Fast campus × upstage

Upstage AI Lab

Information Retrieval Competition Report

[IR 3조]

목차

01

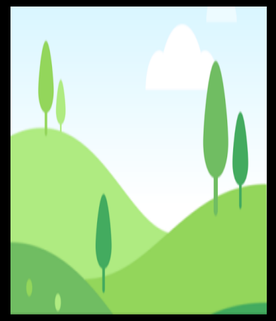
그룹 소개

Upstage AI Lab

그룹소개

김정헌

- 개선 사항에 대한 아이디어 제시
- 프롬프트 튜닝



지수영

- 프롬프트 실험
- 검색엔진 실험
- 임베딩 모델 학습

이강건

- 임베딩 모델 실험
- 프롬프트 수정

이지환

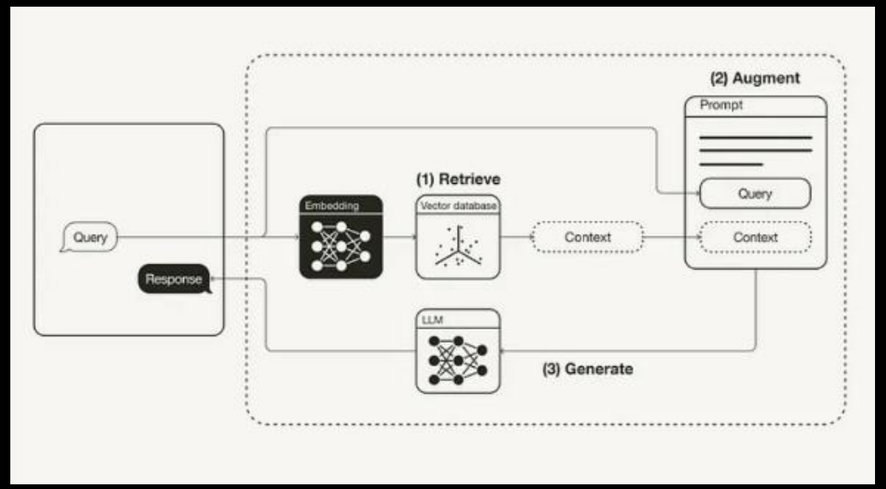
- 임베딩 모델 실험
- 역색인 모델 실험
- 프롬프트 수정

한민규

02

대회 소개

대회 소개



"Scientific Knowledge Question Answering | 과학 지식 질의 응답 시스템 구축"

- 적합한 레퍼런스 추출을 위해 검색엔진과 RAG사용
- 답변 생성을 위해 LLM사용

데이터셋

{"docid": "42508ee0-c543-4338-878e-d98c6babee66", "src": "ko_mmlu__nutrition__test", "content": "건강한 사람이 에너지 균형을 평형 상태로 유지하는 것은 중요합니다. 에너지 균형은 에너지 섭취와 에너지 소비의 수학적 등등성을 의미합니다. 일반적으로 건강한 사람은 1-2주의 기간 동안 에너지 균형을 달성합니다. 이 기간 동안에는 올바른 식단과 적절한 운동을 통해 에너지 섭취와 에너지 소비를 조절해야 합니다. 식단은 영양가 있는 식품을 포함하고, 적절한 칼로리를 섭취해야 합니다. 또한, 운동은 에너지 소비를 촉진시키고 근육을 강화시킵니다. 이렇게 에너지 균형을 유지하면 건강을 유지하고 비만이나 영양 실조와 같은 문제를 예방할 수 있습니다. 따라서 건강한 사람은 에너지 균형을 평형 상태로 유지하는 것이 중요하며, 이를 위해 1-2주의 기간 동안 식단과 운동을 조절해야 합니다."}
{"docid": "7a3e9dc2-2572-4954-82b4-1786e9e48f1f", "src": "ko_ai2_arc__ARC_Challenge__test", "content": "산꼭대기에서는 중력이 아주 약간 변합니다. 이는 무게에 영향을 미칩니다. 산꼭대기에서는 무게가 감소할 가능성이 가장 높습니다. 중력은 지구의 질량에 의해 결정되며, 산꼭대기에서는 지구의 질량과의 거리가 더 멀어지기 때문에 중력이 약간 감소합니다. 따라서, 산꼭대기에서는 무게가 더 가볍게 느껴질 수 있습니다."}

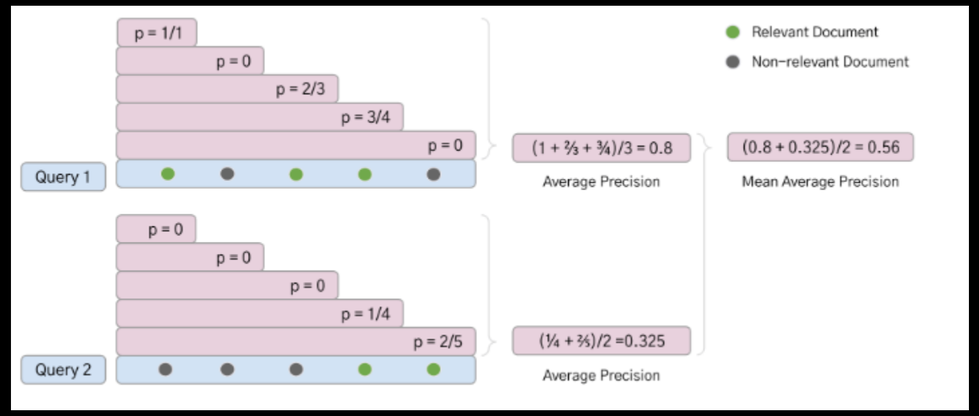
{"eval_id": 0, "msg": [{"role": "user", "content": "나무의 분류에 대해 조사해 보기 위한 방법은?"}]}
{"eval_id": 1, "msg": [{"role": "user", "content": "각 나라에서의 공교육 지출 현황에 대해 알려줘."}]}
{"eval_id": 2, "msg": [{"role": "user", "content": "기억 상실증 걸리면 너무 무섭겠다."}, {"role": "assistant", "content": "네 맞습니다."}, {"role": "user", "content": "어떤 원인 때문에 발생하는지 궁금해."}]}
{"eval_id": 3, "msg": [{"role": "user", "content": "통학 버스의 가치에 대해 말해줘."}]}
{"eval_id": 4, "msg": [{"role": "user", "content": "Dmitri Ivanovsky가 누구야?"}]}
{"eval_id": 36, "msg": [{"role": "user", "content": "니가 대답을 잘해줘서 너무 신나!"}]}

[데이터 건수]

- documents.jsonl: 4272
- eval.jsonl: 220

- doc_id : 대화 고유번호
- src: 출처
- content: 지식 정보

평가 지표




```python
def calc_map(gt, pred):
  sum_average_precision = 0
  for j in pred:
    if gt[j["eval_id"]]:
      hit_count = 0
      sum_precision = 0
      for i,docid in enumerate(j["topk"][:3]):
        if docid in gt[j["eval_id"]]:
          hit_count += 1
          sum_precision += hit_count/(i+1)
      average_precision = sum_precision / hit_count if hit_count > 0 else 0
    else:
      average_precision = 0 if j["topk"] else 1
    sum_average_precision += average_precision
  return sum_average_precision/len(pred)
```

- RAG에 대한 end-to-end 평가 대신 적합한 레퍼런스를 얼마나 잘 추출했는지에 대한 평가만 진행
- 이번 대회에서는 MAP를 약간 변형하여 RAG 평가에 적합하도록 살짝 수정한 형태의 로직을 사용
- 검색이 필요없는 ground truth 항목에 대해서는 검색 결과가 없는 경우를 1점으로 주고 그렇지 않는 경우는 0점으로 계산

03

Exploratory Data Analysis

Exploratory Data Analysis

```
Test 2
Question: [{'role': 'user', 'content': '기억 상실증 걸리면 너무 무섭겠다.'}, {'role': 'assistant', 'content': '네 맞습니다.'}, {'role': 'user', 'content': '어떤 원인 때문에 발생하는지 궁금해.'}]
```

```
Test 21
Question: [{'role': 'user', 'content': '요새 너무 힘들다.'}]
```

- 멀티턴 대화 존재
- 과학 상식과 관련이 없는 대화 존재

04

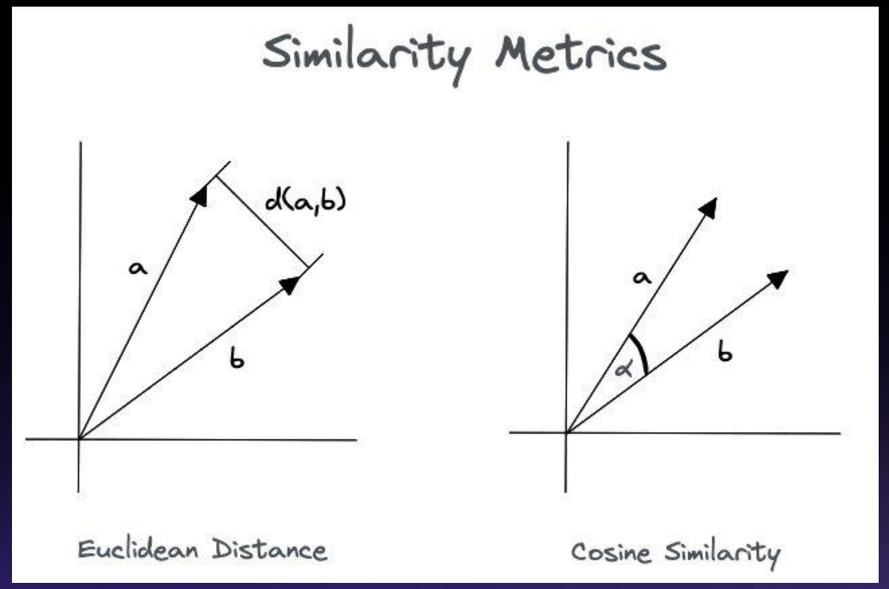
검색 엔진 및 모델

유사도

L2 norm vs Cosine similarity

L2 norm (벡터의 크기에 영향, 위치 기반의 유사도 측정에 사용)

Cosine (벡터의 방향에 영향, 텍스트 유사도 측정 또는 문서 비교 등에 사용)



$$||x||_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$$

where $x \in \mathbb{R}$ is a vector of dimension n with coordinates x_i

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

모델

SBERT 기반 모델과 RoBERTa 기반 모델 실험

BERT 기반

- snunlp/KR-SBERT-V40K-klueNLI-augSTS
- sentence-transformers/sentence-t5-large
- jhgan/ko-sbert-multitask
- jhgan/ko-sbert-nli

RoBERTa 기반

- jhgan/ko-sroberta-multitask (최종 채택)
- jhgan/ko-sroberta-base-nli

역색인

BM25 vs LM Jelinek-Mercer

- BM25 (확률적 언어 모델을 기반으로 한 랭킹 함수, 단어 수 기반으로 점수 계산)
- LM Jelinek-Mercer (언어 모델을 기반으로 한 정보 검색 방법, 쿼리와 문서 간의 확률적 유사성 계산)

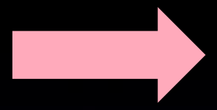
항목	BM25	LM Jelinek-Mercer
기반 모델	확률적 언어 모델 + 빈도 기반 모델	언어 모델
주요 개념	TF, IDF, 문서 길이 정규화	확률적 유사성, 스무딩
파라미터	k1, b	λ
계산 복잡성	상대적으로 낮음	상대적으로 높음
문서 길이 고려	예	아니오 (별도로 처리 필요)
장점	효율성, 문서 길이 고려	문서와 쿼리의 확률적 모델링, 스무딩 제공
단점	파라미터 튜닝 필요, 고정된 모델	계산 복잡성, 스무딩 매개변수 설정 필요

역색인과 임베딩 혼합

Sparse retrieve와 dense retrieve를 혼합하여 사용

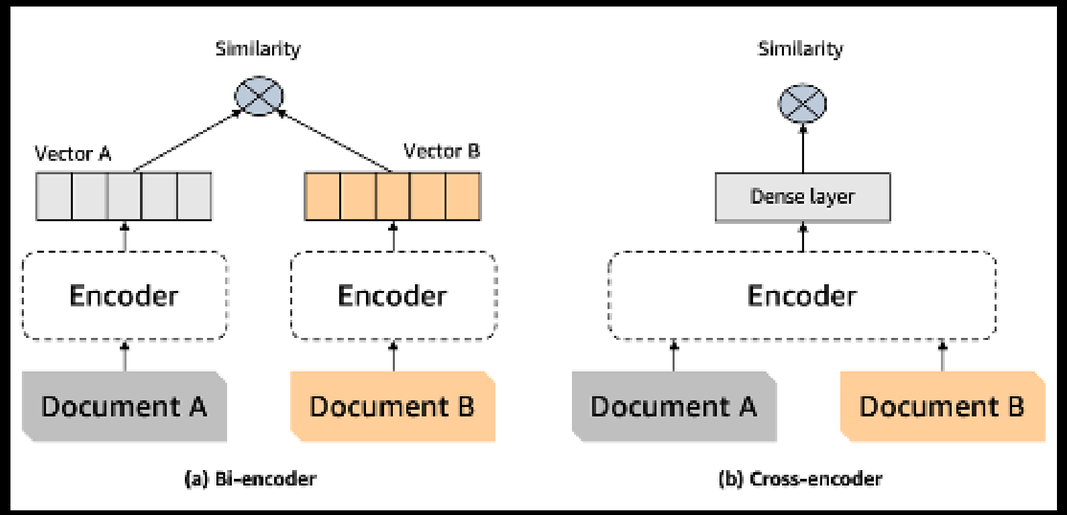
```python
# 전통적인 역색인 기법을 사용하여 단어의 빈도와 위치를 기반으로 검색
# 계산 자원이 적게 들고 빠른 검색이 가능하지만, 의미적 유사성을 잘 포착하지 못할 수 있음
def sparse_retrieve(query_str, size):
    query = {
        "match": {
            "content": {
                "query": query_str
            }
        }
    }
    return es.search(index="test", query=query, size=size, sort="_score")
```

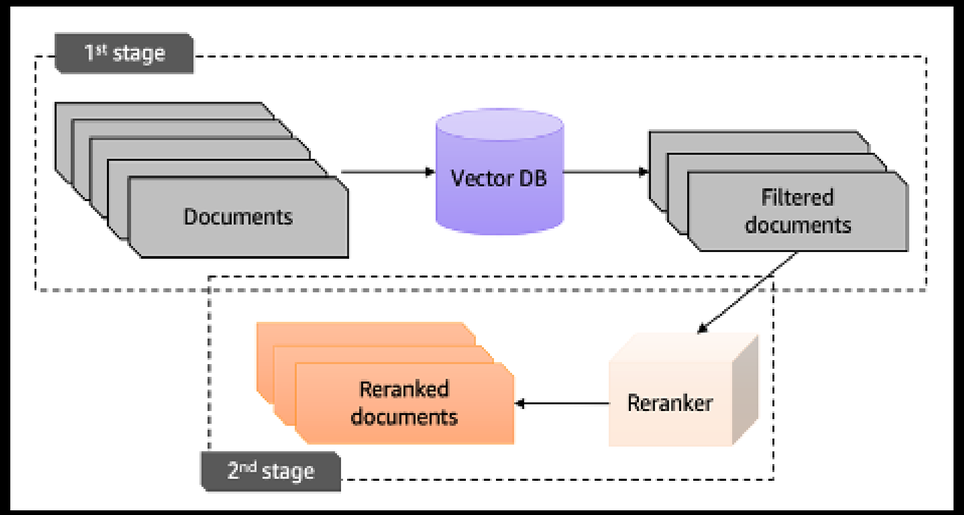
```python
# 딥러닝 기반의 임베딩 벡터를 사용하여 의미적 유사성을 포착한 검색을 수행
# 계산 자원이 많이 들지만, 높은 의미적 관련성을 제공
def dense_retrieve(query_str, size):
    # 벡터 유사도 검색에 사용할 쿼리 임베딩 가져오기
    query_embedding = get_embedding([query_str])[0]

    # kNN을 사용한 벡터 유사성 검색을 위한 매개변수 설정
    knn = {
        "field": "embeddings",
        "query_vector": query_embedding.tolist(),
        "k": size,
        "num_candidates": 100
    }

    # 지정된 인덱스에서 벡터 유사도 검색 수행
    return es.search(index="test", knn=knn)
```



```python
# Hybrid retrieve (역색인 + 벡터 유사도 혼합)
def hybrid_retrieve(query_str, size):
    query_embedding = get_embedding([query_str])[0]

    # Using elasticsearch query DSL
    # 기존 match와 knn이 별도로 나열되어 있던 것을 should를 통해 결합
    body = {
        "query": {
            "bool": {
                "should": [{
                    "match": {
                        "content": {
                            "query": query_str,
                            "boost": 0.002
                        }
                    }
                },

                {
                "knn": {
                    "field": "embeddings",
                    "query_vector": query_embedding.tolist(),
                    "k": 5,
                    "num_candidates": 50,
                    "boost": 1
                    }
                }]
            }
        },
        "size": size
    }

    return es.search(index="test", body=body)
```

Reranker





- Reranker는 질문과 문서 사이의 유사도를 측정하는 것을 목표로 함
- Cosine distance가 유사한 문서들을 뽑아, 이 문서들과 쿼리간의 관계를 Reranker로 파악하여, Rerank score를 따로 계산하여 점수가 높은 순서대로 문서를 정렬

임베딩 모델 학습

- 과학 상식 문서를 기반으로 gpt3.5로 질문을 생성하여 학습에 사용
- optimizer = Adam
- criterion = ContrastiveLoss
- monologg/koelectra-base-v3-discriminator: 0.7152 => 0.7303
- jhgan/ko-sroberta-multitask: 0.7152 => 0.6985

```
prompt = f"""
다음 내용을 바탕으로 답변할 수 있는 질문 3개를 생성해주세요.
반드시 한국어로 질문을 작성해야 합니다.
각 질문은 완전한 문장이어야 합니다.

{content}

질문들은 다음과 같은 JSON 형식으로 반환해주세요:
{{
    "question1": "첫 번째 한글 질문?",
    "question2": "두 번째 한글 질문?",
    "question3": "세 번째 한글 질문?"
}}
"""
```

```
실제 파일 처리 시작
총 처리할 항목 수: 4272
처리 중:   0%|          | 1/4272 [00:01<2:12:47,  1.87s/item]Raw API Response: {
    "question1": "에너지 균형을 유지하는 것이 왜 중요한가요?",
    "question2": "올바른 식단과 적절한 운동을 통해 에너지 균형을 어떻게 달성할 수 있나요?",
    "question3": "에너지 균형을 유지하면 어떤 문제를 예방할 수 있나요?"
}

항목 1에 대해 생성된 질문:
질문 1: 에너지 균형을 유지하는 것이 왜 중요한가요?
질문 2: 올바른 식단과 적절한 운동을 통해 에너지 균형을 어떻게 달성할 수 있나요?
질문 3: 에너지 균형을 유지하면 어떤 문제를 예방할 수 있나요?
```

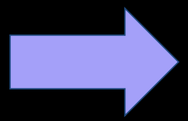
05

Prompt Engineering

Prompt Engineering

프롬프트 변경안 (1)

```
# RAG 구현에 필요한 질의 분석 및 검색 이외의 일반 질의 대응을 위한 LLM 프롬프트
persona_function_calling = """
## Role: 과학 상식 전문가

## Instruction
- 사용자가 대화를 통해 과학 지식에 관한 주제로 질문하면 search api를 호출할 수 있어야 한다.
- 과학 상식과 관련되지 않은 나머지 대화 메시지에는 적절한 대답을 생성한다.
"""
```



```
'''
    질의 분석 변경점
        - 사용자가 지식에 관한 질문을 하는 경우에는 반드시 search 함수를 호출한다.
            1. 과학 지식으로 좁힐 경우, 과학 지식임에도 불구하고 처리가 안되는 경우 존재
                차라리 한정짓지 않고 다양한 주제의 지식 관련 질문을 처리할 수 있도록 변경
            2. 지식 관련 질문임에도 불구하고 함수가 호출되지 않는 경우 존재
                '반드시' 라는 단어를 포함시켜 호출되지 않는 경우를 처리

        - 나머지 메시지에는 함수 호출을 하지 않고 적절한 대답을 생성한다.
            1. 위의 경우를 제외한 나머지 메세지에 대한 처리
                함수를 호출하지 않겠다라는 내용을 포함시켜 함수가 호출되는 경우를 처리
'''

persona_function_calling = """
## Role: 과학 상식 전문가

## Instruction
- 사용자가 지식에 관한 질문을 하는 경우에는 반드시 search 함수를 호출한다.
- 나머지 메시지에는 함수 호출을 하지 않고 적절한 대답을 생성한다.
"""
```

Prompt Engineering

프롬프트 변경안 (2)

기존 프롬프트

```
# RAG 구현에 필요한 Question Answering을 위한 LLM  프롬프트
persona_qa = """
## Role: 과학 상식 전문가

## Instructions
- 사용자의 이전 메시지 정보 및 주어진 Reference 정보를 활용하여 간결하게 답변을 생성한다.
- 주어진 검색 결과 정보로 대답할 수 없는 경우는 정보가 부족해서 답을 할 수 없다고 대답한다.
- 한국어로 답변을 생성한다.
"""

# RAG 구현에 필요한 질의 분석 및 검색 이외의 일반 질의 대응을 위한 LLM 프롬프트
persona_function_calling = """
## Role: 과학 상식 전문가

## Instruction
- 사용자가 대화를 통해 과학 지식에 관한 주제로 질문하면 search api를 호출할 수 있어야 한다.
- 과학 상식과 관련되지 않은 나머지 대화 메시지에는 적절한 대답을 생성한다.
"""
```

0.7045	0.7106

프롬프트 변경

```
# RAG 구현에 필요한 Question Answering을 위한 LLM 프롬프트
persona_qa = """
## Role: 과학 상식 전문가

## Instructions
- 사용자의 이전 메시지 정보 및 주어진 Reference 정보를 활용하여 300자 이내로 답변을 생성한다.
- Reference 정보의 신뢰성을 평가하고, 가장 신뢰할 수 있는 정보를 우선적으로 사용한다.
- 주어진 검색 결과 정보로 대답할 수 없는 경우는 정보가 부족해서 답을 할 수 없다고 대답한다.
- 불확실한 정보가 있다면 명시적으로 언급한다.
- 추가 정보가 필요한 경우, 사용자에게 구체적인 추가 질문을 제안한다.
- 한국어로 답변을 생성한다.

"""

# RAG 구현에 필요한 질의 분석 및 검색 이외의 일반 질의 대응을 위한 LLM 프롬프트
persona_function_calling = """
## Role: 과학 상식 전문가

## Instructions
- 사용자가 대화를 통해 물리, 화학, 생물, 지구과학, 우주과학 등의 과학 지식에 관한 주제로 질문하면 search api를 호출한다.
- search api 호출 기준:
  1) 질문이 구체적인 과학적 사실이나 개념을 요구할 때
  2) 복잡한 과학 현상에 대한 설명이 필요할 때
- 과학 상식과 관련되지 않은 대화 메시지에는:
  1) 예의 바르게 주제를 과학 관련 주제로 전환하도록 유도한다.
  2) 만약 사용자가 계속 다른 주제를 이야기하면, 정중하게 과학 상식에 대해서만 답변할 수 있다고 설명한다.
- 이전 대화 내용을 고려하여 일관성 있는 답변을 생성한다.
- 모든 답변은 한국어로 제공한다.
"""
```

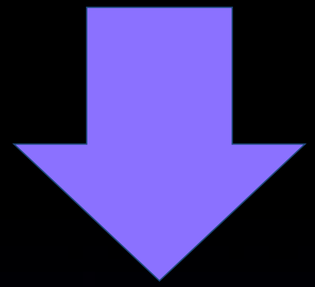
0.7152	0.7182

06

결과 및 회고

Final Leaderboard Score

IR 3조	수영		0.9061	0.9121



IR 3조	수영		0.8644	0.8682

그룹 스터디 인사이트 공유

1

인사이트 1

dense retrieve와 sparse retrieve를 같이 써 보니 성능이 향상되었다.

2

인사이트 2

유사도 계산에 따라 최종 결과에서 생각보다 차이가 있었다.

3

인사이트 3

프롬프트에 따라서도 성능이 차이가 났다.

Upstage AI Lab

대회 소감

김정헌	이 분야 자체가 유망할 것 같다.
이강건	많은 실험을 하지 못해 아쉽지만 IR을 경험해볼 수 있어서 좋았습니다.
이지환	개인 사정으로 3~4일 정도밖에 해보지 못해서 아쉽다.
지수영	RAG를 경험해 볼 수 있어서 좋았습니다.
한민규	

감사합니다. 그리고
다들 고생 많으셨습니다!