



커널 아카데미 : AI 부트캠프

Scientific Knowledge Question
Answering Seminar 2025. 11. 27(목)

목차

01. 팀 소개

02. 경진대회 수행 절차 및 방법

03. 분석 인사이트 및 결과

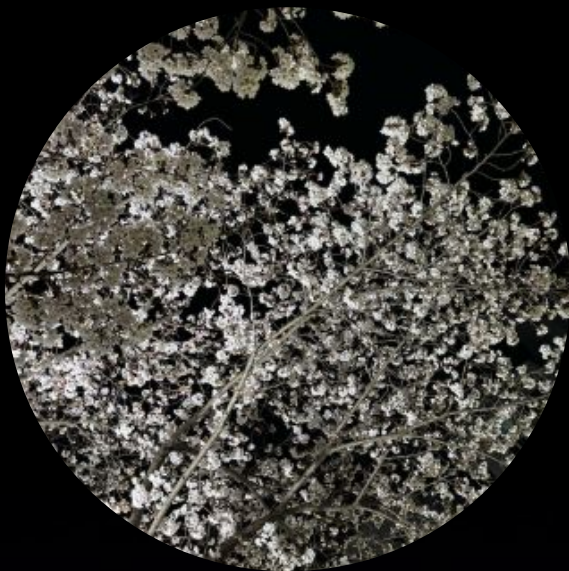
04. 회고

01

팀 소개

팀장/팀원 소개
협업 방식

* IR_AD_2



팀장

김명철
산업공학



팀원

김수환
통계



팀원

김병현
정보보호학



팀원

송인섭
기계공학



팀원


이가은
생명과학

경진대회 협업 방식

Scientific Knowledge Question Answering | 과학 지식 질의 응답 시스템 구축

협업 마인드셋 : 이슈나 자료 공유는 바로 바로!

협업 진행 횟수 및 일정 : 매일 오전 10시 / 이슈 발생 시 미팅 진행



김병현 오전 1:16

README 읽어보시고 랭그래프 리팩토링 및 실험 관련 문서는 "experiments/" 폴더에 상세하게 적혀있습니다.

code/ 폴더 안에서 엘라스틱 서치 설치하시고 바로

```
uv run python code/scripts/rag_with_langgraph.py --skip-index
```

하시면 문제없이 실행됩니다.. 자세한 파라미터 옵션은 문서 참고하십쇼 그럼 이만.

(env도 통째로 첨부했는데 solar 키 쓰실분 그냥 쓰십쇼 free key 선언합니다 100

+

17

미팅/회의

회의

이 이가은

완료

4회차 피어세션

회의

이 이가은

완료

18

미팅/회의

회의

이 이가은

완료

To do list

To do

이 이가은

완료

To do list

To do

명 명철 김

완료

19

미팅/회의

회의

이 이가은

완료

20

미팅/회의

회의

이 이가은

완료

To do list

To do

병 병현 김

완료

02

경진대회 수행 절차 및 방법

목표 수립
수행 내용 / 수행 결과

경진대회 목표 수립

Scientific Knowledge Question Answering | 과학 지식 질의 응답 시스템 구축

주제

Scientific Knowledge Question Answering | 과학 지식 질의 응답 시스템 구축
질문과 이전 대화 히스토리를 보고 참고한 문서를 검색엔진에서 추출 후 이를 활용하여 질문에 적합한 대답을 생성하는 태스크

목표

목표

* 대회니까 1등!!!!

많은 방법들을 적용하며, 최대한 체화하자.

개요

소개 및 배경 설명

* 환각 문제를 가진 LLM의 한계를 극복하기 위해, 과학 상식
문서를 검색·활용하는 RAG 시스템을 직접 구축해 보는
대회입니다.

기간

2025. 11. 14 ~ 2025. 11.27

경진대회 수행 내용

Scientific Knowledge Question Answering | 과학 지식 질의 응답 시스템 구축

1

→

* 개발 환경 구축

* 버전관리: Github

* 패키지 관리/가상 환경: UV

* python 3.12.12

2

→

* 데이터 분석

* documents.jsonl: 4272개

- 자연과학/공학/사회/인문/기타로 이루어진 16개 과목

eval.jsonl: 220개

- 일상대화: 20개

- 멀티턴: 20개

3

→

* Retrieval 모델

* Sparse: Elasticsearch의 BM25

Dense: Upstage Embedding model

- 'solar-embedding-1-large'

4

→

* 구조

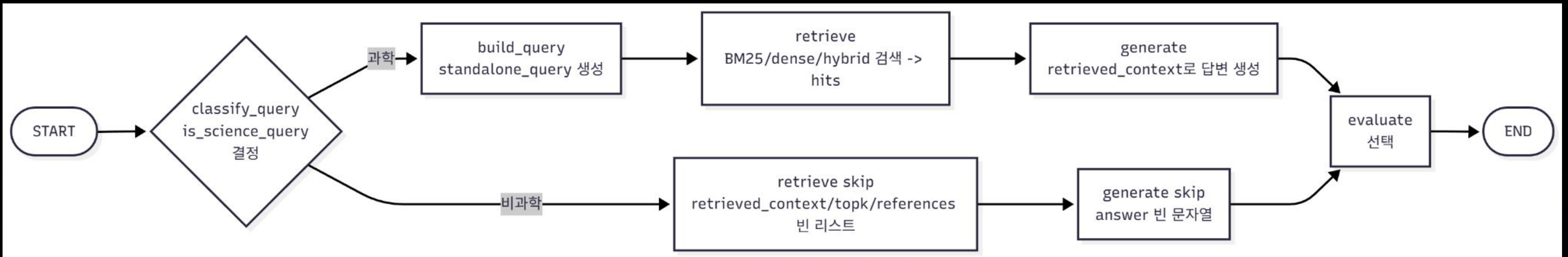
* langgraph 도입

- 일상/과학 질문 필터링후 분기

- sparse/dense hybrid

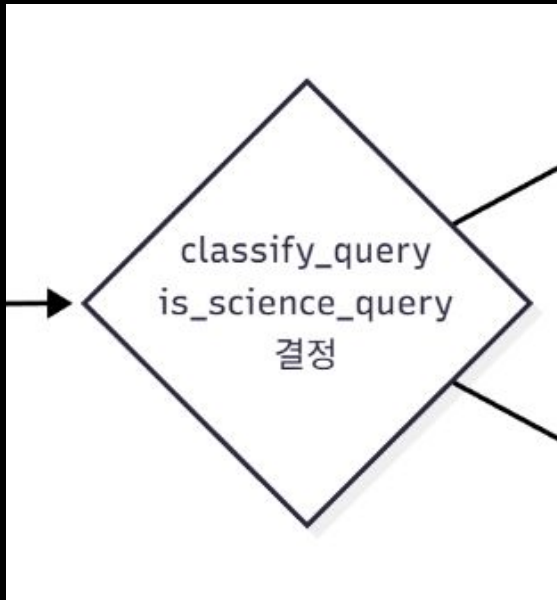
경진대회 수행 결과

Scientific Knowledge Question Answering | 과학 지식 질의 응답 시스템 구축



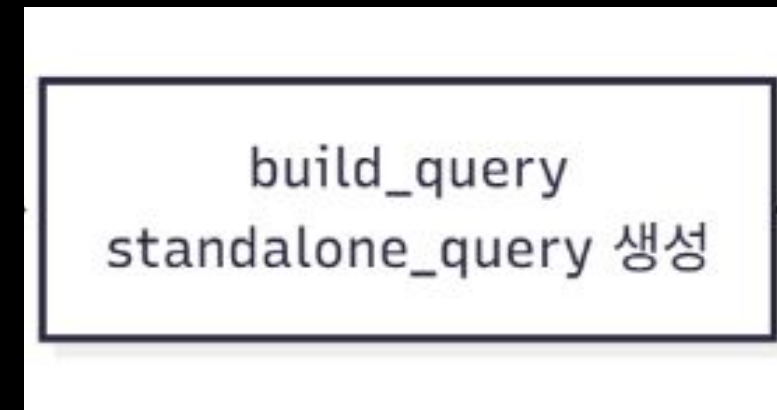
경진대회 수행 결과

Scientific Knowledge Question Answering | 과학 지식 질의 응답 시스템 구축



classify_query

- 비과학 질의 여부를 분류해 `is_science_query` 플래그에 기록
- 입력: `state["message"]` - 사용자 대화나 질문
- 외부 콜백 : `Ture` -> 과학 `False` -> 비과학
- 출력: 콜백 없으면 기본값 `True`, 있으면 반환값으로 설정
- 흐름: 모두 `build_query_node`로 이동 (이후 노드들에서 플래그 결과로 RAG 돌릴지 빈 결과 낼지 결정)



build_query

- `standalone query`를 생성하고 상태에 저장
- 입력: `state["message"]` / `state["is_science_query"]`
- 외부 콜백 : 대화 맥락을 반영한 검색용 질의 문장 생성
- 출력: `is_science_query == False` 시 빈 문자열 / 과학 질문이나 콜백 시 메시지 출력
- 흐름: `retrieve` 노드로 이동

경진대회 수행 결과

Scientific Knowledge Question Answering | 과학 지식 질의 응답 시스템 구축

retrieve

BM25/dense/hybrid 검색 ->
hits

Retrieval 실험 내용

- 초기에 sparse(BM25) + dense 하이브리드 계획
- 하지만, Solar 임베딩을 진행한 후 dense-only 프로젝션만으로 topk가 안정적으로 맞아 최고점수 달성.
- 이 후, sparse 비율을 계속 늘어보면서 하이브리드 실험을 했지만, dense-only 일 때의 점수가 가장 높게 나오는 상황을 확인.
- 로컬 정답지가 없어서 근거를 명확하지는 않지만, topk=3 고정인 rule에서 sparse를 섞으면 topk 변동이 생겨 성능이 하락한 것이라 추측. (토큰화/중복 이슈)

retrieve skip

retrieved_context/topk/references
빈 리스트

generate skip
answer 빈 문자열

비과학 질의 정규화 필터링 내용

- 정상 비과학 질의는 topk를 비워 제출하면 보상을 받지만, 문서를 제출하면 패널티가 적용됨.
- eval.jsonl 전수 조사 후 감정/인사/모델 메타·잡담 패턴을 정규식으로 정의해 is_science_query에서 분기 처리.
- 초기 12건 → 모델 신원/능력 질문, 취향/날씨/상담 등 패턴을 확장해 17건으로 포착, 모두 topk=[], answer=""로 제출. dense-only 일 때의 점수가 가장 높았다는 사실을 확인. (휴리스틱 키워드를 검색하고 메타 질문에 대한 규칙을 추가해 private 데이터에 robust하게 대응)
- 제출 로그에서 비과학 건수와 topk 빈 ID가 일치하는지 매번 검증해 오탐/미탐을 최소화.
- 효과: 불필요한 검색/LLM 호출을 줄이고, 비과학 질의에서 패널티를 피하며 MAP/MRR이 상승(0.79 → 0.88).

경진대회 수행 결과

Scientific Knowledge Question Answering | 과학 지식 질의 응답 시스템 구축

```
def build_rag_graph(callbacks: RagGraphCallbacks):  
    graph = StateGraph(RagPipelineState)  
  
    graph.add_node("classify_query", lambda state: _classify_node(state, callbacks))  
    graph.add_node("build_query", lambda state: _build_query_node(state, callbacks))  
    graph.add_node("retrieve", lambda state: _retrieve_node(state, callbacks))  
    graph.add_node("generate", lambda state: _generate_node(state, callbacks))  
    graph.add_node("evaluate", lambda state: _evaluate_node(state, callbacks))  
  
    graph.add_edge(START, "classify_query")  
    graph.add_edge("classify_query", "build_query")  
    graph.add_edge("build_query", "retrieve")  
    graph.add_edge("retrieve", "generate")  
    graph.add_edge("generate", "evaluate")  
    graph.add_edge("evaluate", END)  
  
    return graph.compile()
```


03

분석 인사이트 및 결과

문제 및 인사이트 도출
해결 방법 및 결과

경진대회 인사이트 공유

Scientific Knowledge Question Answering | 과학 지식 질의 응답 시스템 구축

01. 일상대화 식별 필터 도입

- * eval.jsonl에 일상대화 존재 확인 후 필터링 적용
- 일상 대화 20개 중 17개 분기

```
# 감정/응답 표현을 담은 정규식 목록. eval.jsonl에서 추출한 사례들 기반으로 구성했다.
NON_SCIENCE_PATTERNS = [
    re.compile(pattern)
    for pattern in [
        r"안녕|반가", # 인사/접대
        r"나기\.*대답을\.*알려줘", # 모델 정전/대화 언급
        r"(기분이\.*좋아|신나|즐거워)", # 감정 표현
        r"힘들|힘든|힘드|우울", # 고인/상당 요청
        r"그만\.*해줘", # 대화 종료 표현
        r"너는\.*(누구|어디|무슨)\.*(아|니|나\?)출신(소속)", # 옛날 신문 질문 확장
        r"너\.*(질|무슨\.*말을)\.*알려", # 모델 능력 질문
        r"너\.*정말\.*똑똑", # 모델 칭찬/감탄
        r"(오늘|지금)\.*(기분|상태)\.*(어때|어떻|공감)", # 모델 상태 질문
        r"(농담|재밌는\.*이야기|웃긴\.*얘기)\.*알려", # 농담 요청
        r"(성심|지루)\.*해서", # 심심함 표현
        r"(영화|드라마|노래|게임)\.*추천해줘", # 취향 추천
        r"(오늘|내일)\.*(날씨|기온)\.*알려줘", # 실시간 날씨 질의
        r"(상황|연애|아빠|친구)\.*상당", # 감정 상담 요청
    ]
]

# is_science_query는 user 메시지를 합치 비교해 패턴에 해당하는지 판별한다.
def is_science_query(messages: List[Dict[str, str]]) -> bool:
    user_text = " ".join(msg.get("content", "") for msg in messages if msg.get("role") == "user")
    normalized = user_text.strip()
    if not normalized:
        return True
    for pattern in NON_SCIENCE_PATTERNS:
        if pattern.search(normalized):
            return False
    return True
```

02. Embedding 모델 변경

- * 임베딩 차원을 증가시키면 더 풍부한 의미 정보를 담을 수 있다는 가설
 - 기존 Hugging Face SBERT 계열: 768차원
 - OpenAI ‘text-embedding-3-small’: 1536차원
 - Upstage ‘solat-embedding-1-large’: 4096차원
- 다만, Elasticsearch-8.8.0의 최대 임베딩 차원이 2048차원이므로, 초과하는 고차원 모델은 랜덤 프로젝션 기반의 1536차원으로 차원 축소 후 적용.

> 임베딩 모델 교체 후, 실제로 점수 상승 확인

openai_emb	AI	0.7780	0.7803	2025.11.22 00:01
emb_upstage	AI	0.8515	0.8515	2025.11.22 23:12

경진대회 인사이드 공유

Scientific Knowledge Question Answering | 과학 지식 질의 응답 시스템 구축

03. 시도해본 점

* *sparse retrieve*와 *dense retrieve*를 가중합을 통해 *hybrid* 방식 적용.

* *dense retrieve* 모델이 베이스라인 코드에서 제공된 모델(*SBERT*)일 때, *sparse 0.7 : dense 0.3* 비율이 최적

* *CoBERT* 시도
의존성문제 이슈로 실패

* *RERANK* 기법 시도
모델에 대한 낮은 이해로 인한 시간부족 이슈로 인한 실패

경진대회 인사이트 공유

Scientific Knowledge Question Answering | 과학 지식 질의 응답 시스템 구축

04. 멘토링

Q. 임베딩 모델 변경 후 *hybrid* 방법 적용시, 기존의 비율대신 *dense-only*로 했을 때, 최고 성능이 나온 것에 대한 해석.

A. 기존의 임베딩 모델 성능이 낮은 걸로 보임

Q. 임베딩 모델 선택하는 법

A. 벤치마크를 참고(기준이 되는 데이터셋)

- 벤치마크 결과에서 성능이 우수한 모델 파악
- 수행하려는 *task*와 유사한 벤치마크에서 검증된 모델 사용

04

회고

우리 팀의 목표 달성도
느낀점 및 향후 계획

경진대회 회고

Scientific Knowledge Question Answering | 과학 지식 질의 응답 시스템 구축

Point 1

우리 팀의 처음 목표에서 어디까지 도달했는가

* 모든 과정을 다 경험

Point 2

우리 팀이 잘했던 점

* 분업이 잘 이루어짐

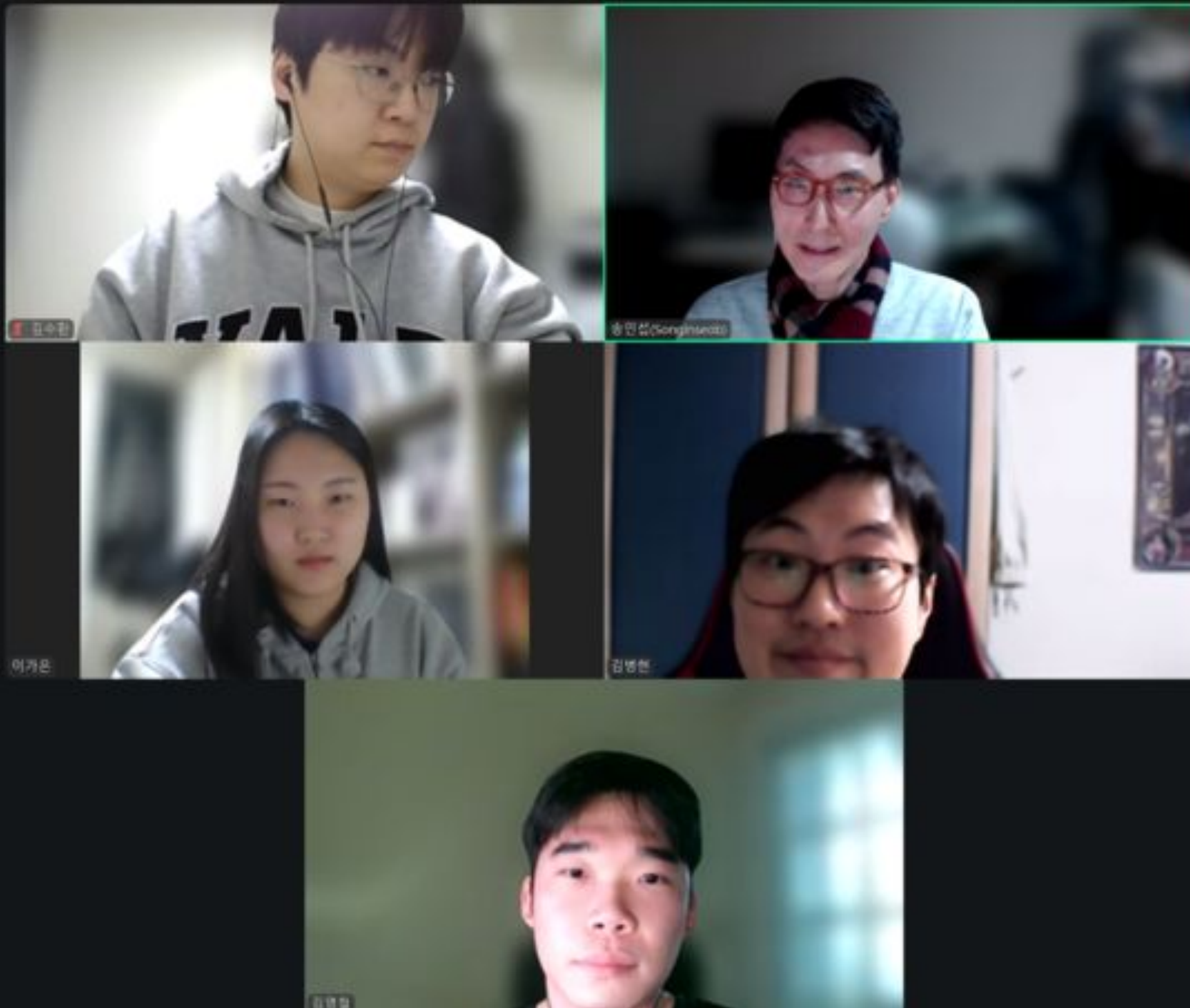
Point 3

협업하면서 아쉬웠던 점

* 다양한 접근 방식을 늦게 알게 되어 도입할 시간 부족

경진대회 진행 소감

Scientific Knowledge Question Answering | 과학 지식 질의 응답 시스템 구축



* **김명철** 직전 프로젝트에서 본 분야여서, 쉽게 생각했는데 생각보다 깊은 분야여서 공부할 것이 많았다.

* **김수환**

* **김병현** 유익한 경험이었고, 모두 행복했으면 좋겠습니다!

* **송인섭** 직전에 RAG 프로젝트를 수행하였고 이번에도 유사한 IR 경진대회를 수행하면서 LLM을 활용한 애플리케이션에 중요한 기술임을 이해하게 되었음.

* **이가은** Agent 프로젝트를 끝내고 하는 경진대회라 간단하게 생각했는데, 생각보다 다양한 내용을 알아야 해서 많이 공부하게 되었던 대회였습니다. 다양한 인사이트를 제공해준 팀원 분들께 감사합니다.

Life-Changing Education

감사합니다.
