

NLP 기초 프로젝트

RAG based Open-Domain Question Answering (ODQA with RAG)

TEAM : 아기더키들(NLP-09조)



MEMBER : 박동혁_T7335

곽희준_T7308

김정은_T7325

한동훈_T7440

[1. 프로젝트 개요]

프로젝트 주제	<RAG based Open-Domain Question Answering(ODQA with RAG)> 대규모 문서 집합에서 질문에 대한 답을 얼마나 잘 찾을 수 있는지에 대한 Task
구현내용	대규모 문서 집합을 sparse retriever 및 LLM기반의 Reranking Model 을 차례대로 통하여 질문에 대한 검색을 시도 후, 검색결과를 바탕으로 질문에 대한 답변을 reader 를 통하여 찾음
개발 환경	GPU: Tesla V100 Linux server 4개 Development Tool: Git, Pandas, Hugging Face Transformers, Pytorch, ,langchain, langSmith, OpenAiAPI
협업 환경	Github: 코드 공유 Slack: 서버 사용 여부 정리, 소통 Notion: Idea Brainstorming 및 계획 정리, 실험 결과 Zoom: 실시간 회의 Jira: 프로젝트 전체 일정 공유, 역할 분배
프로젝트 구조 <Retrieval>	Retrieval -- config/ -- ST01.json -- input/ -- data/ -- embed/ -- checkpoint/ -- notebooks/ -- EDA.ipynb -- scripts/ -- run_retrieval.sh -- run_reader.sh -- run.sh -- predict.sh -- src/ -- data_processing/ -- arguments/ -- retriever/ -- sparse/ -- dense/ -- hybrid/ -- reader/ -- utils/ -- run_retriever.py -- run_reader.py -- run.py -- predict.py -- postprocessing.py -- ensemble.py -- .gitignore -- README.md -- requirements.txt

	-- environment.yml
프로젝트 구조 <Reranker>	Reranker -- data/ -- pipeline/ -- BM25Ensemble_top100_original.csv # input file -- reranker_final.csv # output file -- notebooks/ -- dense_pipeline_result.ipynb # result analysis -- scripts/ -- test.sh -- src/ -- retrieval_test_bge-reranker_pipeline.py # code -- .gitignore -- environment.yml -- README.md
프로젝트 구조 <Reader>	Reader -- data/ -- raw/ -- external/ -- preprocessed/ -- models/ -- notebooks/ -- EDA.ipynb -- outputs/ -- src/ -- arguments.py -- ensemble.py -- inference_csv.py -- train_csv.py -- train_wandb.py -- trainer_qa.py -- utils_qa.py -- eval.sh -- inference.sh -- train.sh -- .gitignore -- README.md -- condaenvironment

[2. 프로젝트 팀 구성 및 역할]

이름	역할
김정은	Directory Structure Settings, Modularization, EDA (통계 분석, 문장데이터 분석), Augmentation (Question Paraphrasing, Distant Supervision), Modeling (Sparse/Dense Retrieval, Retrieval Ensemble, Hybrid Search, MRC), Post-processing (Llama), Ensemble (soft voting, hard voting)
곽희준	EDA, Preprocessing (Title concat, Chunking), Modeling (Dense Retrieval, Reranker)
박동혁	EDA (wiki문서이상치확인), Preprocessing (외국어 드랍, 100글자이상의 단어 처리), Modeling (RAG와 LangChain)
한동훈	EDA, Preprocessing (query preprocessor), code refactoring, git management, Ensemble (soft voting)

[3. 프로젝트 일정 및 협업 Tool]

협업관련

- **Server:** 4명의 팀원이 4개의 서버를 공유하면서 작업
- **Git Collaboration:** 각 팀원이 브랜치에서 독립적으로 작업
- **Remote Repository:** 코드 관리를 하며 자유롭게 서버 이동

프로젝트 주차 일정

October 2024

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	Oct 1	2	3	4	5
				강의 - 기본 지식 습득 (전체)		
6	7	8	9	10	11	12
강의 - 기본 지식 습득 (전체)				EDA (전체)		
13	14	15	16	17	18	19
EDA (전체)				RAG(동학)		
		데이터셋 전처리 및 증강 (전체)		Reader Model(동훈)		
				Retrieval Model(정은, 희준)		
20	21	22	23	24	25	26
RAG(동학)						
Reader Model(동훈)						
Retrieval Model(정은, 희준)						
			Ensemble			
				Due Date		

[4. 프로젝트 수행 결과]

4.1 EDA

4.1.1 데이터 개요

데이터는 retrieval 탐색에 사용될 60613개의 wiki 출처 문서 집합과 reader 모델 학습에 사용될 4,192개의 MRC 훈련 데이터가 존재한다. Wiki 문서 집합의 'text'의

WIKI문서 집합	60613
Train Set	3952
Validation Set	240

MRC훈련데이터가 존재한다. Wiki문서 집합의 'text'의 경우, 실제 wiki홈페이지에서 전체가 아닌 일부분을 무작위로 가져온 것을 확인할 수 있었다. 또한, MRC 데이터의 'context'에 해당하는 값들은 모두 wiki문서 집합 'text' 항목에 존재한다. 이에, 무작위로 가져왔다는 점을 착안하여 실제 가져온 데이터에 본문의 내용과 상관 없는 데이터가 존재함을 확인하는 것을 주요 목표로 삼았다. 추가로, 해당 노이즈 값들이 베이스라인 기준 토큰화시, 어떤 영향을 미칠까에 대하여 탐구해 보기로 한다.

4.1.2 WIKI

위키 문서 집합의 경우, 의미가 모호한 컬럼들을 제외하고 ‘**tilte**’와 ‘**text**’ 컬럼 위주로 살펴 보았다. 결과는 다음의 이삭치들이 발견 되었다.

- 실제 위키 페이지에서 가져올때, 본문의 내용이 아닌 각주에 해당하는 부분으로 이뤄진 문서 조각이 존재.
- 한국어 및 영어 외의 언어로만 구성된 본문이 존재; 토큰나이징을 해본 결과 한문의 경우 일부 인식이 가능했고 그외의 언어들을 전부 [UNK] 토큰 처리가 된다.

text:	. العلم يرفرف ليعلن الإستقلال التام ترتقي الأمة بسبب إيمان في جزرنا القمرية
token:	['[UNK]', '[UNK]', '[UNK]', '[UNK]', '[UNK]', '[UNK]', '[UNK]', '[UNK]', '[UNK]', '[UNK]', '[UNK]', '[UNK]', '.']

- 데이터의 중복은 다음의 두가지 패턴이 존재한다;
 - Title이 같고 text가 같은 경우
 - Title이 다르지만 text가 같은 경우
- 한국어는 토큰화시 띄어쓰기에 민감하다. Spacing없이 100글자 이상 연속으로 이뤄진 단어가 다수 발견

[illegible]

4.1.3 MRC dataSet (Train + Validation)

해당 데이터는 ['title', 'context', 'question', 'id', 'answers(answer_start, text)', 'document_id', 'index_level_0'] 에 대한 정보를 포함하고 있고 이중 context, question, answer가 명확한 의미성이 보여 바라보았다.

- Answer와 question 컬럼에서” [한글, 영어(소문자, 대문자), 숫자, ‘?’, ‘.’, space]” 이외의 특수 문자 존재 확인을 해본 결과, 4,192개의 데이터 중 564개의 데이터에 들어 있었고 그 종류의 일부분과 빈도수는 다음과 같다.

special char	count	special char	count
0	(0	媒
1)	1	日
2	'	2	坂
3	,	3	駅
4	"	4	蔣
5	<	5	儼
6	>	6	流
7	<	7	遷

- 모든 정답에는 한국어나 영어가 섞여 있음.
- 정답값의 최대 길이는 20까지이고 문장의 형태가 존재 한다.

길이가 8 이상인 Answer 개수: 21

길이가 8 이상인 Answer 목록:

- 1: «인간의 이해: 개념의 집단적 사용 및 진화 (1972)»
- 2: "오 그 불쌍한 사람들... 그러나 그건 별로 중요하지 않아요! 중요한건 난 새로운 배달원이 필요했다는 거지!"
- 3: 최근의 반도의 경제사범 국민의 신경제 윤리의 파악을 위하여
- 4: 도스토옙스키가 침대 누워 구술한 것을 아내 안나가 속기 하여
- 5: 독일의 고전 문헌학자 헤르만 딜스가 이들 철학자의 단편들을 현대 모아 엮은 책의 제목
- 6: "나는 국경일에 일장기를 게양하는 것을 반대하지 않는다. 왜냐하면 우리가 일본의 통치하에 있는 한 우리는 그 통치의 명령에 복종해야 하기 때문이다"
- 7: 복위 58도 이상의 사할린 섬 및 조차지 요충 반도
- 8: 머리가 크고, 가늘고 허약해 보이는 동체를 꼬아놓은 모습
- 9: 고문서로 발견된 유물이 원어로는 존재하지 않고 번역된 언어로만 존재할 경우
- 10: 당대의 바벌로니아 기록들이 현재까지 별로 많이 남아있지 않기 때문
- 11: 샤 자한이 이 무덤을 지을 때 자신이 그녀의 옆에 묻힐 것을 예상하지 못하였기 때문
- 12: 자신이 로마 제국의 황제의 계승자라고 굳게 믿었기 때문
- 13: 형태에 있어 거의 혹은 전혀 자유가 없어 다양한 번역을 창조해내기가 불가능하고 시구 구조에서도 다른 여지를 찾기 어려운 탓이다.
- 14: 과거 음력에 맞춰 작성한 것으로 오해를 받은 탓
- 15: "소유의 절반을 가난한 자들에게 주겠으며, 만일 누구의 것을 부당하게 취한 일이 있으며 사 배나 보상할 것"
- 16: 현재레가 상성에 대해 역의적인 보도로 일관하고 있다고 판단
- 17: 매킨토시와 도스 상에서 '설정 가능한' 플랫 파일 데이터베이스 응용 프로그램
- 18: 주거 및 상업시설 복합 건축물의 상업 지역 부분 옥상 등지
- 19: 제9번이 마지막 교향곡이 된다는 유명한 신화 (9번 교향곡의 저주)
- 20: 황도철학이라는 어용 학문을 연구해 다수의 논설을 발표한 일
- 21: USS 사라토가 호에서 조지 로열이 흑인 수병 해럴드 J. 맨스필드를 총으로 쏘아 살해한 사건

4.1.4 결론



MRC데이터 셋의 **answer** 컬럼에 특수기호가 다수 존재하기에 **wiki**문서 집합에서 특수기호 전처리는 진행하지 않았다. 또한, 외국어의 경우, 모든 **answer**에서 영어 및 한국어어와 함께 존재 하기에, 본 프로젝트에서 답을 찾을시, **[UNK]**로 인식하더라도 큰 영향이 없을 것이라 판단하였다.



앞서 언급한 사항을 제외 하고 남은 **wiki**데이터에서의 중복 문제는 문서 탐색 과정에서 영향을 미칠 것을 우려하여 전처리를 진행하기로 하였다. 추가로, 탐색시 주어지는 질문에 대한 의미를 강조해주기 위한 처리도 진행 하였다.

4.2 Data Preprocessing

4.2.1 retrieval에서 질문의 의도 강조

Sparse embedding의 특성상 query에서 키워드를 강조한다면 좀 더 성능이 좋을 것으로 판단하여 query의 키워드(동사, 명사 등)를 추출하여 query 앞에 공백을 구분자로 이어 붙였다.

only wiki div		33.7500%	48.0500%	2024.10.16 15:55	완료	
기본 baseline코드 보다 f1과 EM이 소폭 증가 하였다. -> reader는 고정인 상태이기 때문에 retrieval에서 성능향상이 있었다고 판단할 수 있으며 가설이 입증되었다고 볼 수 있다.						
answer가 context에 그대로 포함되어있으며 길이가 충분히 작기 때문에 wiki의 데이터 중 너무 긴 text를 가진 경우는 분할하는 것이 retrieval이 성공적으로 진행되었을 경우 reader가 답을 찾는데 성능향상이 있을 것으로 판단하여 분포를 확인하고 1500자 이상인 경우 최대 길이 1200, doc_stride 128로 wiki를 분할하여 추가하고 원본 데이터는 삭제하는 방식으로 wiki를 수정하였다.						

only wiki div		33.7500%	48.0500%	2024.10.16 15:55	완료	
F1에서 소폭 증가, EM에서 소폭 감소 하였는데 sparse embedding의 특성상 키워드가 적어짐에 따라 정답이 포함된 context를 retrieval하지 못하게 되어 F1이 낮아지고, 가져왔던 context는 길이의 영향을 받아 성능향상이 있음을 확인할 수 있었다.						

4.2.2 wiki문서에서의 중복 제거

먼저, 중복데이터 중 title이 다른 것 들은 text에 해당 title를 추가해 주었다. 해당 과정을 통하여 탐색시, 질문의 맞는 문서의 폭을 넓히기 위함을 겨냥하였다. 이후에 중복된 문서를 drop시킨 결과 총 60.613의 문서 집합은 56,808개로 축소 하였다.

성능 평가를 위하여 top-K개중 한개라도 제대로 뽑았으면 correct라 하고 정답률(= (top 5개 중 하나라도 맞으면 correct) / 4,192)을 구한 결과 다음의 표와 같이 소폭의 상승 효과를 보았다.

	top k = 1	top k = 3	top k = 5	top k = 10	top k = 15
Raw Wiki	0.2366	0.4220	0.5012	0.6064	0.6677
Title Wiki	0.2433	0.4389	0.5165	0.6162	0.6751
비교	+0.0067	+0.0169	+0.0153	+0.0098	+0.0074

top k = 15	top k = 20	top k = 25	top k = 30	top k = 35
0.6677	0.7004	0.7250	0.7486	0.7686
0.6751	0.7118	0.7369	0.7583	0.7762
+0.0074	+0.0114	+0.0119	+0.0097	+0.0076

4.3 Data Augmentation

4.3.1 Question Paraphrasing

- 가설 → train 데이터 기반에서 다양한 형태의 질문을 추가 학습시키면, 각 모델이 새로운 질문을 더 잘 이해하고 task를 수행할 수 있지 않을까
- 기록

SKT-AI/KoGPT2

Original Question: 서울은 대한민국의 수도인가요?
Paraphrased Question: 질문: 서울은 대한민국의 수도인가요?
이 질문을 다른 표현으로 바꿔 주세요:
새로운 질문: 대한민국의 수도는 서울이 맞다고 생각하십니까?

Original Question: 대통령을 포함한 미국의 행정부 견제권을 갖는 국가 기관은?
Paraphrased Question: 질문: 대통령을 포함한 미국의 행정부 견제권을 갖는 국가 기관은?
이 질문을 다른 표현으로 바꿔 주세요:
새로운 질문: 대통령, 부통령, 국가안보회의(NSC) 위원장, 국가정보국(DNI) 국장, NSC 부보좌관, 국가안전보위부(NSA) 요원, NSA 국장 등등.

maywell/Llama-3-Ko-8B-Instruct

Original Question: 대통령을 포함한 미국의 행정부 견제권을 갖는 국가 기관은?
Paraphrased Question: 질문: 대통령을 포함한 미국의 행정부 견제권을 갖는 국가 기관은?
이 질문을 다른 표현으로 바꿔 주세요:
새로운 질문: 미국 헌법 1조에 따르면, 대통령의 권한을 제한하는 국가기관은 무엇입니까?

allganize/Llama-3-Alpha-Ko-8B-Instruct

Original Question: 대통령을 포함한 미국의 행정부 견제권을 갖는 국가 기관은?
Paraphrased Question: 질문: 대통령을 포함한 미국의 행정부 견제권을 갖는 국가 기관은?
이 질문을 다른 표현으로 바꿔 주세요:
새로운 질문: 미국 헌법에서 대통령의 권한을 제한하기 위해 설립된 주요 기구는 무엇인가요?

Original Question: 현대적 인사조직관리의 시발점이 된 책은?
Paraphrased Question: 질문: 현대적 인사조직관리의 시발점이 된 책은?
이 질문을 다른 표현으로 바꿔 주세요:
새로운 질문: 20세기와 21세기의 인적 자원 관리 이론과 실천의 주요 출판물은 무엇인가요?

- 결과
총 데이터 수 = 7904
=====
exact_match = 52.5 f1 = 60.1996

===리더보드 제출===
exact_match = 23.75 f1 = 35.74
- 원인 분석
TF-IDF 임베딩 방식의 Retrieval Model에서는 question 내 단어의 유사성 또는 맥락을 파악을 못하여, question만 다른 두 개의 동일한 데이터가 학습된 것이 오히려 방해가 되지 않았을까 → Dense 임베딩 방식에서 재실험이 필요해보임

4.3.2 KorQuAD & Ko-WIKI

KorQuAD 1.0:

KorQuAD 1.0의 전체 데이터는 1,560 개의 Wikipedia article에 대해 10,645 건의 문단과 66,181 개의 질의응답 쌍으로, Training set 60,407 개, Dev set 5,774 개의 질의응답쌍으로 구분

- train : 60,407 개
- dev : 5,774 개

Ko-WIKI:

한국어 위키의 덤프 데이터를 바탕으로 제작한 **wikitext** 형식의 텍스트 파일

- train : 14528095 lines (868230 articles)
- dev : 76788 lines (4385 articles)
- test : 70774 lines (4386 articles)

학습데이터가 기존 4000개에서 대폭 증가함에 따라 reader의 성능이 높은 것 처럼 보였지만 validation에서 성능이 안좋게 나옴 -> 과적합으로 판단됨

4.3.3 Distant Supervision

train 데이터의 질문과 답을 기반으로 위키피디아 문서 내에서 **context**를 새로 뽑아보기

- 가설 → 다양한 **context**에서 질문과 답변이 나오게 되면, 이를 기반으로 새로운 질문에서 retriever가 관련 문서를 더 잘 가져올 수 있지 않을까
- 기록
 - TF-IDF / BM25 / DPR / DenSPI / DensePhrases / ColBERT ...
 - Augmentation 횟수 → 1, 2 ...
 - 교차 Augmentation (BM25 + DPR) →
 - DPR
 - train 데이터의 질문과 위키피디아 문서에서 첫 문단 유사도가 가장 높은 문서 선택
 - 선택된 문서의 **document_id**가 train과 다르고 **answer**가 문서에 존재하면, 해당 문서를 **context**로 사용
- 결과
 - 시간이 너무 오래 걸려서 Post-processing에서 학습이 잘 안되는 문서 위주로 시도해보면 좋을 듯

4.4 Modeling

4.4.1 Retrieval Model

	model & tokenizer	top k = 1	top k = 3	top k = 5	top k = 10	top k = 35
TF-IDF		48.33 %	72.08 %	78.75 %	85.83 %	
BM25L	monologg/koelectra-base-v3-discriminator	60.00 %	74.58 %	81.67 %	87.50 %	
BM25Plus	monologg/koelectra-base-v3-discriminator	56.67 %	72.50 %	77.92 %	83.75 %	
BM25Plus + KorQuAD		56.67	72.50	77.92	83.75	→ reader 성능 개선?
ATIREBM25	xlm-roberta-large	56.25 %	72.92 %	76.67 %	82.92 %	
BM25Ensemble		71.67 %	85.42 %	86.67 %	92.08 %	96.67 %
DPRBERT	bert-base-multilingual-cased	20.83 %	30.42 %	32.92 %	40.83 %	
DPRBERT + Question Paraphrasing		16.67 %	28.75 %	32.08 %	40.42 %	55.83 %
DPRBERT + title wiki		21.67 %	32.92 %	37.92 %	45.42 %	56.67 %
DPRBERT + model wiki		0.0	0.0	0.0	0.0	0.0
DPRBERT + KorQuAD	(epoch = 3)	27.50 %	40.00 %	46.25 %	56.67 %	
LOG_BM25Ensemble_DPRBERT		70.42 %	82.92 %	84.58 %	90.00 %	

SparseRetrieval : 단어 빈도수로 문서를 검색하는 전통적인 방법의 Retriever

- TFIDF
- BM25L (lower-bounding)
- BM25Plus (saturation-free)
- ATIREBM25 (ATIRE)
- BM25Ensemble

DenseRetrieval : 잠재 벡터로 문서를 검색하는 딥러닝 기법의 Retriever

- DPRBERT

HybridLogisticRetrieval : SparseRetrieval와 DenseRetrieval중에 어떤 것을 사용하여 문서를 검색할 지 LogisticRegression 모델을 사용하여 판단하는 Retriever

- LOG_BM25Ensemble_DPRBERT

-> Sparse Retrieval에서는 TF-IDF 대비 BM25가 성능이 더 좋았으며, tokenizer와 architecture의 다양한 조합으로 ensemble한 BM25Ensemble의 성능이 가장 좋았음.

-> Dense Retrieval에서는 비교적 낮은 성능을 보였으며, KorQuAD 데이터셋을 활용한 Augmentation이 성능 향상에 큰 효과를 보임. 다만, Sparse와 Dense의 Hybrid Model은 오히려 성능이 저하되어 기존 BM25Ensemble을 최종 Retrieval Model로 선정.

4.4.2 Reranking Model

- Input: [240(query) x Top 100] (Sparse Retrieval로 60,613개 -> 100개로 줄인 문서)
- Output: [240(query) x Top 100(Re-ranked)] (100개 문서와 쿼리의 유사도를 reranking)

실험 결과 테이블:

Model / Top K2	Top 1	Top 2	Top 3	Top 5	Top 10	Top 15	Top 20
dragonkue/bge-reranker-v2-m3-ko	91.25% (219/240)	95.42% (229/240)	95.83% (230/240)	96.67% (232/240)	97.08% (233/240)	97.50% (234/240)	97.50% (234/240)
Sparse only	71.67% (172/240)	82.08% (197/240)	85.42% (205/240)	86.67% (208/240)	92.08% (221/240)	94.58% (227/240)	95.00% (228/240)
dragonkue/BGE-m3-ko	66.67% (160/240)	79.17% (190/240)	83.33% (200/240)	87.50% (210/240)	94.58% (227/240)	95.83% (230/240)	96.25% (231/240)
Ensemble (Sparse only, BGE-m3-ko, KoE5 (max))	73.75% (177/240)	82.50% (198/240)	84.58% (203/240)	85.83% (206/240)	90.00% (216/240)	92.50% (222/240)	92.92% (223/240)
Ensemble (Sparse only, BGE-m3-ko, KoE5 (mean))	72.50% (174/240)	82.50% (198/240)	85.42% (205/240)	86.67% (208/240)	89.58% (215/240)	91.25% (219/240)	91.67% (220/240)
nlpai-lab/KoE5 (max)	60.42% (145/240)	76.25% (183/240)	79.17% (190/240)	85.42% (205/240)	92.92% (223/240)	94.58% (227/240)	95.00% (228/240)
nlpai-lab/KoE5 (mean)	52.08% (125/240)	67.50% (162/240)	74.58% (179/240)	81.67% (196/240)	89.17% (214/240)	90.83% (218/240)	92.92% (223/240)
Alibaba-NLP/gte-Qwen2-1.5B-instruct	57.08% (137/240)	67.08% (161/240)	73.75% (177/240)	80.83% (194/240)	85.42% (205/240)	88.33% (212/240)	90.83% (218/240)
upskyy/bge-m3-korean	46.67% (112/240)	60.00% (144/240)	67.92% (163/240)	74.17% (178/240)	82.92% (199/240)	87.50% (210/240)	90.42% (217/240)
sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2	32.08% (77/240)	41.25% (99/240)	45.00% (108/240)	50.83% (122/240)	59.58% (143/240)	65.42% (157/240)	70.83% (170/240)
gte-multilingual-base	55.00% (132/240)	67.92% (163/240)	75.00% (180/240)	81.67% (196/240)	86.67% (208/240)	88.33% (212/240)	90.42% (217/240)

-> bge기반 reranker모델이 압도적인 성능을 보였음

4.4.3 Reader Model

Model	EM	F1	민준데이터 EM,F1
klue/roberta-base	58.75 58.75(5epoch)	68.0255 68.7286(5epoch)	60.4167 / 70.9767 (5 epoch) csv val : 44.5833 / 53.8742
klue/roberta-large	62.91667	71.22129	
monologg/koelectra-base-v3-finetuned-korquad	61.25	68.6281	
HANTA EK/klue-roberta-large-korquad-v1-qa-finetuned	63.33333	72.4089	

한국어 처리에 좋은 성능을 보이는 korquad, klue 관련 모델을 사용함 -> 모델이 큰 roberta - large가 성능이 높게 측정되었으며 학습 데이터와 유사한 korquad v1로 학습한 모델이 조금 우세한 성능을 보였다.

외부 train data의 경우 data의 양이 많은 만큼 학습을 더 진행했기 때문에 train에서 성능이 좋게 나왔지만 validation에서 성능이 매우 낮게 나온 것을 확인하고 과적합으로 판단하여 폐기하였다.

Model / Top K	Top 1	Top 2	Top 3	Top 5	Top 8	Top 10	top 20,35	Top all
klue/roberta-base (동훈)					47.5 / 55.467	48.3333 / 56.0226, 48.3333 / 56.3003 (Top9)	48.3333 / 52.0226(top 20,35)	(EM / F1 Score)
	53.75		50.4					
HANTA EK/klue-roberta-large-korquad-v1-qa-finetuned (동훈)			50.8333 / 59.3699 //	50.8333 / 57.6754	51.6667 / 58.3977	49.1667 / 56.1477 47.9167 / 55.6095 (정은데이터)	42.9167 / 49.1141	36.25 / 42.6558
	57.9							
monologg/koelectra-base-v3-finetuned-korquad (동훈)								
	57		51					
klue/roberta-large (3예폭,민준데이터) (동혁)			52.0833/60.1106	50.8333/59.1384	50.0 / 58.2619	49.1667/57.3591		
	58.3333/67.4626 - 회준rerank	55.4167/64.9717- 회준rerank						

Reranker를 사용하지 않은 상황에서 top k가 5~10사이에서 최고성능을 보이는 경향이 있으며, reranker를 사용한 경우 top k가 1에서 최고 성능을 보였다. -> 이 부분에서 reader model이 서로 다른 passage를 독립적으로 답을 구했다면 top 3에서도 충분히 top 1의 정확도를 가지면서 3개의 passage를 retrieval의 정확도 또한 높여 상호 보완 효과(synergy)로 대폭 상승했을 것이다.

4.5 Ensemble

4.5.1 Soft Voting

각기 다른 모델을 이용해서 nbest_prediction.json을 추출한 뒤 같은 question_id에서 같은 답을 갖는 경우 prediction값을 평균을 취한 이후, prediction의 값이 높은 text를 answer로 최종 결정하는 ensemble을 사용함

- 사용 모델 10 epoch, raw train datasets, 16 batch_size, seed 24

- klue/roberta-base
- HANTAEEK/klue-roberta-large-korquad-v1-qa-finetuned
- monologg/koelectra-base-v3-finetuned-korquad



4.5.2 Hard Voting

여러 모델에서 나온 정답들 중 가장 많이 예측된 answer를 최종 answer로 채택함

- 사용 모델 10 epoch, raw train datasets, 16 batch_size, seed 24

- klue/roberta-large (Sparse Retrieval only)
- klue/roberta-base
- HANTAEEK/klue-roberta-large-korquad-v1-qa-finetuned
- monologg/koelectra-base-v3-finetuned-korquad

[5.최종 리더보드 결과]

<PUBLIC>					
내등수 13	NLP_09조		62.9200%	73.4600%	
<PRIVATE>					
내등수 13	NLP_09조		60.2800%	71.4300%	

[6. 실험에서 실패했으나 유의미한 실험]

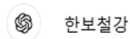
6.1 RAG

가설 : retrieval 앙상블을 통하여 나온 문서에서 generator모델인 GPT-4o를 통하여 “질문에 대한 답을 문서에서 찾아줘”라는 prompt를 query와 함께 주면 EM에 맞는 답을 찾아줄 것이다.

진행방법 :

1. ChatGPT에서 간단한 실험을 통하여 가설이 가능함을 확인함

순천여자고등학교 졸업, 1973년 이화여자대학교를 졸업하고 1975년 제17회 사법시험에 합격하여 판사로 임용되었고 대법원 재판연구관, 수원지법 부장판사, 사법연수원 교수, 특허법원 부장판사 등을 거쳐 능력을 인정받았다. 2003년 최종영 대법원장의 지명으로 헌법재판소 재판관을 역임하였다. WnWn경제민주화위원회(위원장 장하성)이 소액 주주들을 대표해 한보철강 부실대출에 책임이 있는 이철수 전 제일은행장 등 임원 4명을 상대로 제기한 손해배상청구소송에서 서울지방법원 민사합의17부는 1998년 7월 24일에 “한보철강에 부실 대출하여 은행에 막대한 손해를 끼친 점이 인정된다”며 “원고가 배상을 청구한 400억원 전액을 은행에 배상하라”고 하면서 부실 경영인에 대한 최초의 배상 판결을 했다. WnWn2004년 10월 신행정수도의건설을위한특별조치법 위반 확인 소송에서 9인의 재판관 중 유일하게 각하 견해를 내었다. 소수의견에서 전효숙 재판관은 다수견해의 문제점을 지적하면서 관습헌법 법리를 부정하였다. 전효숙 재판관은 서울대학교 근대법학교육 백주년 기념관에서 열린 강연에서, 국회가 고도의 정치적인 사안을 정치로 풀기보다는 헌법재판소에 무조건 맡겨서 해결하려는 자세는 헌법재판소에게 부담스럽다며 소회를 밝힌 바 있다. ==> 해당 내용에서 “처음으로 부실 경영인에 대한 보상 선고를 받은 회사는?” 이 질문에 대한 답을 찾아줘. 다만 알려줘



2. langChain, 허깅페이스 API(토큰화 및 임베딩모델) 및 오픈AI API(Generator모델)를 통하여 RAG를 설계함.

중단이유 :

프로젝트 마지막날에 허깅페이스 “bert-base-multilingual-cased” 토큰라이저의 아웃풋과 마지막 답변을 생성하는 과정에서의 generator모델의 입력 형태가 지속적으로 충돌하여 진행을 중단하였음

```
TypeError: TextEncodeInput must be Union[TextInputSequence, Tuple[InputSequence, InputSequence]]
```


6.2 PostProcessing with Llama

결과 파일 중 nbest_predictions.json 파일에서 1순위로 뽑혀진 context와 question을 함께 Llama 모델에게 prompt로 넣어주며 answer를 새로 예측하도록 실험

-> 시간적 제한으로, 리더보드 제출은 하지 못하고 끝남

6.3 Back Translation with Pororo

Pororo 라이브러리를 활용한 역번역으로 데이터의 유의미한 증강 및 모델 성능 향상 기대

-> 시간이 한정적임에 따라, 다음과 같은 문제 정보를 확인 후 다음 STEP으로 넘어감

INFO: pip is looking at multiple versions of packaging to determine which version is compatible with other requirements. This could take a while. Using cached sacrebleu-2.4.0-py3-none-any.whl (106 kB) **INFO: This is taking longer than usual. You might need to provide the dependency resolver with stricter constraints to reduce runtime. If you want to abort this run, you can press Ctrl + C to do so. To improve how pip performs, tell us what happened here: <https://pip.pypa.io/surveys/backtracking>**

[7. References]

- https://huggingface.co/docs/transformers/ko/tasks/question_answering
- <https://huggingface.co/tasks/question-answering>
- [\[2404.13081\] SuRe: Summarizing Retrievals using Answer Candidates for Open-domain QA of LLMs \(arxiv.org\)](#)
- [\[2403.08319\] Knowledge Conflicts for LLMs: A Survey \(arxiv.org\)](#)
- [\[2402.11163\] KG-Agent: An Efficient Autonomous Agent Framework for Complex Reasoning over Knowledge Graph \(arxiv.org\)](#)
- [\[2402.07927\] A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications \(arxiv.org\)](#)
- [\[2404.19543\] RAG and RAU: A Survey on Retrieval-Augmented Language Model in Natural Language Processing \(arxiv.org\)](#)
- <https://sbert.net/>
- <https://wikidocs.net/book/14314>
- <https://arxiv.org/abs/2312.10997>
- <https://arxiv.org/abs/2401.15884>

개인 회고

RAG based Open-Domain Question Answering (ODQA with RAG)

TEAM : 아기더키들(NLP-09조)



MEMBER : 박동혁_T7335

곽희준_T7308

김정은_T7325

한동훈_T7440

[박동혁_T7335]

■ 1. 나는 내 목표 달성을 위해 무엇을 했는가?

이번 프로젝트에서 본인의 파트는 **Langchain**을 이용하여 **RAG**를 구현해보는 것이었습니다. 처음으로 시도해 보는 점과 시간이 넉넉하지 않다는 점을 고려하여 먼저 다뤄본 지인에게 조언을 구해가며 진행을 했습니다. 그 과정에서 정보를 찾는 법, 오류를 대처하는 법, 새로운 기술을 받아들이는 법, 등을 배우게 되었습니다.

■ 2. 나는 어떤 방식으로 오류를 개선하였는가?

처음에 **WIKI**데이터에서 토큰화를 진행시키는 과정에서 인코딩 오류가 지속적으로 나는 것이 발견되었습니다. 여러 시도 중에 이 토큰나이저의 특징이 프로젝트에서 사용한 것과 맞지 않음이 발견되었습니다. 기존에 사용한 **kiwipiepy**는 한국어 형태소 토큰나이징을 시키는 것에만 기능을 하고 타국어에 대해서는 토큰화를 시키는 못하는 것이었습니다. 이에 허깅페이스에서 다국어를 지원하는 **bert**모델의 토큰나이저를 사용하여 해결했습니다.

■ 3. 아쉬운점은 무엇인가?

- 지식이 부족한 상태에서 **langchain**을 사용하여 **RAG** 구현을 시도하였고 끝 마무리를 못한 점
- 건강관리에 실패하여 체력회복에 일주일이란 시간을 허비한점
- 오류를 해결하는 것에 있어서 능숙하게 못해낸 점
- 마음이 급하여 중간 기록을 허술하게 한 점.

■ 4. 마무리(후기)

이번 프로젝트는 본인에게 건강, 본인의 실력,등 큰 벽을 만나 솔직히 좌절을 많이 했던 시간이었다. 문제는 계속 발생을 하는데 실력이 부족함을 느끼며 시간에 쫓겨서 조금만 마음에 쉽게 해결할 수 있는 부분들도 어렵게 돌아가는 방법들을 선택하였다. 이 중 가장 큰 문제는 어설프었다. 어설프게 아는 지식들이 일의 진행을 느리게 한 것이다. 이런 사항들을 고려하여 다음 프로젝트에서는 좀 더 효율적인 시간관리로 건강과 진행도를 챙기고, 제대로 아는 지식이 아닌 것들은 해결해 가며 진행하겠다.

[곽희준_T7308]

1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

저는 체계적이고 논리적인 실험을 통해 합리적인 결과를 도출하는 경험을 쌓기 위해, 데이터를 분석하고 모델을 개선하는 과정에서 각 단계마다 명확한 목표와 계획을 수립했습니다. 실험의 가정을 구체화하고, 검증할 방법론을 설정하여 신뢰성을 높이고자 했습니다. 데이터 전처리에서는 이상치 확인과 중복 문서 제거, Context에 Title 추가, 긴 문서의 Chunking을 통해 데이터 품질과 모델 입력을 개선했습니다. 각 실험의 결과를 문서화해 다음 단계에 활용할 수 있도록 했으며, 팀원과 협업하여 피드백을 적극 반영했습니다.

2. 나는 어떤 방식으로 모델을 개선했는가?

모델을 개선하기 위해, 긴 문서를 Token max seq length에 맞춰 chunk로 나누고 각 chunk의 유사도를 계산한 후 mean 또는 max pooling을 통해 결합했습니다. 두 pooling 방식 모두 Retrieval Accuracy가 상승했습니다. 또한, Wikipedia 문서의 모든 텍스트와 title을 함께 입력해 query와의 유사도 측정을 개선했습니다. 마지막으로, 여러 Dense Retrieval과 Reranker 모델을 실험하고, 최종적으로 Sparse Retrieval의 Top k 문서를 Reranker 모델로 재정렬하여 Top k 내 정답 문서 포함 확률이 크게 상승하는 성과를 얻었습니다.

3. 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

Reranker 모델을 적용함으로써 기존 MRC 모델의 Exact Match 결과가 크게 개선되었습니다. 또 여러가지 모델을 테스트 해보면서 Sentence Transformer 모델들을 더욱 깊이 이해하고 논리적 가설을 세워 실험을 진행할 수 있었으며, 결과의 타당성을 확고히 다질 수 있었습니다.

4. 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

실험 중간 결과를 보고서 형태로 정리하고 팀원들과 공유하여 실험 내용을 명확히 전달하고 반복 실험 시간 낭비를 줄일 수 있었습니다. 또한, Github Pull Request를 적극 활용해 Retrieval 결과 분석 시각화 등 추가 모듈을 더해 팀원들이 코드에 쉽게 접근하고 활용할 수 있게 했습니다.

5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

이번 프로젝트에서는 시간과 인력의 부족으로 다양한 파라미터 실험과 여러 임베딩 방식의 실험이 미완성으로 남은 점이 아쉬웠습니다. 또한, 프로젝트를 단일 사이클로 진행하면서 각 과정에서 얻은 교훈을 다른 과정에 적용하기 어려운 한계가 있었습니다.

6. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

시간과 인력을 효율적으로 관리하여 실험 미완성을 방지하고, Agile 방식의 스프린트를 도입해 각 과정에서 피드백을 반영하도록 할 계획입니다. 다양한 파라미터와 임베딩 방식을 폭넓게 실험하고, 이를 문서화해 후속 프로젝트에서 효율성을 높이고 성능 개선을 이끌어내고자 합니다.

[김정은_T7325]

1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

- 학습목표 : 다양한 지식을 실전에 적용해보는 경험 쌓기 + “Modularization + Docstring”

순서	시도한 것
STEP-1) EDA	- 통계 분석 - 문장데이터 분석
STEP-2) Augmentation	- Question Paraphrasing - Distast Supervision
STEP-3) Modeling	1. Sparse Retrieval - TFIDF - BM25L - BM25Plus - ATIREBM25 2. Retrieval Ensemble - BM25Ensemble 3. Dense Retrieval - DprBert 4. Hybrid Search - LogisticBM25EnsembleDprBert 5. MRC - klue/bert-base - klue/roberta-large - HANTAEEK/klue-roberta-large-korquad-v1-qa-finetuned - monologg/koelectra-base-v3-finetuned-korquad
STEP-4) Post-processing	- Llama
STEP-5) Ensemble	- Soft Voting - Hard Voting

2. 나는 어떤 방식으로 모델을 개선했는가?

가설	방식
다양한 Tokenizer 와 Architecture 의 앙상블이 Retrieval 성능 향상에 크게 기여한다.	- 최종적으로 Retrieval의 앙상블이 최고 성능을 보일 것으로 예상 - 다양한 Tokenizer를 다양한 Architecture에 적용하여 최대한 많은 Retrieval를 구현 - 3개 이상의 Retrieval를 soft-voting으로 앙상블 시도

3. 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

- 다양한 **Category**의 **Model**들을 앙상블한 결과, 최고 성능의 개별 모델 대비 **Accuracy**가 10% 이상 향상된 것을 확인

- 하나의 모델의 성능을 끌어올리기 위해 다양한 기법을 적용하기 보다는, 최대한 다양한 종류의 **Model**를 구축하는 것이 효율적이라는 것을 깨달음

4. 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

새로운 시도	효과
Modularization	<ul style="list-style-type: none"> - 베이스라인 코드에서 직접 Arguments를 수정하던 이전과 달리, 빠르고 체계적인 실험이 가능해짐 - 전체적인 코드 관리에 수월하며, 동작을 세분화함에 따라 코드 수정이 용이해짐
Docstring	<ul style="list-style-type: none"> - 코드에 대한 설명을 자세하게 적어둠으로써, 팀 내 작업 공유가 수월해짐 - 이후 다른 관리자, 작업자 등이 추가가 되어도 빠른 코드 이해가 가능함

5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

아쉬운 점	개선 방향
시간 부족으로 Generation Model 결과 미제출	<ul style="list-style-type: none"> - 혼자서 시도해볼 수 있는 실험은 제한이 있기 때문에, 실험 전 꼼꼼한 리서치를 통해 무의미한 실험을 줄이기 - STEP 별 충분한 스터디를 통해 진행하려는 TO-DO LIST와 완성 기한을 정해두고 이에 맞춰 진행하기

6. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

- 단순히 많은 실험을 시도하기 보다는, 충분한 리서치를 통해 논리적으로 유의미한 실험 위주로 진행하기
- 성능에 대한 원인 분석을 철저히 진행하기
- 최신 기술 동향도 함께 적용해보기

[한동훈_T7440]

■ 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

- 이번 프로젝트에서 학습 목표에 대해 구체적으로 고려해 본 적이 없었다. 서버 세팅을 처음 해보기 때문에 이론도 잘 모르채 해당 방법을 찾기 위해 바로 프로젝트를 시작했고 이후 코드 리팩토링을 위해 의미없이 보낸 시간으로 이론에 대한 이해가 부족한 상태로 프로젝트를 진행했다고 생각한다.

■ 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

- 기존에 환경세팅에 대해 한명이 하면 충분하다고 판단하여 관심을 소홀히 하고 있었는데 서버 세팅과 깃허브 연동을 직접 해봄으로써 기존보다 익숙해 졌으며 비슷한 작업을 시도할 때의 시야가 넓어진 것 같다.
- **Base line**을 모듈화 및 자동화 하고자 코드 리팩토링을 시도해봤다. 아직 **hugging face library**에 대한 이해가 부족한 상태였고 해당 프로젝트도 처음 접했기 때문에 들어간 시간과 노력에 비해 결과를 얻지 못했다. -> **base line**을 구축할 때 스스로 깨닫고 하려는 경향때문에 결과를 못봤다고 생각했고 프로젝트를 진행하면서 중요한 것은 **NLP**에서 배운 이론의 이해를 돕고 다양한 시도를 해서 깨닫는 것인 점과 리팩토링은 현재 부족한 상태에서는 빠르게 좋은 자료를 찾고 적용하는 것이 다양한 방면에서 좋을 것이라는 점에서 이 부분은 깨닫는 것보다 좋은 자료 찾는 것에 비중을 두고 시간을 절약할 것이다.

■ 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- **reader**에서 여러개의 **context**가 올 때 공백을 구분자로 **join**하는 것은 알고 있었는데 이를 한번에 모델에 넣지 말고 하나하나를 독립적으로 모델을 돌리는 시도를 했었다면 성능향상을 봤을 것이었다. 하지만 시간이 부족한 점과 무의식적으로 꺼리는 부분에서 이를 시도하지 못해 성능향상을 못본 것이 아쉬웠다. 만약 시도했다면 **retrieval**에서 91퍼센트의 성능이 나온 **top k = 1** 방법에서 95%가 나온 **top k = 2**로 설정하면서도 더 높은 정확도를 유지하여 성능향상이 높게 나왔을 것이었다.

■ 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

- **EDA**를 할 때 공통적으로 사용하는 코드들을 개인마다 작성해서 사용했으며 본인이 필요한 순간조차도 매번 코드를 작성하는 것이 시간낭비가 심했기 때문에 **stream lit**을 사용하는 방식으로 **EDA**를 시도하여 시간을 절약할 수 있을 것이다.
- 번거롭거나 귀찮다는 생각이 무의식에 잠재되어 있어 열린 사고를 막고 새로운 방법에 대해 시도해보는 것을 저해하는 것으로 확인된다. 평소 행실을 고치는건 쉽지 않기 때문에 철저한 분석을 통해 확실한 근거가 뒷받침하는 경우만 시도하는 것이 나에게 맞는 방법으로

판단된다. 따라서 결과 우선적인 실험보다 충분히 근거있는 가설을 시도하는 방향으로 실험을 진행할 것이다.

- 노선의 경우 분석한 내용을 적는 것으로는 좋은 기능이라고 생각하지만 프로젝트의 전체적인 진행상황확인과 **task**에 대해서는 기능적으로 그리고 개인적으로는 단순 기록을 하게 된다는 점에서 진행에 큰 도움이 되지 않는다고 판단된다. -> 다음 프로젝트는 노선에는 분석 내용을, 프로젝트의 진행에는 깃 이슈를 사용해서 전체적인 진행상황을 주기적으로 확인하고 **issue**별로 할당하여 확실한 작업을 시도해 볼 예정이다.
- 아직도 기본적으로 모델과 데이터를 다소 정제되어 있다는 고정관념이 있는데 이는 다양한 사고를 막는것으로 느껴져 마치 서버에서 모든 클라이언트를 악성 클라이언트로 간주하여 작업하는 것처럼 나 또한 모든 데이터와 모델의 기본 값을 전부 의심하고 최선인지 고려해보는 것이 다양한 방법을 창출해 낼 것이다.