

수능형 문제 풀이 모델 생성

본 프로젝트에서는 지문과 질문을 읽고 선택지에서 정답을 추론하는 것을 목적으로 한다. 네이버 커넥트재단 부스트캠프 AI Tech 7기 NLP 과정의 일환으로 진행되었으며, Large Language Model (LLM)을 사용한 다양한 방법론을 학습하고, 협업 도구를 활용하여 원활한 의사소통과 작업관리를 경험하였다.

주요 성과

정확도 (Accuracy) 점수 0.7494 달성 (Baseline 0.3954 대비 +0.3540 개선)

내등수 7	NLP_10조		0.7494	45
----------	---------	---	--------	----

접근 방식

LLM을 사용하여 프롬프팅 방법론을 적용하고 사전 학습, RAG 등 추가 지식 학습을 위한 방법론을 적용하였다. 추가로, Scaling law에 따라 높은 파라미터의 모델을 사용하여 추론을 진행하였다.

Retrieval-Augmented Generation

방법 : Wikipedia 문서와 KoreanTextBook 데이터셋을 기반으로한 두 개의 Vector DB를 구축 후 각각 테스트를 진행하였다.

결과 : 동일한 LLM 모델을 기준으로 테스트 데이터셋에 대해 각각 0.6198 (Wikipedia), 0.5379 (KoreanTextBook)의 정확도를 보였다.

Domain-Adaptive Fine tuning

방법 : Aihub 데이터셋과 우리역사넷 크롤링을 통해 데이터를 수집하고 LLM 모델에 사전학습을 하였다. 그 후 기존 학습 데이터셋을 학습시켜 평가(Validation)를 진행하였다.

결과 : 평가 데이터셋 기준으로 정확도 0으로 산출되어 테스트 데이터셋에 대한 평가는 진행하지 않았다.

High Parameter Model

방법 : PEFT-LoRA와 양자화를 통해 32B 모델을 최적화하고 테스트 데이터셋만으로 추론을 진행하였다.

결과 : Few Shot 프롬프트 기반으로 0.7333를 달성하였다.

협업 방식

Github

브랜치 관리 : Git-flow 기반으로 브랜치를 관리해 작업 간 충돌을 최소화하고, 업무 분담을 수월하게 진행하였다.

이슈 및 PR 관리 : 작업 단위별로 이슈를 생성하여 업무 분배 및 코드리뷰를 진행하였다

Notion

프로젝트 진행 상황을 한 눈으로 파악하고 세부 작업 내용을 작성하여 모든 팀원들이 자신이 맡은 작업 외의 내용을 파악할 수 있도록 하였다. 추가로, 제출 횟수를 효율적으로 사용할 수 있도록 코드 버전과 데이터셋 버전을 관리하여 중복되는 제출이 없도록 하였다.

개요

Task 소개

인공지능(AI)의 등장 이후 인간의 사고 능력을 평가하는 각종 시험에서 AI 챗봇이 줄줄이 고득점을 받는 가운데, 올해 대학수학능력시험 국어영역에서 오픈AI의 최신 모델인 o1 프리뷰가 만점에 가까운 점수를 받았다. 이러한 대형 모델들은 한국어에 완벽히 최적화되지 않았지만, 수능에서 높은 성적을 기록했다. 본 프로젝트는 작은 규모의 모델로도 같은 성적을 낼 수 있는지에 대한 도전이다. 우리가 알고 있는 한국어의 특성과 수능 시험의 특징을 바탕으로 수능에 특화된 모델을 구축하고, 정확도(Accuracy)를 기반으로 평가하였다.

팀 구성 및 역할

이름	역할
강경준	EDA(Label 분석, 데이터 유형 분석), 데이터 수집 및 전처리(CLIC Data), Reasoning(Orca)
권지수	데이터 수집(Crawling), 데이터 정제(kiwi)
김재겸	데이터 증강, 데이터 실험(Fine-Tuning), 모델 실험(Fine-Tuning)
박동혁	Prompt-Reasoning(연역적, 귀납적, 단계적 추론), baseline 모듈화
이인설	Domain-Adaptive Fine-Tuning, RAG 구현(Korean Textbooks), baseline 모듈화
이정희	Model 탐색, Prompt 작성(Few-Shot, EN-Prompt), RAG 구현(Wikipedia)

데이터셋 설명

Upstage에서 제공한 수능의 국어, 사회 영역과 비슷한 문제와 KMMLU, MMMLU, KLUE MRC 데이터를 사용하였다. 총 2031개의 학습 데이터와 869개의 평가 데이터로 구성되어 있다. 그 중 KLUE MRC 데이터의 질문과 선지는 지문을 기반으로 GPT-4o-mini를 통해 생성되었다.

분류	구성	샘플 수	컬럼명
Train_Datasets	KMMLU MMMLU(Ko) KLUE MRC	2031	id, paragraph, problems, question_plus - problems(dictionary): 'question', 'choices', 'answer'
Test_Datasets	수능형 문제 KMMLU MMMLU(Ko) KLUE MRC	435 (private)	id, paragraph, problems, question_plus - problems(dictionary): 'question', 'choices', 'answer'
		434 (public)	- 'answer'는 빈 문자열

[표 1] 데이터셋 구성

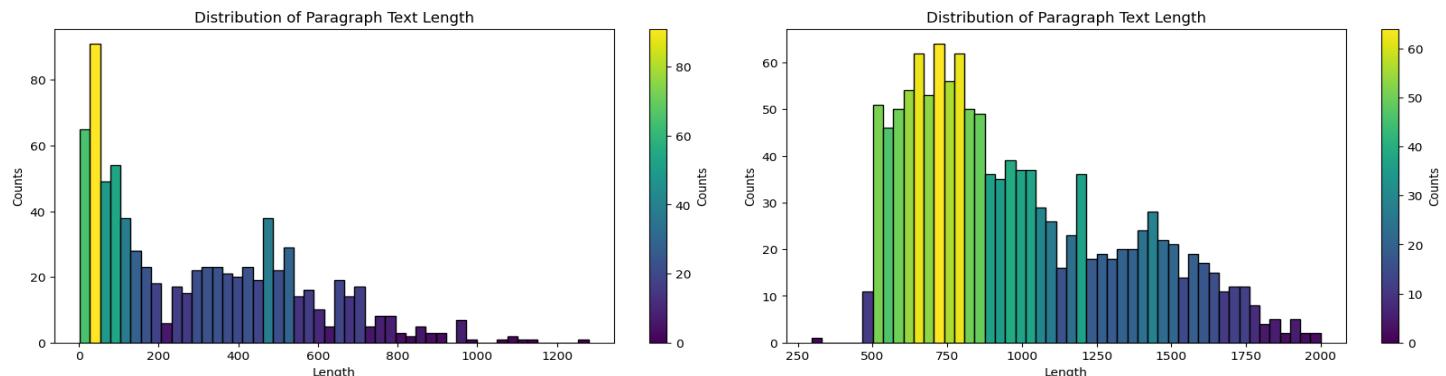
평가방법

평가방법은 전체 데이터에서 모델이 맞게 예측한 데이터로 나누는 정확도(Accuracy)를 사용하였으며, 본 프로젝트에서는 “모델이 맞춘 문제 수 / 전체 문제 수”로 모델을 평가하였다.

데이터 분석 및 전처리

EDA

Paragraph Length Analysis



[그림 1] (left) MMMLU, KMMLU 기반 paragraph length distribution, (right) KLUE MRC 기반 paragraph length distribution

각 문제 별 지문에 대해, 문제 source 별로 text 길이 분포가 다양하고 문제 특징에 차이가 있다. 먼저, MMMLU, KMMLU 기반 문제들은 전반적으로 지문의 길이가 짧고 정답에 대한 직접적인 내용을 포함하지 않아, RAG 혹은 추가적인 pre-training 등을 통한 지식 학습이 필요할 것으로 보인다. 반면, KLUE MRC 기반 문제들은 대체로 길이가 길고, paragraph 내용을 통해 정답을 도출해내는 형태이다.

데이터 전처리

비정상 데이터 제거

Upstage에서 제공한 총 2031개의 학습 데이터에 중복 데이터가 있는지 확인한 결과, 지문과 질문 쌍이 중복되는 데이터가 각 5쌍이 발견되었으며 문제와 선지가 불일치하는 데이터를 확인하였다. 이를 통해 데이터 품질 검수의 필요성을 고려하게 되었고, 전체 학습 데이터에 대해 지문과 질문, 그리고 선택지 자체가 잘못된 문제가 있는지 확인하는 절차를 거쳤다. 그 결과, 중복 데이터 외에도 지문, 질문, 선택지가 잘못 구성되어 매칭이 되지 않는 경우, 답이 틀리거나 여러 개인 경우들이 확인되어, 이를 비정상적인 데이터로 간주하고, 115개의 데이터를 제거해 결과적으로 총 1,916개의 데이터를 분석에 이용하였다.

generation-for-nip-426

지문 : (가)은/는 의병계열과 애국계몽운동 계열의 비밀결사가 모여 결성된 조직으로, 총사령 박상진을 중심으로 독립군 양성을 목적으로 하였다.
문제 : (가)에 대한 설명으로 옳지 않은 것은?

선지 :

- 1) 고려 문종 때에 남경(편호)으로 승격되었다.
- 2) 종루(31), 이현, 칠패 등에서 상업활동이 이루어졌다.
- 3) 정도전은 궁궐 전각()과 도성성문 등의 이름을 지었다.
- 4) 성곽은 거중기 등을 이용하여 약 2년 만에 완성되었다.

정답 : 1

[그림 2] 비정상 데이터의 예시: 지문의 내용으로 보아, (가)는 특정 조직을 뜻하나, 선지는 조직에 대한 내용이 아님

불필요한 단어 제거

문장의 길이를 줄이기 위해, 모델이 문제를 풀 때 필요 없다고 판단한 단어를 제거했다. 제거한 단어는 이메일, url, 전화번호, 기자 정보이다.

띄어쓰기 재배치

정제된 1916개 문제에서도 띄어쓰기가 이상한 데이터가 다수 발견되었고, 이는 모델이 문제를 분석할 때 잘못된 해석을 할 것이라고 판단해 kiwi를 이용하여 띄어쓰기를 재배치하였다.

Accuracy(데이터 전처리 전 → 후)	0.6290 -> 0.6429
-------------------------	------------------

[표 2] 데이터 전처리 전 → 후의 Accuracy 비교

모델 탐색

2~3 Billion Language Model

beomi/gemma-ko-2b

Gemma 모델을 한국어에 최적화한 모델로, 2B라는 작은 파라미터로 학습시간은 총 30분이내가 소요되었다. 하지만, 파라미터가 작은 만큼 모델의 기본 성능 또한 0.4032로 낮은 정확도를 보였다.

meta-llama/Llama-3.2-3B-Instruct

LLaMA 3.2 모델에서 사용자 명령에 반응하도록 특화된 모델(Instruct)로, 명령에 더 정확하게 답변하거나, 특정 작업을 수행하는데 최적화된 모델이다. 하지만, 한국어로 학습되지 않았고 파라미터 또한 높지 않아서 0.3364로 낮은 정확도를 보였다.

7~9 Billion Language Model

Scaling law[1]에 따라 3B 이하의 모델보다 더 높은 파라미터를 가진 모델로 생성한다면 더 좋은 결과가 나올 것이라고 가정하였다. 따라서 양자화를 하지 않고 제한된 서버 자원으로 학습이 가능한 7~9B 모델을 탐색하였다. Gemma, EXAONE, LLaMA 모델은 한국어와 Instruct로 학습된 모델이며, Aya-expansie와 Qwen 모델의 경우 다국어와 Instruct로 학습된 모델이다.

같은 환경과 프롬프트를 사용하여 fine-tuning 후 추론한 결과, 테스트 데이터셋(test)에서 0.6452를 기록한 EXAONE과 Qwen 모델이 탐색한 모델 중 가장 정확도가 높았다. 하지만 EXAONE(4096)의 경우 Qwen(32,768) 대비 짧은 컨텍스트 길이를 가지고 있어 학습속도가 더 빠르고, 한국어에 더 최적화된 토크나이저를 갖고 있어서 EXAONE을 Base Model로 선정하였다.

Model	Accuracy(val)	Accuracy(test)
beomi/gemma-ko-2b	-	0.4032
meta-llama/Llama-3.2-3B-Instruct	0.4729	0.3364
rtzr/ko-gemma-2-9b-it	0.5583	0.5783
LGAI-EXAONE/EXAONE-3.0-7.8B-Instruct	0.4843	0.6452
sh2orc/Llama-3.1-Korean-8B-Instruct	0.5885	0.6106
CohereForAI/ay-a-expansie-8b	0.2760	0.6014
maywell/Qwen2-7B-Multilingual-RP	0.8852	0.6452

[표 3] 모델 탐색 결과

Prompting

본 장에서는 주어진 데이터를 효율적으로 사용하기 위하여 3가지 가이드라인(Reasoning, En-promprtng, FewShot)을 모델에 제시한다.

Reasoning

< COT 기반 >

현재의 LLM은 지문, 질문, 선지 각각의 섹션에 대한 지식적 이해도가 뛰어나다. 반면에, '지문 및 질문'과 같이 다수의 섹션들의 관계성을 추론하는 능력이 떨어진다. 이러한 상황을 보완하는 방법으로 언어학적 접근법(귀납적, 연역적)과 CoT기법을 응용하여 섹션들 간의 관계성을 강화하였다.

< LLM 기반 Reasoning Generation >

LLM은 뛰어난 성능을 보여주지만, 컴퓨팅 리소스의 제한으로 인해, 기업이 아닌 개인이 LLM을 사용하는 데는 어려움이 있다. 따라서, 비교적 작은 모델을 통해 거대 모델의 추론 능력을 흉내내기 위해, [2]의 방식을 참고했다. 논문에서 제시하는 방식은 거대 모델에서 사용자의 질문에 대한 상세한 정답 도출 과정을 생성하고, 정답 도출 과정에 대한 text를 상대적으로 작은 모델을 통해 학습하는 방식이다. 이를 우리 데이터에 적용하여, 거대 모델로 지문, 문제, 보기를 제시하여 상세한 문제 풀이 과정을 생성했다. 이후, 비교적 작은 모델에서 해당 reasoning text를 학습하는 방식으로 fine-tuning을 진행했다. 이 과정에서, 거대 모델은 openAI의 GPT 4o-mini를 활용했고, fine-tuning에는 huggingface의 LGAI-EXAONE/EXAONE-3.0-7.8B-Instruct 모델을 활용했다.

EN-Prompting

한국어 기반의 Reasoning 결과, 교착어와 복잡한 어순 등의 언어적 특성에 의하여 모델의 성능의 한계가 있음이 확인된다. 사전 학습 모델이 한국어에 비하여 제약 조건이 적은 영어 데이터를 많이 사용한 사실을 기반으로 reasoning 부분을 영어로 바꾼 결과, 기존의 한국어 대비 +0.03의 유의미한 상승을 보였다.

FewShot

현실에서 시험지를 보면 일정 지문, 질문, 선지, 답을 내는 구간이 확실하게 나눠져있다. 하지만 모델의 프롬프트에서는 모든 것을 하나의 텍스트 덩어리로 인식을 하는 경향이 나타난다. 이는 각 섹션 간의 관계성을 추론하기에 방해 요인이 농후하다는 결론이 도출되었다.

해당 문제를 해결하기 위한 방법으로, 지문, 질문, 선지, 답 순서의 동일한 형태가 유지되는 n가지의 예시를 추가하였다. 이는 모델이 시험지와 같은 형태를 인식하는 것과 동일한 효과를 나타낸다.

<pre># CoT (System) system_prompt = """ 시험 문제를 푸는 똑똑한 학생으로서 다음 문제의 답을 찾으세요. """ # CoT (User) Messages = """ 1. 지문을 읽고 질문을 확인합니다. 2. 질문을 분석하여 무엇을 묻는지 파악합니다. 3. 선택지를 하나씩 평가하고, 지문과 질문과 가장 잘 맞는 선택지를 결정합니다. 4. 선택지를 결정한 이유를 설명하고 정답을 출력하세요. 5. 정답은 무조건 1개만 존재합니다. """ </pre>	<pre># CoT (System) system_prompt = """ As a smart student solving exam questions, find the correct answer based on the given passage and question. """ # CoT (User) Messages = """ 1. Read the passage and review the question 2. Analyze the question to understand what 3. Evaluate each option and decide which one is correct 4. Explain the reason for your choice and provide the answer 5. There is only one correct answer. """ </pre>	<pre>SYSTEM_PROMPT: "당신은 전교 1등 학생입니다. 지문을 읽고 질문에 해당하는 선택지를 찾으세요." PROMPT_NO_QUESTION_PLUS: """ ### 예제 1 지문: 학교 심리학자인 Mr.Thomas 씨는 특수 교육 교사인 Ms.Ri 질문: 이러한 협업은 어떤 상담 모델의 예시입니까? 선택지: 1 - 체계 2 - 자원 3 - 3 측(triadic) 4 - 적응형 학습 환경 정답: 3 ### 예제 2 """ < reasoning 방식을 적용한 prompt template ></pre>
< reasoning 방식을 적용한 prompt template >	< 영어로 작성한 prompt template >	< few-shot을 적용한 prompt template >

[그림 3] 실험에 사용된 프롬프트

데이터 증강

데이터 증강

Finetuning을 위한 데이터 증강을 위해서 기존 KMMLU, MMMLU, KLUE-MRC이외에 CLICk(Cultural and Linguistic Intelligence in Korean)와 한국사 도메인 문제에 대한 성능을 올리기 위해 7차 교육과정 국사 교과서 데이터를 증강에 추가로 사용했다.

데이터 설명

CLICk(Cultural and Linguistic Intelligence in Korean)은 한국어 대형 언어 모델의 문화적 및 언어적 지능을 평가하기 위해 개발된 벤치마크 데이터셋이다. 이 데이터셋은 총 1,995개의 질문-답변(QA) 쌍으로 구성되어 있으며, 언어와 문화라는 두 가지 주요 범주 아래 11개의 세부 카테고리로 분류되어 있다. CLICk은 기존의 한국어 벤치마크 데이터셋이 영어 데이터를 번역하여 사용함으로써 발생하는 문화적 맥락의 차이를 보완하고자 설계되었다.

이를 위해 공식 한국 시험과 교과서에서 데이터를 수집하고, 각 질문마다 정확한 답변을 위해 필요한 문화적 및 언어적 지식을 세분화하여 주석을 달았다. 7차 교육과정은 대한민국 교육부가 발족한 이래 일곱 번째로 개정된 교육과정이다. 국민 공통 기본 교육과정과 고등학교 선택 중심 교육과정으로 구성되는 것이 특징이며, 교육내용과 방법을 진로와 적성에 맞게 다양화하고 교육내용의 양과 수준을 적정화하여 심도 있는 학습을 할 수 있도록 함을 방침으로 하여 구성되었다.

데이터 정제

수집한 CLICk data를 기존의 데이터와 형태를 맞추기 위해 몇 가지 처리가 요구됐다. 먼저, 단순 개념 문제의 경우 지문이 제공되지 않는 경우가 있었다. 기존의 데이터에도 단순 개념 문제가 있었고, 이러한 경우 기존의 데이터에서는 지문의 의미가 크지 않은 경우나 지문과 질문이 같은 경우가 많았다. 이러한 특징을 반영해, CLICk data에서도 지문이 누락된 경우, 질문으로 지문을 채워 넣었다. 다음으로, 데이터 중 일부가 질문과 보기가 결합된 형태로 제공되어 있어, 이를 분리하는 작업 과정을 거쳤다. 해당 문제들의 패턴을 파악하고, 정규표현식을 활용하여 분리했다. 이후, 지문이 따로 있는 문제의 경우 이를 지문의 뒷 부분에 덧붙여 주었고, 지문이 없는 문제의 경우 해당 보기의 지문으로 설정했다.

데이터 증강 방법

KMMLU, MMMLU, KLUE-MRC 데이터의 paragraph와 CLICk data, 국사 교과서의 context를 이용하여 gpt-4o-mini 모델로 수능형 문제를 생성하는 방식으로 증강에 사용했다. 프롬프트 구성 시 모델에게 본인이 '수능 출제의원'임을 알리고 traindata 상의 수능형 5지선다 문제로 fewshot을 제공했다. 제공받은 paragraph에 해당하는 question, choices, answer를 생성하도록 했다. 이때 생성된 데이터의 answer가 어느 한 번호로 치우치는 것을 방지하기 위해 각 answer를 골고루 제공했다.

데이터 증강

데이터 증강

원본 데이터

paragraph	나는 삼한(三韓)산천의 음덕을 입어 대업을 이루었다.(가)는/은 수덕(水德)이 순조로워 우리나라 지맥의 뿌리가 되니 대업을 만 대에 전할 땅이다. 왕은 춘하추동 네 계절의 중간 달에 그 곳에 가 100일 이상 머물러서 나라를 안녕케하라. – 고려사 –
question	(가)지역에 대한 설명으로 옳은 것은?
choices	<ol style="list-style-type: none"> 1. 이곳에 대장 도감을 설치하여 재조 대장경을 만들었다. 2. 지눌이 이 곳에서 수 선사 결사 운동을 펼쳤다. 3. '망이 .망소이가 이 곳에서 봉기하였다. 4. 몽골이 이 곳에 동녕부를 두었다.
answer	4

[표 4] 원본 데이터 예시

증강 데이터

paragraph	나는 삼한(三韓)산천의 음덕을 입어 대업을 이루었다.(가)는/은 수덕(水德)이 순조로워 우리나라 지맥의 뿌리가 되니 대업을 만 대에 전할 땅이다. 왕은 춘하추동 네 계절의 중간 달에 그 곳에 가 100일 이상 머물러서 나라를 안녕케하라. – 고려사 –
question	위의 인용문에서 언급된 '대업'을 이루기 위해 왕이 해야 할 행동은 무엇인가?
choices	<ol style="list-style-type: none"> 1. 나라를 정복하다. 2. 춘하추동 네 계절의 중간 달에 특정 장소에 100일 이상 머물다. 3. 백성을 다스리다. 4. 외교 관계를 강화하다. 5. 군사 훈련을 실시하다.
answer	2

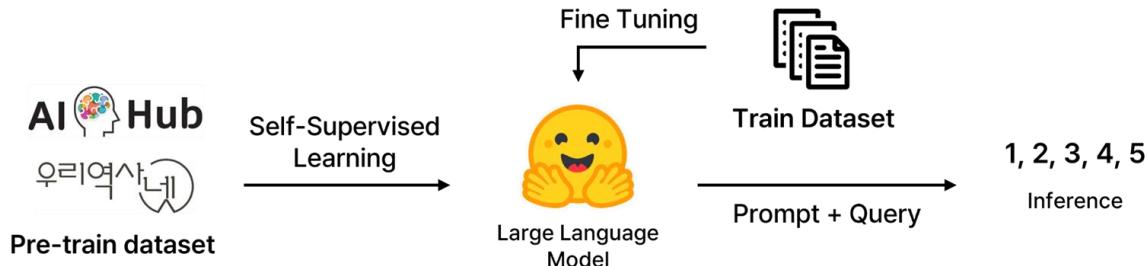
[표 5] 증강 데이터 예시

Accuracy(데이터 증강 전 → 후)	0.6322 → 0.6460 (0.0138 상승)
------------------------	-----------------------------

[표 6] 증강 데이터 기반 실험 결과

Domain-Adaptive Fine-Tuning with LoRA

학습 데이터에는 언어 이해 능력이 바탕이 되는 국어 비문학 지문뿐만 아니라 한국사, 한국 경제 등과 같은 사전 지식이 요구되는 문제들이 있어, 특정 도메인 지식에 대한 학습이 추가적으로 필요할 것이라고 판단하였다. 이에 기존 언어 능력 및 지식을 유지하면서 해당 도메인 지식을 추가 학습시키고자 Continual Learning 방식을 시도하였으나, Catastrophic Forgetting 방지를 위한 추가적인 기술적 지식 및 구현 경험이 부족했고, 모든 파라미터를 학습시키기 위한 자원 제약 (GPU 메모리 한계 등)으로 인해 Domain Adaptation 방식으로 방향을 전환했다.



[그림 4] Pre-train workflow

데이터 구성

도메인과 관련된 추가적인 지식을 학습하기 위해서는 해당 주제의 텍스트 데이터가 필요했다. AI Hub에서 제공하는 문장 유형(추론, 예측 등) 판단 데이터는 다양한 카테고리의 문장으로 구성되어 있고, 이 중 '역사'와 '금융'과 관련된 2,175개의 텍스트 데이터를 이용해 학습 데이터를 구성하였다.

학습 방법

모델의 효율성과 구현 용이성을 고려하여 LoRA(Low-Rank Adaptation)를 활용해 특정 도메인에 최적화된 경량 학습을 수행했다. 도메인 관련 텍스트가 주어지면 이를 기반으로 다음 단어를 예측하는 방식으로 학습을 진행하였다.

평가 방법

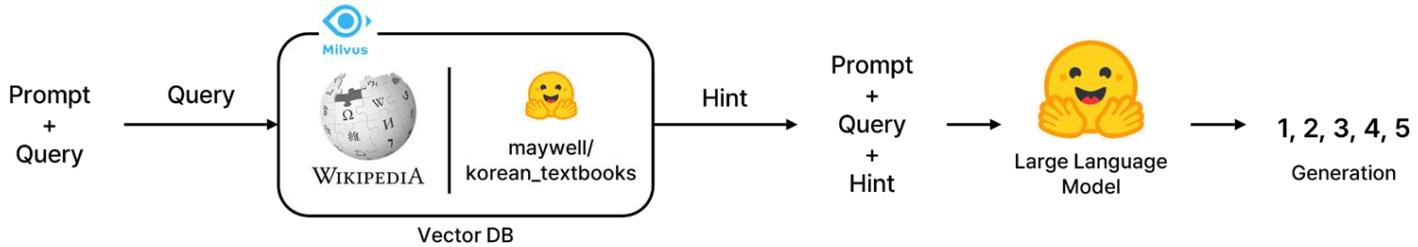
추가적인 학습이 완료된 후에도 기존 언어 능력을 잃지 않으면서, 도메인 관련 지식이 제대로 학습이 되었는지 확인하기 위해 도메인 관련 QA 데이터를 활용하였다. Hugging Face에서 제공하는 CLICk 데이터 중 'Kedu_history'와 'KIIP_economy' 부분을 이용해 학습시킨 지식과 가장 유사한 분야의 QA 데이터를 선정하였다.

결과

기존 모델은 정확도 0.3을 달성하였지만, 학습 후 정확도가 0으로 문제 풀이 성능이 더 떨어짐과 동시에 기존 언어 능력도 저하됨을 확인하였다. 이는 학습 시 사용한 모델링 방식이 평가 방식인 QA Task와 직접적으로 연결되지 않아 발생한 문제로 판단하였다. Catastrophic Forgetting을 방지하는 기술을 적용함과 동시에 자원 제약을 극복하기 위해 모델의 일부 파라미터만 학습하는 방식으로 Continual Learning을 적용했다면 유의미한 성과를 얻을 수 있었을 것이라고 판단된다. 여기에 Quantization을 추가로 적용했다면, 자원 효율성을 더욱 극대화하면서 성능을 개선할 수 있었을 것으로 보인다.

RAG(Retrieval-Augmented Generation)

수능 문제의 경우, 주어진 지문만으로 해결하기 어려운 경우가 많아 문제 풀이에 필요한 추가적인 배경 지식을 보완할 수 있는 RAG 구조를 설계하였다. 이를 위해 BGE-m3-ko 모델을 이용하여 임베딩 벡터를 생성하여 두 가지 방식으로 DB를 구축하고 각각의 접근법을 통해 RAG 성능을 비교하였다.



[그림 5] RAG workflow

[그림 5]은 RAG의 workflow이다. 구축된 지식 정보 Vector DB에서 주어진 Query와 Cosine 유사도가 제일 높은 문서를 Hint로 제공한다. 학습 데이터 중 랜덤으로 3개의 데이터를 Few-Shot으로 활용해 제공된 Hint를 추가하여 LLM의 새로운 입력으로 사용해 추론만 진행하여 정답을 도출하였다.

Wikipedia 기반 RAG

다양한 분야의 지식을 포함하고 있는 Wikipedia 문서를 2~4 문장으로 나눈 Cohere/wikipedia-22-12-ko-embeddings 데이터셋을 활용해 약 123만 개의 데이터를 기반으로 DB를 구축하였다. Hint로 제공되는 문서의 길이가 짧기 때문에 모든 문제에 관련 문서를 제공하였다.

그 결과, 데이터 전처리 및 증강한 결과(0.6460)보다 0.0262 감소된 0.6198의 정확도를 보였다.

Korean Textbooks 기반 RAG

Korean Textbooks의 사회 및 한국사 데이터를 활용해 약 120만 개의 데이터로 DB를 구축하고, 각 문제의 정보를 토대로 요약 모델(eenzeenee/t5-base-korean-summarization)을 이용해 검색 중심의 Query를 재작성하였다. 사회 영역이라고 판단되는 문제들(지문 길이 800자 이하)에게만 관련 문서를 제공하였으며, 상위 3개 문서를 Hint로 제공해 Hard voting 방식으로 정답을 도출하였다.

그러나 상위 첫 번째 문서만 사용한 경우와 3개 문서를 사용한 경우의 정확도가 동일(0.5379)하여, 문서 간 정답이 엇갈리는 현상이 있었으며 이전 방식보다 성능이 떨어짐을 확인하였다.

결과

각 DB를 서로 다른 방식으로 구현하여 정확한 비교는 어려웠으나, 성능 하락의 원인을 다음과 같이 판단하였다.

1. Query 구성
2. DB의 문서 품질
3. Retrieval 성능

Query 방식에 따라 추출 문서의 질이 달라지며, Korean Textbooks는 대화형 텍스트로 구성되어 Wikipedia에 비해 정보 전달력이 낮았을 가능성이 있다.

Wikipedia DB를 이용할 때, 문제에 대한 정보를 요약해 재작성한 Query를 사용하면 성능이 향상될 것으로 보인다.

또한 Retrieval 성능이 RAG에 중요한 영향을 미치므로, Hybrid 방식이나 Re-ranker를 활용하면 정확한 Hint 추출과 성능 향상이 가능할 것으로 보인다.

결과

High-Parameter Model 시도

서버의 하드웨어적 한계를 벗어나 양자화된 32B 모델(unslot/Qwen2.5-32B-Instruct-bnb-4bit)을 사용할 수 있도록 unslot package를 사용해 메모리 사용량을 감소시켰다. 이에 따라 few-shot 기반으로 추론만 하여 단일 모델 테스트 결과 중, 0.7696라는 가장 높은 정확도를 보였다.

양상블 결과

No	Model	Dataset	Acc (middle)	Acc (Final)	Note
1	unslot/Qwen2.5-32B-Instruct-bnb-4bit	x	0.7696	0.7517	Only Inference Few shot(2)
2	unslot/Qwen2.5-32B-Instruct-bnb-4bit	v0.1.6	0.7650	0.7310	Fine-Tuning
3	unslot/Qwen2.5-32B-Instruct-bnb-4bit	x	0.7581	0.7333	Only Inference Few shot(3)
4	LGAI-EXAONE/EXAONE-3.0-7.8B-Instruct	v0.1.6 + click + aug	0.6498	0.6322	.
5	LGAI-EXAONE/EXAONE-3.0-7.8B-Instruct	v0.1.6	0.6452	0.5977	EN-prompt
6	beomi/Qwen2.5-7B-Instruct-kowiki-qa-context	v4.0	0.6452	0.6046	.
7	ensemble (1,2,4,5,6)	.	0.7719	0.7494	.
8	ensemble (1,3,4,5,6)	.	0.7673	0.7517	.
9	ensemble (1,2,3,4,5,6)	.	0.7719	0.7448	.

[표 7] 양상블 결과

[표 7]는 여러 실험 결과 상위 6개의 결과(1번~6번)와 해당 결과들을 조합하여 양상블(7번~9번)을 한 결과이다. 상위 6개의 실험 결과를 통하여 동일한 실험 조건 아래에서 작은 모델 대비 큰 모델인 Qwen2.5 32B가 월등히 좋은 성능을 보였고, 해당 모델에 파인 튜닝 없이 2-shot으로만 진행한 실험이 파인 튜닝한 모델 대비 +0.02(Final ACC 기준)로 가장 좋은 결과를 보인다.

양상블 (1,2,3,4,5,6)은 상위 6개의 결과 모델을 Hard Voting 양상블을 하였고, 양상블 (1,2,4,5,6)은 mid ACC기준 32B중 가장 낮은 수치(0.7581)를 기록한 3번 실험을 제하고 진행하였다. 양상블 (1,3,4,5,6)의 경우 32B 모델 중, 파인튜닝의 실험을 제외한 양상블로 Middle, Final 각각이 0.7673, 0.7517기록하였다. 타 양상블에 비하여 middle score에서는 가장 낮았지만 Final score에서는 가장 좋은 성적을 거둔 것을 볼 수 있다.

결론

2~3B 모델에서 7~9B 모델까지 테스트를 하였고 RAG나 데이터 증강 그리고 여러가지 프롬프팅 기법을 적용하였으나, 성능 상승 효과를 보지는 못하였다. 마지막으로, LLM에서 데이터의 수보다 모델의 사이즈가 크면 성능 또한 향상된다는 Scaling law에 따라 양자화된 32B 모델까지 사용하여 단순 추론만 진행하였을 때, 이전 시도들보다 약 12%가 향상되었다. 이를 통해 LLM은 Scaling law를 크게 영향을 받는 것을 알 수 있었다. 하지만 7~9B의 학습 시간은 2시간 이내였던 것에 비해 32B 모델의 경우 10시간 이상 소요되었다. 따라서, 향후 프로젝트에서 32B 모델을 10B 이내의 모델로 Knowledge Distillation하거나 학습 시간을 단축시킬 수 있는 방법들(unslot package, flash attention 등) 활용하고자 한다.

팀회고

프로젝트 팀 목표

- 아이디어를 적극적으로 공유하며, 모델 score에 집중하지 않고 더욱 건설적인 실험을 목표로 하며 프로젝트를 진행하는 것
- 협업 툴을 적극적으로 활용하여, 회의를 통한 활발한 의견 공유, 프로젝트에 대한 개별 진행 상황 공유, 프로젝트 진행에 대한 적절한 기록 및 공유하고 있는 컴퓨팅 리소스에 대한 효율적인 활용 등 협업 역량을 기르는 것

좋았던 점과 배운 점

- 신생 팀이라고 생각하기 무안할 정도로 프로젝트 진행하며 아이디어나 실험 결과를 공유하는 데에 있어 굉장히 원활한 소통이 이루어진 점이 좋았다. 모든 팀원들이 적극적으로 의견을 제시하고 서로의 정보를 공유하며, 팀워크를 빠르게 키울 수 있었다.
- 협업 툴로 GitHub를 활용함에 있어, Git Flow, Issue, PR 등을 적극 활용할 수 있었다. 이를 통해, 태스크와 repository를 체계적으로 관리할 수 있었고, 무엇보다 git과 github이라는 툴에 대해 더욱 익숙해지고 높은 이해도를 갖게 되는 계기가 됐다.
- 실험에 있어서 많이 좌절했지만, 끝까지 흔들리지 않고 고민과 실험을 반복했다는 점이 좋았다. 비록, 실험이 성공하지는 않더라도, 우리만의 실험을 진행하고 성능을 올리기 위한 고민을 지속한 것이 역량을 키우는데 많이 도움이 됐을 것이라 생각한다.

아쉬운 점

- 전반적인 의사결정 시간이 조금 길었다고 생각한다. 다양하게 의견을 공유하고 회의를 진행하는 것은 좋은 결과물을 만드는데 분명히 도움되지만, 비교적 짧은 시간 내에 결과물을 만들어야 하는 프로젝트의 상황에는 부합하지 않는 것으로 보인다.
- 데이터를 검수하는 데에 너무 많은 시간을 사용한 점이 아쉬웠다. 기초적인 처리를 빠르게 진행하고, 더 많은 실험에 시간을 투자할 필요가 있었을 것이라 생각한다.
- RAG 성능을 더 고도화하거나, 큰 LLM을 파인튜닝해보지 못한 것이 아쉬웠다.
- 각자 task를 나눠 진행했는데도 하다보니 겹치는 부분이 발생하게 되었다. task에 대한 이해도가 부족하여 필요한 작업들을 적절히 계획하고 분배하는 일이 적절히 이루어지지 않았고, 따라서 각자가 진행하는 일이 겹치거나 반복적으로 진행하게 되는 경우가 발생했다.
- 프로젝트 수행 절차를 정확히 결정하고, Task의 순서를 명확히 결정하고 프로젝트를 진행해야한다는 것을 깨달았다. 처음 데이터, 모델을 결정할 때에는 다같이 같은 작업을 하는게 효과적이었을 것 같다는 생각이 들었다.

향후 발전 계획

- 다음 프로젝트에서는 task를 좀 더 세분화하여 시간을 효율적으로 쓰는 방향으로 고심할 것이다.
- 프로젝트의 빠르고 원활한 진행을 위해, 빠르고 명확한 의사결정을 내리는 것이 필요할 것이다. 이를 위해 회의 마무리 시간에 의사결정 시간을 추가하기, 결론이 잘 나지 않는 사안에 대한 투표 시스템 도입 등의 방안들을 고려해볼 수 있다.
- 프로젝트 돌입 전, task에 대한 전체적인 이해도를 높일 필요가 있다. task가 어떤식으로 진행돼야 하는지, 진행해야 할 구체적인 task는 무엇이 있고 어떤 순서로 진행해야 할지 등 전반적인 task에 대한 이해도를 높임으로써, 더욱 구체적인 프로젝트 진행 계획을 세울 수 있을 것이다.

개인회고 - 강경준

프로젝트 개인 목표

- 새롭게 적용하는 협업툴에 적응하는 것. 특히, 팀 노션 페이지에서 개인 별 프로젝트 진행 상황을 공유하는 것이 좋다고 생각했고, 이를 잘 활용할 필요가 있다고 생각했음
- 활용할 수 있는 모델의 크기가 한정돼 있기 때문에, 이러한 한계를 잘 해결하는 방안에 대해 고민해보는 과정이 필요하다 생각함. 큰 모델의 지식을 전달하거나, mixed precision을 활용하는 등의 다양한 방법을 고려해볼 수 있고, 이런 방법들을 이해하고 직접 적용해보는 것이 목표였음

시도 및 결과

- 데이터 EDA 및 전처리 과정에서 LLM을 활용함. LLM의 활용성이 늘어남에 따라 모든 과정에서 LLM의 활용 가능성 이 발생함. 따라서, label이 없는 데이터에 대한 labeling에서 LLM을 활용했음. 결과적으로 신뢰할 수 있을 만큼의 결과를 내기는 어려웠음.
- 텍스트 데이터에서는 정규표현식을 활용한 패턴 검색과 처리가 굉장히 유용함. 데이터 내에서 보기와 질문이 결합된 형태가 종종 발견되었고, 이를 분리하기 위해 문제가 되는 케이스들을 찾아보고 패턴을 규명하여 처리함. 결과적으로 해당 데이터들에 대해서 적절한 처리를 진행할 수 있었음
- 작은 모델이 거대한 모델의 행동을 학습하는 방식 중 하나로서, reasoning 학습 방식을 선택함. 거대 모델이 생성한 reasoning을 작은 모델이 학습함으로써, 작은 모델의 성능을 극대화 하는 방식임. 먼저, huggingface의 CasualLM model의 동작에 대한 이해가 부족해 적절한 학습 코드를 작성하는데 어려움이 있었음. 이로인해, 시간이 부족하여 최종 데이터에 대한 결과를 보지 못함. 뒤늦게 validation 데이터에 대해 결과를 도출했을 때, 만족할만한 결과가 있었기 때문에 다른 태스크에서도 활용해보는 것을 목표로 함

좋았던 점과 배운 점

- 결과물을 많이 내지는 못했지만, 계속해서 고민하고 시도했다는 점이 좋았다고 생각함. 당장에 성능을 내는 일은 어렵지만, 그럼에도 계속해서 고민해 나가는 과정을 통해 사고력을 늘려나갈 필요는 있음. 이러한 점에서 나의 실험이 좌절되더라도, 고민하고 시도하는 일은 멈추지 않는 것이 나에게 더 도움이 될 것이라 생각함.

아쉬운 점

- 그럼에도 불구하고, 결과물을 적극적으로 내는 자세가 너무 부족함. 먼저, 사소한 부분에 너무 연연하여, 결과 도출까지 이어지지 못하는 경우가 너무 많음. 완벽하지 않더라도, 실험 상황을 적절히 통제하고 결과를 남기는 습관을 들일 필요가 있음.

앞으로의 목표

- 프로젝트 기간 내에 마무리 못한 방법론이 있었는데, 해당 방법론에 많은 관심이 생겼음. 해당 방법론에 대해 조금 더 자세하게 이해하고, 명확하게 결과를 도출하여 해당 방법론에 대해 더 확실한 경험을 남기고 싶음.
- 추가적으로, sLLM, 모델 최적화 및 경량화 등 작은 모델의 활용에 대해서 더 많은 관심이 생겼고, 이에 대해 더 공부 할 예정.

개인회고 - 권지수

프로젝트 개인 목표

새로 꾸린 팀에서 첫 프로젝트를 진행하게 됨에 있어, 어떤 분위기인지 파악하고 잘 따라가야겠다고 생각했다. 이전까지의 프로젝트에서는 Notion으로 협업을 진행해서 GitHub가 익숙하지 않았는데 이번에 잘 활용하기로 해서 좀 더 공부해보고, 모르는 건 빨리 물어보기로 다짐했다. 이번 프로젝트는 수능에 특화된 언어 모델을 구축하는 작업으로써, 여러분야, 또 많은 데이터가 필요하겠다고 생각했고, 팀원들의 다양한 도전을 이해하고 그에 맞는 데이터를 수집 및 정제해서 제공하는 게 재밌을 것 같다고 생각했다.

시도 및 결과

프로젝트 초기 단계에서 주어진 데이터에 대한 EDA를 진행하고 데이터를 살펴보며, 사람이 봐도 풀 수 없는 이상한 데이터가 포함되어 있다는 사실을 발견했다. 학습 데이터가 2031개 정도 되었는데, “지문, 문제, 선지, 정답” 서로 간에 매칭이 잘 되지 않아 사람이 봐도 풀지 못하는 문제는 팀원들과 직접 보고 제거했다. 그렇게 제거한 데이터에서 필요없는 단어를 제거하고 잘못된 띄어쓰기를 수정하여 약간의 성능 향상을 도출했다. 또한, 팀원들이 필요한 데이터를 수집하고, 필요에 따라 크롤링을 하거나 PDF를 읽어오는 작업을 했다. 다른 팀원이 우리 프로젝트에 맞는 모델을 찾고, 어떤 모델이 이번 태스크에 잘 맞는지 실험 해놓은 걸 토대로, 성능이 가장 좋은 모델에 적합한 하이퍼파라미터 튜닝을 진행했는데, 내가 했던 결과로는 모델의 과적합을 보완하지는 못했다. 결과 그래프를 보고 빠르게 판단하여 수정하는 방법에 대해 공부가 부족했던 것 같다.

좋았던 점과 배운 점

부스트캠프 과정에서 GitHub에 대한 강의가 있었지만, 막상 프로젝트에서 이용하려고 시도할 때, 낮은 이해도에서부터 오는 약간의 두려움이 항상 있었다. 그래서 이번 프로젝트에서 걱정이 많았지만, 팀원 모두가 도와줘서 조금 사용해볼 수 있게 된 게 좋았다. 지금도 원활하게 이용하기엔 부족한 부분이 많다고 느끼기에, 다음 프로젝트를 위해서 좀 더 공부해야겠다고 생각했다.

그리고 처음 만난 팀원들이지만 프로젝트 계획이나 진행 상황, 또는 질문과 답변과 같은 사소한 것부터 꼭 필요한 부분까지 빠르게 공유하는 점이 좋았다.

아쉬운 점

다른 팀원들의 다양한 방법론 실현하는 데에 도움이 되고자 필요한 데이터를 찾기 위해 많은 시간을 보냈지만, 내가 생각했던 것보다 저작권에 자유로운 데이터를 얻기가 아주 힘들었다. 프로젝트가 끝나고 나서야 사용 가능한 데이터를 몇 개 알게 되었고, 그걸 찾지 못했다는 게 아쉬웠다. 그리고 데이터 검수하는 데 시간을 많이 할애한 것에 대해 다들 아쉬워했는데, 다른 팀의 발표를 듣고 데이터를 자세히 들여다보는 데에 시간을 좀 더 썼어야 했나? 하는 생각이 들기도 했다.

앞으로의 목표

협업을 위해 GitHub에 대한 공부가 더 필요할 것 같다. 그리고 프로젝트를 진행하면서 다른 팀원들이 했던 방법론에 대해 더 깊게 이해하지 못한 부분이 있었던 것 같다. 부스트캠프에서 공부하면서 항상 시간이 모자란 것 같아 답답한 부분이 있지만 마지막 프로젝트 전에 협업 툴이나 그동안 강의에서 배웠던 부분에 대해 다시 생각해보고 정리하는 시간을 가져야겠다.

개인회고 - 김재겸

프로젝트 개인 목표

본 프로젝트에서는 체계적인 실험과 논리적 흐름을 중요하게 생각했다. 또한 체계적인 협업, 시도해보고 싶은 내용을 체계적으로 적용해 보는 것을 목표했다.

시도 및 결과

초반에는 data centric적인 관점으로 데이터오류교정 및 전처리에 집중했다. 이후에는 rag구현을 시도했다. rag구현은 충분한 지식의 부족 및 시간의 부족으로 실패했다. 이후 데이터 증강을 통한 성능개선에 힘썼고, 7B~9B모델 중에서는 가장 높은 성능을 볼 수 있었다. 다만 초반부터 모델서치에 시간을 오래 가져가지 못해서 다양한 모델을 적용해보지 못했어서 32B짜리 큰 모델에 대한 실험이 부족했다. Scalling Laws의 영향으로 파라미터수가 많은 모델에서 가장 높은 성능이 나왔다

좋았던 점과 배운 점

우선 양게나마 rag시스템의 흐름을 배울 수 있었다. 또한 LLM을 파인튜닝해보는 경험을 쌓고, 프롬프트에 대한 고민도 해보면서 해당 task를 어떻게 잘 수행할 수 있을지에 대한 경험을 했다. 그리고 그동안 이론적으로만 알고 있었던 Git을 이용한 협업을 체계적으로 해본 것 같아서 좋았다. 정말 리더보드에 신경을 쓰지 않고 프로젝트를 진행한 것은 처음인 것 같다. 리더보드에 신경을 안쓰니 정말 해보고 싶은 것을 다 해봤다는 부분에서 좋았던 것 같다. 이번 멘토님께서 해주신 말씀중에 NLP분야에는 다른 분야와 다르게 “Scalling Laws”라는 믿음이 있다라고 해주셨는데 이번 프로젝트에서 정말 크게 깨달은 것 같다. 또한 사전학습 vs 파인튜닝 vs rag vs 프롬프팅 등 ILM의 성능을 올리기는 방법에는 정말 많은 방법이 있다고 느꼈고, 조금씩이나마 실험해 볼 수 있어서 좋았다.

아쉬운 점

우선 가진 지식의 부족으로 인해 rag를 실패한 점, 또한 메타인지의 부족으로 인해 너무 시간을 많이 쓴 점이 첫번째로 아쉬웠다. 리더보드에 신경을 안쓰는 부분이 메타인지 저하까지 이어진 것 같다. 두 번째로는 데이터 검수에 시간이 많이 들어가서 실험에 시간을 상대적으로 못 쓴 부분도 아쉬운 것 같다. 여러 방법론을 적용하고 여러 데이터를 사용한 결과보다 32B모델이 inference했을 때 비교안될 정도로 성능이 올라가서 그 부분에 대한 분석도 시도해 보고 싶다. 이렇게 리더보드형 프로젝트가 끝났는데, 앞으로 남은 기업해커톤은 조금 더 큰 그림으로 전체 프로세스에 대한 계획 이후에 시간 분배를 잘 해서 성공적인 프로젝트로 끝내고 싶다.

앞으로의 목표

프로젝트 진행에 있어서 성능을 확인 할 수 있는 지표를 마련하고 싶다. 또한 무언가 막히는 부분이 있을 때 해당 내용에 대한 정보를 찾는 선택지를 늘려서 보다 완성도 높은 결과를 만들고 싶다. 앞으로 남은 기업 해커톤 프로젝트는 주제와 협업이 중요하다고 생각한다. 협업자체는 신생팀이라는게 믿기지 않을 정도로 좋았다고 생각한다. 이전 4번의 프로젝트에서의 경험과 강의 및 학습을 통해 얻은 지식으로 성공적으로 부스트캠프 생활을 마무리 하고 싶다.

개인회고 - 박동혁

<프로젝트 개인 목표>

해당 프로젝트의 표면적 의미는 과연 LLM이 사람의 지능을 평가하는 시험에서도 성과가 있을까에 대한 것이었다. 이 주제를 개인적으로 재 정의를 해본 결과 NLP적으로 확인해야 하는 성과는 다음과 같다고 느꼈다.

“현존하는 LLM의 성능이 한국인 고등학생 수준의 추론이 가능한가를 확인해 보는 TASK.”

본인은 해당 질문을 해결하는 것을 중심점으로 두고 이번 프로젝트를 진행하게 되었다.

<'의문'을 풀어 가는 과정>

우선적으로 8b모델 기준으로 한국어로 학습된 모델의 지식 수준을 몇 가지의 한국사 문제로 테스트를 진행해보았다. 해당 과정에서 모델이 반복적으로 이상한 결과를 내놓는 문제들이 발견되었다. 하나의 문제를 여러번의 시도에 걸쳐 풀어내는데 전혀 다른 답이 나오는 것을 발견하였다. 왜 이런 현상이 발생할까에 대한 의문을 던지며 크게 두 가지의 문제사항을 가정했다. 첫째는 모델이 충분한 지식을 보유하지 못한 것이고, 둘째는 모델이 질문, 지문 그리고 선지의 상관 관계를 파악하지 못하는 가정이다. 역사서의 일부분을 주면서 제시되지 않은 관련 인물을 뽑아내는 실험 결과, 보유 지식 면에서는 우수하다는 것을 알 수 있었다. 하지만 이것이 지문과 질문의 형태로 변화된 순간 모델의 결과가 좋지 않음이 발견되었다. 즉, 지식은 충분하나, 지문과 질문의 관계를 추론하는 것에 어려움이 있는 것이다. 더 정확히는 “지시사”에 관한 추론 능력이 부족함을 볼 수 있었다.

문제를 구체화한 다음 방법을 찾기 위하여 비슷한 사례를 실험한 경우가 있는지 확인해본 결과, CoT의 단계적 리즈닝을 주는 방법이 있음이 발견 되었고 이 것만으로 부족함을 느껴서 언어학적 추론 방식인 귀납적 추론과 연역적 추론 방식을 활용하여 Task를 도전하였다.

그 결과 특정한 문제를 10번에 한번 맞추던 모델이 네번에 한번은 정답 결과를 보여주는 것을 확인 할 수 있었다.

<결론을 통한 “시야의 확장”>

결론적으로 말하자면 LLM의 추론 능력은 놀라웠다. 다만, 아직 “완벽한 추론”을 하기에는 넘어야하는 단계가 남아 있음을 발견할 수 있는 경험이었다. 언어학적으로 “지시사”에 대한 대응력이 부족하였고, 심리학적 및 문화적으로 “인물”的 감정 상태를 추론하는 능력이 부족함이 느껴졌다.

또한, 프롬프트 엔지니어링에 대한 발전도 아직 시작 단계임을 조금이나마 느낄 수 있었다. 같은 내용을 주더라도 어떤 모델을 쓰느냐에 따라서 작성 방식이 달라짐에 정형화가 되지 않을 것을 볼 수 있었고 이를 통하여 좋은 프롬프트가 어떤 것인가에 대한 공통된 수치적 지표가 완성되지 않음이 보였다.

만약, 추후에 프롬프트를 통하여 모델의 성능을 개선 시키고자 한다면, 현재 본인이 키워야 하는 능력은 대표 모델들의 프롬프트 작성 방식의 공통점을 찾아 이를 토대로 결과를 증명해 낼 수 있는 지표를 갖는 것이라 생각이 든다.

개인회고 - 이인설

프로젝트 개인 목표

기본적인 모델 성능에 단순히 의존하지 않고, 다양한 방법론과 아이디어를 적용하며 체계적인 실험을 통해 성능을 비교, 분석할 수 있는 능력을 기르는 것을 목표로 한다. 또한 LLM을 원하는 태스크에 맞게 설계하고 자유롭게 활용할 수 있는 역량을 키우고자 한다. 더불어, 논문 및 관련 자료를 효율적으로 탐색하는 능력을 향상시키며, ChatGPT를 보조적인 도구로 활용하는 데 그치고, 주체적으로 문제를 해결하는 능력을 기르고자 한다.

시도 및 결과

Scaling Laws에 지나치게 의존하지 않고, 다양한 아이디어를 제시하며 이를 팀 프로젝트 전체 과정에 녹아들 수 있도록 노력하였다. 특히, CLM의 특성을 깊이 이해하고, 정답에 가까운 답변을 생성하기 위한 방법을 고민하는데 많은 시간을 투자하였다.

Domain Adaptation과 같이 강의에서 다루지 않은 내용을 적용하기 위해 여러 논문과 사전 연구를 탐색하며 새로운 접근법을 모색하였다. 그러나 자원 제한과 개념에 대한 충분한 이해 부족으로 인해 실제 적용에는 어려움이 있었고, 기대했던 만큼의 성과를 얻지는 못하였다. 그럼에도 주어진 상황에서 최선을 다해 적용 가능한 방안을 찾고자 했으며, 이번 도전에서 얻은 경험은 추후 다른 프로젝트에서도 활용할 수 있을 것이라는 점에서 의의를 두고 있다.

좋았던 점과 배운 점

매 프로젝트에서 모듈화 작업을 반복하며 모듈 설계와 관리 능력이 향상되었다. 초기에는 지나치게 세분화된 모듈과 인자 설계로 인해 사용이 어려웠던 점을 경험하였고, 이를 보완하기 위해 처음부터 큰 틀을 잡는 방식으로 접근하였다. 이후, 새로운 방법론이나 실험을 진행할 때 필요한 모듈을 선택적으로 활용할 수 있도록 유연성과 확장성을 고려한 구조를 설계하였다. 또한, Git Flow를 통해 AI와 데이터 분석 프로젝트에서도 효율적인 코드 관리를 익혔다.

ChatGPT에 대한 의존도가 높았던 초기와 달리, 현재는 이를 보조적인 도구로만 활용하며 직접 논문과 소스 코드를 분석함으로써 지식을 쌓아가고 있다. 이를 통해 문제 해결 능력과 독립적인 학습 태도를 점차 향상시키고 있다.

아쉬운 점

프로젝트를 거듭하면서 실험 설계에 익숙해지고 있었으나, 이번에는 이전보다 더 어려운 업무를 맡으면서 설계 과정에서 많은 어려움을 겪었다. 특히, 시간에 쫓겨 태스크 자체에 대한 충분한 학습이 이루어지지 못했고, Fine-tuning을 진행할 때 학습과 평가 방식이 일치해야 한다는 중요한 점을 간과하여 유의미한 성과를 얻지 못했다. 또한, 실험에 필요한 데이터를 확보하는 과정이 매우 어렵다는 점을 깨달았으며, 데이터를 찾는데 많은 시간을 소모해야 했다.

이전에 Retrieval 시스템을 성공적으로 구현한 경험이 있어, 이번 프로젝트에서 RAG 구현을 담당하였다. 하지만 앞선 태스크들에 많은 시간을 할애하였고 DB 구축에도 많은 시간이 걸려, 추가적인 고도화 및 실험을 충분히 진행하지 못해 성능을 정확히 비교, 분석하는데 한계가 있었다.

앞으로의 목표

실험 설계 및 데이터 확보 과정을 효율적으로 관리하는 능력을 키우고, 학습과 평가 방식의 일치성을 더욱 철저히 검토할 계획이다. 또한 제한된 자원 내에서도 성능을 극대화할 수 있는 방법론을 탐구하며, RAG와 같은 복잡한 시스템을 고도화할 수 있는 역량을 강화하고자 한다.

개인회고 - 이정휘

이번 프로젝트에서의 목표는 무엇인가?

본 프로젝트에서는 새로운 팀과의 협업, 팀원들의 능력, 스타일 파악이 중요했고 “기록”을 중요시하고자 했다. 기록을 기반으로 다음 Task에 대한 의사결정을 하고 순차적으로 프로젝트가 진행되도록 하는 것이 목표였다.

나는 어떤 방식으로 모델을 개선했는가?

Base 모델로 사용될 모델들을 탐색하고 Few-shot 프롬프팅을 적용하였다. 추가로, 32B 모델을 사용하여 Few-Shot Inference로 단일 모델 중 가장 높은 성능을 내었다. 모델이 개선되지는 않았지만 Milvus DB, Langchain을 활용하여 RAG를 구현하였다.

마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

첫번째로, 데이터 품질 검수에 있어서 내가 맡은 부분에 자체적으로 판단하기 어려운 부분이 있었다. ChatGPT에 계속 입력하여 실제 LLM이 문제를 풀 수 있는가를 판단하면서 진행하였으나, 이때 좀 더 체계적으로 검수 조건을 결정했다면, 데이터 전처리 시간을 단축할 수 있을거라고 생각했다. 두번째로, RAG를 구현하면서 Retrieval의 성능보다 Vector DB 구성에 좀 더 집중하였으나 마스터 클래스때 “Retrieval 성능이 가장 중요하다”라는 말씀을 듣고 BM25, DPR 혹은 Hybrid Retrieval 등 다양한 Retrieval 방법들을 적용하고 Re-Ranker도 적용해보면 좋았을 것 같았다라는 생각이 들었다. 그리고 멘토님께서 추천한 방법으로 Query Re-writing에 대해서도 말씀하셨었는데, 시간상 제대로 해보지 못한 점이 아쉽다. 세번째로, Prompt를 사용함에 있어서 Fine-Tuning / Inference에 사용되는 기법들이 각각 달랐다는 점을 늦게 알았던 점이 아쉽다. Self-consistency, Knowledge Generation 등과 같은 다양한 프롬프팅 기법들이 있었는데, 이러한 기법들은 대부분 Inference하는데에 대부분 사용되었다는 것이었다. 위 방법들을 fine-tuning 과정에 사용하려다보니 거기에 어려움이 있었다.

한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점은 무엇인가?

어떤 것을 구현할 때 절차를 우선적으로 설계하고 진행하는 것을 목표로 해야겠다고 생각했다. 특히 새로운 기능을 구현 할 때에는 더더욱 필요하다고 느껴졌다. 그리고 기능을 구현하면서 항상 “1분 이내”로 실행한 결과까지 볼 수 있도록 작은 데이터로 우선적으로 테스트해보는 것을 지켜야겠다고 생각했다. 이번 RAG 구현에서도 특히나 123만개가 넘는 데이터를 기반으로 vector DB를 구성하려다보니 너무 많은 시간이 소요되었는데, 이것이 실제 작동되는지도 확인하지 못하고 우선 DB에 모든 데이터를 다 insert하는 시간을 소비했다. 결국 Field가 맞지않아서 DB에 다시 insert해야하는 문제가 있었다. 따라서, 꼭 실행 결과까지 우선 확인 후 큰 단위로 넓히는 것을 시도해보고자 한다.

협업과정에서 잘된 점 / 아쉬웠던 점은 어떤 점이 있는가?

팀원 모두 잠이 없는 편이라 새벽에도 줌에 들어와서 어려운 점이 있거나 회의가 필요한 점이 있다면 즉각적으로 반영할 수 있는 점이 좋았다. 또한 새롭게 합을 맞춤에도 불구하고 마음이 잘 맞았다. 하지만 모두가 “기록”이 부족하다보니 깔끔하게 정리되지 못함이 보였다. 특히 했던 작업을 다시 또 하는 불상사도 존재했다. 또한, Github의 사용이 아직 미숙함이 있어 코드 버전 관리가 제대로 되지 않았던 것이 아쉬웠다.