# 인간 피드백 기반 강화학습 (RLHF)에서 보상 모델의 효과적인 훈련 방법에 관한 연구

김정욱, Aiyanyo Imatitikua Danielle*, 임희석*
고려대학교 컴퓨터학과
{k0s1k0s1k0, titi, limhseok}@korea.ac.kr@korea.ac.kr

# A Study about Efficient Method for Training the Reward Model in RLHF

Jeongwook Kim, Imatitikua Danielle Aiyanyo, Heuiseok Lim
Department of Computer Science and Engineering, Korea University

**요 약**

RLHF(Reinforcement Learning from Human Feedback, 인간 피드백 기반 강화학습) 방법론이 최근 고성능 언어 모델에 많이 적용되고 있다. 이 방법은 보상 모델과 사람의 피드백을 활용하여 언어 모델로 하여금 사람이 선호할 가능성이 높은 응답을 생성하도록 한다. 하지만 상업용 언어 모델에 적용된 RLHF의 경우 구현 방법에 대하여 정확히 밝히고 있지 않다. 특히 강화학습에서 환경(environment)을 담당하는 보상 모델을 어떻게 설정하는지가 가장 중요하지만 그 부분에 대하여 오픈소스 모델들의 구현은 각각 다른 실정이다. 본 연구에서는 보상 모델을 훈련하는 큰 두 가지 갈래인 '순위 기반 훈련 방법'과 '분류 기반 훈련 방법'에 대하여 어떤 방법이 더 효율적인지 실험한다. 또한 실험 결과 분석을 근거로 효율성의 차이가 나는 이유에 대하여 추정한다.

주제어: 인간 피드백 기반 강화학습 (RLHF), 보상 모델, 효율적 훈련 방법

## 1. 서론

최근 고성능을 보이는 많은 상업용 및 비상업용 채팅 언어 모델들(ChatGPT [1], Bard [2], Claude [3], LLAMA2-CHAT [4])이 강화학습을 적용하고 있다. 강화학습은 일반적으로 보상(reward)이 주어지는 환경(environment)에서 에이전트가 보상을 최대화할 수 있는 행동(action)을 선택할 수 있도록 훈련하는 방법이다. 언어 모델을 강화학습으로 훈련하기 위해서는 환경을 제대로 정의해야만 한다. 최근 채팅 가능한 언어 모델의 고성능화를 이끈 강화학습 적용 방법은 RLHF(Reinforcement Learning from Human Feedback)이다. RLHF 방법론에서는 '사람이 선호할 가능성이 높은 모델의 응답'에 높은 보상을 주는 보상 모델 (Reward model)을 도입하여, 크로스 엔트로피 기반 훈련을 마친 후 추가적으로 강화학습 훈련을 적용한다. 이와 같은 RLHF 방법론을 적용한 모델은 언어 이해 및 생성 능력뿐만 아니라 사람이 더 선호할 수 있는 응답을 생성할 수 있게 된다. 하지만 강화학습을 적용한 최신 상업용 언어 모델의 수에 비해 그와 관련한 연구의 수는 매우 적다. 대표적인 상업용 언어 모델인 ChatGPT는 InstructGPT와 비슷한 방법론을 적용했다는 내용이 발표되었지만 구체적인 구현 방법이 명시되지 않았으며, GPT-4의 technical report에서는 모델의 성능만 제시되고 모델 제작에 대한 정보가 일체 발표되지 않았다. 따라서 강화학습 단계에서 보상 모델의 훈련 시 목적 함수나 하이퍼파라미터, 데이터들은 공개되지 않았다. 강화학습을 언어 모델에 적용하기 어려운 이유는 강화학습이 크로스 엔트로피 기반의 일반적인 훈련 방법보다 환경이나 하이퍼파라미터에 민감하기 때문이기 때문에, 정보가 부족한 일반 연구자들은 더욱 연구가 어렵다. 본 연구에서는 보상 모델의 효과적인 훈련 방법에 대하여 실험하고 결과에 대한 이론적인 근거를 제시한다. 보상 모델의 훈련 방법은 크게 두 가지로 나뉜다. 첫 번째는 훈련 대상 모델이 생성한 응답을 사람이 보고 가장 좋은 응답부터 순위를 매기는 순위 기반 방식으로, [5, 6]에서 사용되었다. 두 번째는 [4, 7] 등에서 제시된 이진 분류 기반 훈련 방법으로, 선호되는 응답과 비선호되는 응답을 분류하는 모델처럼 훈련하는 방법이다. 두 가지 방법으로 훈련된 보상 모델은 강화학습의 환경으로 사용되어 가상의 사람처럼 언어 모델이 생성한 결과에 대하여 스칼라값인 보상을 부여한다. 언어 모델은 보상 모델이 부여한 보상 값을 기반으로 강화학습 알고리즘을 사용하여 총 보상 값의 합을 높일 수 있도록 훈련된다. PPO(Proximal Policy Optimization) [8] 알고리즘은 가장 많이 사용되는 현대 강화학습 알고리즘으로, 행동자(actor)인 언어 모델을 주어진 보상을 높이도록 훈련시킬 수 있다. 2장에서는 연구에서 적용한 언어 모델의 기본적인 RLHF 훈련 프레임워크에 대하여 설명한다, 3장에서는 제안하는 방법에 대하여 설명한다. 4장과 5장에서는 실험 결과에 대하여 분석하고 결론을 짓는다. 6장에서는 본 연구와 관련된 RLHF 연구 및 프레임워크를 설명한다.

## 2. RLHF 프레임워크

RLHF를 언어 모델에 적용할 때는 그림 1과 같이 크게 3단계의 과정을 거친다. 첫 번째로, 일반적인 언어 모델을 파인튜

*교신저자(Corresponding author)

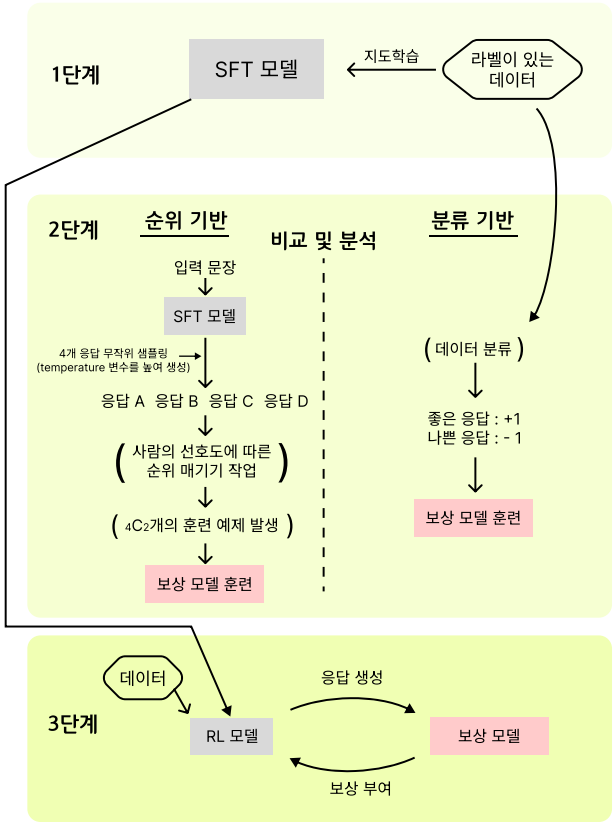


그림 1. 전체 실험 과정

닝(fine-tuning) 훈련할 때와 같은 방법으로 SFT(Supervised Fine-Tuning)을 진행한다. 두 번째로, 문장을 입력받아 스칼라인 보상 값을 출력하는 보상 모델을 훈련한다. 보상 모델을 어떻게 훈련하느냐에 따라 다음 단계에서 모델의 훈련 방향을 조절할 수 있다. 예를 들어, 긍정적인 언어에 높은 보상을 주고 부정적인 언어에 낮은 보상을 주는 보상 모델을 훈련한다면 최종 모델이 긍정적인 언어를 더 많이 생성하도록 조정할 수 있다. 세 번째로, PPO와 같은 강화학습 알고리즘이 보상 모델이 부여하는 보상 값을 높이는 방향으로 SFT 모델 파라미터를 조정한다. PPO는 정책 경사도 (Policy Gradient) 기반 행동자-비평자(actor-critic) 알고리즘이다. 첫 단계에서 훈련된 SFT 모델은 여기서 행동자(actor)가 된다. 비평자(critic)는 보통 행동자 모델의 파라미터를 공유하고 맨 마지막 레이어만 새로 훈련하는 방식으로 구현된다.

## 3. 방법론

상용되는 보상 모델 훈련 방법은 크게 두 가지로 나눌 수 있다. 순위 기반 방식과 분류 기반 방식이다. 본 연구에서는 두 방식을 각각 동일한 조건에서 구현하고 실험하여 각 방식의 특징과 더 나은 보상 모델 훈련 방법을 제시하고자 한다. 상용 언어 모델의 RLHF에서는 사람이 직접 모델의 생성 결과에 대하여 판단한 결과를 사용하지만 비용이 많이 들기 때문에 ChatGPT와 프롬프트를 사용하여 사람을 대체하였다. 최근 [9, 4, 10, 11]과 같은 연구에서는 ChatGPT가 모델 평가 시 사람과 비슷한 수준으로 질 좋은 평가를 내리고, 심지어 사람에 비해 평가의 일관성이 높으므로 좋은 대체제라는 결론을 내렸다.

그림 1는 전체 실험 과정을 설명한다. 1단계에서는 입력 문장에 대한 정답 문장이 제시되어 있는 데이터에 대하여 일반적인 지도학습이 진행된다. 이 결과로 훈련된 모델을 SFT 모델이라고 한다. 본 연구에서는 GPT2-large [12] 모델을 사용하였다.

이어지는 절에서는 그림 1의 2, 3단계를 설명한다. 보상 모델의 훈련 방법에 따른 최종 PPO 모델의 성능 차이를 비교하기 위해, 보상 모델의 훈련 방법 외에는 모두 동일한 실험 조건을 적용하였다. 3.1절에서는 순위 기반 훈련 방법을, 3.2절에서는 분류 기반 훈련 방법을 각각 보상 모델에 적용한다. 두 보상 모델 모두 사전 학습 모델로 LLAMA1-7B [13] 모델을 사용하였다. 보상 모델을 매우 큰 모델로 선택한 이유는 보상 모델이 사람의 선호를 최대한 비슷하게 모방하도록 하였기 때문이다. 그리고 최종 성능을 비교하기 위해 3.3절에서는 동일한 환경에서 동일한 SFT 모델에 PPO를 적용하여 추가적으로 RLHF 훈련을 진행한다. 4장의 실험 결과에서 두 방법으로 훈련하는 과정과 훈련한 모델의 성능을 비교한다.

### 3.1 비교군 1: 순위 기반 보상 모델 훈련

순위 기반 훈련에서는 같은 입력 $x$에 대하여 그림 1의 2단계처럼 SFT 모델의 서로 다른 $k$가지 응답을 수집한다. 그리고 $k$개의 응답을 사람이 더 선호할 만한 순서대로 순위를 매긴다. 다음으로는 $k$개의 응답들 중 2개를 선택한다. 매겨진 순위에 따라 하나를 좋은 응답 $y$, 나머지 하나를 나쁜 응답 $y\prime$으로 간주한다. 그 결과로 총 ${}_kC_2$개의 훈련 데이터 $D$가 생성된다. $D$ 안에서 좋은 응답에 대한 보상은 높게, 나쁜 응답에 대한 보상은 낮게 생성하도록 훈련한다. $\theta$로 파라미터화된 보상 함수 모델 $r_\theta$에 대한 손실 함수는 다음과 같이 정의된다.

$$\text{loss}(r_\theta) = -E_{(x,y,y\prime)\sim D}[\log(\sigma(r_\theta(x,y) - r_\theta(x,y\prime)))]$$

구체적으로는 $k = 4$로, 같은 모델의 입력에 대하여 4개의 서로 다른 응답을 샘플링하여 ChatGPT에게 가장 좋은 응답의 순서대로 순위를 매기도록 프롬프팅하였다. 따라서 하나의 $x$에 대하여 6개의 훈련 예제가 생성되었다. 실험 결과, 생성된 6개의 훈련 예제가 모두 같은 $x$를 공유하고 있기 때문에 1에폭만 훈련했음에도 불구하고 빠르게 과적합되는 현상이 나타났다. 이를 해결하기 위해 [14]에서 제안한 방법대로 6개의 훈련 예제가 모두 같은 미니배치 안에 있도록 환경을 조성하였다. 또한 실험 중간에 환경이 바뀌지 않도록 GPT-4-0613으로 ChatGPT API 버전을 고정하였다.

### 3.2 비교군 2: 분류 기반 보상 모델 훈련

분류 기반 훈련에서는 모델이 좋은 응답과 나쁜 응답을 이진 분류할 수 있도록 훈련한다. 미리 준비된 데이터셋 D 안에는 좋은 응답이 1, 나쁜 응답이 -1로 라벨링되어 있다. 모델은 일반적인 지도학습 방법으로 좋은 응답과 나쁜 응답을 분류할 수 있도록 훈련된다. 훈련된 모델이 내놓은 로짓 값에 시그모이드 함수를 취하면 구간 $(-1, 1)$의 값을 출력한다. 이 값을 추후 PPO 알고리즘에서 보상 값으로 활용한다. 손실 함수는 다음과 같이 정의된다.

$$\text{loss}(r_\theta) = \\ -E_{(x,y,y\prime)\sim D}[p\log(r_\theta(x,y)) + (1-p\prime)\log(1-r_\theta(x,y\prime))]$$

$p$는 y가 좋은 응답이면 1, 나쁜 응답이면 0의 값을 가지는 변수이다. 그 외 각 기호는 3.1절에서 사용된 기호와 동일하다. 실험에서는 주어진 데이터셋의 정답 집합을 좋은 응답으로, SFT 모델이 생성한 응답을 나쁜 응답으로 간주하였다. 따라서 보상 모델은 정답에 가깝게 생성할수록 높은 보상을 주도록 훈련되었다. 두 방법에서 훈련된 보상 모델은 평균이 0이고 분산이 1이 되도록 훈련 후 정규화되었다.

### 3.3 PPO 적용

SFT 모델과 보상 모델이 준비되었다면 PPO 알고리즘을 활용하여 SFT 모델을 추가적으로 훈련한다. PPO 알고리즘의 구현체는 trl [15]을 이용하였다. SFT 모델 $\rho$는 행동자 모델인 $\pi$가 되어 PPO 알고리즘 하에서 훈련된다. PPO 알고리즘은 크로스 엔트로피 기반 훈련보다 불안정하다고 알려져 있기 때문에, [5, 14]에서 제안한 방법대로 $\pi$가 $\rho$의 분포에서 크게 벗어나지 않도록 KL 발산 (Kullback-Leibler divergence) 항을 도입한다. 다음 식은 강화학습 훈련 시 사용된 목적 함수이다.

$$E_{(x,y)\sim D_{\pi_\phi^{\text{RL}}}}\left[r_\theta(x,y) - \beta\log\left(\pi_\phi^{\text{RL}}(y\mid x)/\pi^{\text{SFT}}(y\mid x)\right)\right]$$

$x$는 모델에 들어가는 입력(프롬프트), $y$는 모델이 출력하는 응답이다. $\phi$로 파라미터화된 정책 $\pi_\phi$은 $x$를 입력받아 $y$를 출력한다. $\pi_\phi^{\text{SFT}}$는 레퍼런스 모델로, 강화학습 기간 동안 고정된다. $\pi_\phi^{\text{RL}}$은 훈련 대상인 모델로, PPO 모델로 칭한다. $\beta$는 KL 계수로, KL 발산이 손실 함수에 미치는 영향을 조절한다. KL 계수가 클수록 강화학습 모델 ($\pi_\phi^{\text{RL}}$)의 확률분포가 기준 SFT 모델 ($\pi_\phi{}^{\text{SFT}}$)의 분포가 달라지지 않도록 훈련된다.

## 4. 실험

### 4.1 데이터셋

실험 대상 데이터셋으로 FairytaleQA [16]을 사용하였다. FairytaleQA 데이터셋은 어린이를 위한 동화 지문과 그 지문에 대한 질문, 답변 쌍으로 구성된다. 데이터셋은 총 278개의 책에서 제작된 10580개의 지문-질문-답변 쌍으로 이루어져 있다. 모델은 FairytaleQA 데이터셋에서 주어지는 동화를 읽고 질문을 생성하는 태스크를 수행한다. FairytaleQA 데이터셋의 특징은 교육 분야의 전문가들이 어린이들의 독해 능력 발달을 위하여 추론을 요구하는 질 높은 질문들을 제작하였다는 점이다. 그러므로 모델은 단순한 질문이 아닌 적절한 난이도의 질문을 생성할 것을 요구받는다. 강화학습 단계에서 보상 모델은 더 교육적이고 추론 능력을 요구하며 문법적으로 올바른 질문에 더 높은 보상을 부여하도록 조정된다.

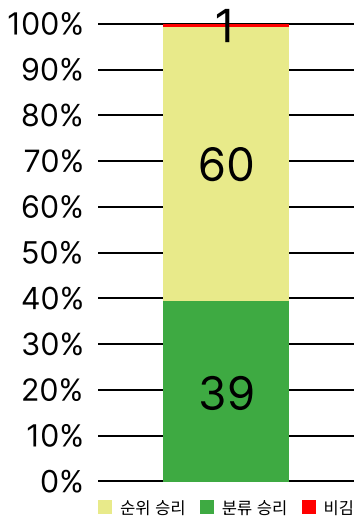


그림 2. ChatGPT가 평가한 승률

### 4.2 실험 결과

그림 2는 각각의 보상 모델로 PPO 훈련한 모델이 FairytaleQA 데이터셋의 테스트용 집합에 대해 생성한 응답을 ChatGPT가 평가한 결과이다. 동일한 사람의 선호도 판단을 가정했기 때문에 보상 모델을 훈련할 때와 동일한 프롬프트를 사용하여 ChatGPT 평가를 진행하였다. 실험 결과, 순위 기반 방법으로 훈련한 보상 모델이 사용된 RL 모델 (이하 '순위 RL')이 분류 기반 방법으로 훈련한 보상 모델이 사용된 RL 모델 (이하 '분류 RL')보다 ChatGPT의 선호도를 더 잘 반영하는 것으로 나타났다. 그림의 '순위 승리'는 순위 RL이 생성한 결과가 분류 RL이 생성한 결과보다 더 좋다고 판단된 비율이다. 그림 3은 훈련 중 옵티마이저의 경사도(gradient) 업데이트 시마다 RL 모델이 받는 보상의 평균을 기록한 그래프이다. 보상이 높아지는 것은 보상 모델이 부여하는 보상을 높일 수 있도록 모델이 잘 훈련되고 있음을 의미한다. 그림 4의 그래프에는 그림 3과 같은 시간 단계마다 RL 모델의 정책 엔트로피를 기록하였다. 엔트로피는 섀넌 엔트로피 (Shannon Entropy)를 말한다. 정책 모델의 엔트로피가 높아지는 것은 모델이 더 높은

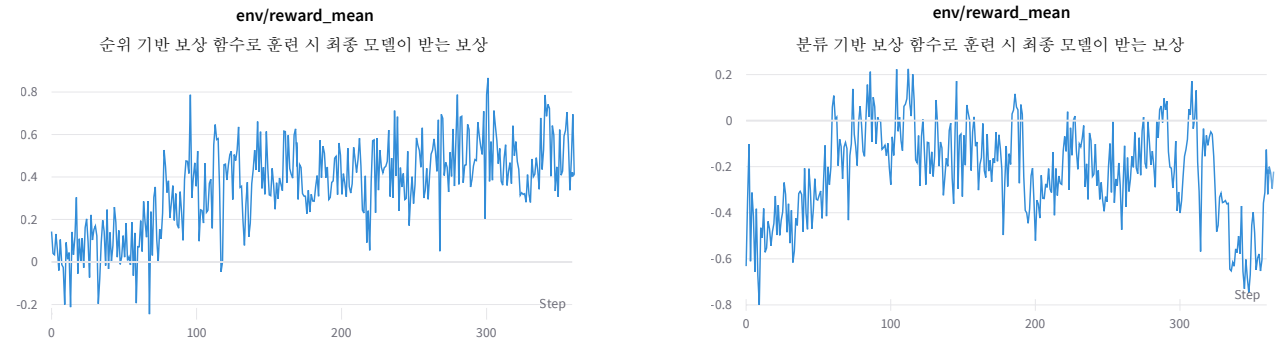


그림 3. RL 모델이 받는 보상 값 비교

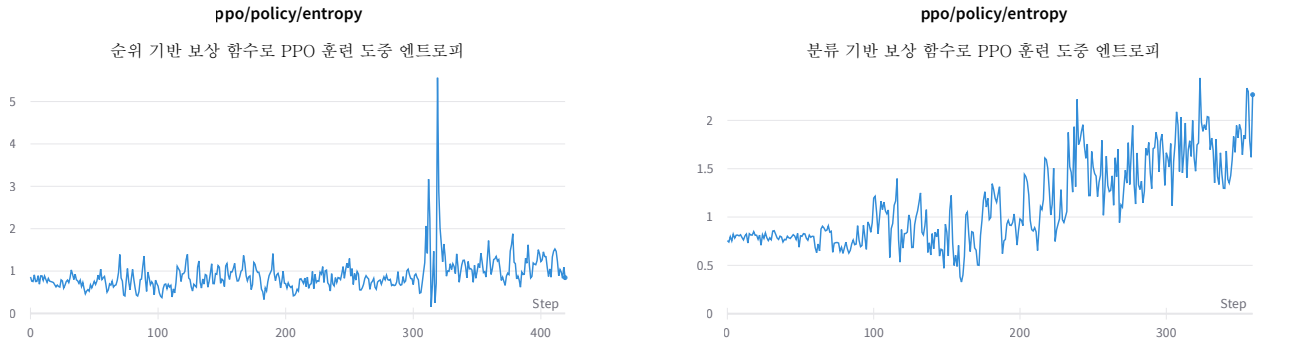


그림 4. RL 모델의 정책 엔트로피 변화 비교

무작위성을 띠게 되었음을 의미한다.

### 4.3 결과 분석

그림 3의 우측 그래프는 분류 기반으로 훈련된 보상 모델이 효과적으로 동작하지 않고 있음을 의미한다. 또한 순위 기반 방법에 비해 훈련 처음부터 더 낮은 보상을 받고 있다. 분류 기반 방법이 순위 기반 방법보다 효과적이지 못한 이유를 실험으로부터 추정해 보고자 한다. 분류 기반 방법으로 보상 모델을 훈련할 때, 보상 모델은 SFT 모델이 생성한 응답을 나쁜 응답으로 간주하여 -1에 가까운 보상을 주도록 훈련된다. 데이터셋의 정답 응답 문장처럼 생성할 경우에 +1에 가까운 보상을 주도록 훈련되었다. 따라서 그림 1의 3단계에서 RL 모델은 보상 모델에게 -1에 가까운 응답을 계속 받게 된다. 그러므로 분류 기반 보상 모델로 강화학습을 진행할 경우 그림 3처럼 낮은 보상만 계속해서 받게 된다. PPO 알고리즘에서는 행동자 모델이 상대적으로 높은 보상을 받도록 조정하게 되는데, 계속해서 낮은 보상이 들어올 경우 의도하지 않은 토큰 생성을 격려하게 되고, 결국 그림 4처럼 모델의 무작위성이 높아지는 결과를 낳는다.

## 5. 결론

본 연구에서는 최근에 고성능 언어 모델의 훈련에 사용되는 RLHF 튜닝 시 효과적인 보상 모델의 훈련 방법에 대하여 연구하였다. 실험 결과, 실제로 ChatGPT나 [14, 6]에 접목되었던 순위 기반 보상 모델이 더 효과적인 방법임을 알 수 있었다. 다만 순위 기반 보상 모델 훈련 방법은 실제로 사람이 모델의 응답에 대하여 순위를 매기는 작업이 많이 필요하기 때문에 비싼 방법이고 일반적인 연구 수준에서는 진행하기 힘들다. 따라서 본 연구에서는 ChatGPT의 프롬프트를 고정시켜 일관적인 사람이라는 가정 하에 실험을 진행하였다. 하지만 분류 기반 보상 모델 훈련이 단점만 있는 것은 아니다. 사람의 작업 없이 주어진 데이터셋만 활용하여 보상 모델을 만들 수 있기 때문에 더 연구해볼 가치가 있는 훈련 방법이다. 최근 상업용 고성능 언어 모델에 RLHF 방법론이 계속 적용되고 있는 것에 비해 학계의 RLHF에 대한 연구의 수는 적다. 또한 RLHF 방법론이 복잡한 훈련 단계와 많은 하이퍼파라미터를 요구하는 것에 비해 상업용 모델은 세부 훈련 과정을 공개하지 않아 일반 연구자들이 접근하기 어렵다. 이 연구가 RLHF 방법론 연구가 더욱 활성화되는 계기가 되었으면 하는 바람이다.

## 6. 관련 연구

RLHF 방법론은 ChatGPT [1]의 등장과 함께 많은 주목을 받았다. 그러나 RLHF 방법론은 ChatGPT가 등장하기 전부터 연구된 분야이다. ChatGPT가 공개되기 이전 OpenAI [17]의 고성능 모델인 davinci-003은 InstructGPT [14] 연구를 기반으로 RLHF 방법이 적용되었다. InstructGPT는 [5, 6]에서 제시한 RLHF 방법론을 기반으로 instruction tuning 방법을 결합하여 발전시킨 방법이다. RLHF에 사용되는 강화학습 알고리즘은 PPO[8]가 대표적이지만 [18]의 경우 어드밴티지 기반 행동자-비평자 정책 경사도 방법인 A2C(Advantage Actor-Critic) [19]를 사용하여 RLHF 방법론을 적용하였다. ChatGPT 이후 RLHF 방법론은 많은 고성능 모델 [4, 3, 18]에 적용되었다. 그러나 상업용 모델의 경우 대부분 세부 RLHF 훈련 방법에 대하여 공개하지 않았다.

오픈소스 RLHF 라이브러리는 구현체마다 모델 훈련 알고리즘이 조금씩 다르다. RLHF 훈련의 구현에서 가장 많이 사용되는 방법은 trl [15] 라이브러리에서 구현된 방법이다. 본 연구에서도 이 라이브러리에서 사용하는 방법을 채택하였다. 그 다음으로는 [7]의 방식이 많이 사용된다. 전자에서는 보상 함수를 밑바닥부터(from scratch) 훈련하고, 후자의 방식에서는 본 논문에서 다룬 분류 기반 보상 함수 훈련 방식을 조합하여 훈련한다. 마이크로소프트社의 [20]에서도 후자의 방식을 사용한다. 이와 같이 RLHF를 적용하는 구체적인 방법은 일관되지 않으며, 구현체마다 조금씩 다르다. 따라서 현재 RLHF를 크로스 엔트로피 기반의 훈련처럼 쉽게 쓸 수 있는 상황은 아니다. 각 구현체의 장점 및 단점도 현재로서 명확하지 않다. 그렇기 때문에, 본 연구가 RLHF 방법론을 더 명확하게 하는 초석이 되었으면 한다.

## 감사의 글

이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기술기획평가원의 지원을 받아 수행된 연구임 (No. 2020-0-00368, 뉴럴-심볼릭(neural-symbolic) 모델의 지식 학습 및 추론 기술 개발). 본 연구는 과학기술정보통신부 및 정보통신기술기획평가원의 대학ICT연구센터지원사업의 연구결과로 수행되었음 (IITP-2022-2018-0-01405). 이 논문은 2021년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2021R1A6A1A03045425). 이 논문은 2022년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2022R1A5A7026673).

## 참고문헌

[1] OpenAI-Blog, "Chatgpt: Optimizing language models for dialogue," 2022. [Online]. Available: https://openai.com/blog/chatgpt/

[2] Google, "An overview of bard: an early experiment with generative ai," 2023. [Online]. Available: https://bard.google.com/

[3] Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon, C. Chen, C. Olsson, C. Olah, D. Hernandez, D. Drain, D. Ganguli, D. Li, E. Tran-Johnson, E. Perez, J. Kerr, J. Mueller, J. Ladish, J. Landau, K. Ndousse, K. Lukosuite, L. Lovitt, M. Sellitto, N. Elhage, N. Schiefer, N. Mercado, N. DasSarma, R. Lasenby, R. Larson, S. Ringer, S. Johnston, S. Kravec, S. E. Showk, S. Fort, T. Lanham, T. Telleen-Lawton, T. Conerly, T. Henighan, T. Hume, S. R. Bowman, Z. Hatfield-Dodds, B. Mann, D. Amodei, N. Joseph, S. McCandlish, T. Brown, and J. Kaplan, "Constitutional ai: Harmlessness from ai feedback," 2022.

[4] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom, "Llama 2: Open foundation and fine-tuned chat models," 2023.

[5] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving, "Fine-tuning language models from human preferences," 2020.

[6] N. Stiennon, L. Ouyang, J. Wu, D. M. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. Christiano, "Learning to summarize from human feedback," 2022.

[7] CarperAI, "trlx," 2023. [Online]. Available: https://github.com/CarperAI/trlx

[8] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017.

[9] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," 2023.

[10] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, "G-eval: Nlg evaluation using gpt-4 with better human alignment," 2023.

[11] B. Naismith, P. Mulcaire, and J. Burstein, "Automated evaluation of written discourse coherence using GPT-4," *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pp. 394–403, Jul. 2023. [Online]. Available: https://aclanthology.org/2023.bea-1.32

[12] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, Vol. 1, No. 8, p. 9, 2019.

[13] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," 2023.

[14] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," 2022.

[15] huggingface, "trl," 2023. [Online]. Available: https://github.com/huggingface/trl

[16] Y. Xu, D. Wang, M. Yu, D. Ritchie, B. Yao, T. Wu, Z. Zhang, T. J.-J. Li, N. Bradford, B. Sun, T. B. Hoang, Y. Sang, Y. Hou, X. Ma, D. Yang, N. Peng, Z. Yu, and M. Warschauer, "Fantastic questions and where to find them: Fairytaleqa – an authentic dataset for narrative comprehension," 2022.

[17] openAI, "openai." [Online]. Available: https://openai.com/

[18] R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen, E. Chu, J. H. Clark, L. E. Shafey, Y. Huang, K. Meier-Hellstern, G. Mishra, E. Moreira, M. Omernick, K. Robinson, S. Ruder, Y. Tay, K. Xiao, Y. Xu, Y. Zhang, G. H. Abrego, J. Ahn, J. Austin, P. Barham, J. Botha, J. Bradbury, S. Brahma, K. Brooks, M. Catasta, Y. Cheng, C. Cherry, C. A. Choquette-Choo, A. Chowdhery, C. Crepy, S. Dave, M. Dehghani, S. Dev, J. Devlin, M. Díaz, N. Du, E. Dyer, V. Feinberg, F. Feng, V. Fienber, M. Freitag, X. Garcia, S. Gehrmann, L. Gonzalez, G. Gur-Ari, S. Hand, H. Hashemi, L. Hou, J. Howland, A. Hu, J. Hui, J. Hurwitz, M. Isard, A. Ittycheriah, M. Jagielski, W. Jia, K. Kenealy, M. Krikun, S. Kudugunta, C. Lan, K. Lee, B. Lee, E. Li, M. Li, W. Li, Y. Li, J. Li, H. Lim, H. Lin, Z. Liu, F. Liu, M. Maggioni, A. Mahendru, J. Maynez, V. Misra, M. Moussalem, Z. Nado, J. Nham, E. Ni, A. Nystrom, A. Parrish, M. Pellat, M. Polacek, A. Polozov, R. Pope, S. Qiao, E. Reif, B. Richter, P. Riley, A. C. Ros, A. Roy, B. Saeta, R. Samuel, R. Shelby, A. Slone, D. Smilkov, D. R. So, D. Sohn, S. Tokumine, D. Valter, V. Vasudevan, K. Vodrahalli, X. Wang, P. Wang, Z. Wang, T. Wang, J. Wieting, Y. Wu, K. Xu, Y. Xu, L. Xue, P. Yin, J. Yu, Q. Zhang, S. Zheng, C. Zheng, W. Zhou, D. Zhou, S. Petrov, and Y. Wu, "Palm 2 technical report," 2023.

[19] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," 2016.

[20] Microsoft, "Deepspeed," 2019. [Online]. Available: https://github.com/microsoft/DeepSpeed