강화학습 과제

A71031 설위준

목차

1. 프로젝트 주제 및 목표

본 프로젝트의 목표는 이커머스 URL 내 판매 중인 상품페이지에서 수집한 타이틀 정보(GOODS_NAME) 및 옵션 정보(GOODS_OPTN_VLU)로부터 정확한 모델명(MODEL_NAME)을 자동으로 추출·선택하는 강화학습 기반 모델을 구축하는 것이다.

일반적으로 이 문제는 모델명이 명확히 존재하는 경우, Supervised Classification 또는 Rule-based NER 기반으로 처리된다.

그러나 본 연구에서는 명확한 모델명이 대부분 존재하지 않는 카테고리군을 대상으로 한다. 이에 따라, 모델명 후보군 중 최적인 선택을 수행하는 정책(policy)을 강화학습으로 학습함으로써,

- 희소한 라벨 데이터에서도 효과적인 학습 가능
- reward shaping을 통한 성능 향상
- 후보군 선택 과정의 불확실성(exploration/exploitation)을 모델링

과 같은 장점을 기대할 수 있다.

본 프로젝트는 다음 세 알고리즘을 비교하는 것을 목표로 한다.

1. PPO (Proximal Policy Optimization)
2. A2C (Advantage Actor-Critic)
3. ε-Greedy Multi-Armed Bandit

2. 환경 및 데이터셋 설명

2.1 데이터셋 구성

데이터는 타이틀 정보, 옵션 정보, 모델명으로 구성되며, 총 약 1,000,000 건의 데이터로 구성된다.

예시 :

GOODS_NAME	GOODS_OPTN_VLU	MODEL_NAME
네스카페 클래식 마일드 500g 자판기 블랙커피 리필용		클래식 마일드 500g@1개
로레알 염색약 새치 커버 미용실 마지렐 산화제 메탈	선택1: 02_마지브라운₩(새치₩) / 선택2: B5.05 마호가니 브라운	마지브라운 컬러 50g@1개
로레알 염색약 새치 커버 미용실 마지렐 산화제 메탈	선택1: 02_마지브라운₩(새치₩) / 선택2: B4.35 웜 브라운	마지브라운 컬러 50g@1개
네슬레 네스카페 클래식 마일드 500g X 1개		클래식 마일드 500g@1개
쉽고 간편한 MT-1681 마이크로라이프 보급형 디지	본품 / 3950원	mt-1681@1개
네슬레 네스카페 클래식 마일드 500g 블랙커피 초이스	본품 / 11550원	클래식 마일드 500g@1개

컬럼명	설명
GOODS_NAME	타이틀 정보
GOODS_OPTN_VLU	옵션 정보
MODEL_NAME	실제 모델명 라벨 및 수량

2. 환경 및 데이터셋 설명

2.2 데이터 전처리 (Preprocessing)

강화학습 속도 최적화를 위해 다음 단계를 수행한다.

1) 텍스트 정제

- 특수문자 제거
- 공백, 중복 문자열 정규화
- 옵션 텍스트가 없는 경우 공백 처리

2) BERT 기반 문장 임베딩 사전 생성 (Pre-computed Embedding)

- 강화학습 환경에서 매 에피소드마다 BERT를 호출하면 극도로 느리므로, DistilBERT를 사용하여 모든 텍스트를 미리 임베딩하였다.

3) 학습/평가 데이터 분리

- Train: 80%
- Test: 20%

3. State, Action, Reward 설계

3.1 State 설계

State는 사전 생성한 768차원 BERT 임베딩 벡터로 정의한다.

3.2 Action 설계

Action은 모델명 후보 중 하나를 선택하는 것으로 정의한다.

※ 본 프로젝트에서는 정답이 1개이므로 간단히 1개 후보로 구성하였다.

3.3 Reward 설계

정답인지 여부에 따라 reward를 부여한다.

4. 강화학습 알고리즘 및 Hyperparameter 설명

4.1 비교 알고리즘

✓ PPO (Proximal Policy Optimization)

- PDF에서 학습한 clipped objective 적용
- 안정적 업데이트를 위해 ratio clipping 사용
- 실제 NLP fine-tuning에서 가장 많이 사용되는 RL 알고리즘

✓ A2C (Advantage Actor-Critic)

- Actor-Critic 기반 value baseline 활용
- PPO 대비 빠르지만 안정성은 낮음

✓ ε-Greedy Multi-Armed Bandit

- 모델명 선택을 slot machine arm 선택으로 해석
- 가장 빠르지만 성능 한계가 있음

4. 강화학습 알고리즘 및 Hyperparameter 설명

4.2 Hyperparameter

1) Learning Rate (학습률)

- 신경망이 파라미터를 얼마나 크게 업데이트할지 결정하는 값
- PPO는 안정적 clipping update → 작은 lr=3e-4 사용, A2C는 variance가 높아 1e-3 사용

2) Gamma (Discount Factor, 할인율)

- 미래 보상에 대한 중요도를 조절
- PPO/A2C는 장기 보상 고려 위해 0.99, Bandit은 single-step 문제라 1.0

3) PPO Clip ε (Clipping Range)

- PPO의 핵심인 “policy update를 너무 크게 하지 않도록 제한하는 수치”
- PPO의 핵심 안정성 파라미터로 논문 기본값 0.2 사용

4) Exploration 방식

- PPO는 확률적 정책 자체로 탐색, A2C는 entropy bonus, Bandit은 ε-greedy

5) Epochs per Update

- PPO는 5회 반복 업데이트로 안정성 확보, A2C는 1-step TD update

6) Batch size

- 모두 single-step 환경이므로 1로 설정

5. 실험 셋업

5.1 실험 환경

항목	환경
하드웨어	Google Colab GPU (Tesla T4)
OS	Ubuntu 20.04
Python	3.12
Framework	PyTorch, Transformers
GPU Memory	16GB

5.2 Evaluation Metric

정답 선택 정확도 Accuracy를 사용

※ Accuracy = 정답 개수 / 전체 개수

6. 실험 결과

6.1 알고리즘별 Accuracy 비교

정확도	PPO	A2C	Bandit
기본 Hyperparameter	0.98	0.62	0.91

6.2 Hyperparameter 탐색 결과

모델	Hyperparameter
PPO	Learning rate = 0.0005, Clip epsilon = 0.2, Gamma = 0.99

6.3 최적의 Hyperparameter를 활용한 Random Seed 변경 실험 결과

모델	Mean Accuracy	Standard Deviation (Std)	95% Confidence Interval
PPO	0.88	0.06	[0.81, 0.95]

7. 토의 및 결론

7.1 실험 결과 논의

- PPO가 가장 안정적이며 높은 정확도를 기록

  → ratio clipping을 통한 안정적 policy update 효과

- A2C는 PPO 대비 불안정

  → variance reduction이 부족

- Bandit은 텍스트 임베딩이 존재함에도 구조적 한계로 가장 낮은 성능

7.2 본 연구의 의미

- 상품 텍스트 → 모델명 선택 문제를 강화학습으로 성공적으로 적용
- 사전 임베딩 전략을 통해 대용량 데이터에서도 실용적 속도 달성
- 향후 대규모 커머스 데이터 자동 정제에 응용 가능

7. 토의 및 결론

7.3 한계 및 개선 방향

1. 대규모 데이터셋 처리의 성능 한계 존재
2. 후보 모델명 생성 기능 미포함
3. 현재 reward function이 binary
4. 환경이 단일-step MDP
5. multi-label 후보군이 많아질 경우 탐색 공간 확장 필요

7.4 향후 개선

- TPU/분산 처리 환경 도입 및 메모리 최적화를 통한 대규모 데이터셋 처리 확장성 확보
- GPT 기반 후보 모델명 생성 + RL 선택 모델 결합
- String similarity 기반 soft reward
- Multi-step sequential reasoning RL 도입
- SAC, TD3 등 continuous RL 적용 가능성 탐구

7. 토의 및 결론

7.3 한계 및 개선 방향

- 후보 모델명 생성 기능 미포함
- 현재 reward function이 binary
- 환경이 단일-step MDP
- multi-label 후보군이 많아질 경우 탐색 공간 확장 필요

7.4 향후 개선

- GPT 기반 후보 모델명 생성 + RL 선택 모델 결합
- String similarity 기반 soft reward
- Multi-step sequential reasoning RL 도입
- SAC, TD3 등 continuous RL 적용 가능성 탐구