Wrap-up report

1. 프로젝트 개요

1.1. 프로젝트 배경 및 목적

- 본 프로젝트는 '한국어'와 '시험'이라는 주제에 맞춰서 작은 모델들로 수능 시험을 풀어보는
- **task** 이다.

대부분의 대형 모델들은 한국어에 완벽히 최적화되지 않았음에도 불구하고 수능에서 꽤 높은 성적을 기록 하고 있다. 이를 작은 모델에서도 수행해보는 것이 목적이다.

- **Input**: 입력 데이터는 수능 국어와 사회 과목의 지문 형태를 따른다. 'id', 'paragraph', 'problems', 'question_plus'의 형태로 되어 있고, 각각은 id, 지문, 문제, 보기를 의미한다. problems 에는 'question', 'choices', 'answer'로 구성되어 있고 각각 질문, 선택지, 정답을 의미한다.
- **Output**: 주어진 선택지 중에서 정답을 맞춰야 한다.

1.2. 데이터셋 특성

1.2.1 데이터셋 구성 수능형 문제 수능의 국어, 사회

- 영역(윤리, 정치, 사회)과 비슷한 문제
  - KMMLU Korean History), MMMLU HighSchool 데이터 중 역사, 경제, 정치, 지리, 심리)
- KLUE MRC 경제, 교육산업, 국제, 부동산, 사회, 생활, 책마을)

1.2.2 데이터셋 세부 정보

- 학습 및 평가 데이터 파일의 header: id, paragraph, problems, question_plus problems 안에는
- 'question', 'choices', 'answer' key 를 가진 dictionary 가 들어있습니다. 예시) {'question':

  '상소한 인물이 속한 붕당에 대한 설명으로 옳은 것만을 모두 고르면?',

  'choices': ['ㄱ, ㄴ', 'ㄱ, ㄷ', 'ㄴ, ㄹ', 'ㄷ, ㄹ'], 'answer' 2}
- 각 header 와 key 가 해당하는 값은 다음과 같습니다.
  - paragraph: 지문
  - question: 질문
  - choices: 선지 answer:
  - 정답。

    question_plus: 보기

1.2.3 데이터 **EDA**

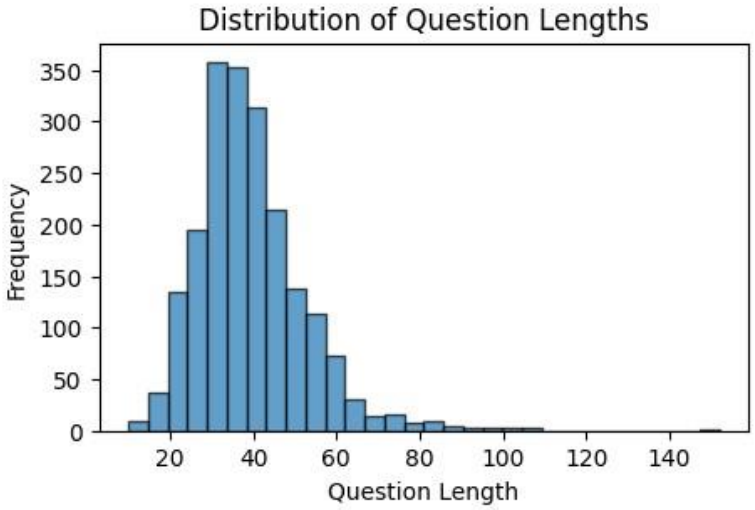
train data 개수는 2031 개이다. test data 개수는 869 개이다.

id	paragraph	question	choices	answer	question_plus
generation-fornlp-425	상소하여 아뢰기를 , “신이 좌참찬 송준길이 올린 차 자를 보았는데 , 상복(喪服)...	상소한 인물이 속 한 붕당에 대한 설 명으로 옳은 것만 을 모두 고르면?	['ㄱ, ㄴ', 'ㄱ, ㄷ', 'ㄴ, ㄹ', 'ㄷ, ㄹ']	2	None
generation-fornlp-426	(가)은/는 의병계 열과 애국계몽 운 동 계열의 비밀결사가 모여 결성된 조직으로, 총사...	(가)에 대한 설명 으로 옳지 않은 것 은?	[고려 문종 때에 남경(南京)으로 승격되었다., 종루(鐘樓), 이현, 칠패 등에서 ...	1	None
generation-fornlp-428	이 날 소정방이 부 총관 김인문 등과 함께 기벌포에 도착하여 백제 군사 와 마주쳤다....	밑줄 친 '그'에 대한 설명으로 옳은 것은?	[살수에서 수의 군 대를 물리쳤다 ., 김춘추 의 신라 왕 위 계승을 지원하 였다 ., ...	2	None

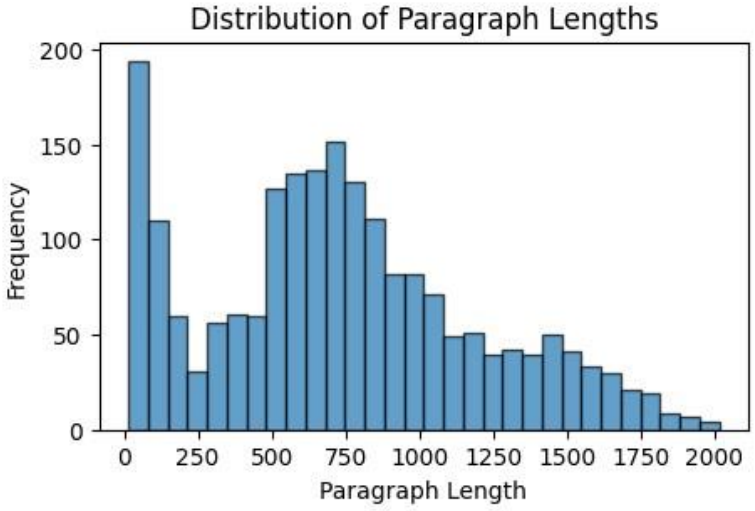
train 데이터 예시 출력은 위와 같다.

```
Missing values in each column:
id              0
paragraph       0
question        0
choices         0
answer          0
question_plus    2031
```

train 데이터 중 존재하지 않는 데이터 개수는 위와 같다. train 데이터에는 question_plus 가 없다는 걸 알 수 있다. test 데이터에도 question_plus 는 없다.

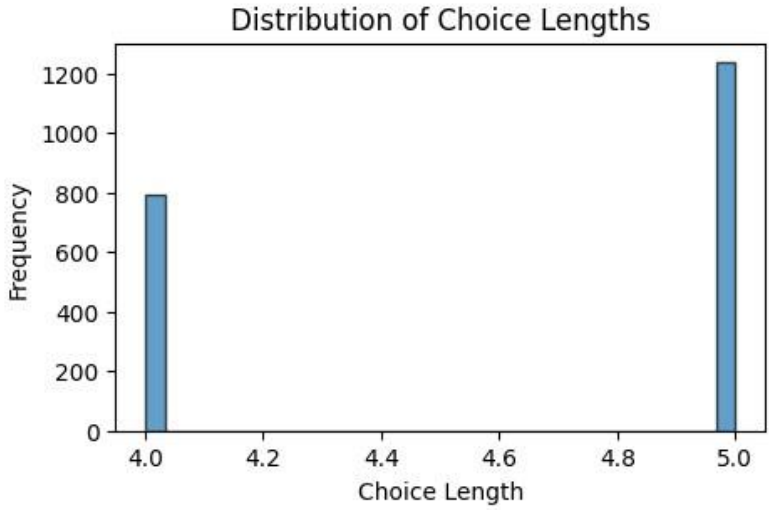


question 의 길이 분포이다. question 을 구성하는 글자 수는 최대 160 글자 이내이며 대부분 데이터는 40 글 자 내외임을 알 수 있다.




```
paragarph length <= 1024: 1535
paragarph length > 1024: 496
min paragarph length: 15
max paragarph length: 2017
```

paragraph 의 길이 분포이다. paragraph 를 구성하는 글자 수는 최대 2017 글자이다.



선택지의 개수 분포이다. 데이터는 선택지가 4 개인 문제와 5 개인 문제로 구성되어 있었으며 선택지가 4 개인 문제는 약 800 개, 선택지가 5 개인 문제는 약 1200 개로 각각 40%, 60% 비율로 존재했다.

<bos>지문을 읽고 질문의 답을 구하세요.<start_of_turn>user 지문:
부산 강서구 김해공항 인근의 '공항마을' 등 그린벨트(개발제한구역)에서 풀린 집단취락(마을) 지역에 상가나 공장을 지을 수 있게 된다. 또 임...주택을 35% 이상 의무적으로 짓도록 한 규제가 사실상 사라지는 등 그린벨트 내 규제가 ...폭 완화된다. 국토교통부는 그린벨트 해 제지역의 개발 사업을 활성화하기 위해 '개발제한구역의 조정을 위한 도시관리계획 변경안 수 립 지침' 및 '도시·군관리계획수립지침'을 이같이 개정해 11 일부터 시행한다고 10 일 발표했다. 개정된 지침은 그린벨트에서 풀린 집단취락이 시가지나 공항, 항만, 철도역 등 거점시설 과 맞닿아 있는 경우 상가나 공장을 지을 수 있도록 했다. 기존에는 자연녹지지역이나 주거지 역으로만 개발이 허용돼 정비사업이 지연됨에 따라 주민의 생활 불편을 초래한다는 지적을 반 영했다. 김정희 국토부 녹색도시과장은 "전국 해제 취락 1656 개(106 ㎢) 가운데 정비가 완료 되거나 진행 중인 곳은 171 개(10%)에 불과하다"며 "이제 부산 공항마을에도 김포공항 인근의 아울렛 같은 쇼핑시설이 들어설 수 있게 돼 정비사업에 속도가 붙을 것"이라고 말했다. 개정 된 지침은 또 그린벨트에서 해제된 땅에 택지개발사업, 공공주택사업 등을 통해 주택을 지을 때 임...주택 용지가 6 개월 넘게 팔리지 않으면 분양주택 용지로 바꿀 수 있도록 했다. 의무적 으로 임...주택을 35% 이상 공급해야 하는 규정을 지키지 않아도 되는 것이다. 이럴 경우 창원 ...전 등의 주택사업에 속도가 붙을 전망이다. 민간업체들의 개발 사업 참여를 장려하는 방안도 담겼다. 민간이 그린벨트 해제 ...상 지역 개발을 위해 설립된 특수목적법인에 출자할 수 있는 비율을 2015 년까지 한시적으로 종전 2 분의 1 미만에서 3 분의 2 미만으로 확...하기로 했다.

질문:
그린벨트 해제지역에서 상가나 공장을 지을 수 있도록 한 개정된 지침의 시행일은 언제인가?

선택지:

1 - 11 일

2 - 10 일

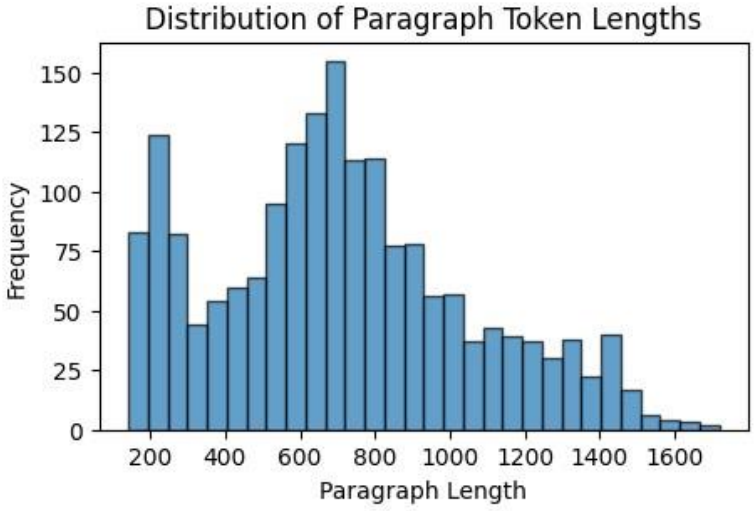
3 - 1 일

```
4· 12 일
5· 15 일

1,2,3,4,5 중에 하나를 정답으로 고르세요.
정답:< end_of_turn >
<start_of_turn>model
1<end_of_turn >
```

베이스라인 코드의 프롬프트를 적용하면 위와 같다. (skip_special_tokens=False 일 때)

train data 에서 eval data 를 추출한 다음 paragraph 토큰 길이 분석은 다음과 같다.




```
max token length: 1718
min token length: 142
avg token length: 708.9025725232622
```

train data 의 paragraph 최대 토큰 길이는 1718 개이다. 평균 토큰 길이는 약 709 개이다.

```
max token length: 1510
min token length: 158
avg token length: 719.7941176470588
```

eval data 의 paragraph 최대 토큰 길이는 1510 개이다. 평균 토큰 길이는 약 720 개이다.

1.3. 평가 방법

본 프로젝트는 수능형 문제를 푸는 task 이다. 이를 평가하기 위해 metric 은 accuracy 를 사용하였다. accuracy 는 (모델이 맞춘 문제 수) / (전체 문제 수)로 나타낼 수 있다.

2. 프로젝트 팀 구성 및 역할

이름	담당 역할
서재덕	프로젝트 일정 관리, 코드 리팩토링/관리, DVC,
이상의	Model 탐색, Quantization, Prompt Engineering, Hint Generation, RAG, Model Merging, Question Classification
김정석	유사 시험 데이터셋 수집 및 전처리
원호영	Code Modularization, w, Model Search
손익준	유사 시험 데이터셋 전처리

3. 프로젝트 수행 절차 및 방법

3.1 프로젝트 시작 전 목표/방법론 설정

3.2 데이터 전처리/증강

- **Shuffling Choices**
  - 이 방법은 선택지에서 정답이 등장하는 번호를 바꾼다.
  - 문제에서 정답은 선택지에 주어지는 텍스트이다. 선택지의 각 번호는 클래스가 아니기 때문에 정답은 어느 번호에도 등장할 수 있다. 따라서 같은 문제일지라도 정답은 기존과 다른 번호에 등장시킨다면 유연하게 학습할 수 있을 것으로 기대하였다. 본 프로젝트에서는 shuffling choices 를 한 번 수행하 였다.

3.3 Techniques

3.3.1 Model Optimization

- LoRA Low-Rank Adaptation)
  - LoRA 는 사전 학습된 LLM 을 적은 resource 로 효율적으로 fine tuning 하기 위한 기술이다.
  - 해당 방법은 LLM 의 가중치는 freeze 하고 저차원의 layer 를 추가해서 해당 layer 를 학습시킨다. 그 로 인해 학습에 사용되는 parameter 수를 크게 줄어들게 되고 학습 시간과 메모리 사용량을 줄일 수 있다.
  - 본 프로젝트에서는 rank 는 6, alpha 는 8, dropout 확률은 0.05 로 설정하였다.
- MixedPrecision Offloading
-
  -
  -

Deepspeed

Quantization

양자화는 모델의 연산 비트수를 줄여서 모델의 크기를 줄이고 메모리 사용량도 줄일 수 있는 방법이 다.

- LLM 은 parameter 수가 billion 단위로 비대해지고 있어 그만큼 메모리 사용량 또한 기하급수적으로 늘어나고 있다. 하지만 그에 따라 필요한 gpu 수도 늘어나기 때문에 큰 비용 문제를 겪게 된다. 해당 문제를 해결하기 위해 모델 경량화가 필요하며 양자화는 경량화하기 위한 방법 중 하나이다. 이를 적 용하기 위한 라이브러리로 bitsandbytes 가 있으며 8bit, 4bit 양자화를 지원한다.
- 본 프로젝트에서는 INT4 로 양자화된 모델인 **Qwen2.5 32B Instruct-GPTQ Int4** 를 사용하였다.

3.3.2 Prompt Engineering

Prompt engineering 은 LLM 을 효과적으로 활용하기 위해 입력을 설계하는 방법론이다.

적절한 형식, 구문, 단어 및 기호 등을 사용하여 입력을 구성함으로써 LLM 이 적절한 출력을 생성할 수 있다.

본 프로젝트에서는 구체적인 질문, 질문과 연관된 지문, 질문에 대한 정답을 포함하는 선택지, 정답을 유추할 수 있는 힌트, 제약 조건 등을 명시하여 정확한 정답을 생성하도록 유도하였다.

3.3.3 Hint Generation

Hint generation 은 주어진 수능형 문제를 풀 때 도움이 되는 힌트를 생성하는 과정이다.

힌트를 제공함으로써 LLM 의 추론 능력을 향상시키고 수능형 문제 해결 능력 향상을 꾀해볼 수 있다. 본 프로젝트에서는 **Qwen2.5 32B Instruct-GPTQ Int4** 를 활용해 힌트를 생성하였다.

3.3.4 Model Merging

Model merging 은 여러 task 에 fine tuning 된 모델을 병합하여 성능을 향상시키는 방법이다. 최근에 제안된 방법으로 DARE 과 TIES 가 있다.

- DARE Drops And REscales)
  - DARE 는 모델 간의 차이점을 나타내는 delta parameter 에서 중복된 정보를 제거하고 rescale 하는 방법이다. delta parameter 중 중요한 정보만 남김으로써 각 모델이 fine tuning 된 task 에서 보인 성능을 병합된 모델에서도 유지할 수 있다.
- TIES TRIM, ELECT SIGN
  - TIES 도 DARE 와 마찬가지로 중복된 parameter 들을 제거하는 방법이다. Trim, Elect Sign, Merging 총 3 단계로 진행된다. Trim 단계에서는 fine tuning 된 모델들의 parameter 들 중에서 변 화가 작은 parameter 들을 초기값으로 reset 한다. Elect Sign 단계에서는 paramter 들의 변화가 큰 쪽으로 merging 방향을 정한다. Merging 단계에서는 앞서 정해진 방향과 일치하는 parameter 들 만 가지고 병합한다.

- DARE TIES
  - DARE TRIES 는 DARE 방식에 TIES 의 Elect Sign 을 결합한 방식이다.

본 프로젝트에서는 shuffling choices 를 사용한 모델과 사용하지 않은 모델을 DARE TIES 방식으로 병합하 였다.

3.3.5 RAG

RAG Retrieval-Augmented Generation)은 외부의 지식을 이용해 LLM 에 입력되는 프롬프트의 맥락을 더 풍부하게 만드는 기술이다.

본 프로젝트에서 사용된 데이터는 수능 관련 문제 영역은 주로 국어와 사회 영역이다. 특히 사회 영역은 문제에 서 주어지는 지문의 정보 뿐만 아니라 암기하고 있는, 외부의 정보를 요구하기도 한다. 이를 해결하기 위해 RAG 가 사용될 수 있다.

본 프로젝트에서 RAG 에 사용되는 지식베이스는 wiki 합성 데이터 및 KoBEST 로 구성하였다. 대량의 문서 데 이터에 대한 검색 속도를 개선하기 위해 Faiss 를 적용했다. 문서 embedding 모델은 KoELECTRA, 최종 답 변 생성 LLM 은 INT4 로 양자화된 모델인 **Qwen2.5 32B Instruct-GPTQ Int4** 를 본 프로젝트에 사용된 데 이터에 fine tuning 하여 사용하였다.

3.3.6 Question Classification

본 프로젝트에서 사용된 데이터는 수능 관련 문제 영역은 주로 국어와 사회 영역이다. 사회 영역 문제의 경우 외부 지식이 필요할 때가 많지만 국어 영역 문제의 경우 대부분 문제에 나타난 지문의 정보만 가지고 정답을 찾 을 수 있다. 만약 RAG 를 사회 영역 문제 뿐만 아니라 국어 영역 문제에도 사용하게 된다면 불필요한 정보가 RAG 로부터 추출됐을 때 LLM 이 주어진 정보에 혼동을 일으킬 수 있다. 따라서 각 문제를 국어 영역 문제와 사 회 영역 문제로 분류할 필요가 있다.

본 프로젝트에서는 **Qwen2.5 32B Instruct-GPTQ Int4** 를 분류 모델로 사용해 각 문제를 국어 영역 문제 와 사회 영역 문제로 나누도록 하였다. 국어 영역 문제일 경우 1 을, 사회 영역 문제일 경우 0 을 출력하도록 하여 RAG 를 사회 영역 문제에만 적용하도록 했다.

4. 프로젝트 수행 결과

4.1. 리더보드 결과

mid test 에선 accuracy 0.7627 를 달성하며 10 위를 기록했다.

final test 에선 accuracy 0.7103 를 달성하며 11 위를 기록했다. mid test 에 비해 accuracy 가 0.0524 감소하였다.

4.2 모델 실험 결과

Model	**train**	**test**	**Accuracy(mid test)**	**Accuracy(final test)**
beomi/gemma-ko2b 3epoch)	baseline	baseline	0.4032	0.3793
meta-llama/Llama3.2 1B Instruct 2epoch)	Prompt	Prompt	0.3733	0.4161
meta-llama/Llama3.2 1B Instruct 2epoch)	Prompt	Prompt Hint1	0.3963	0.4483
meta-llama/Llama3.2 1B Instruct 2epoch)	Prompt Hint1	Prompt Hint1	0.4631	0.4368
meta-llama/Llama3.2 1B Instruct 3epoch)	Prompt Hint1	Prompt Hint1	0.4516	0.4690
Qwen/Qwen2.5 32B Instruct-GPTQ Int4	Prompt	Prompt	0.7604	0.7103
Qwen/Qwen2.5 32B Instruct-GPTQ Int4	Prompt	Prompt Hint2	0.7604	0.7103
Qwen/Qwen2.5 32B Instruct-GPTQ Int4	Prompt	Prompt Hint2 RAG KoELECTRA	0.7373	0.7172
Qwen/Qwen2.5 32B Instruct-GPTQ Int4	Prompt	Prompt Hint2 RAG MPNet)	0.7465	0.6851
Qwen/Qwen2.5 32B Instruct-GPTQ Int4	Prompt	Prompt Hint2 RAG MPNet) Question Classification	0.7488	0.7034
Qwen/Qwen2.5 32B Instruct-GPTQ Int4	Prompt	Prompt Hint2 Model Merging	0.7627	0.7080

Qwen/Qwen2.5 32B Instruct-GPTQ Int4	Prompt	Prompt Hint2 Model Merging RAG MPNet) Question Classification	0.7465	0.7034

표에서 Hint1 은 maywell/EXAONE 3.0 7.8B Instruct-Llamafied 으로 생성한 힌트이다. Hint2 는 Qwen/Qwen2.5 32B Instruct-GPTQ Int4 로 생성한 힌트이다.

Llama-3.2 1B 모델에서 2epoch 로 학습했을 때 test 에만 Hint1 을 사용한 경우가 Hint1 을 사용하지 않은 경 우보다 final 에서 0.0322 더 높은 점수를 기록했다. train 과 test 둘 다 Hint1 을 사용한 경우가 test 에만 Hint1 을 사용한 경우보다 final test 에서 0.0115 낮은 점수를 기록했지만 3epoch 로 학습했을 때 train 과 test 둘 다 Hint1 을 사용한 경우가 다시 final 에서 2epoch 로 학습한 경우보다 final 에서 0.0322 더 높은 점수를 기록했 다.

Qwen/Qwen2.5 32B Instruct-GPTQ Int4 모델에서 학습할 때 2epoch 로 고정하고 test 에만 Hint2 를 사용하여 실험했다.

Hint2 만 사용했을 때 Hint2 를 사용한 경우와 사용하지 않은 경우 성능에서 차이가 없었다.

Hint2 RAG 만 사용했을 때 embedding model 로 MPNet 을 사용한 경우가 KoELECTRA 를 사용한 경우보다 mid test 에서 0.0092 더 높은 점수를 기록했지만 KoELECTRA 를 사용한 경우가 MPNet 을 사용한 경우보다 final test 에서 0.0321 더 높은 점수를 기록했다.

Hint2 RAG Question Classification 만 사용했을 때 Hint2 RAG 만 사용했을 때보다 final test 에서 0.0183 더 높은 점수를 기록했다.

Hint2 Model Merging 만 사용했을 때 Hint2 만 사용했을 때보다 mid test 에서 0.0023 더 높은 점수를 기록했지만 Hint2 만 사용했을 때가 Hint2 Model Merging 만 사용했을 때보다 final test 에서 0.0023 더 높은 점수를 기록했다.

Hint2 Model Merging RAG Question Classification 만 사용했을 때 Hint2 Model Merging 만 사용했을 때보다 final test 에서 0.0046 더 낮은 점수를 기록했다.

위의 결과로 보았을 때 Qwen/Qwen2.5 32B Instruct-GPTQ Int4 모델 같이 큰 모델을 사용한 경우가 Llama-3.2 1B 모델 같이 작은 모델을 사용한 경우보다 매우 높은 성능을 보임을 알 수 있었다.

Hint 만 사용했을 때 Llama-3.2 1B 모델의 경우 2epoch 로 학습하는 것보다 3epoch 로 학습하는 것이 일반 화가 더 잘 되는 것으로 보인다. 또한 더 큰 모델로 생성한 힌트는 작은 모델에 효과적임을 알 수 있다.

Qwen/Qwen2.5 32B Instruct-GPTQ Int4 모델의 경우 힌트를 Qwen/Qwen2.5 32B Instruct-GPTQ Int4 모델로 생성해서 사용했으나 효과가 전혀 없었다. 동일한 모델로 생성한 힌트는 동일한 모델에 효과가 없 는 것으로 생각된다.

Hint2 RAG 만 사용했을 때 KoELECTRA 을 사용한 경우가 MPNet 을 사용한 경우보다 final test 에서 더 높 은 성능을 보인 것은 KoELECTRA 가 MPNet 보다 한국어 데이터셋에 잘 pretraining 된 모델이기 때문인 것으 로 추측된다.

Hint2 RAG Question Classification 만 사용했을 때 Hint2 RAG 만 사용했을 때보다 final test 에서 더 높은 성능이 나왔는데 이는 사회 영역 문제에만 RAG 가 사용돼서 국어 영역, 사회 영역 각각 적절한 프롬프트가 적용됐기 때문인 것으로 보인다.

Hint2 Model Merging 만 사용했을 때 Hint2 만 사용했을 때보다 mid test 에서 점수가 더 높았으나 final test 에서 더 낮은 점수를 보인 것은 병합된 모델이 train data 에 과적합됐기 때문인 것으로 생각된다.

Hint2 Model Merging RAG Question Classification 만 사용했을 때 Hint2 Model Merging 만 사용했을 때보다 final test 에서 점수가 더 낮게 나온 것은 RAG Question Classification 를 사용할 때 task 에 덜 적합한 지식베이스, 문제 분류 방식을 사용했기 때문인 것으로 추측된다.

5. 자체 평가 의견

5.1. 긍정적 요소

본 프로젝트에서 다양한 실험을 통해 다음과 같은 사실을 알 수 있었다.

V100 32GB 1 대의 서버 환경에서 모델 크기가 32B 일지라도 INT4 로 양자화됐다면 모델 전체를 fine tuning 할 수는 없지만 LoRA, RAG 등 사용하는 것이 가능하다.

더 큰 LLM 로 생성한 Hint 는 더 작은 LLM 에 유효하게 작용한다.

다국어 모델보다 한국어에 특화된 모델로 embedding 하면 RAG 를 더 잘 수행할 수

있다. 각 문제를 영역에 맞게 분류를 해서 RAG 를 수행하면 더 높은 성능을 보인다.

5.2. 개선 요소

본 프로젝트에서 train data 개수는 2031 개이며, test data 개수는 869 개이다. 만약 train data 를 더 증강시 키거나 LLM 을 이용한 합성 데이터를 생성해서 더 많은 데이터에 대해 모델을 fine tuning 했다면 성능을 더 개 선시킬 수 있었을 것으로 생각된다. 또한, RAG 에 사용된 지식베이스를 수능형 문제와 연관성 높은 문서들로 구성하고 RAG 에 사용된 embedding model 을 본 프로젝트에 fine tuning 할 수 있었다면 더 좋은 결과를 보 였을 것으로 여겨진다. Question Classification 역시 국어 영역 문제와 사회 영역 문제를 분류할 때 LLM 을 사용했는데 더 적합한 분류 모델을 사용하거나 데이터에 맞게 fine tuning 했다면 개선됐을 것으로 생각된다.

Hint 의 경우 힌트를 적용했을 때와 적용하지 않았을 때 성능 차이가 전혀 없었다. 본 프로젝트에서 사용된 모델 인 **Qwen2.5 32B Instruct-GPTQ Int4** 가 힌트 없이 수능형 문제를 풀 수 있는 모델이거나 동일한 모델로 생성한 힌트는 동일한 모델에 덜 효과적인 것으로 추측된다.

Model Merging 의 경우 오히려 성능이 낮아졌다. 단순히 같은 데이터에서 정답 번호만 바꾼 shuffling choices 한 모델을 병합했기 때문인 것으로 보인다. 이것 역시 다양한 데이터를 학습한 모델들로 병합했다면 좋은 결과가 나왔을 것으로 여겨진다.

6. 참고 문헌

7. 개인 회고

개인 회고 -

이상의

나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

Quantization, Prompt Engineering, Hint Generation, RAG, Model Merging, Question Classification 를 구현하는 것에 중점을 두었다.

양자화를 하기 위해 vLLM, DeepSpeed, Unsloth, bitsandbytes, AutoGPTQ, flash-attention 등, 그리 고 RAG 를 사용하기 위해 Faiss, LangChain 등 다양한 라이브러리들을 알아보았다.

나는 어떤 방식으로 모델을 개선했는가?

Quantization 의 경우 양자화 라이브러리 중 bitsandbytes 를 제외한 다른 라이브러리들은 cuda 관련 패키지 를 설치하는 도중 서버가 터졌다. bitsandbytes 로 4bit 양자화를 시도했고 성공적으로 학습하는 것을 확인하 였다.

Prompt Engineering 의 경우 프롬프트를 구체적인 지시와 제약 조건으로 구성하여 정답을 출력하도록 하였 고 baseline code 에 있던 프롬프트보다 약간 좋은 성능을 보였다.

Hint Generation 의 경우 큰 모델로 생성한 힌트가 작은 모델에 대해 효과가 있음을 확인하였다. 하지만 모델 이 생성한 힌트를 동일한 모델에 주었을 땐 효과가 없었다.

RAG 의 경우 다국어 모델로 임베딩하는 것보다 한국어에 특화된 모델로 임베딩할 때 좋은 성능을 보였다.

Model Merging 의 경우 기존 데이터에 학습한 모델과 기존 데이터에서 정답 번호만 바꾸고 학습한 모델을 병 합했으나 mid test 에선 점수가 약간 높게 나왔다가 final test 에선 더 안 좋은 점수를 보였다.

Question Classification 의 경우 국어 영역 문제와 사회 영역 문제를 분류해서 RAG 를 수행했을 때가 분류하 지 않았을 때보다 더 높은 성능을 보였다. 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

Quantization, Prompt Engineering, Hint Generation, RAG, Question Classification, Model Merging 등 다양한 방법을 구현하고 적용해볼 수 있었다. 큰 모델을 양자화해도 여전히 준수한 성능을 내는 것을 확인할 수 있었고 더 큰 모델이 생성한 힌트가 더 작은 모델의 성능을 높이는데 도움이 되는 걸 알 수 있었

다. RAG 를 적용할 때는 도메인마다 다른 전략이 필요함을 깨달았고 Question Classification 이 RAG 사용할 때의 하나의 전략으로 쓸 수 있음을 알았다. Model Merging 은 일반적으로 더 다양한 task 를 수행한 모델들 에 적합함을 확인할 수 있었다.

전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

Quantization 을 이용해 이전보다 더 큰 모델을 써볼 수 있었고 LangChain 을 활용해 간단하면서 직관적으로 RAG 를 구현 및 수행할 수 있었다. Model Merging 을 통해 모델들을 병합해보고 성능을 확인해볼 수 있었다.

마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

vLLM, DeepSpeed, Unsloth, AutoGPTQ, flash-attention 등 다양한 라이브러리를 사용해보지 못 했다.

다양한 기법들을 적용했으나 성능이 크게 좋아지지 않거나 오히려 안 좋아졌다. 특히 Hint Generation 이 최 종적으로 성능에 영향이 없었고 RAG 의 지식베이스를 적절히 구성하지 못 한 것 같다. Model Merging 을 다 양하게 해보지 못 했고 Question Classification 을 적절히 fine tuning 하지 못 했다.

한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

vLLM, DeepSpeed, Unsloth, AutoGPTQ, flash-attention 등 cuda 를 사용하는 라이브러리들을 설치해 볼 수 있길 기대한다.

LangChain 에 대해 더 깊은 이해를 기대한다.

코드의 재사용성을 높여볼 수 잇길 기대한다.

추가적인 개선 기법에 대해 탐구해보길 기대한다.

개인 회고 - 원호영

시도 및 결과

Code Modularization Baseline code 를 재사용이 용이하게 code 를 Modularization 을 시도하였고, yaml 에서 파인튜닝이 용이하게 수정 가능하게 코드를 재구성 하였고 개인적으로 만족하는 결과를 내었다.

Quantization - 처음에 시도했던 방법은, int8 양자화를 시도했지만 V100 서버에서는 int8 양자화가 지원하 지 않아 사용하지 못하였고, fp16 오프로딩 방법을 적용하였다, 하지만 fp16 오프로딩으로 적용시 7b 모델 이 상은 서버에서 사용하지 못하였고 bitandbytes 를 적용하여 int4 양자화가 사용이 가능하다는것을 알게되어 적용하였고 Qwen-32B-int4 모델을 서버에서 사용할 수 있게되었다

Model Search - 베이스라인 모델 외 성능을 이끌어 낼 수 있는 모델을 서치하였고 만족할만한 성과를 내는 모델을 찾을 수 있엇다.

좋았던 점과 배운 점

Quantization 방법론에 대해 자세하게 알 수 있게 되었고, 서버 환경에 따라 달리 적용해야하는 점을 알게되 었다. 또한, GPU 사용시 어느정도 여유의 GPU 를 남기고 다른 방법론을 시도할 수 도 있다는것도 새롭게 배웠 다.

Code Modularization 적용을 3 명의 팀원이 각각 나누어 진행 하였는대 각 방법들 마다 스타일이 달랐고 새 로운 시각으로 배울수 있어 좋았다.

아쉬운 점

Code Modularization 을 진행하느라 시간을 많이 소모하여 여러가지 방법론을 적용하지 못한게 너무 아쉽 다, 목표한 Quantization 는 진행이 가능했지만, RAG, Prompt Engineering 등 시도해보고 싶은 방법론을 적 용 못한게 아쉽다.

앞으로의 목표

이번 프로젝트를 통해 모델 최적화에대해 중요성을 알게되었고, 앞으로의 기업 해커톤에서 온디바이스에서 사 용가능 할 수 있게 최적화를 잘 진행하는게 목표이다. 빠른 코드 최적화와 철저한 분업으로 일정 기간에 전반적인 틀을 잡고 프로젝트를 진행하는것이 목표이다.

개인회고 - 서재덕

프로젝트 목표

수능형 문제를 해결하는 모델을 구현하는 Task 이다. 새 팀 결성 후 첫 프로젝트인 만큼 협업 강화에 중점을 두 었고, LLM 을 효과적으로 활용하기 위해 모델 경량화를 주된 목표로 설정하였다. 시도 및 결과

- 코드 리팩토링

  Huggingface Transformers 프레임워크 기반의 Baseline 코드를 객체지향 원칙에 맞춰 EDA, Train, Inference 모듈로 분리하였다. 또한, JSON 형식의 Config 파일을 통해 실험 관리를 용이하게 하였다.

  - 타 팀원과 작업물을 쉽게 Merge 할 수 있었고, **Label Normalization** 및 가속 라이브러리 추가 등 확장성 개선에도 긍정적인 영향을 미쳤다.

- **Quantization**

  QAT Quantization Aware Training) 방식과 PTQ Post-Training Quantization) 방식을 모두 시도 하여 경량화와 모델 성능 간의 균형을 도모하였다.

  - QAT 는 경량화와 함께 성능 유지에 강점이 있었으나 훈련 시간이 늘어나는 단점이 있었고, PTQ 는 훈 련 없이 간단히 적용 가능했으나 성능 저하가 관찰되었다.

- **Deepspeed**

  ZeRO Zero Redundancy Optimizer)를 활용한 Offloading 을 중심으로 분산 처리 환경을 구성하였다.

- Upstage 서버 환경에서 설정 난이도가 높았으나, 메모리와 연산량을 효율적으로 줄일 수 있었다. 그 러나 기대했던 성능 최적화 효과는 제한적이었다.

좋았던 점과 배운 점

- 경량화 및 가속 기술 습득

  Quantization, Offloading 등 다양한 모델 경량화 및 학습 가속 기법을 실제 프로젝트에 적용해보며 효율적인 모델 학습 관리를 경험할 수 있었다.

- 협업 및 코드 관리 코드 모듈화와 실험 관리 체계를 통해 협업 효율성을 높였고, 이 과정에서 팀원들과의 원활한 커뮤니케이 션과 프로젝트 방향성 설정의 중요성을 배웠다.

아쉬운 점

- 초기 계획 부족 프로젝트 관리 툴 선정에 과도한 시간을 소모하여 프로젝트를 체계적으로 관리하지 못했고, 초기 구성 지 연으로 인해 진행 속도가 더뎌졌다.
- 문서화의 부족 전반적인 문서화 부재로 인해 작업 공유와 진행 상황 파악에 어려움을 겪었다. 이는 협업 과정에서 혼란을 유발하고, 작업 중복이나 누락으로 이어지기도 했다.

앞으로의 목표

- 프로젝트 관리 역량 강화 초기 기획 단계에서 명확한 방향성을 설정하고, 효율적인 관리 도구를 빠르게 선정하여 프로젝트의 진행 속도를 높일 계획이다. • 문서화 시스템 확립 작업 과정을 세분화하여 기록하고, 가이드라인과 공유 문서를 작성해 작업 간소화와 협업 효율성을 증대시 키겠다.
- 효율적인 모델 경량화 탐구

  QAT, PTQ, Offloading 외에도 다양한 경량화 기법을 연구하고 적용하여, 실질적인 성능과 자원 효율성을 동시에 극대화할 방안을 모색하겠다.