford Encyclopedia of Philosophy

History

External Documents

AP World History: Modern Course and Exam Description

Psychology

External Documents

Psychology 2e

Simply Psychology

Wikipedia

Wikipedia extraction based on topic-specific keywords

DATA PREPROCESSING

1. PDF → TEXT
2. Chunking
3. Extract asterisks/appendixes/tables
4. Retriever Configuration and k Value Tuning
5. Smart EWHA Retriever

1

PDF → TEXT

1-1. Upstage Layzer vs PyMuPDF

Convert PDF to text

Upstage Layzer

PROS

- Favorable for RAG pipeline integration
- Basic paragraph extraction is stable

CONS

- Table structures and complex layout information were not extracted correctly
- Table, line structures, and separators were all flattened into plain text in the extracted output

VS

PyMuPDF

PROS

- Tables, line structures, and format layouts are reliably reflected.
- Document structure is well restored.

CONS

- Text files require post-processing when linked via embedding.

EWHA PDFs require precise layout retention, so **PyMuPDF** proved more accurate than Upstage Layzer

We used **PyMuPDF** as the final text extraction method

1

PDF → TEXT

1-2. Text vs Markdown

Choose the output format for extracted PDF text

Text

PROS

- Simple and compatible across all environments
- Fast embedding and tokenization

CONS

- Difficult to express segmentation
- Contextual distinctions can be difficult due to the lack of document structure

VS

Markdown

PROS

- Document structure can be expressed

CONS

- Resulting in a larger number of tokens and complex chunking tasks
- Post-processing is required for Markdown formatting after PDF extraction

PyMuPDF already preserved the paragraph structure well.

Storing the documents in **text** was more efficient in both performance and speed.

1

PDF → TEXT

1-3. Rule-based postprocessing for clause/chapter/paragraph unit structure recovery

Problems

- Key units like **“Article X / Chapter X / Section X”** appeared without line breaks

-> Merging important structural boundaries

Recovery Rules

- Insert a line break before **‘Article’**
- Insert a line break before **‘Chapter’**
- Insert a line break before **‘Section’**
- Insert a line break before numbered items (①②③④⑤)

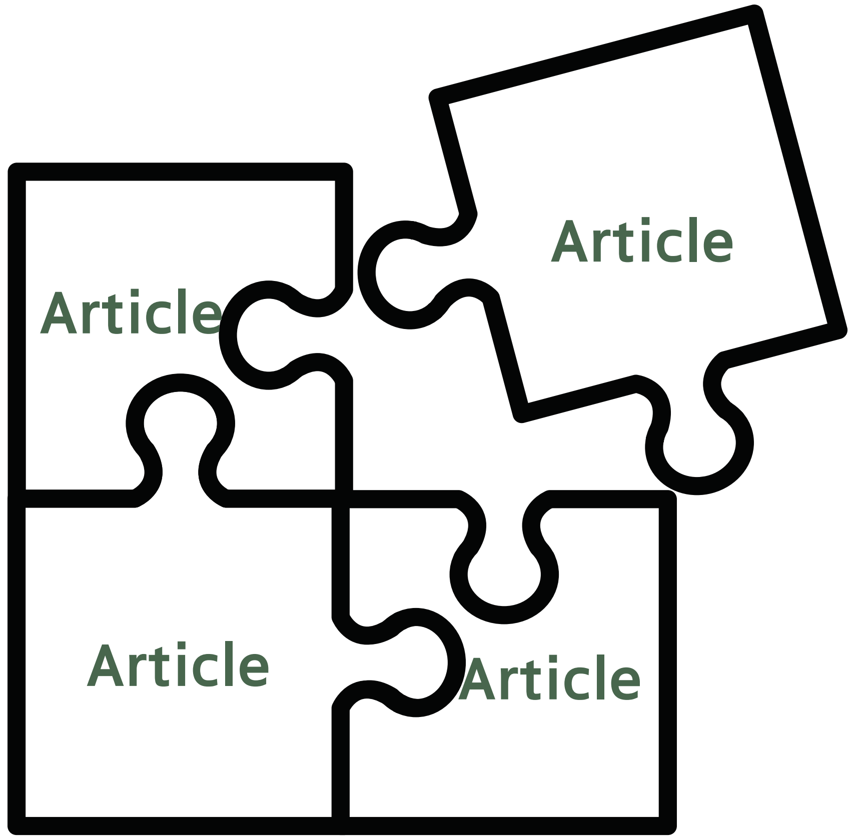
Text restored to a form nearly identical to the clause **structure of the original PDF**

Increased accuracy in searching for related clauses in RAG

Reduced semantic confusion by clearly separating units during chunking

2 CHUNKING

Article-level Chunking



Why ?

- EWHA PDFs use “Article units” as their most meaningful structural boundary
- Structure-based chunking is more suitable than fixed-length chunking

Method

1) Splitting by Article based on the "Article X" pattern
2) Assigning Metadata to Each Chunk
→ Allow queries to be matched instantly and improve recall in our RAG

3 EXTRACT ASTERISKS/APPENDIXES/TABLES (NUMBER TABLES)

3-1. Automatic detection of asterisks, appendices, and attached documents

Problems

- EWHA PDFs include appendices, supplements, attachments, and numeric table-like data in addition to the main articles.
- These sections follow different structural patterns, they cannot be captured using simple “Article X”– based chunking

Detection Logic

- Detect keywords such as **“Appendix”, “Supplement”, “Attachment”** as the starting point
- Extract everything until the next “Article X” as a single block

Previously missed or mixed sections can now be stored as independent chunks,
preventing retrieval errors and improving accuracy

3

EXTRACT ASTERISKS/APPENDIXES/TABLES (NUMBER TABLES)

3-2. Automatic Extraction of Number-Based Table Structures

Problems

- PyMuPDF preserves line breaks, numeric alignment, and spacing better than Upstage Layzer
- But it still cannot fully reconstruct actual table structures (rows/columns)

→

Detection Logic

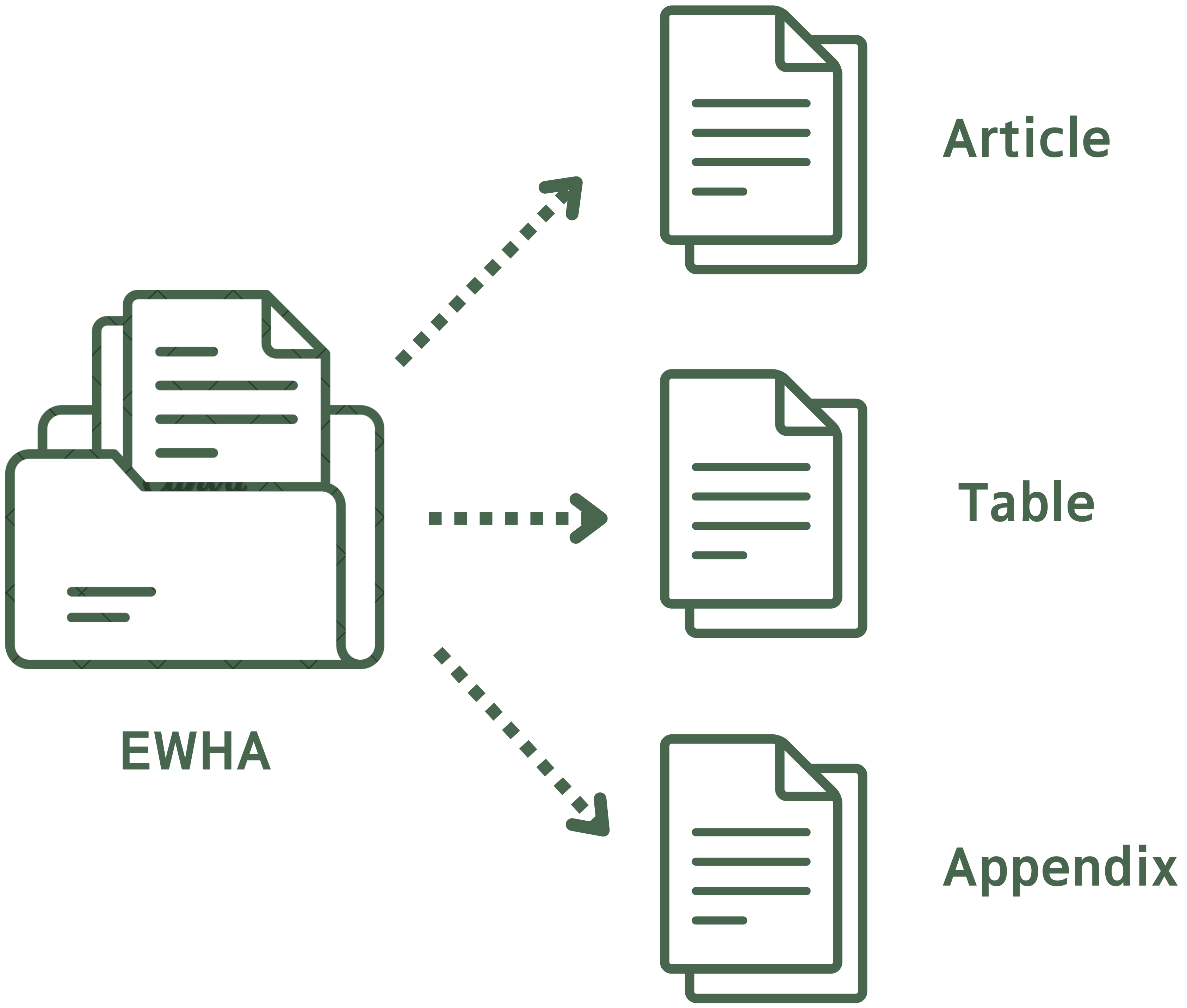
- Split the extracted text into lines
- If a line contains 2–3 digit numbers (years, credit units), mark it as a potential table line
- Extract two or more consecutive numeric lines as a single table block

Separately preserves broken table data in PDF extraction process
Prevents tables from mixing with article chunks, **improving retrieval performance**

3

EXTRACT ASTERISKS/APPENDIXES/TABLES (NUMBER TABLES)

3-3. Extracted blocks are stored as separate chunks



Extracted table/appendix blocks were stored **as separate chunks with metadata** → Prevent them from mixing with article chunks and improves retrieval accuracy through metadata-based filtering

4

RETRIEVER CONFIGURATION AND K VALUE TUNING

k=5

- Fast search speed
- Missing answers

VS

k=10

- Insufficient evidence is found when multiple clauses are linked in the school regulations

VS

k ≥ 25

- Increased noise
- No performance improvement

k = 18

Retrieval of most related articles

Inclusion of appendix/table chunks when relevant

→ Provide the **best balance between accuracy and recall**

5

SMART EWHA RETRIEVER

1. Query-Based Keyword Extraction

- Extract meaningful words (Korean/English) from user-entered questions
- Remove irrelevant articles early
- Perform semantic filtering prior to vector search

2. Candidate Filtering

- Iterates over each clause/appendix/table chunk and calculates a score based on how many times the query keyword is included

3. Using the Entire Corpus When Too Few Candidates

4. Local Chroma Generation

- A temporary **Chroma VectorStore** is created using the extracted candidate chunks

5. Top-k Vector Search

- Performs a top-k search on the temporary chroma
- **Hybrid method**

DOMAIN SPECIFIC TESTING

01 MMLU Domain Specific Test

- Get domain-specific test data from mmlu
- Data selection(Documents, Wikipedia) based on results

02 Text Extract

- Extract text from txt and pdf files, create documents, and merge them with Wikipedia data

03 Chunking

- Adjust chunk size and overlap size
- chunk_size: 1000 → 500
- chunk_overlap: 100 → 50

04 Prompts

- Modify the prompt to output a **clear answer** other than the description

STRATEGIES

1. Vector DB Performance Comparison
2. Multi-Retriever Implementation
3. Question Classifier
4. Handling Classification Failure
5. Retriever Fallback
6. Self-Verification

1

VECTOR DB PERFORMACE COMPARISON

Chroma

PROS

- Document + metadata can be stored integrally.
- High integrity with LangChain
  - The process is a one-line flow:
    Embedding → Storage → Search

CONS

- Performance may be lower than FAISS at large

VS

FAISS

PROS

- Overwhelming search performance for large-scale vectors

CONS

- No metadata storage function
- Complex conversion process when used with LangChain

✓ The core of this project lies in **structure-based chunking and metadata filtering**.

✓ **The data scale is not that large** to necessitate the high performance of FAISS.

Considering development efficiency and search accuracy,

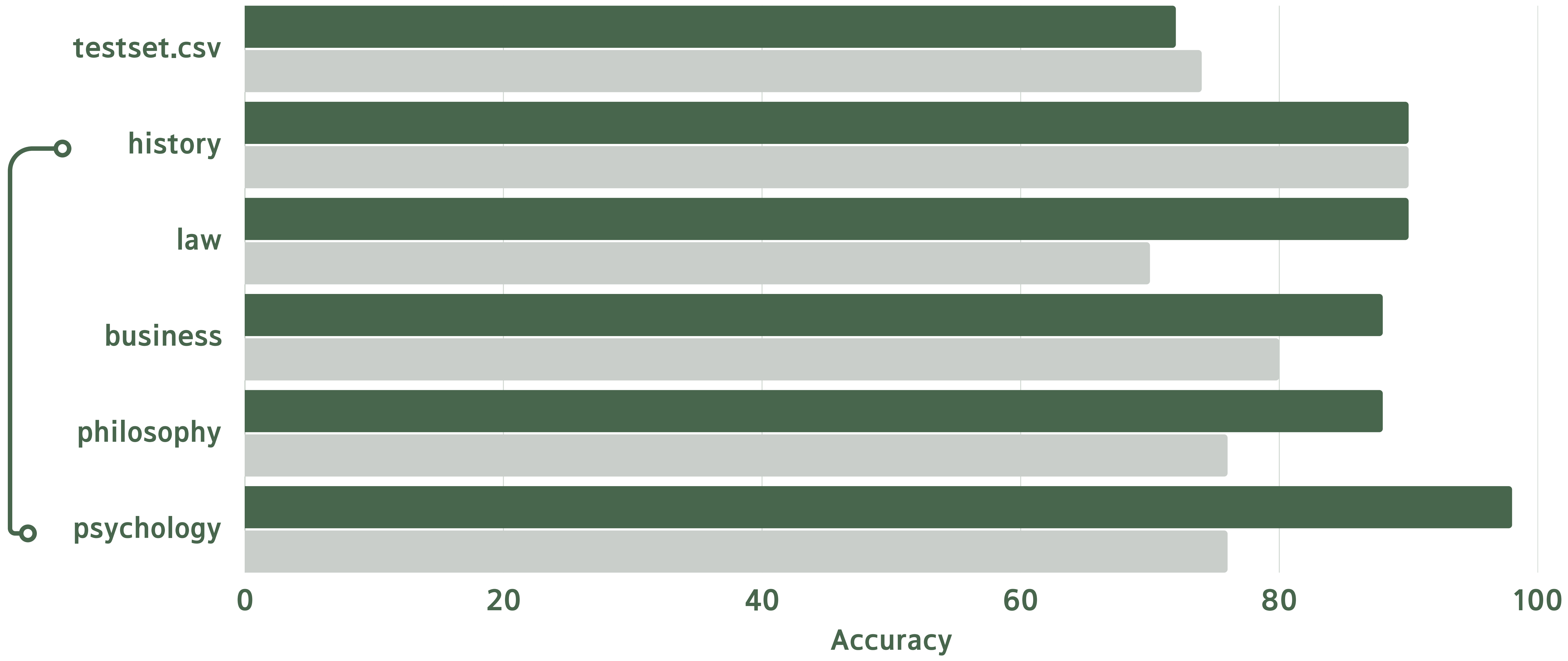
Chroma is better vectorspace for this project.

1

VECTOR DB PERFORMACE COMPARISON

Chroma > FAISS

The performance when using **Chroma** was marginally **higher** than that of **FAISS**.

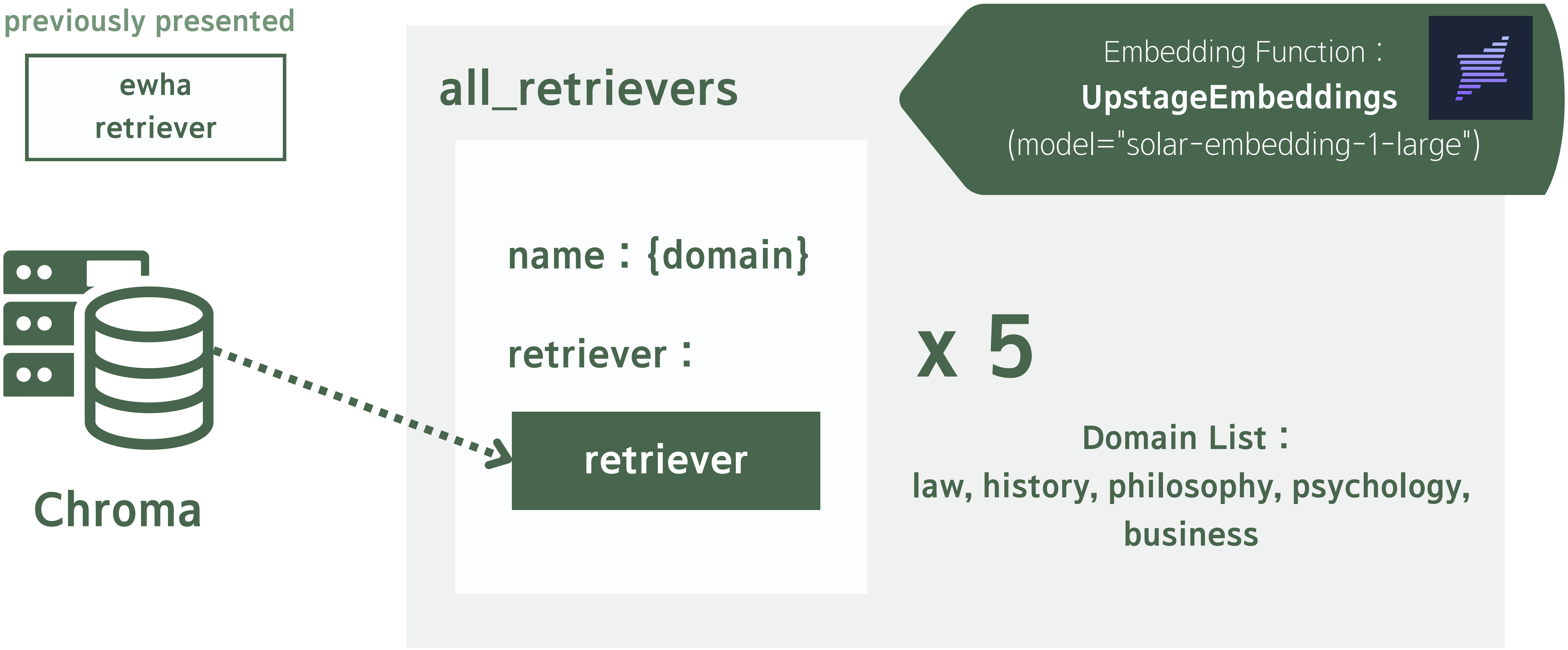


MMLU
test Datasets

2 MULTI-RETRIEVER IMPLEMENTATION

Specialization by creating a separate Retriever for each domain field.

Purpose : Enhance the specialization and accuracy of retrieval for specific queries.



previously presented

ewha retriever

Chroma

all_retrievers

name : {domain}

retriever :

retriever

x 5

Embedding Function :
UpstageEmbeddings
(model="solar-embedding-1-large")

Domain List :
law, history, philosophy, psychology, business

3 QUESTION CLASSIFIER

Analyze the query and classify it into a domain.

Purpose : Maximize search efficiency by reducing unnecessary searches.

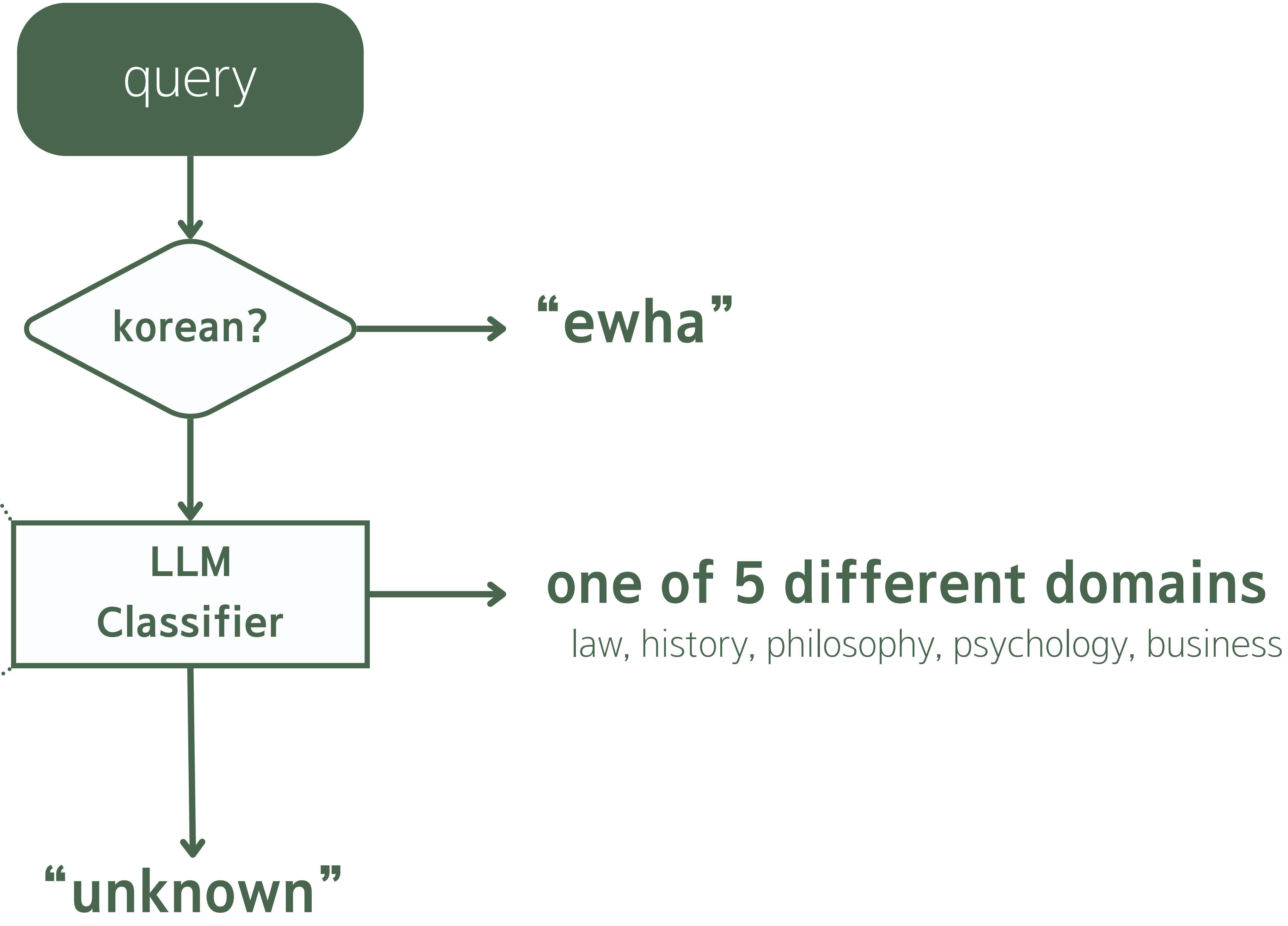
LLM :

ChatUpstage

(model="solar-pro2")

system_prompt

"Classify the input question into one of the following fields: {field_list}. Output must be only one of the field names or '{UNKNOWN_FIELD}' if ambiguous/irrelevant. Output only the name, with no extra text or punctuation."



query

korean?

"ewha"

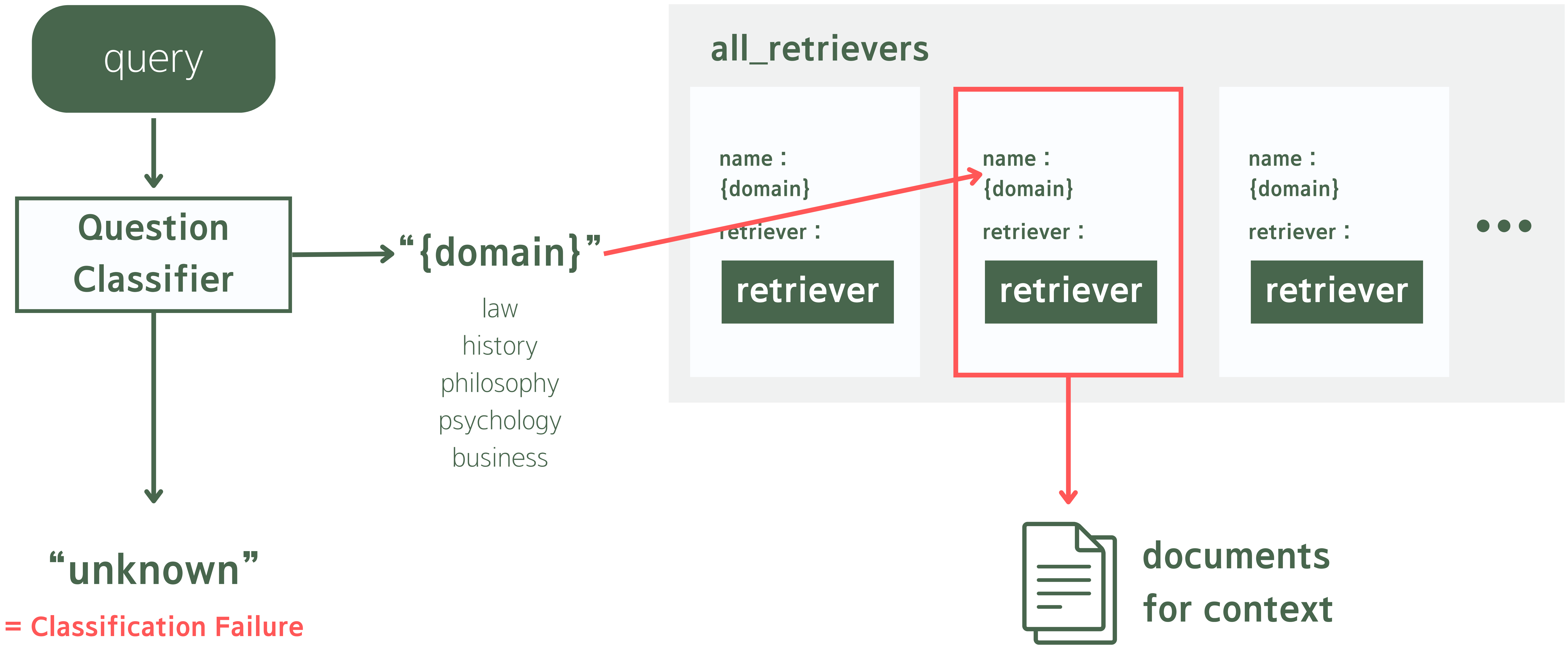
LLM Classifier

one of 5 different domains

law, history, philosophy, psychology, business

"unknown"

3 QUESTION CLASSIFIER



query

Question Classifier

"{domain}"

law

history

philosophy

psychology

business

"unknown"

= Classification Failure

all_retrievers

name : {domain}	name : {domain}	name : {domain}
retriever : retriever	retriever : retriever	retriever : retriever

•••

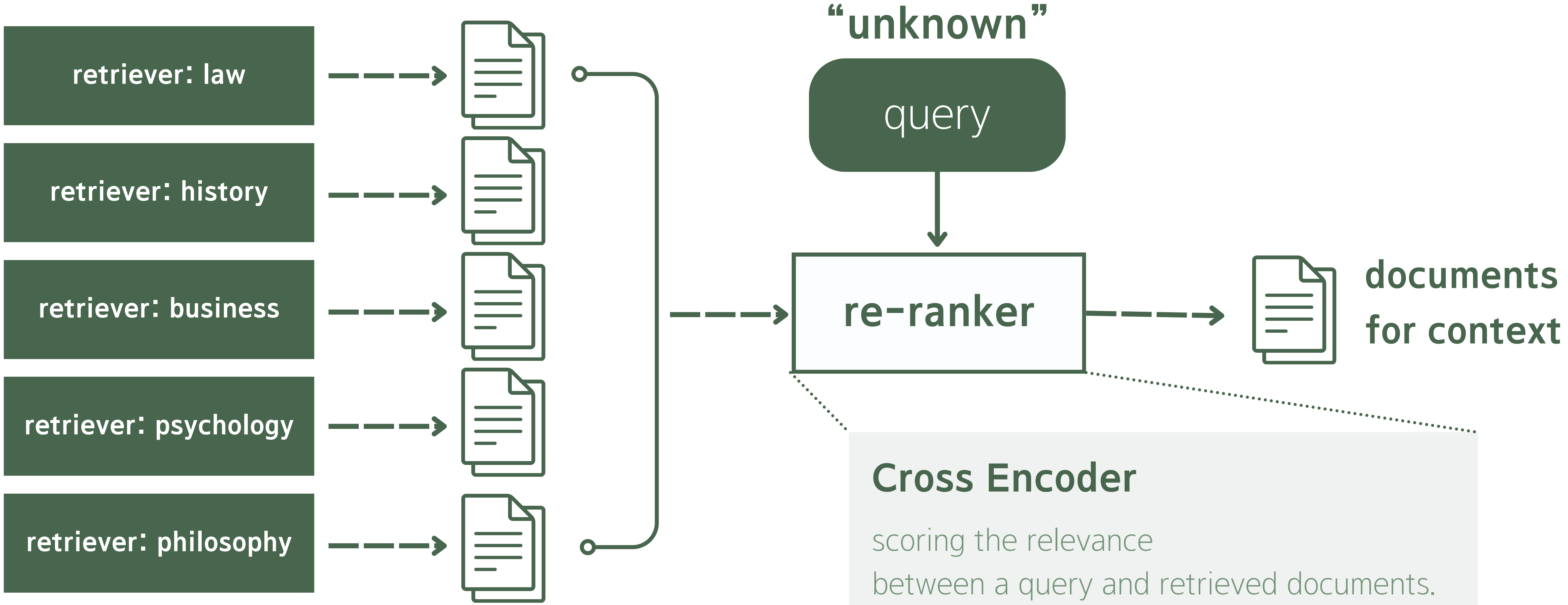
documents for context

4

HANDLING CLASSIFICATION FAILURE

Purpose : Achieve System Stability and Coverage.

Ensemble Search and Re-ranking



Cross Encoder

scoring the relevance
between a query and retrieved documents.

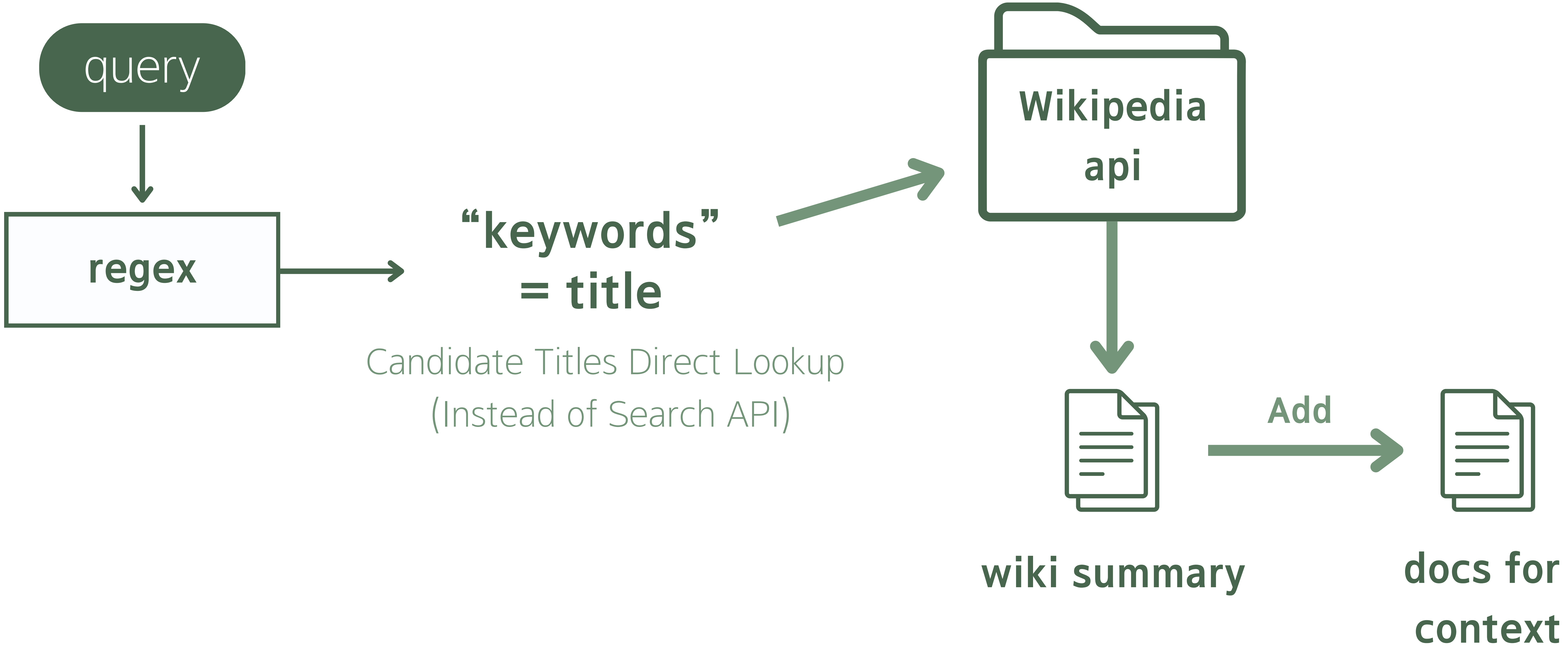
5 RETRIEVAL FALLBACK : WIKIPEDIAAPI

Purpose : Supplement Chroma db with **Wikipediaapi**

Keyword-Based RAG Problems

- Keywords do not fully match the actual problem
- Keywords not in vector DB appear

=> Retrive failed

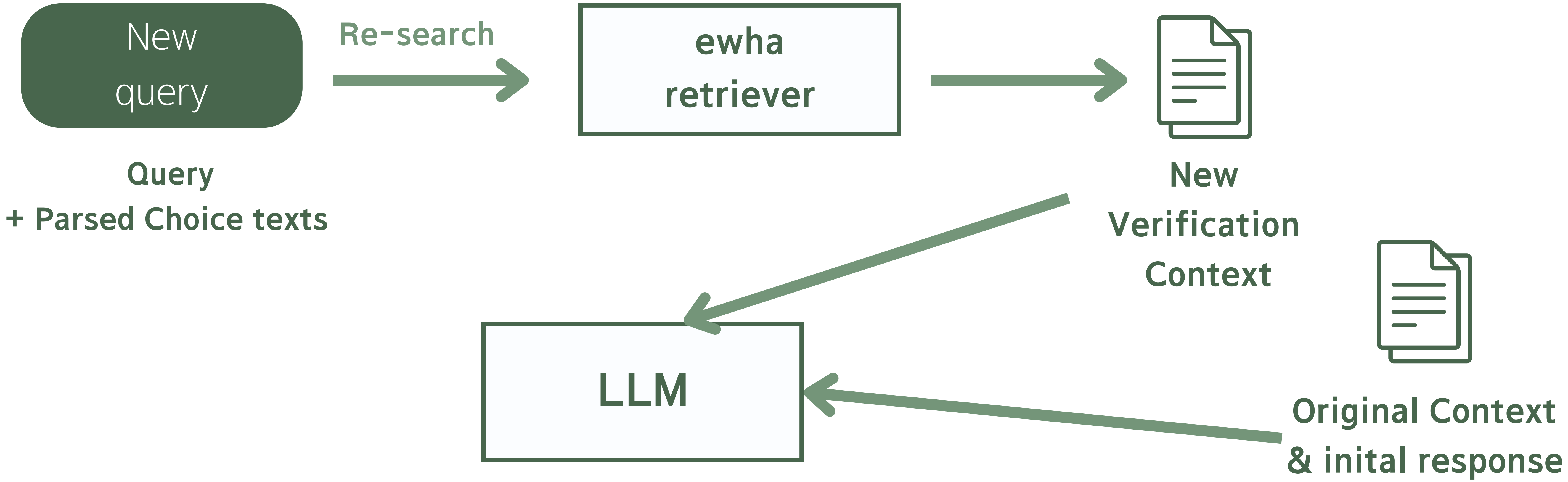


Candidate Titles Direct Lookup
(Instead of Search API)

6 SELF-VERIFICATION

Purpose : Reverify the **reliability** of the initial answers in **EWHA questions**

Reliability-based verification function



Task:

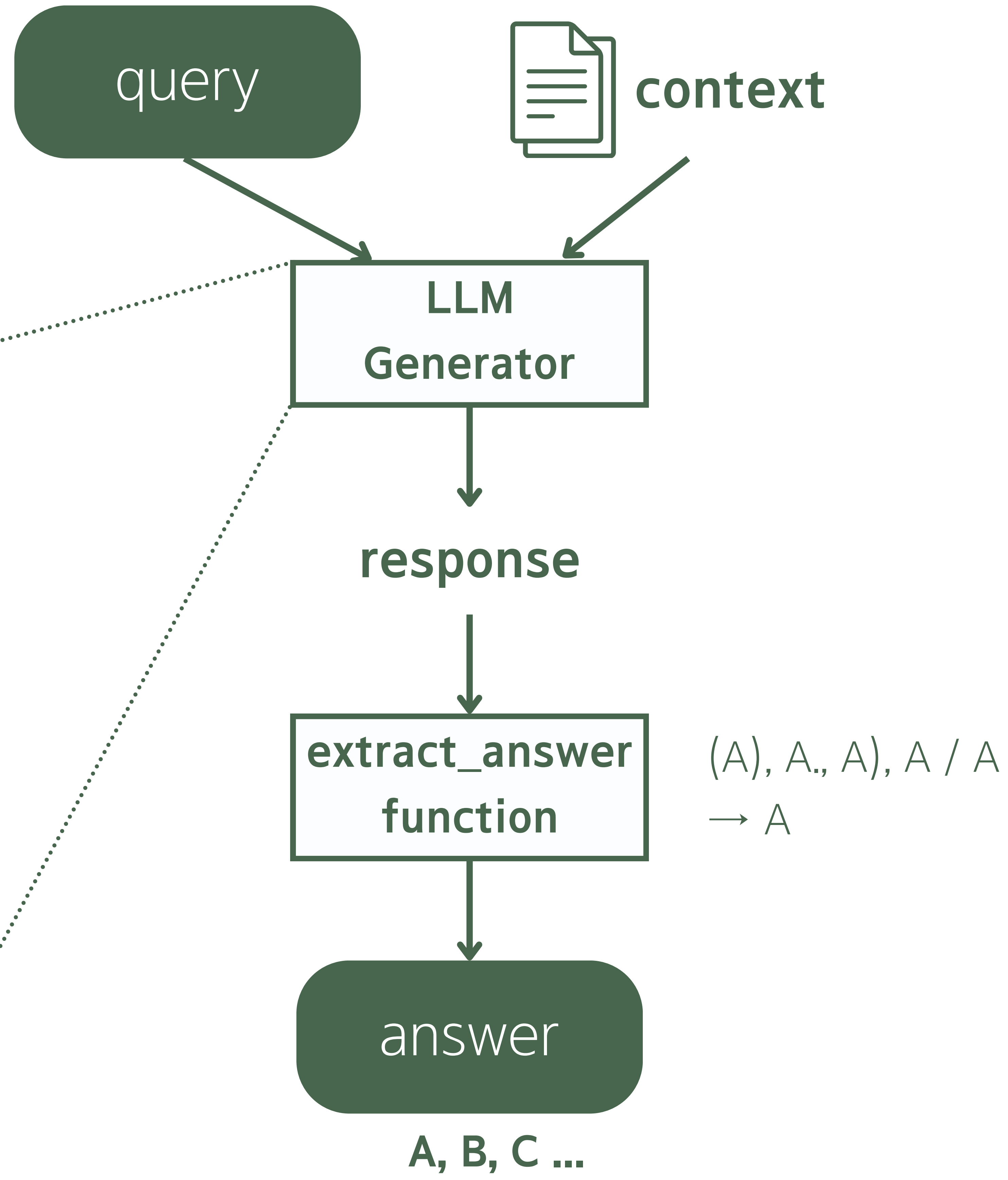
1. Determine if the new context STRONGLY contradicts the original answer
2. Only suggest a different answer if you are VERY confident (>80%)

Generator

LLM :
ChatUpstage
(model="solar-pro2")

prompt

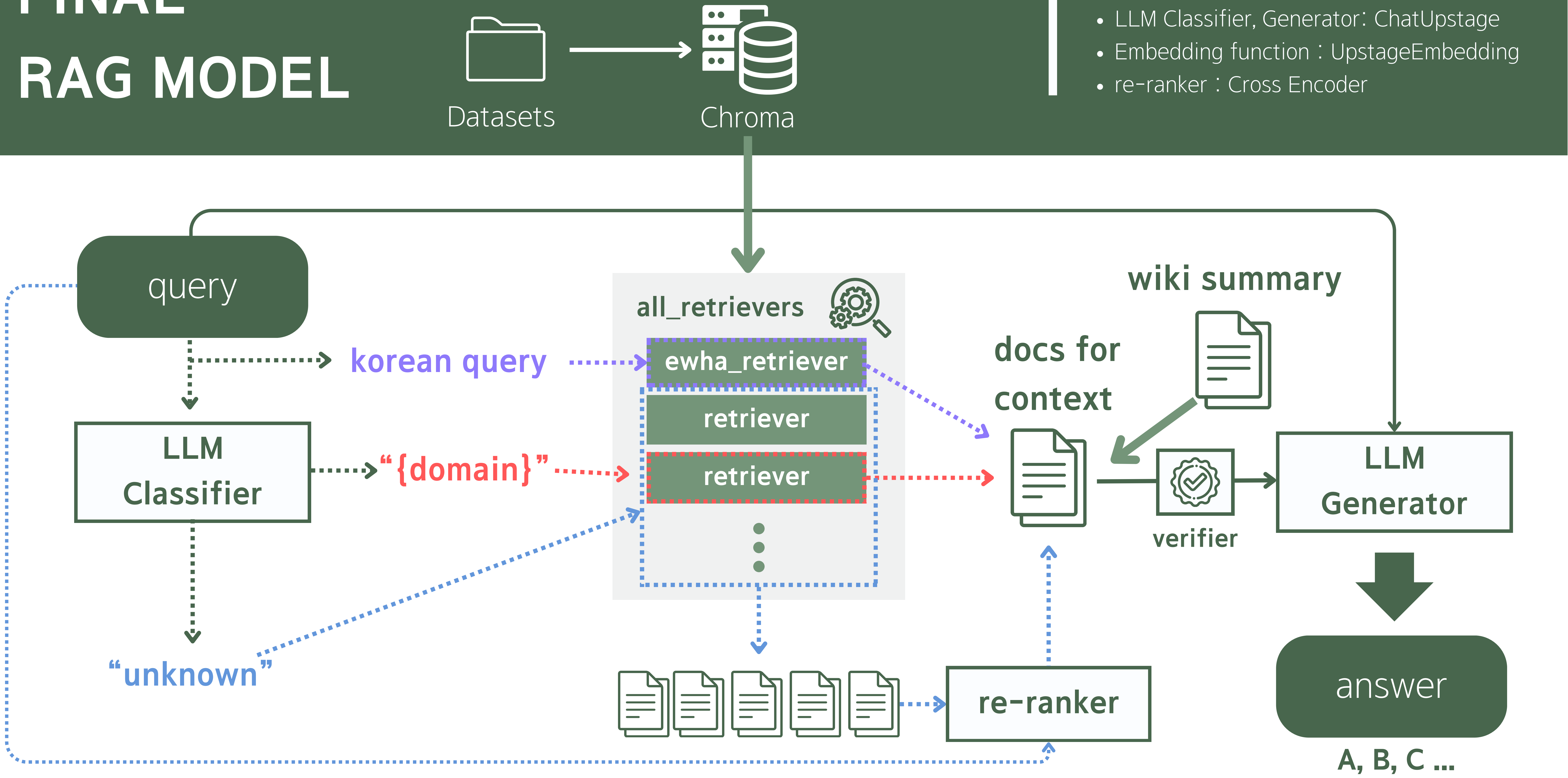
Select the correct multiple-choice answer.
Question: {question}
Context: {context}
Respond ONLY with the best option letter (A, B, C, ...).



FINAL RAG MODEL

Model

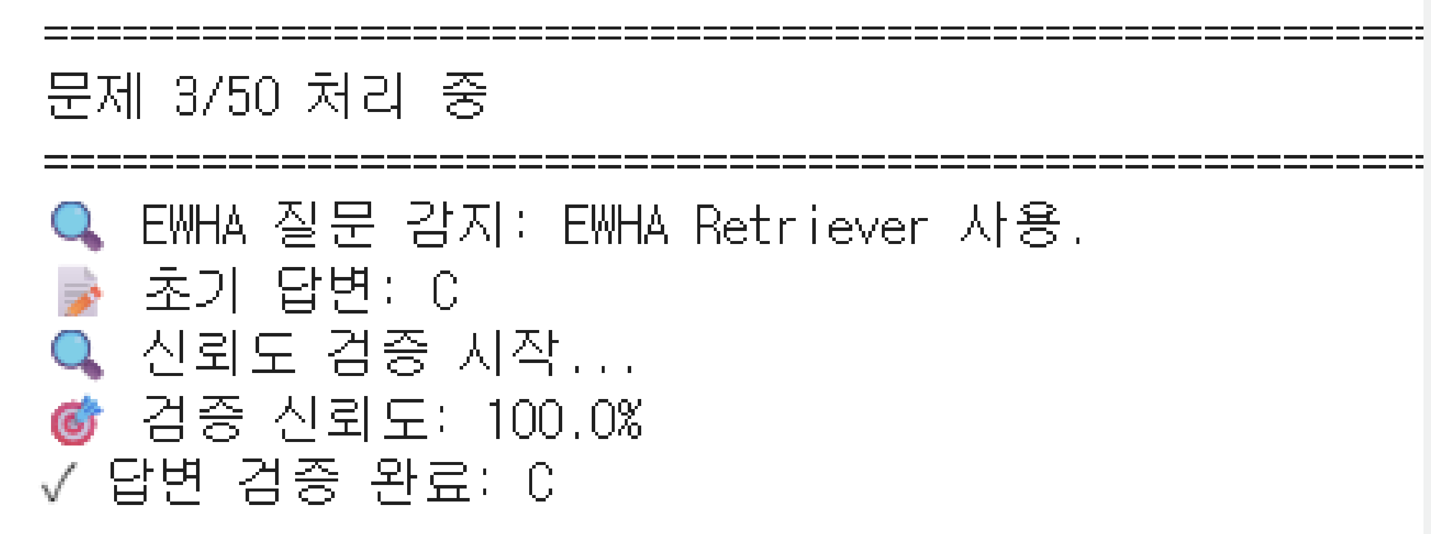
- LLM Classifier, Generator: ChatUpstage
- Embedding function : UpstageEmbedding
- re-ranker : Cross Encoder



RESULTS

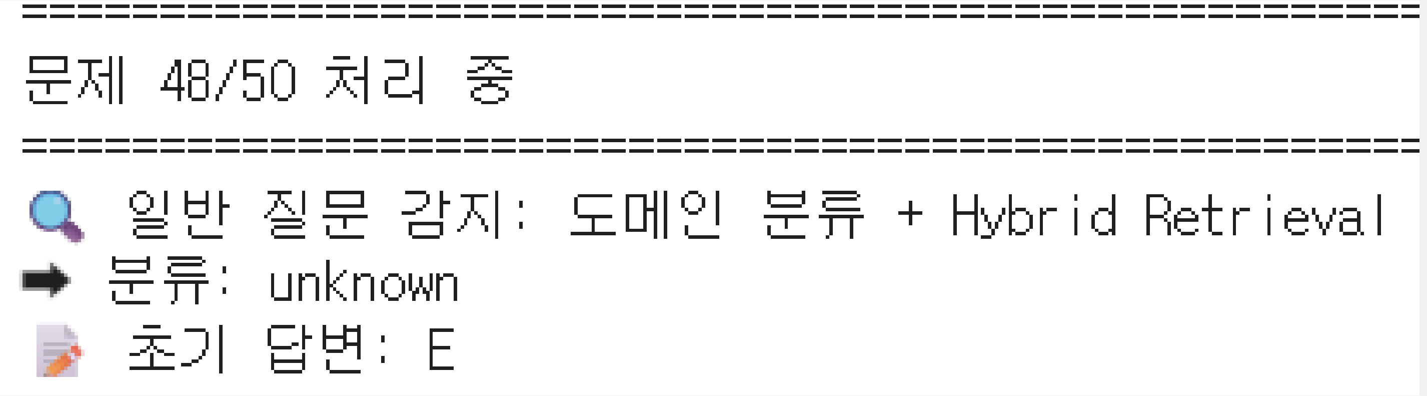
testset.csv

ewha

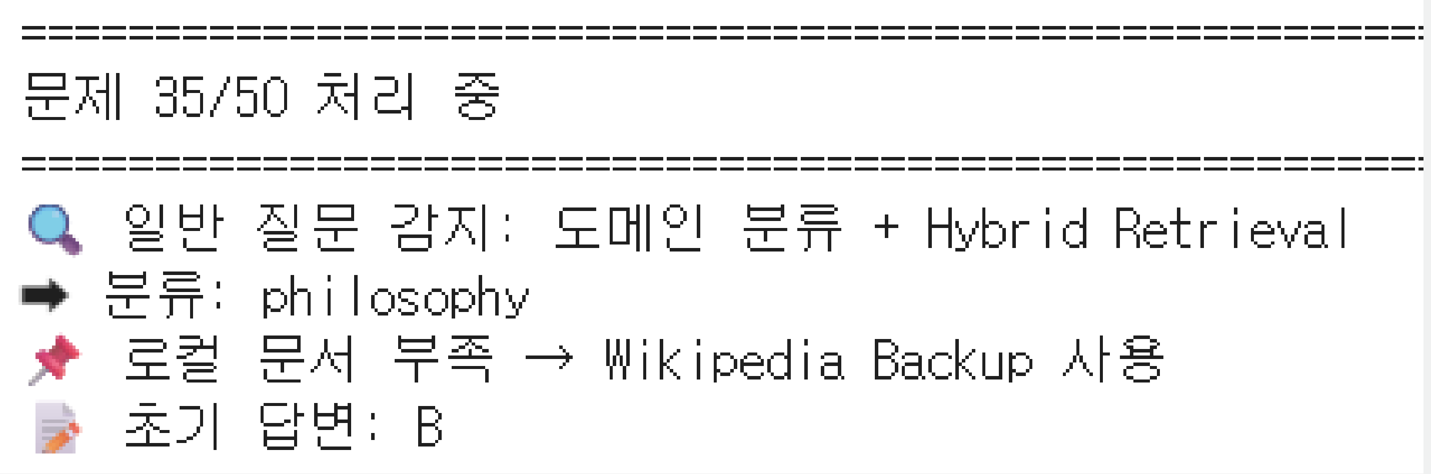


```
============================================================
문제 3/50 처리 중
============================================================
🔍 EWHA 질문 감지: EWHA Retriever 사용.
📝 초기 답변: C
🔍 신뢰도 검증 시작...
🎯 검증 신뢰도: 100.0%
✓ 답변 검증 완료: C
```

MMLU

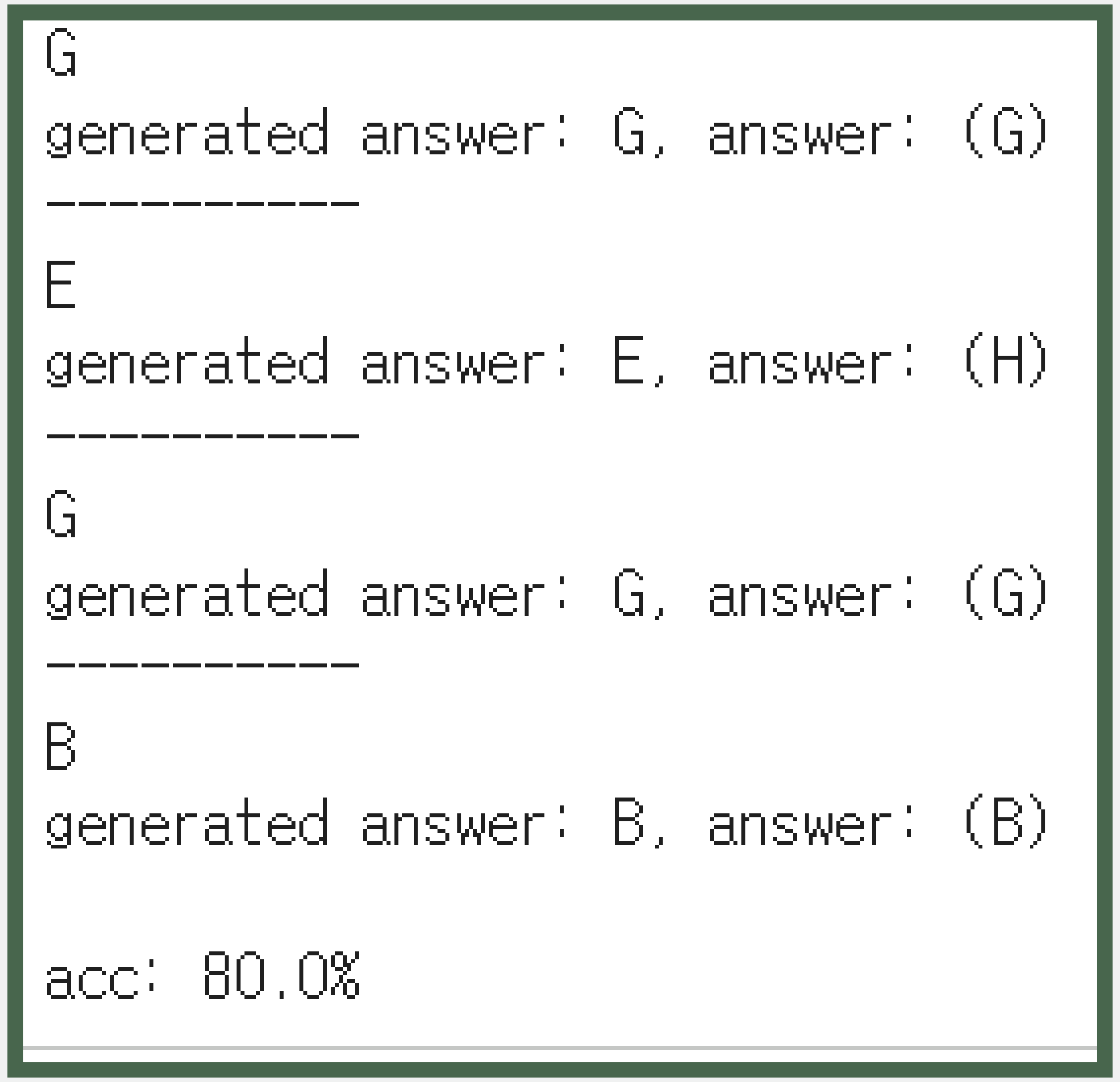


```
============================================================
문제 48/50 처리 중
============================================================
🔍 일반 질문 감지: 도메인 분류 + Hybrid Retrieval
➡ 분류: unknown
📝 초기 답변: E
```



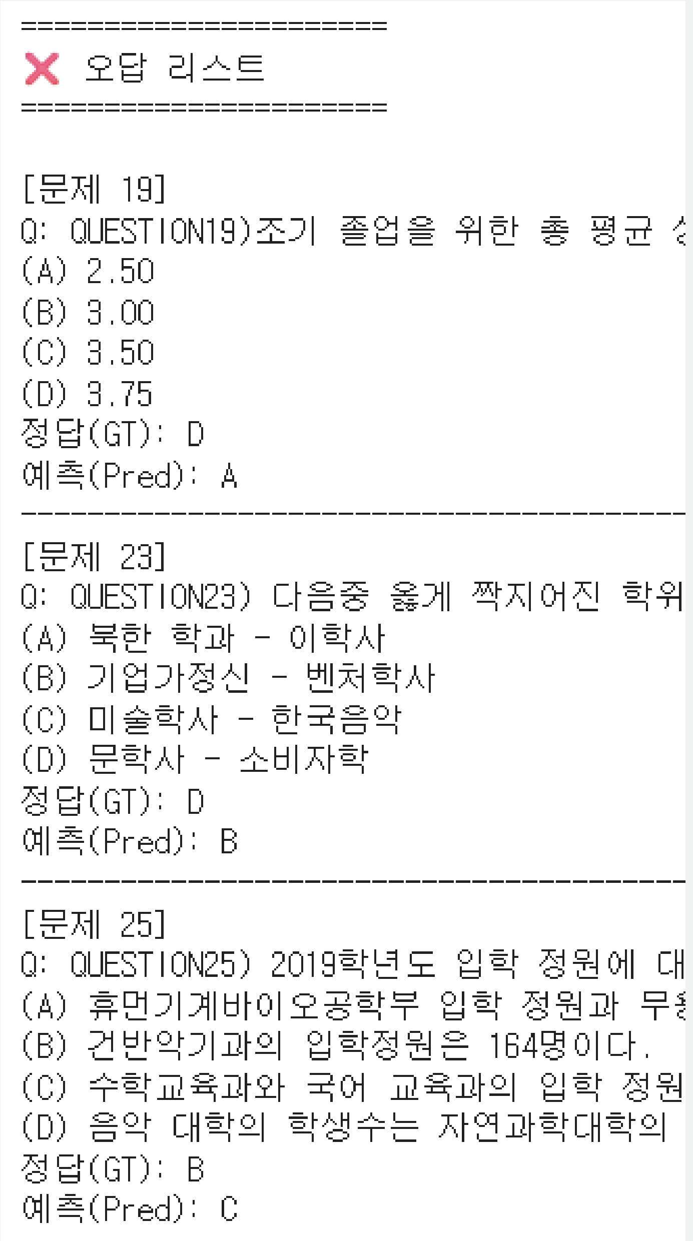
```
============================================================
문제 35/50 처리 중
============================================================
🔍 일반 질문 감지: 도메인 분류 + Hybrid Retrieval
➡ 분류: philosophy
📌 로컬 문서 부족 → Wikipedia Backup 사용
📝 초기 답변: B
```

extract answer



```
G
generated answer: G, answer: (G)
-----------
E
generated answer: E, answer: (H)
-----------
G
generated answer: G, answer: (G)
-----------
B
generated answer: B, answer: (B)

acc: 80.0%
```



```
======================
❌ 오답 리스트
======================

[문제 19]
Q: QUESTION19)조기 졸업을 위한 총 평균
(A) 2.50
(B) 3.00
(C) 3.50
(D) 3.75
정답(GT): D
예측(Pred): A
--------------------------------------------
[문제 23]
Q: QUESTION23) 다음중 옳게 짝지어진 학위
(A) 북한 학과 - 이학사
(B) 기업가정신 - 벤처학사
(C) 미술학사 - 한국음악
(D) 문학사 - 소비자학
정답(GT): D
예측(Pred): B
--------------------------------------------
[문제 25]
Q: QUESTION25) 2019학년도 입학 정원에 대
(A) 휴먼기계바이오공학부 입학 정원과 무
(B) 건반악기과의 입학정원은 164명이다.
(C) 수학교육과와 국어 교육과의 입학 정원
(D) 음악 대학의 학생수는 자연과학대학의
정답(GT): B
예측(Pred): C
```

Final Accuracy : 80%

- ewha : 22/25
- MMLU : 18/25

Any Questions?

Team contribution

	Data Collection	Modeling Ideas	Presentation
2391004 김남우	business, philosophy	• Choice-Guided Verification • Context-Contrast Scoring • Confidence-Calibrated Correction	PPT (Front part)
2391013 위다현	law, psychology	• Multi-Retriever System • Question Classifier • Fallback Strategy : re-ranking	PPT (Back part)
2391045 조유민	history, ewha	• Initial Baseline Model • Ewha Data Preprocessing • Ewha Retriever • Retrieval Fallback : Wikipedia API Fallback	PPT Editing Presentation

THANK YOU

- 2391004 김남우
- 2391013 위다현
- 2391045 조유민