Традиционные подходы RLHF обычно строятся с алгоритмами обучения с подкреплением (PPO), которые максимизируют функцию награды, которая оценивает качество ответа модели, основанную на человеческих предпочтениях, при этом контролируя отклонение новой модели от референсной через KL-дивергенцию. Однако такой подход имеет некоторые сложности: нестабильность обучения, требуется тщательная настройка гиперпараметров, ограниченная масштабируемость, высокая стоимость из-за обучения дополнительной модели.

Стандартный метод RLHF включает несколько шагов. Сначала человек собирает данные в виде подсказок и примеров ответов, после чего предварительно обученная модель донастраивается на этих данных. Далее обучается модель вознаграждения: она получает пары ответов (лучший и худший) и учится оценивать, какой из них предпочтительнее, обычно с помощью модели Брэдли-Терри. При этом минимизируется функция потерь - кросс-энтропия, она используется для обучения модели вознаграждения путем сравнения предпочтительных и не предпочтительных завершений, рассматривая ее как задачу бинарной классификации. Эти оценки затем используются как функция награды в алгоритме обучения с подкреплением, где языковая модель обучается выдавать ответы с более высокой оценкой, но при этом не слишком отклоняться от исходной версии - это контролируется через KL-дивергенцию.

Статья предлагает метод, который поможет упростить процесс обучения, сохранить и даже превысить качество с помощью метода DPO. Вместо того, чтобы явно обучать отдельную модель вознаграждения и использовать сложные алгоритмы RL, DPO напрямую оптимизирует политику модели, увеличивая относительную логарифмическую вероятность предпочтительных ответов по сравнению с не предпочтительными. Для этого используется бинарная кросс-энтропия, которая сравнивает вероятности новых ответов с вероятностями старой политики, что позволяет контролировать отклонение новой модели от исходной через KL-дивергенцию без дополнительной явной модели вознаграждения. То есть основная идея DPO - напрямую оптимизировать вероятность модели для предпочтительного варианта по сравнению с не предпочтительным, используя следующую функцию потерь:

$$L_{dpo} = -\ log\sigma(\beta[log\pi_{\Theta}(y_w|x)\ -\ log\pi_{\Theta}(y_i|x)\ -\ log\pi_{ref}(y_w|x)\ +\ log\pi_{ref}(y_i|x)]).$$

Эксперименты показывают, что DPO действительно работает лучше или не хуже PPO во многих задачах. В одном из примеров - генерация отзывов с нужной эмоциональной окраской - DPO достигает более выгодного баланса между качеством и стабильностью модели, одновременно меньше отклоняясь от исходной. В задачах суммирования текстов и диалогов DPO тоже показывает лучшие результаты, причем более устойчив к выбору параметров семплинга. Особенно интересно, что DPO хорошо работает и на данных из другой области, где PPO обычно тоже неплохо справляется, хотя у него есть доступ к большему объему вспомогательных данных.

В ходе изучения были выявлены следующие преимущества DPO над RLHF: отсутствие сложных RL алгоритмов, эффективность и устойчивость из-за минимальной настройки гиперпараметров, быстрее сходится, лучший компромисс между наградой и KL, проще масштабируется благодаря простоте DPO, дешевле в реализации.

DPO, как и методы RLHF, обучается на парах предпочтений, созданных либо другой моделью, либо ранней версией той же модели, из-за этого могут быть следующие проблемы: из-за того, что модель учиться на ответах, которые она сама не генерировала, новая модель будет генерировать ответы, не покрывающие обучающей выборкой, снижая качество и обобщаемость; если пары chosen/rejected содержат ошибки, то новая модель может унаследовать и усилить их; отсутствие обучение на собственных ответах ограничивает возможности модели в адаптации к новым ситуациям; качество и разнообразие обучающих пар сильно влияют на обучение; требуется тщательный сбор и обновление данных предпочтений.

Ссылка на публичный репозиторий с кодом:
https://github.com/Elina117/Direct-Preference-Optimization-gpt2

(описание практической части в readme.md)