

K-SAT Problem Solver Wrap-up Report

NLP-01조: NLPing

팀원 이름: 육지훈 전진 정준한

허윤서 이수진 이금상

1. 프로젝트 개요

A. 개요

대회 특징	설명
대회 주제	이번 대회에서는 '한국어'와 '시험'이라는 주제에 맞춰서 작은 모델들로 수능 시험을 풀어보는 도전을 시작해보려 합니다. 대부분의 대형 모델들은 한국어에 완벽히 최적화되지 않았음에도 불구하고 수능에서 꽤 높은 성적을 기록하고 있습니다. 그렇다면, 작은 모델로도 같은 성적을 낼 수 있을까요? 혹은, 우리가 알고 있는 한국어의 특성과 수능 시험의 특징을 바탕으로 수능에 특화된 우리만의 AI 모델을 만들어볼 수 있을까요? 수능에 최적화된 모델을 만들어, GPT, Claude, Gemini 같은 대형 모델들을 뛰어넘어 봅시다!
대회 설명	<ul style="list-style-type: none">Input: 입력 데이터는 수능 국어와 사회 과목의 지문 형태를 따릅니다. 'id', 'paragraph', 'problems', 'question_plus'의 형태로 되어 있고, 각각은 id, 지문, 문제, 보기를 의미합니다. problems에는 'question', 'choices', 'answer'로 구성되어 있고 각각 질문, 선택지, 정답을 의미합니다.Output: 주어진 선택지 중에서 정답을 맞추어야 합니다. 정답은 지정된 submission 형식에 맞게 csv 파일로 작성하여 제출해야 합니다.
대회 기간	2024년 11월 13일 (월) 10:00 ~ 2024년 11월 28일 (목) 19:00
데이터 구성	<ul style="list-style-type: none">수능형 문제<ul style="list-style-type: none">수능의 국어, 사회 영역(윤리, 정치, 사회)과 비슷한 문제KMMLU (Korean History), MMMLU (HighSchool 데이터 중 역사, 경제, 정치, 지리, 심리)KLUE MRC (경제, 교육산업, 국제, 부동산, 사회, 생활, 책마을)
평가지표	$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} = \frac{TP + TN}{TP + TN + FP + FN}$

B. 환경

(팀 구성 및 컴퓨팅 환경) 6인 1팀, 인당 V100 서버를 VSCode와 SSH로 연결하여 사용

(협업 환경) Notion, GitHub

(의사 소통) 카카오톡, Zoom, Slack, Jira

2. 프로젝트 팀 구성 및 역할

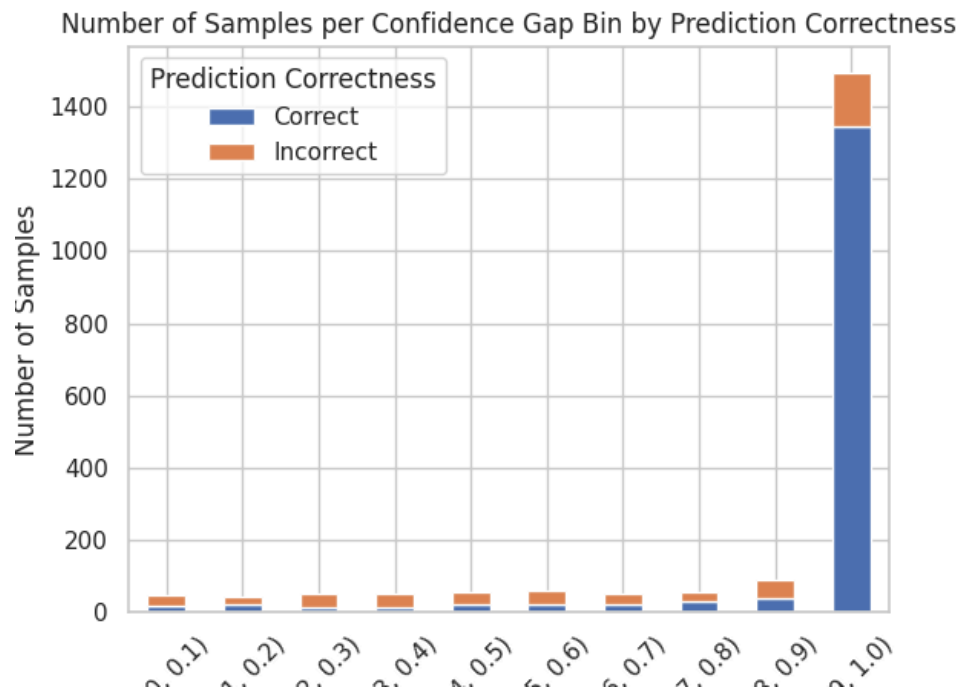
팀원 명	역할
육지훈	EDA , 이상치 탐지, 모델 실험, EDA(Easy Date Augmentation) , 백트랜슬레이션 데이터 증강, choice 순서 변경 데이터 증강, 제로샷 & 제로샷 CoT 성능 분석
전진	EDA , 데이터 라벨링 및 전처리, 데이터 증강, RAG 구현, 제로샷 CoT
정준한	과목 분류, 모듈화, 추론속도 향상, 제로샷 & 제로샷 CoT 성능 분석
허윤서	모델 취약점 분석, 제로샷 & 제로샷 CoT 분석, 프롬프트 튜닝, 기타 utils 구현
이수진	이상치 탐지, 과목분류 기반 모델 취약점 분석, 프롬프트 튜닝, 제로샷 CoT 성능 분석
이금상	외부데이터셋 탐색, 데이터 증강, RAG , 제로샷 CoT 성능 분석

3. 프로젝트 수행 절차 및 방법

A. EDA(Exploratory Data Analysis)

a. 모델의 **confidence** 분석

기존의 데이터로 학습된 LLaMa 3.2 3B 모델을 바탕으로 Train 데이터 셋의 문제에 대해 어느정도의 확신을 가지고 답을 선택하는지를 확인하였다. 선택지 별 확률을 구해 가장 우세한 확률을 가진 선택지와 그 다음의 선택지 간의 확률 차이를 바탕으로 **confidence gap**을 계산하여 맞힌 문제와 틀린 문제의 분포를 확인하였다.



강한 확신을 가진 문제 중 맞은 문제와 틀린 문제의 차이를 비교하고 확신이 낮은 문제를 분석하였다.

강한 확신을 가지고 틀리는 경우 문제 자체 오류나 모델의 추론 능력 부족으로 인해 지문을 표면적으로만 파악하는 경향이 있음을 발견하였다.

약한 확신을 가진 문제는 지문의 내용이 부족하거나 지문에서 관련 내용을 찾기 힘든 경우가 많이 포진되어있음을 발견하였다.

b. 이상치 탐지

제공된 train 데이터셋에서 KMMLU 기반 데이터 오류인 문제와 선지가 다른 문제와 엇갈려 잘못된 데이터를 확인하였다. 또한 **question**에는 “밑줄 친”이라는 단어가 있지만 **paragraph**에는 실제로 밑줄이 안쳐져있어 학습에 방해가 될 수 있다고 생각하였다. 그외에도 데이터를 살펴보면 `\x0`와 같은 노이즈가 낀 데이터, '[', ']' 기호가 섞여있는

paragraph, 너무 짧은 paragraph와 같은 데이터 이상치가 있다는 점을 확인하였다.

c. Category classification 기반 모델 취약점 분석

모델이 어떤 과목(Category)에서 잘 맞추고, 못 맞추는지 분석하기 위해 train데이터셋에 대해 과목을 분류하고 분석했다.

train데이터셋은 KLUE-MRC, KMMLU, MMMLU로 구성되어있었는데, 원본데이터셋을 이용해 일치하는 행들을 찾고 분류하였다.

다음과 같이 16개로 Category를 분류하고 분석을 진행하였다.

- KLUE-MRC (경제, 교육산업, 국제, 부동산, 사회, 생활, 책마을)

- KMMLU (korean history)

- MMMLU (european_history, geography, goverment_and_politics, macroeconomics, microeconomics, psychology, us_history, world_history)

‘gemma-ko-2b로 train데이터셋을 훈련한 결과’와 위의 ‘a. 모델의 confidence 분석’을 기반으로 ‘llama 3.2 3B의 훈련결과’를 카테고리 라벨링하여 정확도를 측정하였다. 두 모델에 대한 결과는 다음과 같다.

이중질문(예: “(가)에 대한 설명으로 옳지 않은 것은?”라는 문제에서 1. (가)를 찾고, 2. (가)에 대하여 보기에서 옳지 않은 것이 무엇인지 찾는 경우와 같이 총 두 단계로 이루어진 문제)이 없는 KLUE-MRC 데이터의 경우 gemma모델에서는 정확도가 모두 0.6 이상이었고, Llama모델에서는 높은 confidence를 가지는 문제의 상위권에 KLUE-MRC가 분포하여 두 모델 모두 KLUE-MRC에 대한 정확도는 높다는 것을 파악했다. 그러나, gemma모델에서는 kmmlu-한국사(0.43), mmlu-world history(0.40), mmlu-european history(0.39), mmlu-us history(0.37)와 같이 역사 관련 카테고리에서 정확도가 낮았고, Llama모델에서는 confidence가 낮은 문제 카테고리에 역사 관련 문제가 상위권에 분포했다. 따라서 이러한 점을 보완하기 위해 RAG와 같은 방법론으로 배경지식을 함께 학습시켜야겠다는 결론을 얻었다.

B. 데이터 전처리

a. 데이터 정제

앞서 데이터에서 확인한 데이터 이상치를 바탕으로 데이터를 정제하였다.

question에서는 “밑줄 친”이라는 단어를 제거하였고, \xa0과 같은 노이즈가 많은 행들을 제거하였다. paragraph에 ‘[과]’ 기호가 섞인 경우를 살펴보았을 때 대괄호 안에 있는 단어와 관련된 문제가 없다는 점을 확인하여 paragraph에서 대괄호도 제거하였다.

이렇게 총 27행을 제거하여 만든 데이터셋인 ‘cleaned_data.csv’를 학습에 활용하였다.

b. 라벨 평탄화

train.csv 파일을 로드한 후, **problems** 열의 데이터를 파싱하여 문제와 선택지 정보를 딕셔너리로 변환하고, 원본 데이터를 기반으로 선택지 순서를 교체하여 정답 선택지를 업데이트한 뒤 순서를 교체하여 5배 증강된 데이터로 새로운 데이터프레임을 생성하여 **train_x5.csv** 파일로 저장하여 라벨 평탄화를 진행하였다.

c. 질문 형태 일반화

데이터 내 일부 **question**이 일반적이지 않은 질문의 형태인것을 발견하였다. 이러한 형태의 질문들은 모델이 학습하고 정답을 생성하는 것에 어려움이 있어 LLM을 활용하여 해당 질문들을 일반적인 질문의 형태로 재구성 하는 작업을 수행하였다.

C. 증강

a. AGIEval 벤치마크 데이터셋 이용

이 데이터셋은 미국 SAT, 로스쿨 입학 시험 등 20개의 공식, 공개 및 고급 입학·자격 시험에서 파생된 데이터로 구성되어 있다. 해당 데이터 중 영어 데이터셋을 활용하여 증강을 진행하였다.

영어-한국어 번역역 과정에서는 Google 번역과 Llama의 여러 모델을 활용하였으나, 완벽한 번역에는 실패하였다. 최종적으로 GPT-4o-mini 모델을 사용하여 번역을 완료하였다. Batch API를 이용해 paragraph, question, choice 컬럼을 개별적으로 번역한 뒤, 후처리 과정을 통해 정제된 증강 데이터셋 1581개를 구축하였다.

b. facebook/belebele 데이터셋 이용

Huggingface Datasets에서 찾은 facebook/belebele 데이터셋을 사용하였다. 지문, 질문, 4지선다형으로 구성되어있고 한국어 데이터셋이 900개 존재하였다.

D. 모델 선택

호랑이 리더보드, ko-llm 리더보드(upstage), LogicKor 리더보드 등을 참고해 모델들을 사용해보았다.

모델 크기별 좋았던 모델은 다음과 같다.

3B: Bllossom/llama-3.2-Korean-Bllossom-3B

8B: NCSoft/LLaMA-Varco-8B-Instruct, CohereForAI/aya-expanse-8b

14B 이상: Qwen/Qwen2.5-14B-Instruct, Qwen/Qwen2.5-32B-Instruct,

Qwen/Qwen2.5-72B-Instruct-AWQ

E. Zero-Shot 분석

모델이 틀린 문제에 대한 confidence 분석 과정 중 한 train 데이터 샘플에서 모델의 크기가 증가할수록 추론능력이 증가한다는 점을 확인하였다.

“지난 **2018년 400**호점 달성 **2년** 만에 **500**호점을 돌파하며 가파른 성장세를 보여준 것”

위의 지문에 대해 “크린업24가 500호점 계약을 성사시킨 연도는 언제인가?” 라는 질문의 답변을 LLaMa 3.2 3B (한국어 파인튜닝 버전), Gemma-9b (한국어 파인튜닝 버전), Qwen 2.5 14B 모델에 대해 비교 분석을 진행하였다.

llama 3B: 지문에 2005년 셀프빨래방 프랜차이즈 가맹 사업을 시작했기 때문에 2005년이다.

gemma 9B: 지난 2018년 달성 2년만에 500호점을 돌파했기 때문에 2018년이다. 나머지 선택지들은 지문에 언급이 안되었다.

qwen 14B: 선택지의 2020년이 지문에 언급은 없지만 2018년 2년 뒤 500호점을 돌파한 것으로 보아 2020년이다.

위의 분석을 통해 모델의 크기에 따라 순차적으로 추론이 발전하는 것을 확인하고 Qwen 2.5 14B 모델의 제로샷 성능을 측정하였다. 그 결과 8B 모델 학습 대비 13% 성능 향상을 보였다.

F. Prompt 튜닝 & CoT

zero shot CoT 성능 향상을 위해 프롬프트 튜닝을 진행하였다. CoT 프롬프트를 위해 한국어, 영어로 다양한 프롬프트를 작성하였다.

모든 프롬프트에는 정답이 숫자가 아닌 경우로 출력되거나, 숫자가 여러개 출력되는 것을 방지하기 위해서 단 하나의 숫자 정답을 출력할 것을 강조했다. 또한 문제를 푸는 풀이과정을 단계별로 제시했다. 한국어, 영어의 풀이과정 지시사항의 예시는 다음과 같다.

[한국어 Prompt]

1. 문제를 분석하고 질문의 요구사항을 파악합니다.
2. 관련 정보를 지문과 질문에서 추출합니다.
3. 선택지를 하나씩 검토하며 논리적으로 평가합니다.
4. 최적의 답을 선택한 뒤, **Structured output** 형식으로 작성합니다.

[중요] 정답은 단 하나의 숫자(1, 2, 3, 4, 5)로 생성합니다.

[영어 Prompt]

1. After the [Answer] format, generate the correct answer number.
2. The correct answer must be one of the numbers: (1), (2), (3), (4), (5).
3. After the [Justification] format, provide the reasoning for the correct answer.
4. The justification must be based on the content of the passage.

G. RAG

Paragraph 내 정보만으로 정확한 문제 풀이가 불가능한 유형의 문제가 존재한다. 이러한

문제들에 대한 풀이 능력을 상승시키기 위하여 외부 데이터를 참조할 수 있도록 **RAG**를 구현하였다.

먼저 **RAG**의 기반이 될 데이터베이스는 저작권 문제가 없으면서, 모든 도메인 정보를 얻을 수 있는 **Wikipedia** 한국어 **Dump** 데이터를 활용하였다.

Query와 연관된 문서를 연결하는 **Retrieval**은 다음 방식들을 테스트 해보았다. 먼저 일반적인 방법론인 임베딩 벡터를 기준으로 **Query**와 **Wiki** 간 연관성이 높은 문서를 찾는 **Dense Retrieval**, **BM25**를 활용한 **Sparse Retrieval**을 테스트하였다. 그러나 직접적으로 문제풀이에 필요한 문서를 찾는 것에는 어려움이 있었다. 따라서 **LLM**을 활용하여 질문과 보기로 이루어진 **Query** 내에서 문제풀이에 직접적으로 도움이 될만한 **Keyword**를 찾는 방식으로 테스트해보았고, 이전 방식보다 연결된 문서의 정확도가 높다고 판단하여 해당 방식으로 진행하였다.

H. Gpt4o distillation


기존 train데이터셋에는 **reasoning**이 존재하지 않았지만, **Gpt4o**로 **reasoning**을 생성하여 **reasoning**도 같이 학습시키려고 시도하였다.

학습 시에, **reasoning**을 학습시키고 추론시에는 **CoT** 기법으로 **reasoning**과 **answer**를 도출시키려고 하였다.


하지만, 메모리 부족(토큰이 길어짐)에 따라, **14B** 이상의 모델에서는 훈련이 불가능했다.

4. 프로젝트 수행 결과

Leaderboard [Mid]

내등수 7	NLP_01조		0.7834	85	21h
----------	---------	---	--------	----	-----

Leaderboard [Final]

내등수 8	NLP_01조		0.7402	85	1d
----------	---------	---	--------	----	----

5. 자체 평가 의견

- **JIRA**를 처음 도입하면서 프로젝트 진행 상황을 체계적으로 관리할 수 있었고, 역할 분배와 작업 우선순위 설정에 많은 도움을 받았다. 특히, 각 작업의 세부사항을 명확히 정의하고 개인에게 할당하면서 마감 기한을 효과적으로 준수할 수 있었다. 다만, **JIRA**의 **Sprint** 기능을 충분히 활용하지 못한 점이 아쉬웠으나, 이를 통해 프로젝트 관리 툴에 대한 경험과 중요성을 깨달았다. 다음 프로젝트에서 전체적인 프로젝트 일정 계획을 세우고, 작업의 유기적인 연결이 원활히 될 수 있도록 개선하면 좋을 것이다.
- 대규모 모델 사용이 성능에 긍정적인 영향을 미친다는 것을 직접 확인하였으며, 이는 향후 모델 선택과 설계에 중요한 기준이 될 것이다. 또한, **OOM** 문제를 겪으면서 리소스 관리와 경량화 기법의 필요성을 절실히 느꼈다. 이를 통해 **QLoRA**나 기타 최적화 기법에 대한 추가 학습 의지가 생겼다.
- 시간 부족으로 **RAG** 구현에 참여하지 못한 점과 일부 실험이 **OOM** 문제로 중단된 점은 아쉬움으로 남았지만, 다음 프로젝트에서는 이러한 한계를 극복하기 위한 구체적인 계획을 세울 것이다. 특히, 경량화 기법을 적극적으로 도입하고 **RAG**처럼 데이터 활용을 극대화할 수 있는 기법을 학습하여 시도할 예정이다.
- **Qwen2.5-30B** 모델에 제로샷과 제로샷 **CoT**를 활용하며 최소한의 학습 데이터로도 모델의 추론 능력을 효과적으로 확장할 수 있음을 경험했다. 특히, 제로샷 **CoT**를 통해 복잡한 문제에서도 단계적 사고 과정을 모델이 스스로 학습하고 적용할 수 있도록 유도하며, 다양한 상황에서 일반화된 성능을 발휘할 수 있는 잠재력을 확인할 수 있었다. 이를 통해 적은 자원으로도 높은 성과를 달성할 수 있는 방법론의 가능성을 체감하게 되었다.
- 협업과 분업을 통해 성과를 도출하며 팀워크의 중요성을 깨달았다. 특히, 프로젝트를 체계적으로 분할하고 우선순위를 정한 덕분에 효율적인 작업 진행이 가능했다. 이러한 협업 경험은 향후 프로젝트에서도 큰 자산이 될 것이다.
- **Jira** 협업 툴을 이용해 협업과 분업을 시도해보니, 팀원들의 진행 방향을 한눈에 파악할 수 있어 매우 유용했다. 각 작업의 세부사항을 확인할 수 있어 프로젝트의 방향성을 더욱 명확히 잡을 수 있었다. 또한 분업을 통해 팀원들이 각자 맡은 부분에서 최선을 다한 덕분에 좋은 성과를 이끌어낼 수 있었으며, 나 역시 맡은 일에 최선을 다하고자 하는 마음가짐을 가지게 되었다. 이러한 경험은 향후 프로젝트에서도 큰 자산이 될 것이라 생각한다.

개인 회고

육지훈_T7404

1. 나는 내 학습 목표 달성을 위해 무엇을 어떻게 했는가?

- **EDA** 및 이상치 탐지 수행: 데이터의 전반적인 분포를 파악하고, 이상치로 인해 모델 성능에 악영향을 줄 수 있는 항목들을 제거하거나 보완했다.
- 제로샷(**CoT** 포함) 성능 실험: 사전 학습된 언어 모델의 제로샷 능력을 평가하며, 체계적으로 성능 분석을 진행했다.

2. 나는 어떤 방식으로 모델을 개선했는가?

- 백트랜슬레이션 증강: 기존 데이터에 번역 증강 기법을 적용하여 모델이 다양한 표현을 학습할 수 있도록 데이터를 확장했다.
- 선택지 순서 변경 증강: 문제의 선택지 순서를 무작위로 바꿔 모델이 특정 순서에 의존하지 않도록 데이터 다양성을 확보했다.

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

- 백트랜슬레이션 결과: 데이터의 표현 방식 다양성이 증가하여 모델의 일반화 성능이 향상되었으며, 언어적 편향에 덜 민감하게 작동하는 것을 관찰했다.
- 선택지 순서 변경 결과: 특정 위치에 과적합된 예측 패턴이 줄어들고, 다양한 문제 구성에서도 안정적인 성능을 보여줬다.

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

- 백트랜슬레이션 도입: 기존 **EDA** 기반 증강 대신, 의미는 유지하되 표현이 달라진 데이터를 생성하여 모델이 더 유연한 패턴을 학습할 수 있었다. 이는 성능 개선과 표현 다양성 증대에 크게 기여했다.
- 선택지 순서 변경 적용: 기존 데이터의 일관된 패턴을 깨뜨려, 다양한 문제 구성에도 더 강인한 성능을 보여주는 효과를 얻었다.

5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 모델 성능 한계: 모델의 크기가 작아 데이터 증강을 하였음에도 높은 정확도를 달성하는 데에는 어려움이 있었다.
- 제로샷 성능 한계: 복잡한 문제나 추론 능력을 요구하는 질문에서는 여전히 모델 성능이 제한적이었다.

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

- 더 큰 모델 사용: 성능 한계를 극복하기 위해, 더 큰 사전 학습 모델(예: **GPT-4**, **PaLM**, 또는 **LLaMA 3**)을 사용하여 더 높은 언어 이해와 추론 능력을 확보할 계획이다.
- 경량화 기법 도입: 큰 모델 사용으로 인한 계산 비용 증가를 완화하기 위해, 경량화 기법을 도입하여 모델의 효율성을 높이고 실제 활용 가능성을 향상시킬 계획이다.

정준한_T7437

1. 나는 내 학습 목표 달성을 위해 무엇을 어떻게 했는가?

- 모듈화
- 지문 순서를 바꿔 증강 사이즈별로 성능 테스트
- 데이터 증강
- 3B~72B 모델까지 다양한 모델 성능 테스트
- Gpt4o를 이용한 reasoning 데이터 생성 및 학습
- 프롬프트 튜닝 & CoT 추론 (학습모델 & 제로샷)

2. 나는 어떤 방식으로 모델을 개선했는가?

- 데이터 증강(facebook/belebele)을 하였고, 기존 train데이터셋에 합쳤을 경우 성능이 향상하는 것을 확인하였다.
- 데이터셋에 1번 정답이 많아, 평탄화하거나 지문순서를 바꿔 증강하는 방식을 사용했지만 성능향상은 일어나지 않았다.
- 프롬프트 튜닝 & CoT를 이용해 성능향상을 이뤄냈다.
- 72B 모델까지 다양한 모델을 테스트해보았다.
- GPT4o를 이용한 reasoning 합성데이터를 생성하고 distillation을 시도해보았다. 하지만 메모리 부족으로 인해, 14B 이상 모델에서는 훈련이 불가능했다.

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

- 모듈화를 하여 config 파일들을 이용해 학습과 추론이 가능했었다.
이를 이용해 다양한 모델들의 성능을 측정해보고 실험결과를 알아낼 수 있었다. config파일만 50개 쌓인 것 같다.
- 모델 파라미터가 커질수록, 확실히 성능이 많이 오르는 것을 확인했다.

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

- 8B 이상의 LLM을 활용해본 적은 처음인데, 4bit 양자화와 LoRa를 이용해 학습하는 과정을 처음으로 제대로 겪어보았다. 72B 모델도 huggingface에서 양자화한 버전으로 제공되고 저장공간도 적다는 것도 이번에 새로 알게되었다.
이렇게 거대한 모델들을 사용해보면서, 모델크기가 커짐에 따라 성능향상이 일어나는 것을 몸소 경험해보았고, 프롬프트 튜닝과 CoT기법을 사용해보면서 성능이 변화하는 것도 확인해볼 수 있었다.
- 협업을 할 때, JIRA를 처음으로 도입해보았다. JIRA를 활용해보니 잘 사용한다면 프로젝트 진행 관리부터 역할 분배까지 잘 할 수 있는 Tool이라고 생각했다. 이번에 활용하면서 본인이 해야할 일을 명확히 정할 수 있어서 효과적이었던 것 같다. 하지만, JIRA의 기능중에 Sprint를 제대로 활용하진 못한 것 같아서 더 잘 활용해보면 좋을 것 같다고 생각한다.

5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 시간이 부족해 RAG를 구현하는 과정에 참여를 못해서 아쉬웠다.

- 모델 훈련이나 추론 과정에서 **OOM**으로 인해 진행하지 못했던 경험이 있다. 좀 더 공부해서 **QLoRA**를 사용하거나 다른 경량화 기법들을 사용해봤으면 좋았을 것 같은데, **LLM**을 본격적으로 쓰는 것은 이번이 처음이라 시간도 부족했고 많은 다양한 기법을 수행해보지 못해서 좀 아쉽다.

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

- **RAG**나 기타 경량화기법들을 사용해 볼 것 같고, 프롬프트 튜닝이나 **CoT**도 깊게 가면 더 자세한 기법들이 있을텐데 이번에는 시도해보는데 의의를 두었지만 다음에는 더 깊게 공부하고 진행할 것 같다.

허윤서_T7442

1. 나는 내 학습 목표 달성을 위해 무엇을 어떻게 했는가?

- 이번 프로젝트에서는 생성모델의 **SFT**를 통해 모델이 원하는 출력을 낼 수 있게 해보는 것이 목표였고 모델에 따른 **Instruct Template**를 구성하는 연습과 베이스라인에 있던 **logit** 기반의 방식 학습 뿐만 아니라 생성 기반의 **CLM** 방식 학습을 구현해보았다.

2. 나는 어떤 방식으로 모델을 개선했는가?

- 모델의 개선이 쉽지는 않았다. 제한된 자원에서 기존의 **logit** 방식의 학습에는 한계가 있어보였고 **CoT**나 프롬프트를 통해 모델의 추론과정을 이끌어내려는 시도를 했다. 추론과정을 확인하고 모델이 어떻게 문제를 해결하는 지를 바탕으로 모델이 못하는 부분을 분석했다.

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

- 점수 증가폭이 미미했던 기존의 **8B** 모델 **logit** 학습 방식 대비 더 큰 모델의 제로샷만으로도 기존 점수를 웃도는 것을 발견했다. 깨달음까지는 아니지만 좀 더 범용적인 문제를 다루기 위해서는 문제를 나누어서 해결하거나 아니면 이미 대규모 코퍼스로 사전학습된 모델에게서 답을 원하는 형태로 이끌어 낼 수 있도록 하는 것이 효과적이라고 생각했다.

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

- 전과 비교해서 새로운 시도는 **Jira**와 같은 툴을 사용해 협업과정을 좀 더 체계화하여 소통의 장벽을 줄이고 일괄된 방향으로 나아가고자 하였다. 협업과정이 가시화되었고 소통도 지난 프로젝트에 비해 더 잘 되었다고 느낀다. 하지만 일괄된 진행방향과 한정된 시간 내의 완성도있는 프로젝트를 위해서는 개선할 부분이 더 남아있다고 생각한다.

5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 문제마다 유형이 조금씩 달라서 일부 데이터에 성능 향상이 있는 프롬프트를 전체 데이터에 일괄적으로 적용하는 것이 어려웠고, 학습을 할 때는 제한된 메모리와 모델크기와 학습세팅의 적절한 비율을 조정하는 것이 쉽지 않았다. 그리고 학습이나 추론이 오래걸리다보니 결과를 바탕으로 다음 전략을 세우는 과정이 더뎠다. 아쉬웠던 점은 모델의 학습이나 추론을 경량화하지 못해봤다는 점이다. **unsloth**나 **qlora**, 여러 양자화 기법을 사용한 학습을 통해 문제 유형에 따른 전문가 모델을 만들어 봤더라면 의미있는 성능 향상이 있지 않았을까 싶다.

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

- 사실상 이번 프로젝트에서 가장 문제였던 점은 제한된 자원과 모델의 성능을 타협하지 못했다는 점인것 같다. 1등 팀이 발표에서 **unsloth** 환경 세팅을 하다 서버를 8번정도 터트린 경험을 말해주었는데 나는 서버 한 두번 터져서 포기한 경험이 있었다. 그런 태도적인 부분에서 차이가 났다고 생각을 하고 다음에는 안되는 건 없다는 마음가짐으로 최종 프로젝트에 임해야겠다.

전진_T7431

1. 나는 내 학습 목표 달성을 위해 무엇을 어떻게 했는가?

- 질문의 구조를 중점적으로 분석하여 구조적 특징을 기준으로 데이터의 라벨링 및 전처리
- 외부 학습 데이터 소스 참조를 위한 RAG 구현

2. 나는 어떤 방식으로 모델을 개선했는가?

- 답을 잘 내지 못하는 데이터들의 공통된 특징을 찾아 데이터 전처리를 통해 성능 개선 시도
- 여러 RAG 기법에 따른 결과를 분석하여 RAG 성능 개선

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

- 크기가 큰 모델로 프로젝트 방향을 전환하면서 이전 데이터를 기반으로 한 성능 개선은 명확하게 측정하지 못하였음
- RAG 구현 또한 여러 한계점에 부딪히며 유의미한 성능 개선은 이루지 못하였음

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

- 체계적인 프로젝트 진행을 위해 JIRA를 활용
- 일정 관리와 task 분담을 체계적으로 진행할 수 있었고 다른 팀원들이 수행한 내용을 파악하기 용이

5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 성능 개선에 가장 큰 부분을 작용하는 크기가 큰 모델 활용을 늦은 시기에 반영하여, 관련하여 직접적으로 성능을 높일 수 있을만한 task를 수행하는데에 시간이 부족
- 특히 RAG 같은 경우도 시간이 조금 더 주어졌다면 훨씬 좋은 성능으로 구현할 수 있었을 거란 아쉬움이 남음

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

- 기간 전반적인 일정 관리를 더욱 면밀히 계획 및 수행
- 여전히 개인 단위로 분리적인 프로젝트 진행 과정을 개선하여 여러 팀원이 서로 유기적인 협업을 수행

이수진_T7411

1. 나는 내 학습 목표 달성을 위해 무엇을 어떻게 했는가?

- 데이터 이상치 탐지 및 정제
- 데이터 정제 한 후 모델 성능 측정
- 데이터의 카테고리 라벨 별 모델의 취약점 분석
- 프롬프트 튜닝 & 제로샷 CoT 추론

2. 나는 어떤 방식으로 모델을 개선했는가?

- 그동안의 프로젝트에서 데이터의 품질만 개선해도 성능이 향상되는 경우를 보았기 때문에 이번에 데이터 구조를 자세히 보고 데이터를 정제하려고 하였다. 하나하나 살펴보면서 이상치 데이터를 삭제한 후에 기존 **train** 데이터셋과 성능을 비교했을때 **private** 결과 기존 성능이 향상된 것을 볼 수 있었다.
- 프롬프트를 조금씩만 바꿔도 제로샷 CoT의 성능 차이가 있었다. 다양한 언어, 구조의 프롬프트 튜닝을 시도하며 모델 성능 향상을 시킬 수 있었다.
- 데이터의 카테고리 분류를 기반으로 모델이 어떤 카테고리를 잘못추는지를 분석한 결과 **RAG**가 필요하다고 생각했다. **RAG**를 시도해보았지만 **retrieval**을 구현하면서 문서를 가져오는 부분의 성능이 좋지 않았고, 이에 어려움을 느껴 큰 도움이 되지는 못하였다.

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

- 모델의 크기에 따라 학습을 하지 않고 제로샷임에도 큰 성능 향상을 볼 수 있었다. 또한 프롬프트 튜닝으로도 결과가 달라지는 것을 보고 프롬프트 튜닝을 처음 시도해보았고, 더 좋은 프롬프트 튜닝은 무엇일 지 고민하게 되었다.
- **RAG**를 구현하는 과정에 대해서 관련 책을 통해 공부해서 더 좋은 **retrieval** 구현이나 **external knowledge**를 잘 검색할 수 있게 하고 싶다.

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

- LLM을 제대로 처음 사용하다보니 학습해야할 것이 많았다. 제로샷 추론을 처음 해봤고, CoT하는 과정도 처음 알게 되었다.
- **Jira**라는 협업 툴을 처음 시도해보았다. 그동안 서로 학습하는데 집중해왔고, 팀원별로 분업을 하여 진행한 것은 이번에 처음인 것 같다. 협업 과정을 상세히 관리할 수 있고, 피어세션을 통해서 서로의 업무 현황을 공유하기에도 수월했던 것 같다.

5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- **Jira**를 통해 분업을 하여 체계화 시킨 것은 너무 좋았다. 더 개선해야할 부분은 전체적인 일정에 대한 계획을 짜고, 세부적으로 들어갔다면 **RAG**나 CoT에 대한 내용을 좀 더 깊이있게 논의하고 공부하고 구현해보았을 것 같다.
- **Jira**를 사용하면서 서로 분업을 하고 이를 유기적으로 연결하는 노력이 더 필요할 것 같다. 이번

프로젝트에서 큰 문제는 없었지만 최종 프로젝트에서는 서로의 업무가 자연스럽게 연결되어야 완성도 있는 결과가 나올 것 같다.

- 시간이 더 있었다면 **RAG** 관련해서 좀 더 공부하고 개선할 수 있었을 텐데 그 점이 아쉽다.
- 모델 훈련 과정에서 **unsloth**와 같은 경량화 기법을 다양하게 시도해보지 못한 것이 아쉽다. 제한된 자원을 최대한 활용해서 더 큰 모델을 훈련해보지 못한 것이 아쉽다.

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

- 데이터를 정제할 때도 노이즈가 많은 데이터를 보고 이를 어떻게 개선할 수 있을까하는 고민이 좀 더 필요할 것 같다. 단순히 행을 제거하는 것이 아니라 적은 훈련 데이터셋인 만큼 이를 최대한 활용할 수 있는 방법에 대해서 고민이 필요하다.
- 제한된 자원을 최대한 활용해서 모델을 훈련하려는 노력이 참 중요하다는 것을 깨달았다. 현업에서도 효율적으로 자원을 활용해서 모델을 훈련하고 추론하는게 중요할 것 같고, 이를 구현하는 과정에서 다양한 경량화 기법에 대해서 1등팀의 발표를 들으면서 많이 배웠다. 이 경량화 기법을 시도해서 **OOM**의 한계를 극복하려는 다양한 노력을 하고, 방법론을 적극적으로 찾고 싶다.

이금상_T7407

1. 나는 내 학습 목표 달성을 위해 무엇을 어떻게 했는가?
 - 외부 데이터셋을 탐색하고 **AGIEval**, **SAT** 등의 데이터를 이용해 추가 데이터를 구축하였다.
 - 역사 데이터셋을 이용해 **RAG**를 구현하였다.
 - 제로샷 **CoT**를 위해 프롬프트를 작성하고 성능을 분석하였다.
2. 나는 어떤 방식으로 모델을 개선했는가?
 - 수능 문제를 잘 맞추게 하기 위해 해당 데이터셋과 비슷한 유형의 데이터셋을 구축하고, 고품질의 한국어 **AGIEval** 데이터셋을 제작하였다.
 - 모델이 역사 문제를 잘 맞추지 못하는 것을 확인하고 추가적인 역사 데이터를 이용하여 역사 문제에 특화된 **RAG**를 구현하고 성능을 확인하였다.
 - **CoT**를 통해 모델의 성능을 높이려 영어 및 한국어 프롬프트를 작성한 후 성능을 분석하고 개선하였다.
3. 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?
 - **CoT**를 진행하기 위해 여러 프롬프트를 작성하였다. 같은 의미의 프롬프트여도 다른 결과를 가져올 수 있음을 확인하였다.

모델이 잘 이해할 수 있는 프롬프트를 작성하기 위해 길게 작성하기보다는 간결하게 작성하고, 입력 데이터를 확실히 구분할 수 있는 디테일을 포함하여 높은 성능(**0.7975**)을 낼 수 있었다.

단지 많은 정보를 담은 프롬프트보다 모델이 더 잘 이해할 수 있도록 고민하고 작성한 프롬프트를 통해 좋은 결과를 얻을 수 있음을 알게 되었다.
 - 역사 문제에서 성능이 저조한 모델을 개선하고자 **RAG**를 활용하여 사전 지식이 필요한 문제를 해결하려는 시도를 하였다. 결과적으로 대부분의 문제에서 모델이 적절한 문서를 참조하지 못했지만, 일부 문제에서는 문서를 잘 참조하여 정답을 맞추는 것을 확인하였다.

리트리벌 성능을 더욱 향상시켰다면 더 좋은 성과를 얻을 수 있었을 것으로 판단하였다. 또한, 역사 **RAG**는 한정된 문서를 이용해 실험했지만, 위키와 같은 폭넓은 문서를 활용한다면 충분히 좋은 성과를 낼 수 있을 것이라는 가능성을 확인하였다.
4. 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?
 - **CoT**를 배우고 처음 사용해보았다. 이론적으로 성능 향상이 가능하다는 것을 알고 있었지만, 실제로 얼마나 성능이 향상되는지 체감하지 못했었다. 이번 기회를 통해 **CoT**가 실제로 성능 향상을 이끌어낼 수 있음을 확인하였고, 프롬프트의 중요성도 배울 수 있었다.
5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?
 - 데이터셋 구축을 위해 여러 모델로 외부 데이터셋을 번역했으나 번역 품질이 낮았다. 이후 **GPT-4-mini**를 이용해 번역을 진행하기 위해 배치 **API**를 활용하였고, 낮은 비용으로도 좋은 결과물을 얻어낼 수 있었다.

결과적으로 **GPT**를 활용하여 우수한 결과물을 얻었지만, **Llama**와 같은 오픈소스 **LLM**을

효과적으로 사용하지 못한 점이 아쉬웠다.

여러 프롬프트를 활용해 모델의 성능을 끌어올리고자 노력했으나, 성능을 획기적으로 개선하지는 못했다. 프롬프트나 방법론을 어떻게 활용하면 더 나은 결과를 낼 수 있을지에 대한 명확한 답을 얻지 못해 아쉬움이 남는다.

- 역사 **RAG** 구축에 시간이 부족해 추가적인 개선을 진행하지 못한 점도 아쉽다. 문제점을 발견하고 발전 방향을 설정했지만, 프로젝트 마감에 임박해 여러 시도를 하지 못한 점이 아쉬움을 남겼다.

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

- **RAG**를 더욱 발전시켜 향후 프로젝트에 활용하고 싶다.
- **CoT** 프롬프트를 더 많이 실험해보고 싶다.