

Upstage AI Lab

IR 경진대회

Scientific Knowledge Question Answering

24.12.20

www.fastcampus.co.kr

Copyright © FAST CAMPUS Corp. All Rights Reserved. 무단전재 및 재배포 금지

목차

- 01. 오프닝
- 02. 그룹 스터디 주제 소개
- 03. 실습 및 예시
- 04. TroubleShooting
- 05. 그룹 스터디 회고

01

오프닝

02

그룹 스터디 주제 소개

프로젝트 소개

Scientific Knowledge Question Answering

주제

과학 지식 질의 응답 시스템 구축

목표

질문과 이전 대화 히스토리를 보고 참고할 문서를 검색엔진에서
추출 후 이를 활용하여 질문에 적합한 대답을 생성하는 태스크


개요

소개 및 배경 설명
과학 지식 질의 응답 시스템 구축

기간
2024. 12. 16. ~ 2024. 12. 20.

1. Idea 공유

- 디테일한 아이디어 실시간 기록 및 공유
- 팀원 간 빠른 피드백 및 개선사항 논의
- 효율적으로 방향성 조정 및 개선점 도출



Idea 공유

▼ 결과 분석 #1

[1] :

- eval_id : 213
- standalone_query : "각 나라에서의 공교육 지출 현황"
- topk : [...]
- answer : "죄송합니다, 제가 현재 각 나라의 공교육 지출 현황에 대한 정보를 가지고 있지 않습니다. 더 많은 정보를 제공해 주시면 도와드릴 수 있을 것 같습니다."
- references : [...]

[8] :

- eval_id : 81
- standalone_query : "대학 버스의 가치"
- topk :
 - [0] : "bd91bda8-351e-4683-bb1a-8254f93e2376"
 - [1] : "5d48fa4e-1999-45dc-b768-b15b30a87cc5"

JuYeong Jin Dec 16 (edited)

가중치 줄 수 있더라고요!

SPARSE_WEIGHT = 0.6

DENSE_WEIGHT = 0.4

이런 식으로요.

이 부분도 조정해가면서 실험해보면 좋을 것 같아요 ^^

Ji Hwan Kim 2d

그럼 RAG를 할때 hybrid search를 적용하면 저렇게 되는건가요??

↳ 댓글 소통

2. Experiments 공유

- submission 결과 즉시 업데이트
- 사용한 모델 등 상세 스펙 기록
- 중복 실험 방지로 자원 낭비 최소화

Experiments									
IR Competition Board +									
Status	Person	Created Time	Aa Hypothesis	# mAP (Io...)	# mAP (L...	# MRR (L...	Retriever	Search	
Done	Juha Lee	December 17, 2024 8:24 PM	sparse → dense로 변경		0.8538	0.8561	Elasticsearch	Dense	
Done	JuYeong Jin	December 18, 2024 8:48 PM	categories + 검색 기능 개별 추가		0.8167	0.8182	Elasticsearch	catego	
Done	east_star	December 18, 2024 12:43 PM	test4/임베딩 변형, 모델 변형 / HNSW		0.8129	0.8152	Elasticsearch	Dense	
Done	Ji Hwan Kim	December 17, 2024 9:45 AM	prompt 개선		0.6061	0.6121	Elasticsearch	Sparse	
Done	Ji Hwan Kim	December 16, 2024 3:51 PM	openAI 임베딩 모델을 사용 #1		0.5985	0.603	Elasticsearch	Sparse	
To Do	성범 한	December 17, 2024 11:32 PM	dense_FAISS, colbert 임베딩		0.55		FAISS	Dense	

CV 구축

: 자체 검증 프로세스 구현 (Team Cross Validation 구축)

검색 실행

- 팀 SOTA 모델 기반 topk 결과의 정확한 순서 매칭
- 문서 ID 기반 비교
- 기준점: 70% 이상 일치율

```
# 두 모델의 검색 결과 비교
for idx, (ref_vals, comp_vals) in enumerate(zip(reference_topk, comp_topk)):
    set_ref = set(ref_vals)
    set_comp = set(comp_vals)

    # Intersection and differences
    intersection = set_ref & set_comp
    unique_to_ref = set_ref - set_comp
    unique_to_comp = set_comp - set_ref

    # 일치율 계산
    match_percentage = round(len(intersection) / len(set_ref.union(set_comp)) * 100, 2)
```

↳ 코드 일부분

예시)

=== 기준 파일 : SOTA(Solar) 모델 ===

[비교 파일 : OpenAI GPT 3.5 모델]

- 총 비교 행 수: 220
- 완전 일치 행 수: 190
- 평균 일치율: 86.36%

결과가 저장됨 : /home/IR/data/reference_comparison_results.csv

일치율 70% 이상 제출

03

실습 및 예시

실험으로 직접 확인하기

과학이 아닌 것 같은 아이템

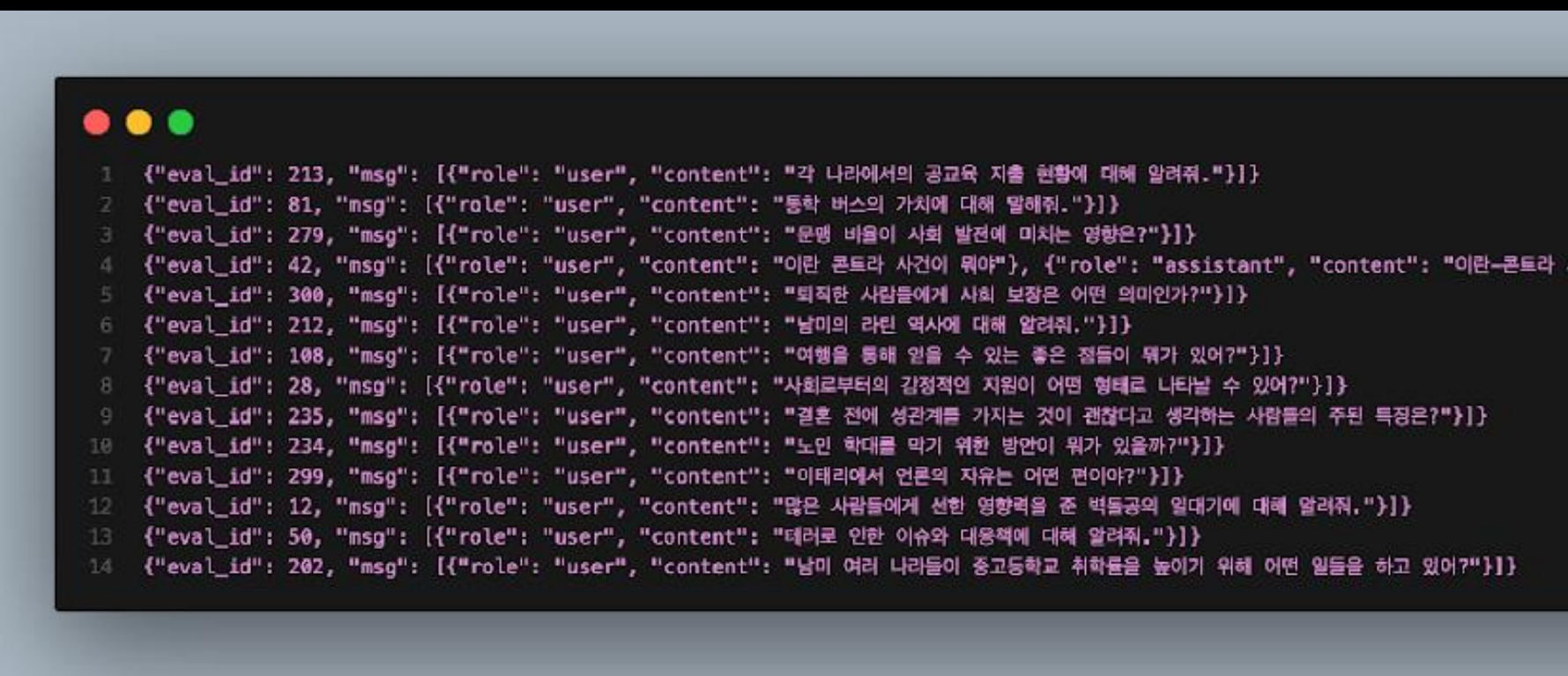
제출 결과 중 해당하는 것의 topk 제거

Upstage AI Lab

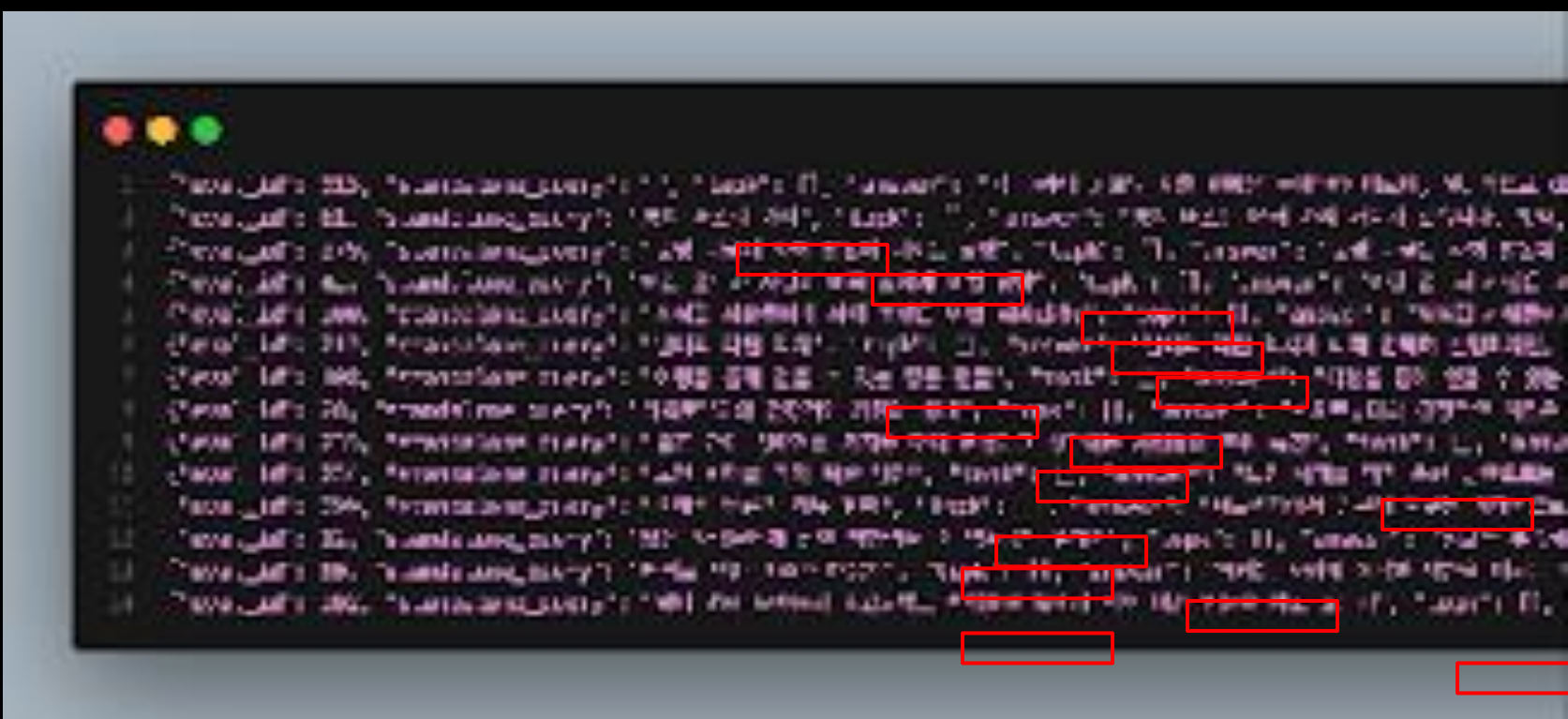
실습 및 예시

과학의 정의?

실험으로 직접 확인하기



과학이 아닌 것 같은 아이템



제출 결과 중 해당하는 것의 topk 제거

- 노인 학대... 관련 query: 사회 과학으로 분류 될수 도 있음 -> 과학 상식은 아니나 큰 category 면에서 과학으로 분류된다면 topk 반환해야함
- 과일샐러드에 대해 알려달라 -> 절대 과학상식이 아님 :

실험으로 직접 확인하기 (결과)

0.7333	0.7394	2024.12.17 08:56
-	-	

원본 결과

0.6848	0.6909	2024.12.17 16:02
-	-	

삭제 후 결과

Upstage AI Lab

실습 및 예시

Prompt Engineering

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.
"""
```

- `search api`를 호출할 수 있어야 한다>> 일부 `topk`, `standalone query`를 불러오지 못하는 상황 발생>> 무조건적으로 호출로 변경
- 적절한 응답>> 과연 ‘적절한’의 기준이 무엇인가>> 과학 상식과 관련 없을시에 검색 제외 (정보를 찾을 수 없음 및 출력 x)>> MAP가 내려감>> 기준을 일상대화, 과학상식등 좀 더 세분화해서 프롬프트 수정
- 4o mini 사용

0.7015	0.7076
0.6970	0.7015

원본 prompt
MAP

```
persona_function_calling = ""
## Role: 과학, 사회, 인문, 공학 등 전반적으로 모든 분야의 지식과 상식을 갖춘 전문가

## Instruction
- 카테고리별로 질문(fact), 조연 (reference 기반 혹은 질문자의 마지막 대화를 기준으로 ), 일상대화를 나눈 뒤
  쿼리: 메릴랜드의 대표적인 천연 자원은? // 질문
  연구실에서 무엇인지 파악이 안된 가루를 저울로 옮기는 좋은 방법은? // 조연
  요새 너무 힘들다 // 일상 대화

- 사용자가 질문한 내용을 검색하기 위해 최종 검색 질의(standalone query)를 생성합니다.
- standalone query는 사용자 질문의 핵심만을 포함하며, 명확하고 간결해야 합니다.
- 사용자와 여러번 대화를 했을 경우, 대화 맥락을 바탕으로 마지막으로 대화를 나눈 의미를 추론합니다.
- 사용자가 각종 과학 지식을 주제로 질문을 하거나 설명을 요청하면, 시스템은 검색 API를 무조건 호출합니다.
- 과학 지식과 관련되지 않는 다른 일상적인 대화 메시지에 대해서는 적절한 응답을 생성합니다.
- 대화 내용이 과학 지식에 대한 주제가 아니면 검색 api를 호출하지 않습니다.
- 영어로 질문할 때 (예시: Dmitri)는 한국말로 번역하여 인식하여도 됩니다 (예시: 드미트리)
"""
```

- 어차피 일상대화20개를 제외한 대부분의 과학(사회과학 등 포함)에 답변이 필요하다면 단순 과학 상식 전문가보다 전반적으로 모든 분야의 지식의 전문가로 수정

0.7333	0.7394	2024.12.17 08:56
0.6924	0.6955	

수정 후 MAP

Upstage AI Lab 실습 및 예시

전략 수정

Embedding Model 중점

- 어차피 Prompt engineering은 llm으로 넘겨서 진행하기 때문에 기존 baseline에서 dramatic한 변화가 더 이상 없을거라고 판단.
- 시간이 촉박했기에 우선적으로 임베딩 모델, chat model (3.5turbo, 4o mini 등) KR-Sbert보다 나은 성능을 보이는 것을 찾는것, 그리고 3.5turbo보다 최신 버전모델 테스트 등을 중점으로 실험해보기 시작
- Hugging Face에서 bge-m3 model을 발견해서 이모델 중 한국어에 특화되어있으면서 가장 top-k1 부터 topk-3의 f1,precision, recall score가 가장 높은 dragonkue/bge-m3-ko로 실험.

#	mAP (Le...	#	MRR (L...	Retriever	Search ...	Embedding Model	Chat Mo...
	0.8538		0.8561	Elasticsearch	Dense	dragonkue/bge-m3-ko	gpt 3.5turbo
	0.8167		0.8182	Elasticsearch	categories	dragonkue/bge-m3-ko	solar-pro
	0.8129		0.8152	Elasticsearch	Dense	dragonkue/bge-m3-ko	gpt 3.5turbo
	0.6061		0.6121	Elasticsearch	Sparse	openai/text-embedding-3...	gpt-4o-mini
	0.5985		0.603	Elasticsearch	Sparse	openai/text-embedding-3...	gpt-4o-mini
	0.55		0.5515	Elasticsearch	Sparse	snunlp/KR-SBERT-V40K-kl...	gpt-4o-mini
	0.5485		0.5561				
	0.5197		0.5258	Elasticsearch	Sparse	snunlp/KR-SBERT-V40K-kl...	gpt-4o-mini
	0.428		0.4303	Elasticsearch	Dense	snunlp/KR-SBERT-V40K-kl...	gpt-4-turbo
	0.347		0.3485	hybrid	Hybrid	snunlp/KR-SBERT-V40K-kl...	
	0.3341		0.3333	hybrid	Hybrid		
				FAISS	Dense	jhgan/ko-sroberta-multit...	

- map와 mrr은 upstage embedding 이 더 높아서 시도해볼만 했지만 일 단 대회기간이 4 일밖에 안되므로 임베딩 모델을 하나로 fix 하고 다 른 실험을 해보기로 결정

• Top-k 3

Model name	F1	Recall	Precision	mAP	mRR	NDCG
paraphrase-multilingual-mpnet-base-v2	0.2368	0.4737	0.1579	0.2032	0.2032	0.2712
KoSimCSE-roberta	0.3026	0.6053	0.2018	0.2661	0.2661	0.3515
Cohere embed-multilingual-v3.0	0.2851	0.5702	0.1901	0.2515	0.2515	0.3321
openai ada 002	0.3553	0.7105	0.2368	0.3202	0.3202	0.4186
multilingual-e5-large-instruct	0.3333	0.6667	0.2222	0.2909	0.2909	0.3856
Upstage Embedding	0.4211	0.8421	0.2807	0.3509	0.3509	0.4743
paraphrase-multilingual-MiniLM-L12-v2	0.2061	0.4123	0.1374	0.1740	0.1740	0.2340
openai_embed_3_small	0.3640	0.7281	0.2427	0.3026	0.3026	0.4097
ko-sroberta-multitask	0.2939	0.5877	0.1959	0.2500	0.2500	0.3351
openai_embed_3_large	0.3947	0.7895	0.2632	0.3348	0.3348	0.4491
KU-HIAI-ONTHEIT-large-v1	0.4386	0.8772	0.2924	0.3421	0.3421	0.4766
KU-HIAI-ONTHEIT-large-v1.1	0.4430	0.8860	0.2953	0.3406	0.3406	0.4778
kf-deberta-multitask	0.3158	0.6316	0.2105	0.2792	0.2792	0.3679
gte-multilingual-base	0.4035	0.8070	0.2690	0.3450	0.3450	0.4614
KoE5	0.4254	0.8509	0.2836	0.3173	0.3173	0.4514
BGE-m3	0.4254	0.8508	0.2836	0.3421	0.3421	0.4701
bge-m3-korean	0.3684	0.7368	0.2456	0.3143	0.3143	0.4207
BGE-m3-ko	0.4517	0.9035	0.3011	0.3494	0.3494	0.4886

- 출처 : <https://huggingface.co/dragonkue/BGE-m3-ko>

BGE - m3 - ko model

- *baseline code는 sparse retrieval 이 기본으로 되어 있었기 때문에 dense 에 더 특화된 bge - m3 - ko model 에 dense retrieval 적용만으로 MAP가 0.12 정도 성능 향상*

0.7333	0.7394	2024.12.17 08:56
-	-	

prompt engineering 적용> Prior Highest

0.8538	0.8561	2024.12.17 20:25
0.8742	0.8773	

embedding model change / dense retrieval / chat model :3.5 turbo

Upstage AI Lab

실습 및 예시

: LLM 모델 test

```
1 {"eval_id": 78, "standalone_query": "나무 분류 조사 방법", "topk": ["c63b9e3a-716f-423a-9c9b-0bcaa1b9f35d", "191c4b9f-6feb-49c
2 {"eval_id": 213, "standalone_query": "각 나라에서의 공교육 지출 현황", "topk": ["88439180-e442-4ecf-a485-88c7de0296e9", "9f9981ff
3 {"eval_id": 107, "standalone_query": "기억 상실증 원인", "topk": ["df495f22-6315-42a8-9553-b43ab707b683", "b66b9bc2-58c4-4677
4 {"eval_id": 81, "standalone_query": "통학 버스의 가치", "topk": ["bd91bda8-351e-4683-bb1a-8254f93e2376", "7404b07e-7b4d-4668-
5 {"eval_id": 280, "standalone_query": "드미트리 이바노프스키", "topk": ["61e22317-2240-4aaf-9498-917e23f4466e", "e007e8db-08b9-4
6 {"eval_id": 10, "standalone_query": "피임약의 효과와 부작용", "topk": ["99a07643-8479-4d34-9de8-68627854f458", "f312f1fe-28fe-4
7 {"eval_id": 100, "standalone_query": "헬륨이 다른 원소들과 반응을 잘 안하는 이유", "topk": ["49b2beab-b08d-479b-bc98-309c29911e03",
8 {"eval_id": 279, "standalone_query": "문맹 비율이 사회 발전에 미치는 영향", "topk": ["de1ab247-9d48-48f7-8499-31606f53c108", "0f0c
9 {"eval_id": 42, "standalone_query": "이란-콘트라 사건 미국 정치 영향", "topk": ["4b49f3a2-32c9-4b2e-89c4-4719f98e7a74", "111ad5f4
10 {"eval_id": 308, "standalone_query": "자기장의 세기를 표현하는 방식", "topk": ["cedb5d80-b620-465a-89b2-3e4ada64eeb2", "63a3f2d9-
11 {"eval_id": 205, "standalone_query": "피를 맑게 하고 몸 속의 노폐물을 없애는 역할을 하는 기관", "topk": ["2a669d8e-5617-443c-9c4a-18c1
12 {"eval_id": 289, "standalone_query": "글리코겐 분해 인체 필요성", "topk": ["421aac6b-49ce-4697-a68f-850152f323d7", "0a3c5a53-d6
13 {"eval_id": 268, "standalone_query": "빛방울이 점점 커지게 되는 요인", "topk": ["885b7d3a-d152-4d8c-a40d-e197a6aa9bd5", "0e5f4ec
14 {"eval_id": 18, "standalone_query": "기체의 부피나 형태가 일정하지 않은 이유", "topk": ["63846d07-8443-4bf8-8cd9-bc6cc7826555", "b
15 {"eval_id": 9, "standalone_query": "식물이 빛을 에너지로 변환하는 과정", "topk": ["2d57c6de-aaa6-4461-9747-8f48afcd499e", "4534c0e
16 {"eval_id": 101, "standalone_query": "직류와 교류 전류의 차이", "topk": ["144f5e5e-8069-425f-80b3-6388195ba4ee", "07ba99c0-c36a
17 {"eval_id": 236, "standalone_query": "기름과 물이 섞일 수 있는지", "topk": ["6acde723-5e70-4789-9fa2-dbee23d5a250", "e0cc986d-d
18 {"eval_id": 59, "standalone_query": "인간 DNA 결합 과정", "topk": ["a19f30a9-eb0a-4985-9ecd-0cda8e54e549", "ac826442-684c-4e
19 {"eval_id": 25, "standalone_query": "금성에서 달의 관측", "topk": ["35c5dcc7-4720-4318-901e-770105ae63fd", "59a8259f-4a39-4ab
20 {"eval_id": 5, "standalone_query": "차량 연비 개선 긍정적 효과", "topk": ["abf99ff1-d6bf-4020-b752-da7cb8611915", "4b2de0fa-6a8
```

```
1 {"eval_id": 78, "standalone_query": "나무의 분류 방법에 대해 조사해 보기 위한 방법", "topk": ["c63b9e3a-716f-423a-9c9b-0bcaa1b9f35d", "191c4b9f-6feb-49dd-
2 {"eval_id": 213, "standalone_query": "각 나라에서의 공교육 지출 현황", "topk": ["88439180-e442-4ecf-a485-88c7de0296e9", "9f9981ff
3 {"eval_id": 107, "standalone_query": "기억 상실증 원인", "topk": ["df495f22-6315-42a8-9553-b43ab707b683", "b66b9bc2-58c4-4677
4 {"eval_id": 81, "standalone_query": "", "topk": [], "answer": "통학 버스는 여러 가지 가치가 있습니다. \n\n1. **안전성**: 통학 버스는 학
5 {"eval_id": 280, "standalone_query": "Dmitri Ivanovsky", "topk": ["0b5888eb-e3d9-44ed-9c9c-1b88101dd694", "de246588-07fe-
6 {"eval_id": 10, "standalone_query": "", "topk": [], "answer": "피임을 위한 약물 방법은 매우 효과적일 수 있습니다. 일반적으로 사용되는 피임약
7 {"eval_id": 100, "standalone_query": "헬륨이 다른 원소들과 반응을 잘 안하는 이유", "topk": ["49b2beab-b08d-479b-bc98-309c29911e03",
8 {"eval_id": 279, "standalone_query": "문맹 비율이 사회 발전에 미치는 영향", "topk": ["de1ab247-9d48-48f7-8499-31606f53c108", "0f0c
9 {"eval_id": 42, "standalone_query": "", "topk": [], "answer": "이란-콘트라 사건은 미국 정치에 여러 가지 중요한 영향을 미쳤습니다:\n\n1. *
10 {"eval_id": 308, "standalone_query": "자기장 세기를 표현하는 방식", "topk": ["cedb5d80-b620-465a-89b2-3e4ada64eeb2", "59b15893-e
11 {"eval_id": 205, "standalone_query": "", "topk": [], "answer": "피를 맑게 하고 몸 속의 노폐물을 없애는 역할을 하는 기관은 \신장\입니다.
12 {"eval_id": 289, "standalone_query": "", "topk": [], "answer": "글리코겐의 분해는 인체에서 에너지를 공급하는 중요한 과정입니다. 글리코겐은 간
13 {"eval_id": 268, "standalone_query": "빛방울이 점점 커지게 되는 요인", "topk": ["885b7d3a-d152-4d8c-a40d-e197a6aa9bd5", "0e5f4ec
14 {"eval_id": 18, "standalone_query": "", "topk": [], "answer": "기체의 부피나 형태가 일정하지 않은 이유는 기체 분자들이 서로 간의 거리가 멀고
15 {"eval_id": 9, "standalone_query": "식물이 빛을 에너지로 변환하는 과정", "topk": ["2d57c6de-aaa6-4461-9747-8f48afcd499e", "4534c0e
16 {"eval_id": 101, "standalone_query": "직류와 교류 전류의 차이", "topk": ["144f5e5e-8069-425f-80b3-6388195ba4ee", "07ba99c0-c36a
17 {"eval_id": 236, "standalone_query": "", "topk": [], "answer": "기름과 물은 서로 섞이지 않는 성질을 가지고 있습니다. 이는 기름이 비극성 물
18 {"eval_id": 59, "standalone_query": "인간 DNA 결합 과정 2세 생산", "topk": ["c9c30246-4512-4fda-96bf-0bc34d878cf1", "242db65d-
19 {"eval_id": 25, "standalone_query": "금성에서 달이 어떻게 보일까?", "topk": ["35c5dcc7-4720-4318-901e-770105ae63fd", "59a8259f-4
20 {"eval_id": 5, "standalone_query": "", "topk": [], "answer": "차량의 연비가 좋아질 때 나타나는 긍정적인 효과는 여러 가지가 있습니다:\n\n1.
```

4-turbo

topK를 잘 불러오지만 속도도 느리고 비용도 많이 들어감

topK를 잘 불러오고 속도와 비용 모두 가장 최고의 선택

3.5-turbo

```
1 {"eval_id": 78, "standalone_query": "나무의 분류에 대해 조사해 보기 위한 방법", "topk": ["c63b9e3a-716f-423a-9c9b-0bcaa1b9f35d", "191c4b9f-6feb-49dd-
2 {"eval_id": 213, "standalone_query": "각 나라에서의 공교육 지출 현황", "topk": ["88439180-e442-4ecf-a485-88c7de0296e9", "9f9981ff
3 {"eval_id": 107, "standalone_query": "기억 상실증 원인", "topk": ["df495f22-6315-42a8-9553-b43ab707b683", "b66b9bc2-58c4-4677
4 {"eval_id": 81, "standalone_query": "통학 버스의 가치", "topk": ["bd91bda8-351e-4683-bb1a-8254f93e2376", "7404b07e-7b4d-4668-
5 {"eval_id": 280, "standalone_query": "Dmitri Ivanovsky", "topk": ["0b5888eb-e3d9-44ed-9c9c-1b88101dd694", "de246588-07fe-
6 {"eval_id": 10, "standalone_query": "피임을 하기 위한 약으로 처리하는 방법", "topk": ["99a07643-8479-4d34-9de8-68627854f458", "d89
7 {"eval_id": 100, "standalone_query": "헬륨이 다른 원소들과 반응을 잘 안하는 이유", "topk": ["49b2beab-b08d-479b-bc98-309c29911e03",
8 {"eval_id": 279, "standalone_query": "문맹 비율이 사회 발전에 미치는 영향", "topk": ["de1ab247-9d48-48f7-8499-31606f53c108", "0f0c
9 {"eval_id": 42, "standalone_query": "이란-콘트라 사건이 미국 정치에 미친 영향", "topk": ["4b49f3a2-32c9-4b2e-89c4-4719f98e7a74", "3
10 {"eval_id": 308, "standalone_query": "자기장이 얼마나 센지 표현하는 방식", "topk": ["59b15893-ea8a-44e3-b33f-04c3ce9b2e10", "e4091
11 {"eval_id": 205, "standalone_query": "피를 맑게 하고 몸 속의 노폐물을 없애는 기관", "topk": ["2a669d8e-5617-443c-9c4a-18c187157569",
12 {"eval_id": 289, "standalone_query": "글리코겐의 분해는 인체에서 왜 필요한가?", "topk": ["421aac6b-49ce-4697-a68f-850152f323d7", "5
13 {"eval_id": 268, "standalone_query": "빛방울이 점점 커지는 요인", "topk": ["885b7d3a-d152-4d8c-a40d-e197a6aa9bd5", "2826db1e-83
14 {"eval_id": 18, "standalone_query": "기체의 부피나 형태가 왜 일정하지 않을까?", "topk": ["63846d07-8443-4bf8-8cd9-bc6cc7826555", "b
15 {"eval_id": 9, "standalone_query": "식물이 빛을 에너지로 변환하는 과정", "topk": ["2d57c6de-aaa6-4461-9747-8f48afcd499e", "4534c0e
16 {"eval_id": 101, "standalone_query": "직류와 교류 전류의 차이", "topk": ["144f5e5e-8069-425f-80b3-6388195ba4ee", "07ba99c0-c36a
17 {"eval_id": 236, "standalone_query": "기름과 물이 섞여 있을 수 있나?", "topk": ["6acde723-5e70-4789-9fa2-dbee23d5a250", "e0cc986
18 {"eval_id": 59, "standalone_query": "인간이 2세를 생산할 때 DNA의 결합 과정", "topk": ["c9c30246-4512-4fda-96bf-0bc34d878cf1", "8
19 {"eval_id": 25, "standalone_query": "금성에서 달이 어떻게 보일까?", "topk": ["35c5dcc7-4720-4318-901e-770105ae63fd", "59a8259f-4
20 {"eval_id": 5, "standalone_query": "차량의 연비가 좋아질때 나타나는 긍정적인 효과는?", "topk": ["abf99ff1-d6bf-4020-b752-da7cb8611915
```

3.5-turbo(baseline 모델), 4o-mini, 4-turbo 모델 3개를 변인 통제후 실험 함.
팀 자체 CV 데이터 셋으로 평가 결과 3.5-turbo model이 topK도 잘 불러오고 결과가 가장 좋았

은

4o-mini

비용은 가장 적게 들어가지만 topK를 거의 불러오지 못하는 수준

Embedding model 선정 후 중점적으로 시도한 MAP 개선 시도 방법

1. 쿼리 라우팅 (Query Routing)
2. Chunking
3. Retrieval 개선 (보통 단순 sparse / dense보다 Hybrid의 성능이 좋음)
4. Reranking

실습 및 예시

쿼리 확장과 MMR을 활용한 문서 검색 개선

1. 쿼리 확장 (Query Expansion)

- 목적: 쿼리에 관련된 추가 단어를 포함시켜 검색 성능 향상
- 동의어 filter (dimitri : 드미트리) 등을 manually 넣어서 따로 적용시켜 볼까 했지만 시간적이나 데이터셋이 적어서 가능한 부분이기에 pass 함
- CountVectorizer를 사용하여 쿼리에 가장 자주 등장하는 단어 추출 및 원래의 쿼리에서 가장 높은 빈도를 가진 단어를 추가하여 확장된 쿼리 생성

>> 단순 쿼리 라우팅: document의 문서별 토픽도 중요할 수 있지만 쿼리의 검색 성능 자체도 향상되면 더 좋은 결과가 나오지 않을까 예상으로 시작

- 관련 단어를 쿼리 추가 및 검색 결과 의 recall 향상
- 의미적으로 유사한 내용을 포함한 문서를 더 잘 검색할 수 있음

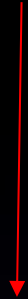
- 초기 검색 단계 강화: 관련된 문서 검색 향상

실습 및 예시

쿼리 확장과 MMR을 활용한 문서 검색 개선

2. MMR(Maximal Marginal Relevance)

- 목적: 검색 결과에서 관련성(Relevance) 와 다양성(diversity) 를 균형있게 유지
- 쿼리와 관련성을 기준으로 문서를 선택하되 이미 선택된 문서와 중복되지 않도록 다양성 추가
- 코사인 유사도를 활용하여 관련성, 및 다양성 계산



- 쿼리와 관련성이 높은 문서를 선택하면서 결과의 중복성을 줄임(다양성 강조)
- 검색 결과의 품질을 높임

```
while len(chosen_docs) < max_results and remaining_docs:
    scores = {}
    for i in remaining_docs:
        relevance = cosine_similarity([query_vec], [doc_vecs[i]])[0][0]
        diversity = (
            min(
                [cosine_similarity([doc_vecs[i]], [doc_vecs[j]])[0][0] for j in chosen_docs]
            ) if chosen_docs else 0
        )
        scores[i] = diversity_weight * relevance - (1 - diversity_weight) * diversity
```

- diversity weight: 0.5 (관련성/ 다양성 가중치), max_results: 반환 문서 최대 개수



- query expansion과 mmr을 통해 유의미한 document를 반환하리라 예상

Sparse (희소) 검색과 Dense(밀집)의 장점을 결합

- 코드 구현 알고리즘

1. 쿼리 확장 (query expansion) : 희소 검색 성능 강화
2. Sparse 검색 : BM25 기반 검색 수행
3. Dense 검색: 쿼리 임베딩을 사용해 벡터 유사도 기반 검색 수행
4. 점수 결합 (Score combination) : sparse와 Dense를 조합함
>> 가중치 (relevance weight는 0.3:0.7로 조정) : 아무래도 모델이 Dense에 조금 더 최적화 되어있음
5. MMR 기반 재정렬
>> score combination을 통해 나온 점수를 기준으로 문서 정렬< MMR을 적용함: 관련성 높은 문서중 중복되지 않는 결과 반환
6. 최종 출력 (hit 구조)

```
sorted_docs = sorted(combined_scores.items(), key=lambda x: x[1]['score'], reverse=True)

query_embedding = get_embedding([query])[0]
doc_embeddings = [info['vector'] for _, info in sorted_docs]
doc_ids = [doc_id for doc_id, _ in sorted_docs]

reranked_results = maximal_marginal_relevance(query_embedding, doc_embeddings, doc_ids, diversity_weight=0.7, max_results=result_limit)
```

- MMR 기반 정렬 코드

실습 및 예시

Hybrid Retrieval ver 2

검색 실행

- Sparse 검색(BM25)과 Dense 검색(임베딩) 각각 실행

```
def hybrid_retrieve(query_str, size):
    sparse_results = sparse_retrieve(query_str, size)
    dense_results = dense_retrieve(query_str, size)

    combined_results = {}

    # Sparse 결과 처리
    max_sparse_score = max(hit['_score'] for hit in sparse_results['hits']['hits'])
    for hit in sparse_results['hits']['hits']:
        docid = hit['_source']['docid']
        normalized_score = hit['_score'] / max_sparse_score
        combined_results[docid] = {
            'content': hit['_source']['content'],
            'sparse_score': normalized_score
        }
```

가중치 최적화 (sparse:dense)

- 0:10 → 순수 Dense 검색
- 3:7 → dragonkue 모델 최적
- 4:6, 6:4, 8:2, 10:0 ... 등

```
SPARSE_WEIGHT = 2.0
DENSE_WEIGHT = 8.0
```

Score 정규화

- Min-Max 정규화 적용
- sparse와 dense 각각 다른 스케일의 점수를 0-1 범위로 통일

```
sparse_scores = [hit['_score'] for hit in sparse_results['hits']['hits']]
if use_min_max_norm and len(sparse_scores) > 1:
    max_sparse_score = max(sparse_scores)
    min_sparse_score = min(sparse_scores)
    score_range = max_sparse_score - min_sparse_score
else:
    max_sparse_score = max(sparse_scores) if sparse_scores else 1
    score_range = max_sparse_score
```

KNN 쿼리

- dense 검색 구현 시 "num_candidates": 300 으로 후보 문서 설정

```
# KNN 쿼리 구성
knn = {
    "field": "embeddings",
    "query_vector": query_embedding[0], # 첫 번째 임베딩 사용
    "k": size,
    "num_candidates": 100, # 후보 문서 수
}
```

실습 및 예시

ReRanking

Rerank Model

- 다양한 모델을 기반으로 실험 생성
- bongsoo/klue-cross-encoder-v1
- ross-encoder/ms-marco-MiniLM-L-12-v2
- cross-encoder/ms-marco-MiniLM-L-6-v2
- klue/roberta-base or klue/roberta-large



결과

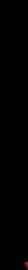
- 자체 검증 결과에서 SOTA 모델 비교 후 30%를 벗어나지 못함

```
=== output_final vs sample_submission_dense(re) ===  
총 비교 행 수: 220  
완전 일치 수: 22  
평균 일치율: 25.30%  
결과가 저장됨: comparison_results.csv
```

임베딩 모델

- dragonkue/bge-m3-ko(고정)

- skt/ko-bert-cross-encoder
- Dongjin/kr/ko-reranker



결과

- 자체 검증 결과에서 SOTA 모델 비교 후 70%를 이상이어서 채택

실습 및 예시

ReRanking...Continued

Rerank Model

- Dongjin/kr/ko- reranker
- skt/ko-bert-cross-encoder 중 성능이 조금 더 좋았던 Dongjin/kr/ko- reranker로 선정

- function 작동 원리

A. < rerank_with_model>

1. 문서 - 쿼리 페어 생성 (쿼리와 문서 내용 조합: 모델 입력 준비)
2. 점수 계산 (문서별 관련성)
3. 점수 정규화 (normalize)
4. 문서 정렬 (내림차순)

B. < answer_question>

1. 입력처리 (쿼리가 list일시 string 변환)
2. 검색 수행(hybrid_search) : topk 상위 10개 문서 검색
3. Reranker 적용 (A function)
4. 결과 반환 (상위 문서 3개 id topk에 추가하여 return)

Hybrid Retrieve & Reranking 적용 결과

- 단순 hybrid retrieve를 적용했을때는 효과가 미미했음 > 아마 bge m3 ko 모델이 dense에 최적화 되어있어서 sparse의 의미가 크게 없던 거 같음. (오히려 final 때 보니 결과가 더 떨어졌음)

- 사실상 Rerank적용이 순위 상승에 큰 기여를 했다고 봄

0.8538	0.8561	2024.12.17 20:25
0.8742	0.8773	

Prior Top score

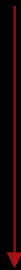


hybrid_exp	J	0.8568	0.8667	2024.12.19 13:08
		0.8545	0.8682	

After Hybrid Retrieval

hybrid_exp	J	0.8568	0.8667	2024.12.19 13:08
		0.8545	0.8682	

Prior Top score



	J	0.8947	0.8970	2024.12.19 21:33
		0.9167	0.9227	

After Hybrid Retrieval & Reranking

04

TroubleShooting

TroubleShooting

: CoBERT를 활용한 도메인 성능 최적화

검색엔진 또는 검색엔진에서 사용하는 모델 고도화를 통한 문서 추출 성능 최적화

- Dense Retrieval 활용
 - 고성능 임베딩 생성 모델(Vector Encoder)을 활용한 색인 및 검색
- CoBERT를 활용한 도메인 성능 최적화
 - 제공된 문서 집합에 대하여 pseudo train 데이터셋 구축한 후 CoBERT 모델 학습
 - 모델 학습 비용이 추가로 들지만 학습이 잘 이루어진다면 고성능의 검색엔진을 얻을 수 있음

```
## Role
가상 데이터 생성기

## Instructions
- 주어진 레퍼런스 정보를 보고 이 정보가 도움이 될만한 질문을 가상으로 3개 생성해줘.
- 아래 JSON 포맷으로 생성해줘.

## Output format
{"questions": [$question1, $question2, $question3]}
"""
```

- 제공된 문서 집합에 대해서 220개의 query에 따른 positive 문서, negative 문서 추출해 pseudo 데이터셋 생성
- pseudo 데이터셋을 통해 CoBERT을 학습
- CoBERT를 활용한 dense_retrieve 시도(FAISS 기반)
- 자체 성능 검증 결과 SOTA 모델과 일치율 30%미만으로 폐기
- 멘토링 후, 성능 하락에 대한 회고
 - 에포크를 여유롭게 설정했어야 했다(5로 두고 학습)
 - CoBERT는 문서 임베딩을 사전에 계산하는데 4200개는 적은 것 같았다, CoBERT보다 다양한 임베딩 모델을 시도해봤어야 했다

TroubleShooting

: documents.jsonl 데이터 셋 분석

멘토링 진행 후 query routing에 대한 생각을 함.

query routing을 하려면 documents.jsonl 데이터셋이 가까운 의미를 가진 문장끼리 잘 나뉘어져 있어야 함.

but... 대회 마감까지는 하루 남음...

그때 발견한 것...!

```
"src": "ko_mmlu__nutrition__test", "content": "건강한 사람이  
"src": "ko_mmlu__conceptual_physics__test", "content": "수  
"src": "ko_ai2_arc__ARC_Challenge__test", "content": "종이와  
"src": "ko_ai2_arc__ARC_Challenge__test", "content": "마이에  
"src": "ko_ai2_arc__ARC_Challenge__validation", "content":  
"src": "ko_mmlu__human_sexuality__test", "content": "AIDS  
"src": "ko_mmlu__virology__test", "content": "헤르페스 감염은  
"src": "ko_mmlu__human_aging__test", "content": "노인들이 기  
"src": "ko_ai2_arc__ARC_Challenge__train", "content": "강한  
"src": "ko_mmlu__conceptual_physics__test", "content": "광  
"src": "ko_mmlu__high_school_biology__test", "content": "등  
"src": "ko_mmlu__high_school_physics__test", "content": "일  
"src": "ko_ai2_arc__ARC_Challenge__validation", "content":  
"src": "ko_ai2_arc__ARC_Challenge__train", "content": "겨울  
"src": "ko_ai2_arc__ARC_Challenge__test", "content": "고체기  
"src": "ko_ai2_arc__ARC_Challenge__validation", "content":  
"src": "ko_mmlu__college_biology__test", "content": "쌍떡잎  
"src": "ko_mmlu__conceptual_physics__test", "content": "원  
"src": "ko_ai2_arc__ARC_Challenge__test", "content": "환경에  
"src": "ko_mmlu__nutrition__test", "content": "기후 변화는 미
```

```
"src": "ko_mmlu__nutrition__test", "content": "건강한 사람이  
"src": "ko_mmlu__conceptual_physics__test", "content": "수  
"src": "ko_ai2_arc__ARC_Challenge__test", "content": "종이와  
"src": "ko_ai2_arc__ARC_Challenge__test", "content": "마이에  
"src": "ko_ai2_arc__ARC_Challenge__validation", "content":  
"src": "ko_mmlu__human_sexuality__test", "content": "AIDS  
"src": "ko_mmlu__virology__test", "content": "헤르페스 감염은  
"src": "ko_mmlu__human_aging__test", "content": "노인들이 기  
"src": "ko_ai2_arc__ARC_Challenge__train", "content": "강한  
"src": "ko_mmlu__conceptual_physics__test", "content": "광  
"src": "ko_mmlu__high_school_biology__test", "content": "등  
"src": "ko_mmlu__high_school_physics__test", "content": "일  
"src": "ko_ai2_arc__ARC_Challenge__validation", "content":  
"src": "ko_ai2_arc__ARC_Challenge__train", "content": "겨울  
"src": "ko_ai2_arc__ARC_Challenge__test", "content": "고체기  
"src": "ko_ai2_arc__ARC_Challenge__validation", "content":  
"src": "ko_mmlu__college_biology__test", "content": "쌍떡잎  
"src": "ko_mmlu__conceptual_physics__test", "content": "원  
"src": "ko_ai2_arc__ARC_Challenge__test", "content": "환경에  
"src": "ko_mmlu__nutrition__test", "content": "기후 변화는 미
```

ko_mmlu__로 시작하는 데이터들은 이미 라벨링이 되어 있음. (2225개)

ko_ai2_arc__ARC_Cgallenge__로 시작하는 데이터들만 라벨링 하면 됨. (2047개)

시도1

라벨링 기준은 ko_mmlu__로 이미 되어 있는 라벨을 기준으로 4o-mini 모델을 이용해서

ko_ai2_arc__ARC_Cgallenge__ 데이터를 LLM이 자동으로 라벨링 하게 함

TroubleShooting

: documents.jsonl 데이터 셋 분석

시도1 결과
4o-mini가 잘못 라벨링 한 것들이 많아서 라벨링된 결과를 빠르게 다시 보면서 다시 라벨링함. (휴먼 라벨링... 결국... 500개 정도는 힘들어서 안 함...)

시도1을 통해 라벨링한 결과를 기준으로 시메틱 기반 쿼리 라우팅 진행함.
(나뉘어진 데이터셋의 임베딩 중심점을 계산하고 쿼리가 들어오면 가장 가까운 임베딩 지점으로 이동하여 해당 데이터 셋 안에서 document를 찾는 방식)

<input type="checkbox"/>	쿼리 라우팅, 청킹 단위 수정	김	0.6924 0.7606	0.6955 0.7652
<input type="checkbox"/>	쿼리 라우팅	김	0.7152 0.7242	0.7182 0.7288

baseline 코드의 흐름대로 따로 청킹을 하지 않다가 나중에 content단위로 청킹을 진행해서 실험을 진행함.
(Public 점수 기준으로 점수가 더 낮았지만 Private 점수 기준으로는 점수가 많이 올라감.)

하지만 baseline에 dragonkue/bge-m3-ko임베딩 모델 바뀌서 작동시킨 모델보다 점수가 잘 나오지 않는 것을 보고 documents.jsonl 데이터 셋을 잘 나누지 못해 결과가 좋지 않은 것은 아닐까 생각함.

TroubleShooting

: documents.jsonl 데이터 셋 분석

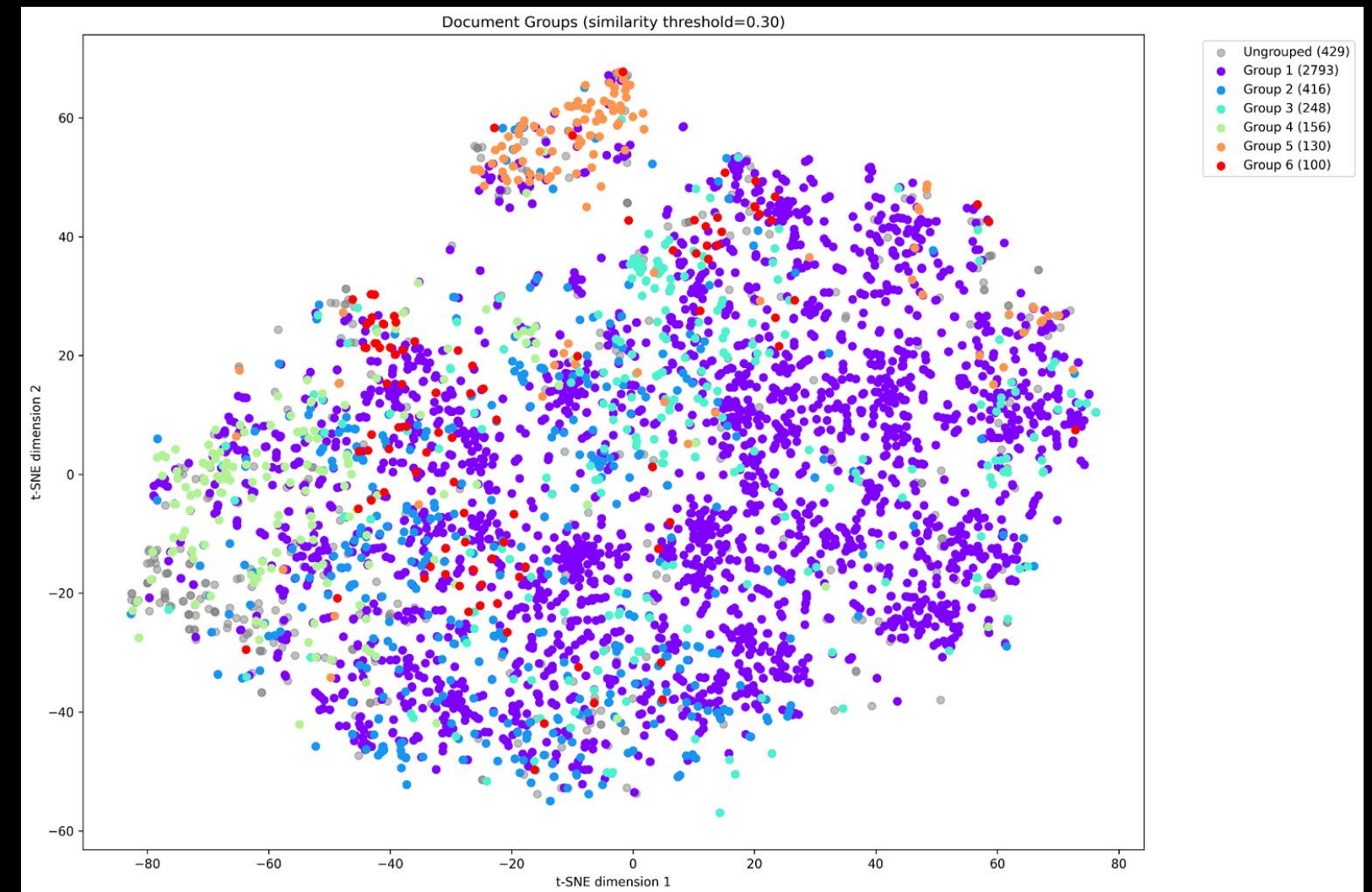
시도2

임베딩 모델을 이용하여 임베딩 거리가 가까운 문서끼리 묶어서 데이터 셋을 형성함.

dragonkue/bge-m3-ko모델과 OpenAI의 text-embedding-ada-002모델로 documents.jsonl 데이터 셋을 나눔.
PCA분석과 여러 변인 통제 실험을 통해 아래의 데이터 셋이 구축됨.

dragonkue/bge-m3-ko

```
{
  "total_documents": 4272,
  "total_groups": 6,
  "ungrouped_documents": 429,
  "group_statistics": [
    {
      "group_id": 1,
      "size": 2793,
      "avg_similarity": 0.37305125521580634
    },
    {
      "group_id": 2,
      "size": 416,
      "avg_similarity": 0.3400805425615265
    },
    {
      "group_id": 3,
      "size": 248,
      "avg_similarity": 0.3474404726537966
    },
    {
      "group_id": 4,
      "size": 156,
      "avg_similarity": 0.357325577774109
    },
    {
      "group_id": 5,
      "size": 130,
      "avg_similarity": 0.3587545190866177
    },
    {
      "group_id": 6,
      "size": 100,
      "avg_similarity": 0.35501504957675933
    }
  ]
}
```



TroubleShooting

: documents.jsonl 데이터 셋 분석

시도2

임베딩 모델을 이용하여 임베딩 거리가 가까운 문서끼리 묶어서 데이터 셋을 형성함.

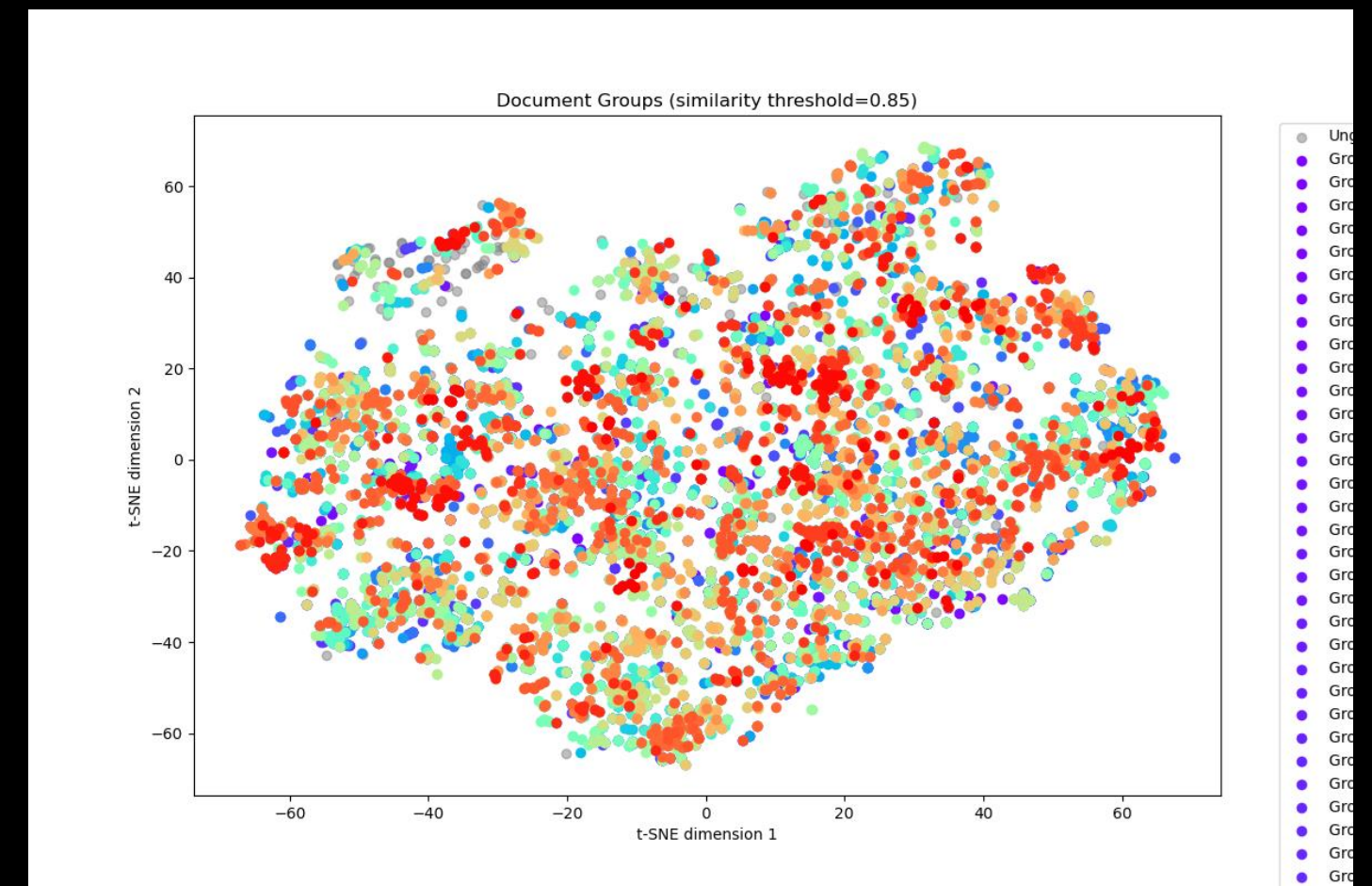
dragonkue/bge-m3-ko모델과 OpenAI의 text-embedding-ada-002모델로 documents.jsonl 데이터 셋을 나눔.
PCA분석과 여러 변인 통제 실험을 통해 아래의 데이터 셋이 구축됨.

text-embedding-ada-002

나뉜 데이터 셋을 기준으로 변인 통제 실험 진행 결과
RAG를 진행할 때 documents.jsonl 데이터를 나눌 때
사용했던 임베딩 모델과 같은 임베딩 모델을 사용하고,
유사성 관련 threshold값도 같은 값으로 설정해야
성능이 좋게 나옴.

```
{
  "total_documents": 4272,
  "total_groups": 554,
  "ungrouped_documents": 112,
  "group_statistics": [
    {
      "group_id": 1,
      "size": 49,
      "avg_similarity": 0.8662813201431799
    },
    {
      "group_id": 2,
      "size": 67,
      "avg_similarity": 0.8673649436812395
    },
    {
      "group_id": 3,
      "size": 31,
      "avg_similarity": 0.8802396346169584
    },
    {
      "group_id": 4,
      "size": 53,
      "avg_similarity": 0.8609948406115343
    },
    {
      "group_id": 5,
      "size": 34,
      "avg_similarity": 0.8687009973864218
    },
    {
      "group_id": 6,
      "size": 69,
      "avg_similarity": 0.8640457535288815
    },
    {
      "group_id": 7,
      "size": 29,
      "avg_similarity": 0.8748294554075934
    },
    {

```



TroubleShooting

: documents.jsonl 데이터 셋 분석

최종 결과

너무 시메틱 기반 쿼리 라우팅을 시도했음. 일정 수준 이상으로 성능이 오르지 않음.
팀 CV로 비교 결과 성능이 그렇게 좋지 못함. 모델이 가져오는 topk 3개를 보니 같은 문서 docid를 가져오는 경우도 있었음.

시메틱 기반 쿼리 라우팅을 할 때 같은 문서는 가져오지 못하도록 하고
가져온 문서들을 리랭킹, 하이브리드 retrieve방식을 사용하면 성능을 더 올릴 수 있지 않았을까 하는 아쉬움...

대회 기간이 짧아 그 방식은 시도해보지 못함.

TroubleShooting

Chunking

Semantic Chunking

- 목적: 긴 문서 자체를 *embedding* 하는 것보다 문장 혹은 문맥 단위로 더 의미있는 검색 및 재정렬을 수행하고자 적용
- *Semantic Chunking* 을 선택한 이유:
 1. '문맥' 별로 긴 문서를 여러 청크로 나누면서 문서 내 특정 부분이나 흐름에 초점을 맞춘 검색이 가능 할 것이라 예상
 2. 청크 내 문장들은 서로 유사하며, 청크 간에는 다름으로 구분 될 수 있도록 *cosine similarity* 를 기준으로 그룹화

```
chunk_doc.jsonl
1 {"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 동등성을 의미합니다. 일반적으로 건강한 사람은 1-"}
2 {"docid": "42508ee0-c543-4338-878e-d98c6babee66", "src": "ko_mmlu_nutrition_test", "content": "이 기간 동안에는 올바른 식단과 적절한 운동을 통해 에너지 섭취와 에너지 소비를 조절해야 합니다. 식단은 영양가 있는 식품을 포함하고, 적절한 칼로리를 섭취해야 합니다."}
3 {"docid": "4a437e7f-16c1-4c62-96b9-f173d44f4339", "src": "ko_mmlu_conceptual_physics_test", "content": "수소, 산소, 질소 가스의 혼합물에서 평균 속도가 가장 빠른 분자는 수소입니다. 수소 분자는 가장 가볍고 작은 원자로 구성되어 있기 때문에 다른 분자들보다 더 빠"}
4 {"docid": "4a437e7f-16c1-4c62-96b9-f173d44f4339", "src": "ko_mmlu_conceptual_physics_test", "content": "수소 분자는 화학 반응에서도 활발하게 참여하며, 수소 연료로도 널리 사용됩니다. 따라서 수소 분자는 주어진 온도에서 평균 속도가 가장 빠른 분자입니다."}
5 {"docid": "d3c68be5-9cb1-4d6e-ba18-5f81cf89affb", "src": "ko_ai2_arc_ARC_Challenge_test", "content": "종이와 플라스틱은 재활용 가능한 자원입니다. 중학교 과학 수업에서 우리는 종이와 플라스틱 가방을 재활용하는 프로젝트를 진행하고 있습니다. 이 프로젝트를 통해 국"}
6 {"docid": "d3c68be5-9cb1-4d6e-ba18-5f81cf89affb", "src": "ko_ai2_arc_ARC_Challenge_test", "content": "종이는 나무에서 추출되는 천연 자원으로, 나무를 재배하고 재생하는 과정을 통해 생산됩니다. 반면에 플라스틱은 석유에서 추출되는 인공적인 자원으로, 석유의 가공"}
7 {"docid": "910107a6-2a42-41a2-b337-fbf22d6440fe", "src": "ko_ai2_arc_ARC_Challenge_test", "content": "마이애미파랑나비는 남부 플로리다에서 멸종 위기에 처한 종입니다. 이 나비의 개체수 감소를 초래했을 가능성이 가장 높은 요인은 주택 건설 증가입니다. 남부 플로리"}
8 {"docid": "910107a6-2a42-41a2-b337-fbf22d6440fe", "src": "ko_ai2_arc_ARC_Challenge_test", "content": "주택 건설 증가를 제한하거나 대안적인 서식지를 마련하는 등의 방안이 고려되어야 합니다."}
```

- *Chunk 된 doc.jsonl 예시*

TroubleShooting

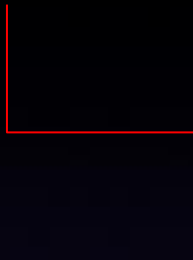
Chunking

결과

- 기대와는 달리 평가 점수가 하락

csv수정_chunkedNew	J	0.8356	0.8394	2024.12.19 18:19
		0.8667	0.8682	

- 과도하게 문서 분할로 인해 검색 성능이 저하 되었을 가능성이 있음: 문서 단위 검색보다 청크 단위 검색에서 결과의 응집도가 낮아짐
- bge-m3-ko 모델이 임베딩 성능이 좋아서 굳이 청크로 세분화 해서 하는 시도보다 그 자체가 나왔던 거 같음.
- 다른 모델 실험으로 청킹을 시도 해봤으면 성능이 올랐을 가능성이 있음.



- 개선 방향
 1. 다른 모델로 바꿔서 진행 (시간적 여유때문에 실험 x)
 2. 문서 분할 기준 조정(유사도 및 크기: 최소 크기 150으로 실험했었음)
 3. 다른 chunking 방식 시도 (semantic 이 텍스트를 의미론적 유사성에 기반하여 분할하기 때문에 시도해본것: 예시: agentic chunking)

TroubleShooting

Hybrid Retrieval

. 시도1

- Position-aware 스키닝 → 자체 검증 평균 일치율 48.44%
 - 상위 결과에 가중치 부여
 - 양쪽 검색에서 모두 높은 순위 받은 문서가 더 높은 점수

```
# 6. Position-aware 스키닝
if use_position_aware:
    for idx, hit in enumerate(sparse_results['hits']['hits']):
        docid = hit['_source']['docid']
        if docid in combined_results:
            combined_results[docid]['position_score'] = 1.0 / (idx + 1)
```

. 시도2

- 쿼리 라우팅 → 성능 개선 x
 - science/computer/medical/engineering/general 로 카테고리 분류해 각각 검색 엔진 적용

```
def get_retrieval_strategy_with_category(eval_id, query):
    """카테고리를 고려한 검색 전략 결정"""
    # 기본 전략 결정
    strategy = get_retrieval_strategy(eval_id)
    categories = determine_query_category(query)
```

. 시도3

- 결과 후보군 확대 → 성능 개선 x

```
# 2. 기본 검색 수행
sparse_results = sparse_retrieve(search_query, size * 2)
dense_results = dense_retrieve(query_str, size * 2)
```

. 시도4

- elasticsearch 내장함수 → 성능 개선 x

```
def hybrid_retrieve(query_str, size):
    """하이브리드 검색 구현"""
    try:
        # 쿼리 임베딩 생성
        query_embedding = model.encode(query_str)

        # Elasticsearch 복합 쿼리 구성
        query = {
```

. 시도5

- 쿼리 확장 → 자체 검증 평균 일치율 44.62%
 - 형태소 분석으로 핵심 키워드 추출
 - 동의어 사전 일부 활용

05

그룹 스터디 회고

그룹스터디 진행 느낀점/소감

: Scientific Knowledge Question Answering

분담의 중요성

Point 1

이유 : 만약 현업에서 4일 안에 성과를 내야 한다면 어떻게 효율적으로 역할을 맡아서 프로젝트를 진행할 수 있을까에 대한 고민을 할 수 있었음. 주어진 타임라인 안에 각자 시도한 방법들로 결과가 나온다 안나온다를 빠르게 정하고 다음 전략을 수립할 수 있었음.

향후 계획 : 앞으로의 프로젝트에서도 협업시에 다 같이 진행하는 것도 좋지만 실험기록들을 잘 기록하고 분담을 조금 더 세분화를 해서 효율을 낼 수 있었으면 좋겠다.

실무 환경을 경험할 수 있는 좋은 기회

Point 2

이유 : 데이터가 그렇게 많지 않은 환경, 기준이 애매모호해서 직접 기준을 설정해야 하는 것들을 경험할 수 있어 좋았다. 다만, 대회 기간이 짧아서 더 다양한 방법론들을 시도해보지 못했던 것이 아쉽다.

향후 계획 : 개인적으로 IR과 RecSys를 결합해서 진행하고 싶은 프로젝트가 있는데 그 프로젝트를 위한 좋은 경험이었다.

깊이 있게 학습할 수 있었던 기회였다

Point 3

이유 : 비록 시간이 짧은 대회였지만, 많은 것을 알게된 대회였던 것 같다. 특히 멘토링 시간에도 현업에서 사용할 수 있는 해결책등을 알려주셔서, 비록 대회에서는 실제 구현하기가 실력이 부족했지만 추후 학습할 때에는 도움이 될 것 같다

향후 계획 : 청킹, reranking, hybrid retrieve, query 라우팅 등 다양한 것을 적용해보고 실습해보고 싶다

시간 부족으로 계획한 실험 못 끝난 아쉬움

Point 4

이유 : 해보고 싶은 실험들이 4일동안 전부 구현하기 어려워서 아쉬웠다. 그래도 팀원분들과 배분이 잘 이뤄져서 효율적이었다.

향후 계획 : hybrid retrieve에서 추가 가중치 설정 등 실험들을 진행해보고 싶다.

Life-Changing Education

감사합니다.
