25. 12. 07

강화학습의 기초 - 팀 프로젝트 보고서

논문 교정 AI 에이전트
최적화 강화학습

서강대학교 AI · SW 대학원 데이터사이언스 · 인공지능전공

A72040 김진산 | A72080 차시명 | A72085 한다현

팀명 : 외유내강 (겉으론 부드러워 보이나, 내가 강화되고 있음)

https://github.com/itcasim0/reinforcement-with-llm

CONTENTS

프로젝트 개요

주제 및 목표

프로젝트 주제

- LLM 기반 논문 교정 AI 에이전트의 **최적화를 위한 강화학습**

프로젝트 목표

- 논문 교정을 위한 다양한 AI 에이전트 간의 상호작용을 **강화학습을 통해 자동화**
- 정확성, 속도, 비용을 **균형적으로 최적화**
- 상황에 맞는 **최적의 교정 액션** 학습

프로젝트 개요

문제 정의

배경

LLM 기반 AI 에이전트 서비스의 증가

- 최근 LLM 기반의 AI 에이전트 서비스 또는 시스템이 많이 개발되고 있음
- AI 에이전트 서비스란, 특정 작업에 대해 다양한 AI 에이전트 간의 상호작용을 통하여 사용자가 원하는 결과를 가져다 주는 서비스
- 이러한 상호작용을 효과적으로 이끌어내는 것이 서비스 성공의 핵심

서비스 구조

논문 교정 AI 에이전트의 운영 방식

- 본 프로젝트는 논문을 교정하는 AI 에이전트 서비스에 초점을 맞춤
- 문법 교정, 명확성 개선, 간결성 등 다양한 교정 작업에 특화되어 있는 에이전트가 입력된 논문을 순차적으로 작업하여 품질 개선 진행

문제 구조

순차적 의사결정 기반 교정 과정

- 한 번의 프롬프팅으로 교정하는 방식이 아닌, 논문 상태에 따라 여러 교정 작업 중 하나를 단계적으로 선택하는 방식으로 접근함
- 이러한 단계적 작업 선택 과정을 강화학습으로 학습해, 각 단계에서 가장 적합한 작업을 고르는 정책을 모델이 익힘

프로젝트 개요

핵심 과제

효율

LLM 교정 작업의 비용 (API 비용),
레이턴시를 고려한 효율적인 전략 학습

최적

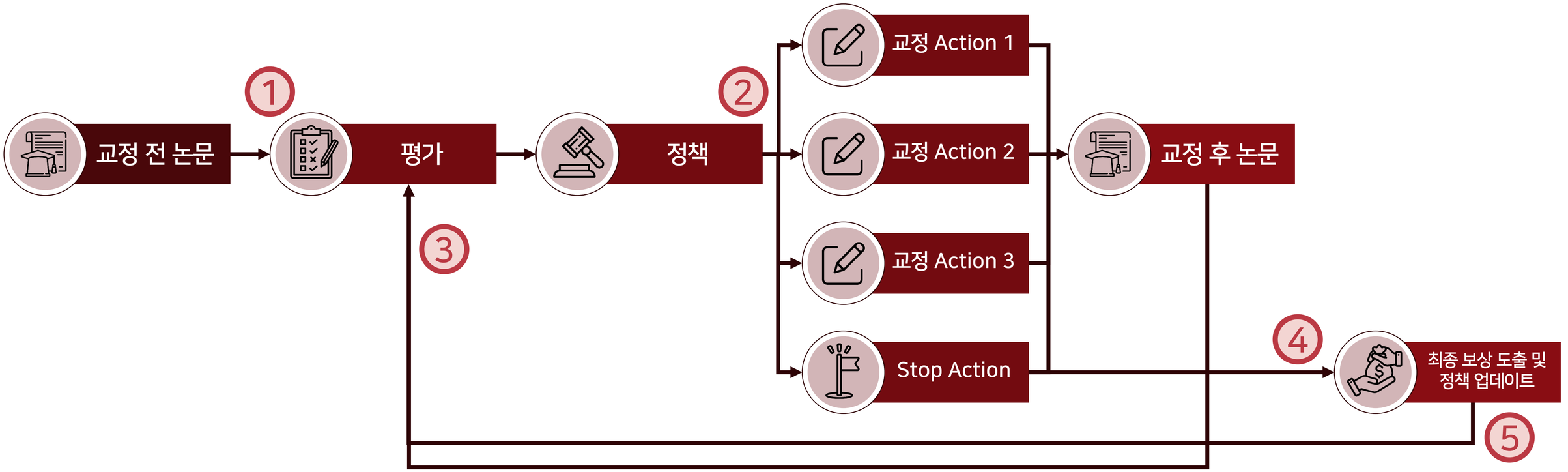
여러 교정 작업의 조합 및 순서 최적화

자동

논문 품질 평가 지표를 기반으로 한
자동화된 의사결정

강화학습 환경

학습 프로세스



① 교정 전 논문을 적절한 **평가 기준에 따라 품질 점수 도출**

② 품질 점수를 포함한 state로 **학습된 정책을 사용하여 action 결정**

③ action에 따라 **교정 된 논문 재평가**

④ 1~3 과정을 반복하다가, stop 조건에 따라 **에피소드를 종료하고 최종 보상 도출**
stop 조건 1 : stop editing action 선택
2 : 최대 스텝 수 도달 (max = 3)
3 : 품질 임계값 도달 (전체 품질 >= 9.5)

⑤ 최종 보상에 따라 **정책 업데이트 진행**

강화학습 환경

state

state 차원 : 논문 품질 점수 + 현재 진행 단계 + 이전 액션

Structure (구조적 완성도)

✓ 핵심 구성요소(배경·방법·결과)의 균형
✓ 문단 간 자연스럽고 일관된 논리 흐름

Length (문장 길이)

✓ 문장당 평균 길이의 적절성
✓ 과도하게 긴 문장에 대한 패널티

Academic Style (학술적 스타일)

✓ 형식적·객관적 문체 유지
✓ 구어체·비학술적 표현 배제

Information Density (정보 밀도)

✓ 핵심 정보·개념의 압축도
✓ 불필요한 수식어·중복 표현 제거

Clarity (명확성)

✓ 모호한 표현 없이 명확한 의미 전달
✓ 불필요한 반복 및 애매한 문장 제거

Overall (전체 품질)

✓ 상위 항목 가중치를 반영한 종합 점수
✓ 초록 전반적 완성도 평가

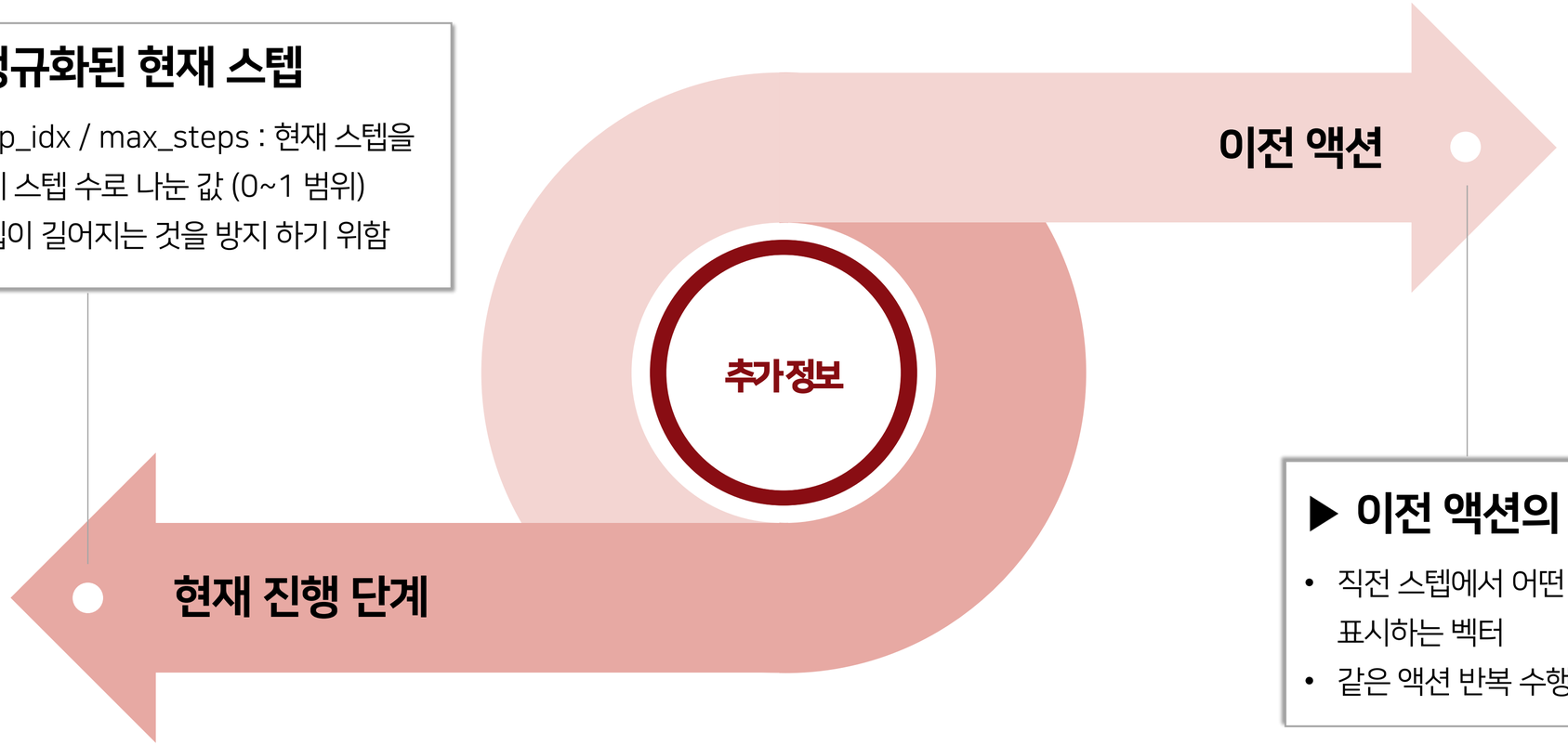
강화학습 환경

state

state 차원 : 논문 품질 점수 + 현재 진행 단계 + 이전 액션

▶ 정규화된 현재 스텝

- step_idx / max_steps : 현재 스텝을 전체 스텝 수로 나눈 값 (0~1 범위)
- 스텝이 길어지는 것을 방지 하기 위함



▶ 이전 액션의 원-핫 인코딩

- 직전 스텝에서 어떤 액션을 수행했는지 표시하는 벡터
- 같은 액션 반복 수행을 방지 하기 위함

강화학습 환경

action

에이전트는 입력된 논문을 수정하기 위해 6개 중 1개의 프롬프트 전략 선택

fix_grammar

문법/맞춤법 오류 교정

01

improve_clarity

표현/구조의 명확성 개선

02

make_concise

중복/군더더기 제거 + 간결화

03

improve_structure

논리 흐름과 같은 구조 개선

04

make_academic

학술적 표현으로 변경

05

stop_editing

교정 종료

06

강화학습 환경

reward

Step Reward (step 마다 보상)

품질 변화량 보상

- 각 평가 지표의 변화량 ÷ 2
  (10점 스케일 조정을 위함)
- 6개 지표의 평균 계산
- 개선 시 : 변화량 그대로
- 악화 시 : -0.2 × |변화량|

비용 패널티

- cost_lambda
  × used_cost_usd
- 기본 cost_lambda = 1.0

step 패널티

- current_step
  × step_penalty
- 기본 step_penalty = 0.1

반복 패널티

- 에피소드 내 동일 액션
  재사용시 : -repeat_penalty
- 기본 repeat_penalty = 0.2

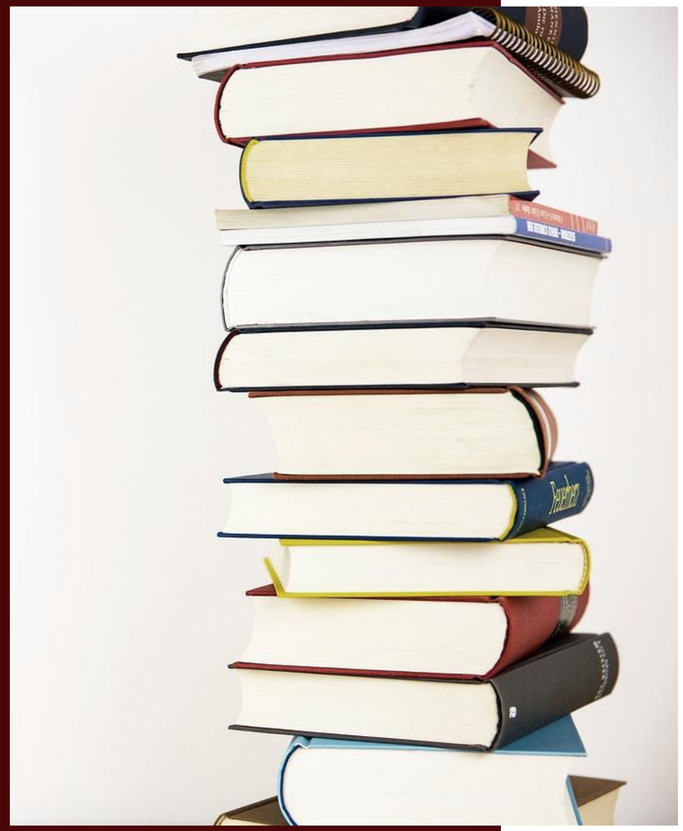
Terminal Reward

종료 보상

- terminal_reward =
  (avg_score – 5.0) ÷ 5.0
  (-1.0~1.0 범위)
- 5개 평가 기준의
  평균(overall)이
  5.0보다 높으면 양수 보상,
  5.0보다 낮으면 음수 보상

데이터셋

원문 및 재가공 데이터



원문

교정이 필요한 실제 논문 데이터를 확보하기 어려워
국내 논문 초록 500건을 크롤링하여 사용
→ 500건 통계 분석을 통해 품질 평가 기준에도 활용

재가공

원문 논문 초록을 LLM(GPT-4o-mini)을 사용하여 재가공 진행
→ 의도적으로 품질을 낮춘 **저품질 텍스트 데이터셋**으로 변환

저품질화 방법

1. 문법, 맞춤법, 시제 오류 삽입
2. 모호한 표현 삽입 ..... ex) 일지도 모르는, 같은 것 등
3. 어색한 종결어미 사용 ..... ex) 인 것이다. 하는 바이다 등
4. 구어체, 감정적 표현 추가 ..... ex) 사실, 굉장히 등
5. 불필요한 수식어 삽입 ..... ex) 매우, 약간 등

데이터셋

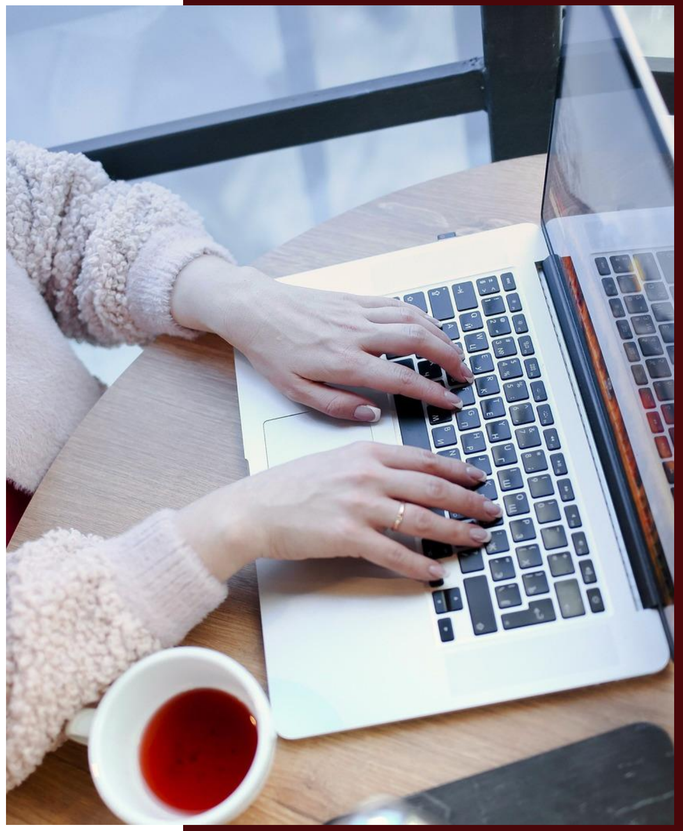
오프라인 데이터

오프라인 데이터셋

매번 학습할 때마다 LLM API를 호출할 경우,
비용이 많이 들고 학습 시간이 오래 걸림
최초 1회만 API를 호출하여 오프라인 학습용 데이터셋을
구축한 뒤 이를 기반으로 **지속적인 튜닝, 재설계, 학습 진행**

데이터 생성 파이프라인

1. 원본 초록 선택
2. 사용 가능한 액션 조합 생성 (길이 1~3)
   액션 5개 기준 : $5^1 + 5^2 + 5^3 =$ **155개 시퀀스**
3. 각 액션 시퀀스를 순차적으로 적용하여 데이터 생성
4. 하나의 원본 데이터는 JSONL 형태로 시퀀스를 가짐
5. 각 시퀀스는 원문, 액션 조합, 단계별 교정 결과,
   비용 정보 포함



학습 패러다임

MLP 기반의 오프라인 학습 / 온라인 학습

- LLM을 파인튜닝하는 강화학습은 고비용이기 때문에 비용을 줄이기 위해 MLP로 정책을 설계하고자 함
- 또한, Action에서 사용하는 LLM도 시간 및 비용이 많이 소모되므로, 2가지의 학습 방법을 중심으로 다양한 실험을 해보고자 함

- 오프라인 학습
  - 학습 시 LLM을 매번 호출하는 것은 시간과 비용이 많이 소모됨
  - 미리 생성된 Trajectory를 활용하여 시뮬레이션 속도를 높이고 샘플 효율성을 확보

- 온라인 학습 (하이브리드 학습)
  - 오프라인 학습을 통해 어느 정도 정의된 환경을 기반으로 온라인 학습 진행
  - 온라인 학습 시 도출되는 Trajectory를 캐시에 저장하여 동일 상황에서 재사용 진행
    - 동일한 조합이 발생하면 LLM 호출 없이 캐시 결과를 바로 사용
    - 불필요한 호출을 줄여 온라인 학습 비용 절감

실험 개요

알고리즘

PPO

Proximal Policy Optimization

- Clipping을 적용해 정책 업데이트 폭을 제한하고 안정적으로 학습
- Actor-Critic 구조로 정책(Actor)과 가치 함수(Critic)를 함께 학습
- 확률적 정책 기반 방식으로 수렴 과정이 비교적 안정적이고 재현성이 높음
- 환경 상호작용 비용이 높거나 파괴적 업데이트가 위험한 상황에서 적합

A2C

Advantage Actor-Critic

- Advantage 기반 정책 업데이트로 효율적으로 학습 수행
- 구조가 단순하고 구현이 쉬우며 병렬 환경에서 학습 속도가 빠름
- Actor-Critic 방식이지만 업데이트 안정화 장치는 상대적으로 단순
- 빠르게 시도하고 baseline을 구축하기 적합

DQN

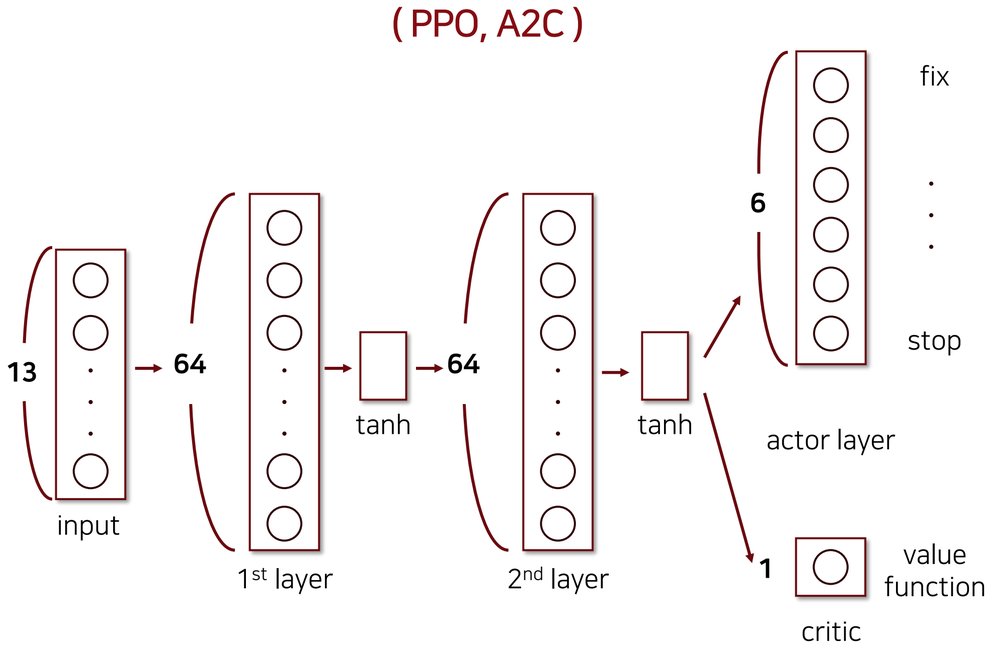
Deep Q-Network

- 정책을 직접 학습하지 않고 Q-value를 학습하는 가치 기반 방식
- Replay Buffer와 Target Network로 데이터 효율성과 안정성 확보
- 이산적 행동 공간에서 높은 성능 발휘
- 연속 행동 공간에서는 적용을 위해 추가 기법 또는 변형 모델 필요

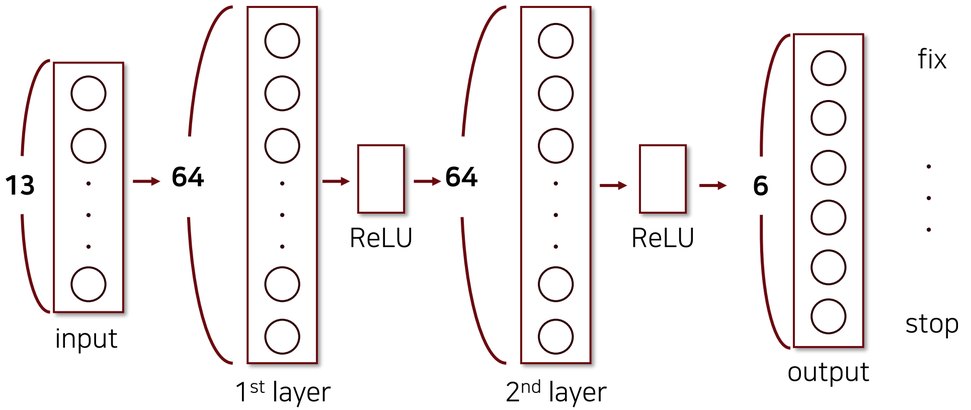
실험 개요

모델 구조

인공신경망 구조



DQN 구조



실험 개요

환경 및 설정

01

하드웨어 / 소프트웨어 환경

- **OS** : Windows 10
- **Python** : 3.x
- **주요 라이브러리** : pyTorch, OpenAI SDK (LLM API)
- **GPU** : CUDA 지원 시 GPU 사용, 미지원 시 CPU 사용

02

모델 설정

- **문서 교정 모델** : Qwen3-8B
- **문서 평가 모델** : 룰 기반 평가기

03

체크포인트 및 로깅

- **체크포인트 저장** : 에피소드 별 모델 저장
- **Trajectory 저장** : 에피소드별 상세 정보 JSON 저장
- **최고 성능 추적** : Best checkpoint 별도 저장

04

환경 파라미터 설정

- terminal_threshold : 9.5 (종료 판단 품질 임계값)
- cost_lambda : 1.0 (LLM 비용 패널티 가중치)
- repeat_penalty : 0.5 (반복 행동 감점)
- step_penalty : 0.1 (스텝 증가 패널티)
- editor_model : qwen3-8b (논문 교정용 LLM)

실험 개요

강화학습 성능 지표

Episode Return

에피소드당 누적 보상

평균 품질 점수

에피소드별 최종 품질

LLM 비용

에피소드당 호출 비용

액션 효율성

선택된 액션의 다양성 및 반복률

수렴 속도

학습 안정화까지 소요 에피소드 수

실험 개요

입력 데이터 흐름

원본	STEP 1	STEP 2	STEP 3
초기 문장	**improve_structure**	**make_academic**	**stop_editing**
뭐랄까, 이 글는 **장애인 취업과 관련된 어떤 느낌를** 다룬 것 같아. 글쎄,(중략...) 그래서인지 **직장 경험이나 임금 같은 게 중요**하다고 하더라. (중략...)	**장애인 취업과 관련된 어떤 느낌을** 다룬 것 같아. 이에 앞서 선행연구들에서는 **직장 경험이나 임금 같은 요소들이 중요**할 수 있다고 말하고 있어. (중략 ...)	**장애인의 취업에 대한 주관적 인식에 대한 연구가** 이루어지고 있다. 이에 앞서 선행 연구에서는 **직장 경험이나 임금과 같은 요소들이 중요한 역할**을 할 수 있다고 제시하고 있다. (중략 ...)	교정 종료 및 마지막 교정 데이터 output으로 반환

* 실제 학습 후 도출되는 결과 중 하나를 샘플로 흐름을 나타내었음

실험

실험 1 : 오프라인 PPO

학습 하이퍼파라미터

- num_episodes : 1000 (총 학습 에피소드 수)
- Batch Size : 3
- Buffer Size : 3
- Learning Rate: 0.003
- Max Steps : 3 (에피소드 당 최대 스텝 횟수)
- Gamma : 0.95 (스텝당 보상 할인율)
- GAE Lamda : 0.95
- Clip Epsilon : 0.2 (PPO Clip Ratio)
- Entropy Coefficient : 0.01
- K Epochs : 3 (배치 재사용 횟수)
- Output Activation : softmax (action 확률 분포)

loss 함수

- $\boldsymbol{Loss = L_{actor} + 0.5 \cdot L_{critic} - 0.01 \cdot H}$
- Actor Loss : Clipped PPO Objective

  (정책이 급격하게 바뀌는 것 방지)

$$L_{actor} = -E_t[\min(r_t A_t,\ clip(r_t,\ 1-\epsilon,\ 1+\epsilon)\ A_t)]$$

- Critic Loss : Value Function Loss

  (상태의 실제 return 과 예측 value 간 오차 (MSE)

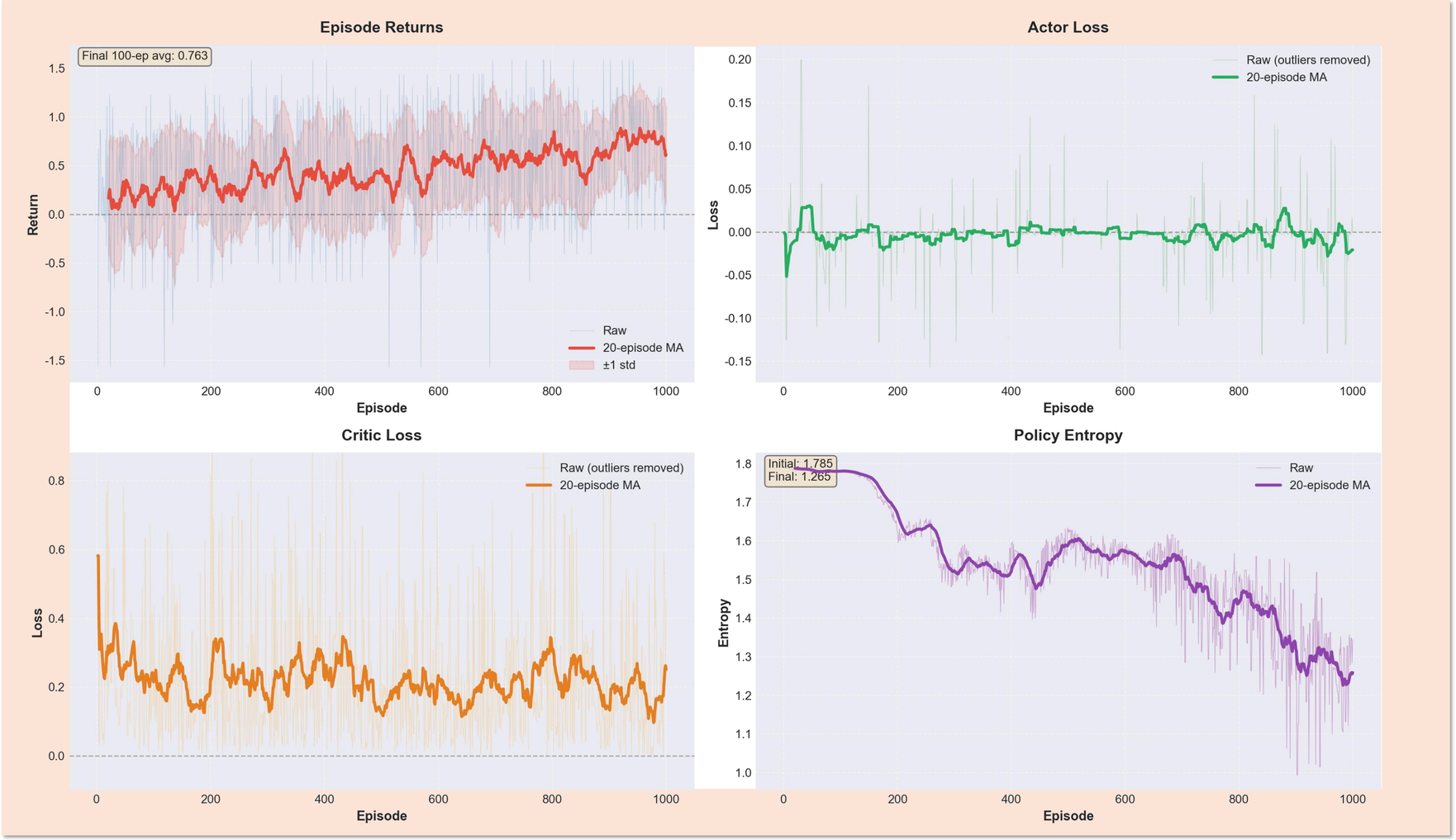
$$L_{critic} = (V_\theta(s_t) - R_t)^2$$

- Entropy : Exploration Term

  (정책의 엔트로피(탐험 장려))

실험

실험 1 : 오프라인 PPO



- ✓ 5개의 데이터셋을 사용하여 PPO 알고리즘으로 오프라인 학습 진행
- ✓ 설계된 환경의 구성 요소 (상태 정의, 보상 형태, 액션 구성)가 실제 학습 과정에서 정상적으로 작동하는지 사전에 검증하고자 배치사이즈와 버퍼사이즈도 작게 설정함
- ✓ 최종적으로 학습 가능성을 확인하여, 데이터 및 step 수를 확장한 후속 실험 진행

실험

실험 2 : 온라인 PPO

학습 하이퍼파라미터

- num_episodes : 1000 (총 학습 에피소드 수)
- Batch Size : 16
- Buffer Size : 32
- Learning Rate: 0.003
- Max Steps : 5 (에피소드 당 최대 스텝 횟수)
- Gamma : 0.95 (스텝당 보상 할인율)
- GAE Lamda : 0.95
- Clip Epsilon : 0.2 (PPO Clip Ratio)
- Entropy Coefficient : 0.03
- K Epochs : 3 (배치 재사용 횟수)
- Output Activation : softmax (action 확률 분포)

loss 함수

- $Loss = L_{actor} + 0.5 \cdot L_{critic} - 0.01 \cdot H$
- Actor Loss : Clipped PPO Objective
  (정책이 급격하게 바뀌는 것 방지)

$$L_{actor} = -E_t[\min(r_t A_t,\ clip(r_t,\ 1-\epsilon,\ 1+\epsilon)\ A_t)]$$

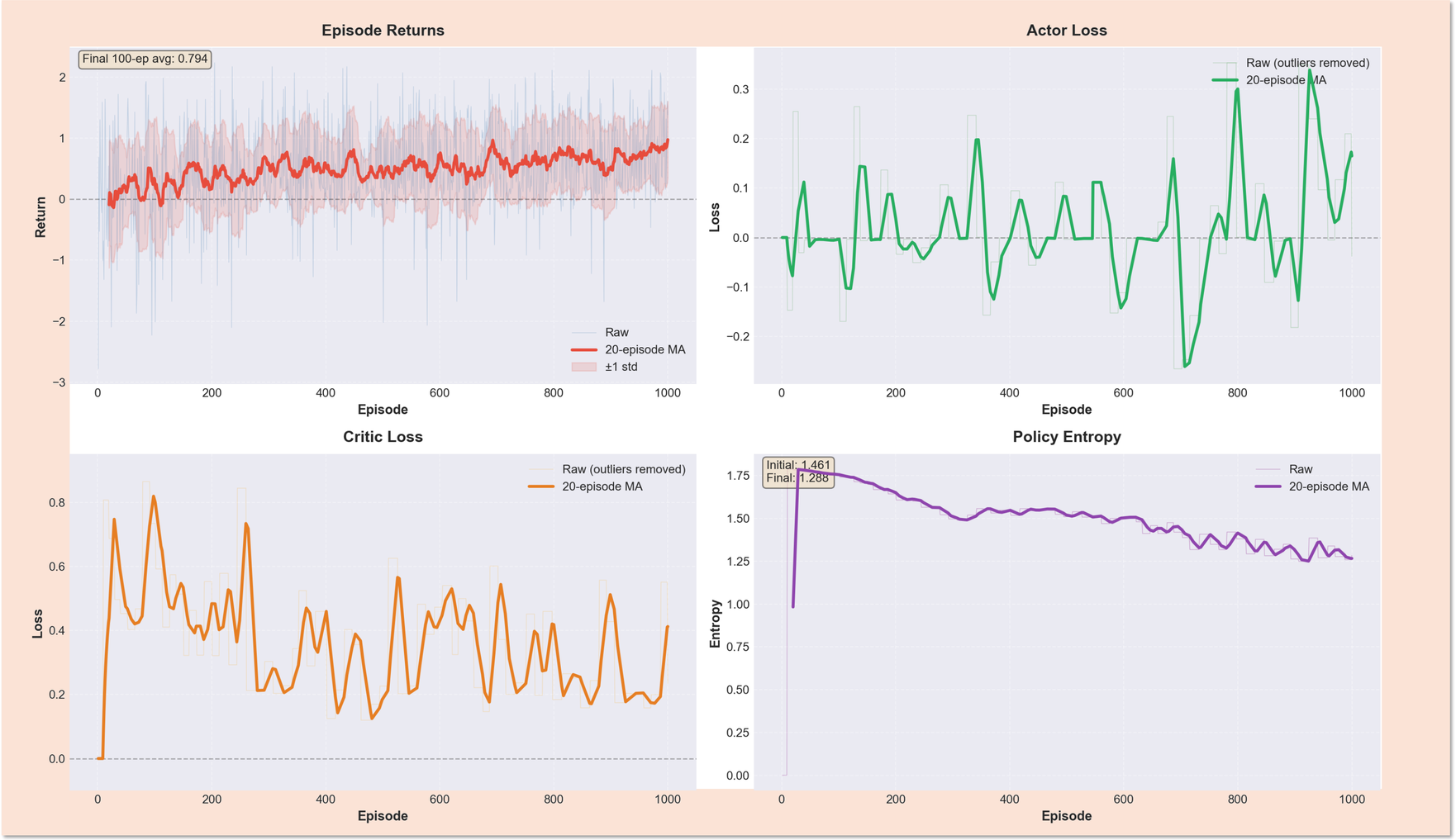
- Critic Loss : Value Function Loss
  (상태의 실제 return 과 예측 value 간 오차
  (MSE)

$$L_{critic} = (V_\theta(s_t) - R_t)^2$$

- Entropy : Exploration Term
  (정책의 엔트로피(탐험 장려))

실험

실험 2 : 온라인 PPO



- ✓ 50개의 데이터셋을 사용하여 PPO 알고리즘으로 온라인 학습 진행
- ✓ PPO 특성 상, 새로운 데이터를 지속해서 수집해야 하므로 배치사이즈와 step 규모가 커질수록 부담스러움
- ✓ 1000개의 Episode 에서는 많은 결과를 가져오지 못함

실험

실험 3 : 온라인 A2C

학습 하이퍼파라미터

- num_episodes : 1000 (총 학습 에피소드 수)
- Learning Rate: 0.003
- Max Steps : 5 (에피소드 당 최대 스텝 횟수)
- Gamma : 0.95 (스텝당 보상 할인율)
- GAE Lamda : 0.95
- Entropy Coefficient : 0.03
- K Epochs : 3 (배치 재사용 횟수)
- Output Activation : softmax (action 확률 분포)

loss 함수

- $Loss = L_{actor} + 0.5 \cdot L_{critic} - 0.01 \cdot E$
- Actor Loss : Clipped PPO Objective
  (정책이 급격하게 바뀌는 것 방지)

$$L_{actor} = -E_t[(A_t \cdot \log \pi_\theta(a_t \mid s_t))]$$

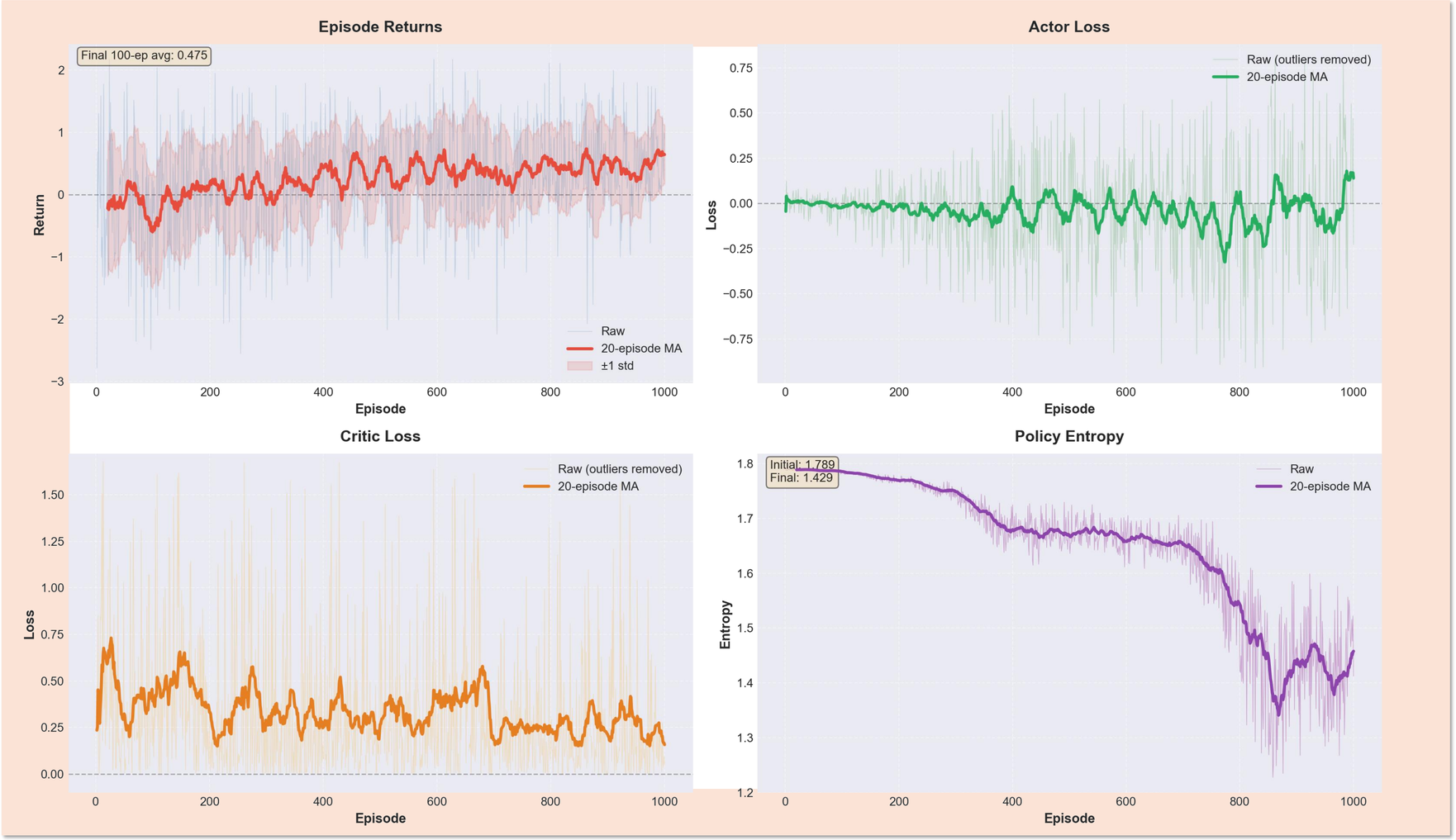
- Critic Loss : Value Function Loss
  (상태의 실제 return 과 예측 value 간 오차
  (MSE)

$$L_{critic} = (V_\theta(s_t) - R_t)^2$$

- Entropy : Exploration Term
  (정책의 엔트로피(탐험 장려))

실험

실험 3 : 온라인 A2C



- ✓ 50개의 데이터셋을 사용하여 A2C 알고리즘으로 온라인 학습 진행
- ✓ A2C는 에피소드마다 업데이트가 이루어져 학습 변화를 세밀하게 확인할 수 있으나 변동성이 PPO보다 크게 나타남
- ✓ 전반적으로 상승하는 Return 흐름 확인
- ✓ Actor Loss에서는 변동성이 크게 증가하는 양상 확인
- ✓ Critic Loss와 Entropy는 점진적으로 감소하며, 전체적인 수렴 양상은 PPO와 유사하게 나타남

실험

실험 4 : 온라인 DQN

학습 하이퍼파라미터

- num_episodes : 1000 (총 학습 에피소드 수)
- Learning Rate: 0.003
- Batch Size : 32
- Buffer Size : 10000
- Max Steps : 5 (에피소드 당 최대 스텝 횟수)
- Gamma : 0.95 (스텝당 보상 할인율)
- GAE Lamda : 0.95
- target update frequency: 10
  (10개 에피소드마다 업데이트)
- epsilon_decay : 0.995 (탐색 감소율)
- K Epochs : 3 (배치 재사용 횟수)
- Output Activation : softmax (action 확률 분포)

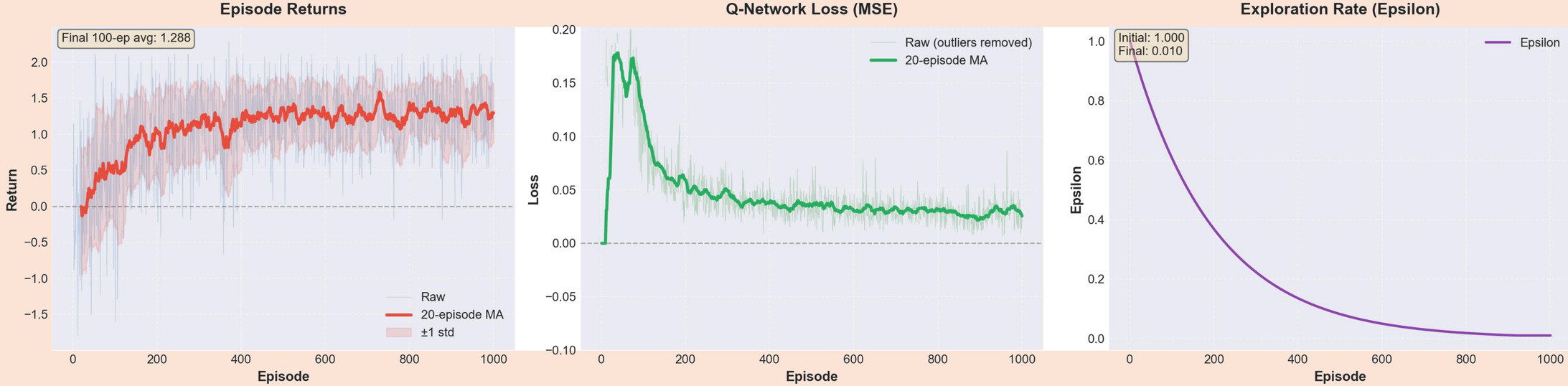
loss 함수

- MSE loss 사용

$$\boldsymbol{Loss = (Q(s_i, a_i) - y_i)^2}$$

실험

실험 4 : 온라인 DQN



- ✓ DQN에서는 가치 기반 방식으로 Q-value가 빠르게 안정화되면서 초기 return이 빠르게 증가
- ✓ 학습이 진행될수록 MSE Loss가 지속적으로 감소하며 안정적인 수렴 패턴을 보임
- ✓ 비교적 적은 상호작용에서도 기본적인 성능의 정책을 빠르게 확보할 수 있음을 확인

결론

토의

방향

최근 LLM 기반 에이전트 서비스가 많이 출시되고 있으며,
에이전트 간 최적 선택을 위한 LLM Router 개념의 강화학습 연구도 활발하게 진행되고 있음

아이디어

이러한 연구를 실제 현업에서 적용하기 위해서는 고성능의 인프라가 필요하고
LLM을 강화학습 하기 위한 시간과 비용이 많이 필요함

목표

그래서 본 프로젝트에서는 저비용 인프라 환경에서도 강화학습을 통해
기존 AI 에이전트 서비스의 성능을 향상시킬 수 있는 가능성을 탐색하고자 함

한계

최근 NeurIPS에 채택된 Router-R1 논문을 기반으로 개념적 스케치 진행
(해당 논문은 일반적인 상황에서의 LLM을 강화학습을 통해 최적화 하는 방법)

에이전트

MLP 구조의 간단한 인공신경망으로 정책을 설계하고, 짧은 시간 안에 가능성을 찾기 위해
석 · 박사 과정에서 요구되는 '논문 교정' 태스크를 중심으로 최적화 프로젝트 진행

결론

결론

프로젝트 주요 성과

1. LLM 기반 논문 교정 에이전트의 **단계적 의사결정**을 강화학습으로 최적화할 수 있는 가능성 보임
2. PPO 기반 정책 학습을 통해 **품질 · 비용 · 속도 간 균형 잡힌 전략**을 도출함
3. 반복 행동, 불필요한 교정, 과도한 API 호출을 억제하는 보상 설계가 효과적으로 작동하여 **효율적인 교정 패턴**을 학습함
4. 저품질 텍스트 재가공 기반 데이터셋은 다양한 오류 패턴을 포함하여 **정책의 일반화 성능 향상**에 기여함

프로젝트 한계

1. 규칙 기반 스코어에 의존하는 평가 방식과 학습에 사용된 제한된 데이터 양으로 인해 **실제 품질과의 완전한 정합성 확보에는 한계가 존재**함
2. **실제 문서 전체를 활용하기 위해서는 필요한 자원 또한 고비용이 될 수 있음**

결론

향후 발전 방향

1. 실제 **사용자 입력 데이터 기반** 정책 사용
2. LLM 기반 품질 평가기로 **보상 정밀도 향상**
3. 모델 · 가격 구조 반영한 **현실적 비용 제약 강화**
4. 프롬프트 전략 뿐만 아닌, Tool, RAG, LLM 까지도 액션에 포함하여 **더욱 복잡하고 정교한 에이전트 설계**
5. state에 **텍스트 임베딩을 포함**하여 자연어처리에 능통한 BERT 기반의 Transformer 모델 활용

부록

참고 문헌

Zhang, H., Feng, T., & You, J. (2025, June). Router-r1:
Teaching llms multi-round routing and aggregation via reinforcement learning.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., et al. (2025).
Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017).
Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ⋯ & Hassabis, D. (2015).
Human-level control through deep reinforcement learning.
Nature, 518(7540), 529–533.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., ⋯ & Kavukcuoglu, K. (2016).
Asynchronous methods for deep reinforcement learning.
International Conference on Machine Learning (ICML).

부록

팀원 역할

김진산

1. 프로젝트 구체화 →
   레퍼런스 논문 기반으로, 특정 도메인 중심으로 강화학습 설계 (state, Action, Reward)
2. 데이터 수집 크롤링 기능 구현 및 수행
3. 룰 기반 문서 품질 평가 구현
4. 시각화 코드 구현 및 PPT 작성용 그래프 자료와 해석 제공

차시명

1. 레퍼런스 논문 (Router-R1) 구현 → LLM 파인튜닝을 사용하지 않는 방법으로 구현
2. 프로젝트 구체화 내용을 반영하여 오프라인 · 온라인 학습과 다양한 알고리즘(DQN, A2C, PPO) 적용 및 확장을 위한 코드 구조 캡슐화, 추상화 진행
3. 학습 과정의 디버깅을 위한 체크포인트 및 로깅 구현
4. PPT 초안 내용 작성

한다현

1. 프로젝트 구체화 →
   레퍼런스 논문 기반으로, 특정 도메인 중심으로 강화학습 설계 (state, Action, Reward)
2. 오프라인 데이터 셋 생성 코드 구현 및 수행
3. 문서 품질 평가 검토 및 구체화
4. PPT 내용 구체화 및 디자인

감사합니다

서강대학교 AI · SW 대학원 데이터사이언스 · 인공지능전공

A72040 김진산 | A72080 차시명 | A72085 한다현

팀명 : 외유내강 (겉으론 부드러워 보이나, 내가 강화되고 있음)

https://github.com/itcasim0/reinforcement-with-llm