SK네트웍스 **Family AI** 과정 **13**기

데이터 전처리 인공지능 학습 결과서

산출물 단계	데이터 전처리
평가 산출물	인공지능 학습 결과서
제출 일자	2025.08.08
깃허브 경로	https://github.com/SKNETWORKS-FAMILY-AICAMP/SKN13-FINAL-3TEAM.git
작성 팀원	강지윤, 기원준, 우민규, 전진혁, 최호연

1. 모델 비교 및 선정 이유

- 비교 대상 모델:

모델명	종류	선정 이유
gemma-2b-it	Transformer 아키텍처	구글 개발, 우수한 한국어 범용성
EXAONE-4.0-1.2B	하이브리드 AI	한국어 특화, 전문 분야(수학/코딩) 추론 강점
kanana-1.5-v-3b	멀티모달 아키텍처	카카오 개발, 한국어 이미지-텍스트 이해
Mistral-7B-Instruct	효율적 Transformer	긴 컨텍스트 처리 및 지시 이행 능력 우수
SOLAR-10.7B-Instruct	심층 확장형 Transformer	복잡한 한국어 지시 이해 및 추론 능력

- 실험 모델 수: 총 5종
- 최종 선정 모델: EXAONE-4.0-1.2B

2. 모델 구조 및 아키텍처

2.1 모델 아키텍처 도식 :

- exaone-4.0.1.2B:

  입력층 → 임베딩층 → Transformer 블록(N개) → 분류기 출력층 (Softmax)

- SD-3.5-Medium :
  텍스트 인코더 (CLIP-G/L, T5-XXL) + Latent representation + MMDiT-X Transformer + VAE 디코더 계층

2.2 구성 요소 설명:

계층명	역할	구성 요소
Embedding	입력 문장을 벡터화	● BBPE tokenizer 사용 ● Vocab size: 102,400 ● 한국어, 영어, 소량의 다국어 토큰으로 구성
Encoder (Transformer block)	의미 표현 학습	● Global Attention ● GQA(Grouped Query Attention) 적용 ● QK-Reorder-LN
Classification Head (Final Layer)	토큰 예측	● 다음 토큰 확률 계산을 위한 Softmax 함수 사용

계층명	역할	구성 요소
입력 텍스트 인코더	텍스트 프롬프트를 의미론적 임베딩 벡터로 변환. 여러 인코더를 사용해 텍스트 길이, 디테일, 컨텐츠 복잡성에 유연하게 대응	● CLIP-ViT/G, CLIP-ViT/L:짧은 문장 내 의미와 스타일을 파악하므로 기초적 이미지 스타일/콘셉트 이해 ● T5-XXL : 긴 프롬프트 처리를 담당해, 풍부한 디테일과 명령 정확도를 보강
패치화 + 위치 임베딩	이미지 라텐트를 Transformer 처리에 적합한 시퀀스로 변환하고 공간적 위치 정보를 보존하도록 준비.	노이즈 라텐트 → 패치화 (2x2) → 선형 변환 → 위치 임베딩 추가
MMDiT-X Transformer 블록	텍스트와 이미지 정보를 함께 처리해 프롬프트에 일관된, 고해상도 이미지를 생성. Dual-attention은 텍스트와 이미지 임베딩을 결합한 joint attention을 구현, QK-Norm은 학습 안정성을 제공. Mixed-Resolution 학습으로 다양한 해상도 대응력을	● 초기 12~13개 레이어에 **Dual-attention / Self-attention** ● **QK-Normalization** 기법 ● **Mixed-Resolution Training** (256 → 512 → 768 → 1024 → 1440), 확장된 384×384 포지셔널 임베딩, 랜덤 크롭

	확보..	
모듈화 및 라텐트 출력	Transformer에서 처리된 라텐트에 조건 정보(텍스트 임베딩 등)를 조절 방식으로 첨가한 후, 원래 공간 형태로 복원하는 중간 단계	Transformer 출력 → Modulation (→ 선형 변환 → Unpatching (패치 병합)
VAE 디코더 계층	최종적으로 latent 표현을 픽셀 이미지로 복원해 출력 이미지를 생성	• 16채널 VAE 디코더

3. 학습 설정 및 하이퍼파라미터*

3.1 Exaone 4.0

항목	값
학습 데이터 수	266건
검증 데이터 수	50건
에폭(Epoch) 수	5
배치 크기 (Batch Size)	4 (GPU 당), Gradient Accumulation: 6 → 총 24개 샘플 기준으로 1 step
학습률 (Learning Rate)	2e-4
옵티마이저	AdamW (Trainer 기본값)
손실 함수	CrossEntropyLoss (Causal LM 기본)
조기 종료 기준	없음 (Early stopping 미사용)

3.2 SD-3.5-medium

항목	값
학습 데이터 수	100개
검증 데이터 수	10개
에폭(Epoch) 수	2
배치 크기 (Batch Size)	1
학습률 (Learning Rate)	1e-4
옵티마이저	adamw_bf16
손실 함수	L2 Loss (MSE)

조기 종료 기준	없음 (Early stopping 미사용)

항목	값
학습 데이터 수	100개
검증 데이터 수	10개
에폭(Epoch) 수	5
배치 크기 (Batch Size)	1
학습률 (Learning Rate)	1e-4
옵티마이저	adamw_bf16
손실 함수	L2 Loss (MSE)
조기 종료 기준	없음 (Early stopping 미사용)

4. 학습 결과 및 성능 평가*

4.1. 학습 결과 요약

4. 1. 1. Exaone 4.0

4. 1. 1. 2. BERTScore - Base Model

Model	**Precision**	**Recall**	**F1 Score**
EXAONE-4.0-1.2B	43.12	37.32	39.82

4. 1. 1. 2. BERTScore - after fine-tuning

Model	**Precision**	**Recall**	**F1 Score**
EXAONE-4.0-1.2B	57.05	58.92	57.83

4. 2. 1. SD-3.5-medium

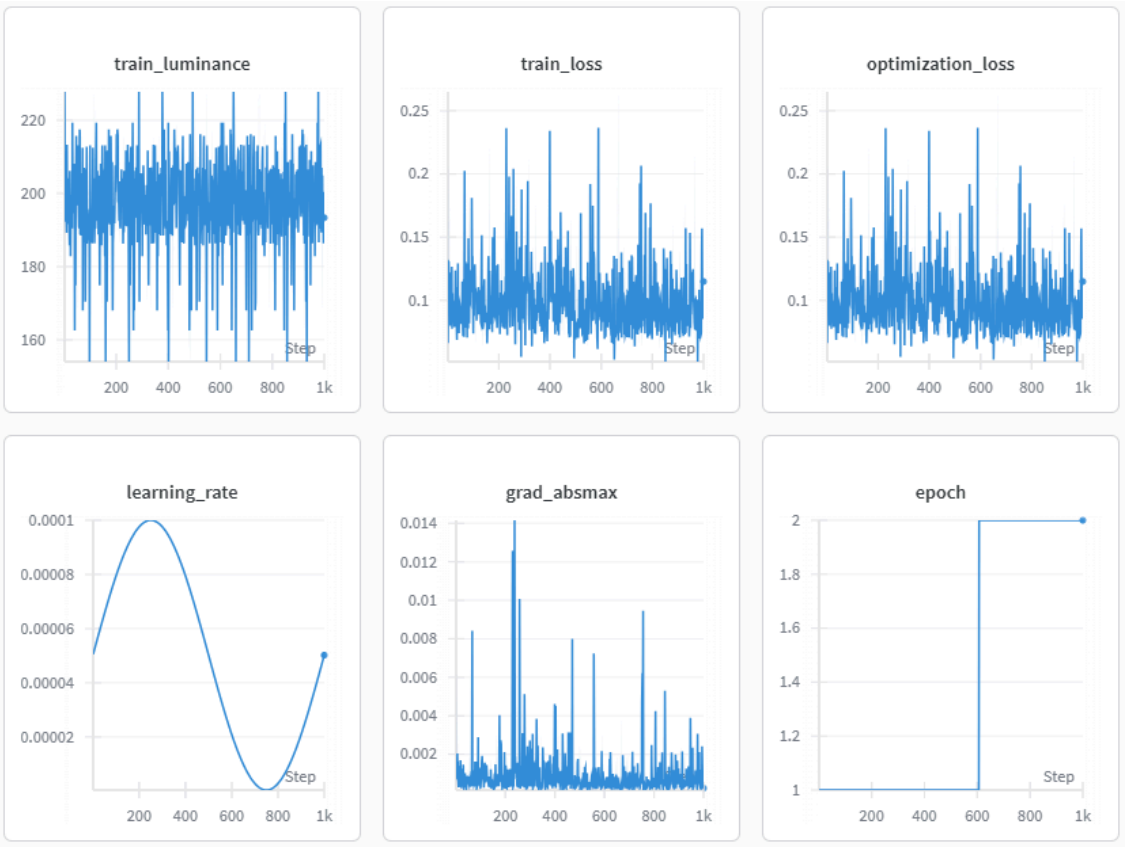
4. 2. 1. 1. 2epoch, 5epoch 진행 결과.

Model	CLIP	FID
Base	26.6794	7.9497
2 epoch train	27.7876	4.0336
5 epoch train	27.0537	8.6304

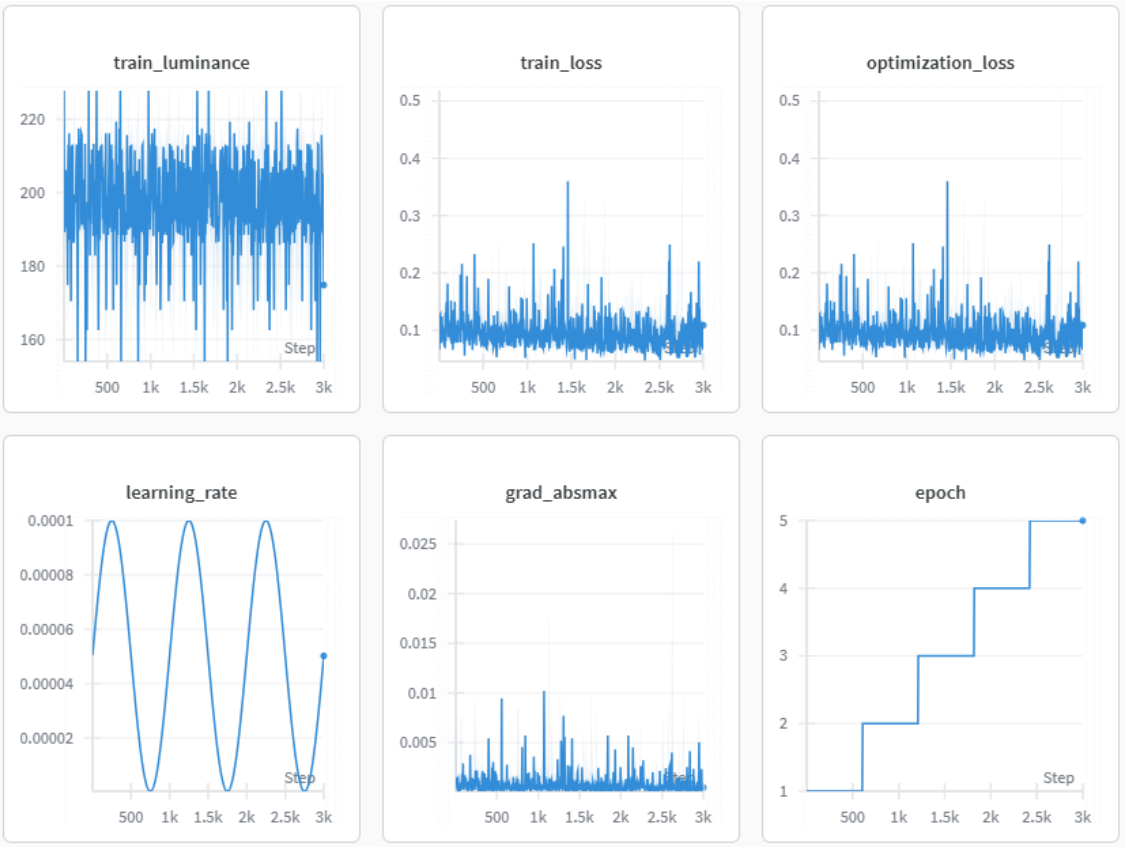
4.2. 그래프 (선택): 학습/검증 loss 변화, accuracy 변화 등

(예: matplotlib 또는 캡처 첨부)

2 epoch)



5 epoch)



4.3 해석 및 분석

- 전체적으로 파인튜닝 후 BERTScore가 상승하여 성능 개선은 있었지만, 여전히 만족스러운 수준은 아님.
- 이는 모델 자체의 적은 파라미터 수, 혹은 적합하지 않은 튜닝 rate를 적용했다고 생각됨.
- 답변이 여전히 간단하고 표면적인 수준에 머무르는 경우가 많음. 보다 상세하고 심층적인 정보가 필요한 질문에 대해서는 충분한 깊이를 제공하지 못해 모델을 더 파악해 볼 필요.

5. 과적합**/**과소적합 대응*****

- 적용 기법:

기법	설명	적용 여부
Dropout	과적합 방지	O (0.1 사용)
조기 종료 (Early Stopping)	성능 저하 시 종료	O
학습률 감소	Plateau 시 자동 감소	O (ReduceLROnPlateau 사용)
교차 검증	데이터 분산 고려	X (단일 validation set 사용)

- 결과: 학습/검증 간 loss 차이가 적고 성능 안정

6. 결론 및 향후 계획*

- 최종 선정 모델: EXAONE-4.0-1.2B
- 활용 방안: 사용자 챗봇, 이미지 생성에 알맞는 프롬프트 생성,
  파인튜닝된 답변(기업 특화 데이터) 제공

- 향후 계획:
  - 데이터셋 확장 및 성능 개선
    - 학습 데이터셋 보강: 현재 데이터셋에 새로운 질문-답변 페어를 추가로 생성하여 모델의 답변 품질과 다양성을 높일 계획입니다.
    - 파인튜닝 최적화: 미미했던 성능 향상의 원인을 분석하여, 모델의 파라미터 수와 튜닝 레이트 등을 재조정하여 파인튜닝 효과를 극대화할 것입니다.
  - 서비스 활용 및 확장
    - COT(Chain of Thought) 적용: 모델의 추론 과정을 개선하는 COT 기법을 도입하여 더 복잡하고 논리적인 답변을 생성할 수 있도록 연구할 계획입니다.
    - REST API 서비스화: 학습된 모델을 웹에 연동하고 REST API로 서비스화하여 다양한 플랫폼에서 손쉽게 활용할 수 있도록 할 것입니다.

7. 부록*

- 전체 학습 로그 캡처 또는 파일 (예: TensorBoard, wandb 스크린샷)
- Github 링크
- Exaone4.0 파인튜닝(adapter_config.json)


```
{
 "alpha_pattern": {},
 "auto_mapping": null,
 "base_model_name_or_path": "./exaone_4.0_1.2b",
 "bias": "none",
 "corda_config": null,
 "eva_config": null,
 "exclude_modules": null,
 "fan_in_fan_out": false,
 "inference_mode": true,
 "init_lora_weights": true,
 "layer_replication": null,
 "layers_pattern": null,
 "layers_to_transform": null,
 "loftq_config": {},
 "lora_alpha": 16,
 "lora_bias": false,
 "lora_dropout": 0.05,
 "megatron_config": null,
```

```
  "megatron_core": "megatron.core",
  "modules_to_save": null,
  "peft_type": "LORA",
  "qalora_group_size": 16,
  "r": 8,
  "rank_pattern": {},
  "revision": null,
  "target_modules": [
    "up_proj",
    "down_proj",
    "v_proj",
    "o_proj",
    "k_proj",
    "gate_proj",
    "q_proj"
  ],
  "target_parameters": null,
  "task_type": "CAUSAL_LM",
  "trainable_token_indices": null,
  "use_dora": false,
  "use_qalora": false,
  "use_rslora": false
```

```
    }
```