TTRL

对于采样生成的N个候选输出，采用多数投票确定答案y^*作为共识
每个采样的输出根据其与共识答案的一致性获得reward
能直接集成PPO，GRPO等算法
Example:
Predictions[1,1,2,2,2,4,5,6]

- True Label(3): reward [0,0,0,0,0,0,0,0]
- Vote Label(2): reward [0,0,1,1,1,0,0,0]，reward hit rate:62.5% (negative reward)

```
# TTRL in GRPO
def correctness_reward(prompts, completions, **kwargs):
        res = [extract_answer(c[0]['content']) for c in completions]
        counter = Counter(res)
        most_common = counter.most_common(1) # [(res, cnt)]
        ans = most_common[0][0]
        answers = [ans] * len(completions)
        return [1  if r == a else -1  for r, a in  zip(res, answers)]
```

RLIF

使用置信度作为reward，通过模型输出的概率分布和均匀分布的差异衡量模型的自我确定性

$$\mathcal{J} = \max_{\pi_\theta} \mathbb{E}_{o\sim\pi_\theta(q)}\left[u(q,o) - \beta\mathrm{KL}\left[\pi_\theta(o|q)||\pi_{\mathrm{ref}}(o|q)\right]\right]$$

RENT

使用熵作为reward，最小化熵 – 提高置信度

$$H(p_t) = -\sum_{v\in V} p_t(v)\mathrm{log}p_t(v)$$

上述方法随训练进行都会出现熵坍缩的现象，使模型失去探索能力.