Projeto 1 – Metodologias em Modelos de Linguagem

This Summary Report aims to review the following article:

Training language models to follow instructions with human feedback

António Cruz (140129), Ricardo Kayseller (95813), Ricardo Pereira (120052), Ivan Magalhães (106586), Erik Daskalyuk (120062)

Section 1 - Objective of the Study

Introduction

Large language models like GPT-3 are powerful but often fail to follow human instructions accurately. Trained to predict the next word in internet text, they tend to produce outputs that are unhelpful, misleading, or even harmful. This study addresses the core problem of misalignment between model behavior and human intent, proposing a method to align language models using human feedback.

Objective of the Study

To address the problem, the primary objective of the study by Ouyang et al. (2022) is to investigate whether large language models (LLMs) can be effectively and scalably aligned with human intent using Reinforcement Learning from Human Feedback (RLHF). The authors aim to demonstrate that training models with human preferences, not just more data or larger architecture, can lead to outputs that are more helpful, honest, and harmless (HOH).

To validate this, the authors compare a small, aligned model, InstructGPT (1.3B parameters), with the original GPT-3 (175B), showing that the smaller model is consistently preferred by human evaluators. This suggests a shift in paradigm: alignment can outperform scale when optimizing language models for real-world use.

Section 2 - Methodology Phase 1: Supervised Fine-Tuning & Data Collection

Overview

The InstructGPT methodology addresses a critical limitation of large language models: simply scaling model size doesn't guarantee outputs that follow user intent or avoid harmful, untruthful, or unhelpful responses. The approach employs Reinforcement Learning from Human Feedback (RLHF) across three stages to align GPT-3 with the goals of being helpful, honest, and harmless.

The Three-Step RLHF Pipeline

1. **Supervised Fine-Tuning (SFT)**: Fine-tune GPT-3 on human-written demonstrations
2. **Reward Model Training**: Train a model to predict human preferences from ranked outputs
3. **Reinforcement Learning via PPO**: Optimize the policy using the reward model

Step 1: Supervised Fine-Tuning Details

Core Approach

SFT establishes the base policy through behavior cloning. Human labelers demonstrate desired output behavior for various prompts, and GPT-3 is fine-tuned on these prompt-demonstration pairs

using standard supervised learning. This creates the initial instruction-following capability that serves as the foundation for subsequent RLHF steps.

Data Collection Strategy

The SFT dataset (~13,000 training prompts) was assembled from two complementary sources:

- **API Prompts (Bulk Data):** Real user prompts submitted to OpenAI's API, providing authentic use cases. These underwent rigorous cleaning to remove PII and eliminate duplicates, with a 200-prompt limit per user.
- **Labeler-Written Examples (Bootstrapping):** Essential for initialization since base GPT-3 initially received few instruction-like prompts. These included plain tasks, few-shot instruction pairs, and prompts based on waitlist use cases.

The resulting dataset emphasized generative tasks (45.6%), open QA (12.4%), and brainstorming (11.2%).

Human Workforce

Approximately 40 contractors, hired through Upwork and ScaleAI, generated demonstrations and rankings. Selection criteria emphasize sensitivity to diverse demographic preferences and ability to identify harmful outputs. The team achieved 72.6% inter-annotator agreement, validating data quality and consistency.

Key Finding

A counterintuitive discovery emerged: while SFT models began overfitting after just 1 epoch (validation loss increased), training for up to 16 epochs improved both reward model scores and human preference ratings. This revealed that standard validation loss is not an optimal proxy for alignment quality - a crucial insight for training models aligned with human preferences rather than purely optimizing statistical metrics.

This SFT phase establishes the critical foundation for the subsequent reward modeling and reinforcement learning stages, transforming a general language model into one capable of following instructions while maintaining the flexibility for further alignment optimization.

Section 3 - Reward Model Training via Ranking

After training the SFT model, the authors wanted to build a new model capable of predicting which results humans prefer.

Human labelers ranked the model's output by preference, from 1 (worst) to 7 (best).

These rankings were then used to train a Reward Model (RM) that predicts a numerical value ("reward") representing how much a given output is preferred.

First, the previous model generates multiple responses for the same prompt. Then, labelers rank those responses, and the RM is trained to assign higher rewards to the preferred outputs.

The loss function used is:

$$loss(\theta) = -\mathrm{E}_{(x,y_w,y_l)}[\log\left(\sigma\left(r_\theta(x,y_w) - r_\theta(x,y_l)\right)\right)]$$

where:

- $\mathrm{E}_{(x,y_w,y_l)}$: average loss over all samples.
- $r_\theta(x, y)$: reward that the RM gives to response $y$for prompt x;
- y_w: preferred output.
- y_l: less preferred output.
- σ: sigmoid function, which ensures the model assigns a higher reward to the preferred response.

Reinforcement Learning via PPO

The Reward Model now acts as a judge for the SFT model. The process works as follows:

SFT model generates outputs → RM evaluates them → RLHF model learns to maximize the reward.

To achieve this, the authors used PPO (Proximal Policy Optimization), with the SFT model as the initial policy. However, during training, the model might "forget" what it learned during SFT. To prevent this, a penalty term was added to ensure the model stays close to the original behavior:

$$penalty = \beta \cdot log \frac{\pi_{RL}(y|x)}{\pi_{SFT}(y|x)}$$

This term is included in the PPO objective:

$$Objective(\Phi) = \mathrm{E}_{x,y}[r_\theta(x, y_w) - penalty]$$

where:

- $r_\theta(x, y)$: reward from the RM.
- $\pi_{RL}(y|x)$: current policy (the model being updated);
- $\pi_{SFT}(y|x)$: base SFT Model.
- β: controls how strongly the model is penalized for deviating from SFT.

Reinforcement Learning via PPO-ptx

After training, the authors found that the model followed instructions better but performed worse on general NLP tasks like summarization and question answering.

To fix this, they introduced a variant called PPO-ptx, which mixes part of the pretraining gradients with the PPO training process. This approach allowed the model to keep learning from the Reward Model while also preserving its general NLP abilities and prior knowledge.

Section 4 - Evaluation Strategy

To assess alignment, the researchers divided evaluation into three main categories:

Helpfulness

Helpfulness was primarily evaluated through human labeler judgments, using labeler preference ratings as the main metric. However, there can be differences between what users intend and what labelers interpret from a prompt.

Honesty

Honesty was measured by assessing the truthfulness of the model's statements about the world using two metrics:

- **Hallucinations**: The tendency to invent information in closed-domain tasks.
- **TruthfulQA Dataset**: Evaluates factual accuracy.

Harmlessness

Two approaches were used to measure toxicity:

- Human labelers judged whether an output was inappropriate for a customer assistant context (e.g., offensive, sexual, or violent content).
- Benchmark datasets such as **RealToxicityPrompts** and **CrowS-Pairs** were used for quantitative evaluation.

Evaluation Methodology

Quantitative evaluations were divided into two parts:

- Evaluations on API Distribution
- Evaluations on Public NLP Benchmark Datasets

Results on API Distribution

The 175B InstructGPT outputs were preferred over GPT-3 outputs 85% of the time, and over few-shot GPT-3 outputs 71% of the time. Larger PPO-ptx models performed slightly worse.

Overall, InstructGPT achieved the best results across all evaluated domains, suggesting it is more reliable and easier to control than GPT-3.

Generalization

To test for bias and overfitting, the researchers used held-out labelers — evaluators who did not participate in creating the training dataset. Results showed that InstructGPT generalizes well and does not overfit to the preferences of its training labelers.

Results on Public NLP Datasets

Truthfulness and Hallucination

InstructGPT models were more truthful and informative than GPT-3 on the TruthfulQA dataset, even without explicit instructions to "tell the truth." Improvements remained strong on non-adversarial prompts, though slightly smaller.

When instructed to respond with "I have no comment" when uncertain, PPO models followed this instruction better than GPT-3.

InstructGPT also halved hallucination rates (21% vs. 41%) on closed-domain tasks.

Toxicity and Bias

When given explicit instructions, InstructGPT generated less toxic output than GPT-3. However, when asked to produce toxic text, it was more toxic than GPT-3.

In terms of bias, InstructGPT and GPT-3 performed similarly. The PPO-ptx model displayed comparable bias levels but showed higher bias when instructed to act "respectfully."

Alignment Tax

During RLHF fine-tuning, some performance regressions were observed on public NLP datasets compared to GPT-3 — a phenomenon known as alignment tax.

By mixing pretraining gradients (PPO-ptx), these regressions were largely mitigated without reducing alignment quality. This method helped maintain or even improve performance while minimizing the alignment tax.

Section 5 - Limitations, Generalization, and Broader Implications

Implications for alignment research

The researchers talked about some inherent implications for the creation of this type of model

- The cost of increasing model alignment is modest compared to pre-training, specifically where this model uses less data to train compared to other models.
- The model generalizes instructions to settings that it wasn't supervised on, when given non-English tasks and code-related tasks.
- Performance degradation was mitigated by fine-tuning.
- The techniques were validated from research from real world.

Who are we aligning ourselves to?

Several factors influence the alignment of the model trained, in terms of its responses to the user, mainly the training data, the fine-tuning data, and the alignment method used. The researchers discussed some influences of their model:

- The model is aligned to the labeler's demonstrations and preferences. The labelers were mostly English-speaking people (US, Southeast Asia) hired via Upwork/Scale AI. The authors report inter-labeler agreement of about 73%.
- The labelling instructions that labelers use as a guide when writing demonstrations are created by the researchers, creating an inherent bias to their preferences.
- The training data is determined by prompts sent by customers, and as such, implicitly aligning what customers think what their end user's think is valuable for the use of the model, which may not always be the best solution for the end user.
- The customers of OpenAI are not representative of all potential users, nor by all individuals and groups impacted by language model use.

One way to mitigate this is to train models on the preferences of certain groups, or fine tune or prompted to easily represent different groups. Yet these models may still affect broader society, meaning that more discussions are needed to create harmony between several groups.

Limitations

The researchers talked about several limitations of their study and model:

In terms of methodology the labeler population is not fully representative of all users or all cultural/linguistic backgrounds. Most prompts and data were in English, and many of the

comparisons were labelled by only one annotator. The researchers note that having examples labelled by different people could help identify areas where their labelers disagree.

Even after fine-tuning, the models still make errors. They sometimes fail to follow instructions, hallucinate facts, generate biased or toxic outputs, or comply with harmful user instructions. For example: when instructed to be maximally biased, the model produced more toxic output than baseline.

When it comes to the prompts used, since the labelers are not the ones who generated said prompts, there can be a difference between what the user intended and what the labeler thought was intended.

Open questions

The researcher talks about several open questions to explore to further align language model behavior.

Methods to reduce the propensity to generate toxic outputs like collection of worst-case scenarios, filtering pre-training data, and combining different methods that worked better for other models.

Training the model to be less harmful is important, but whether an output is harmful will depend on the context. Also, harmful outputs can be beneficial for data augmentation.

To improve the controllability of the model it may be useful to allow users to specify preferences or adjusting behavior via control codes or sampling mechanisms.

Making comparisons of text may not be the best way to align models. There are several ways to criticize model responses, this being a human-computer interaction problem.

Broader Impact

Finally, the authors consider the broader societal and ethical implications of their work.

The fact that model outputs depend on the training data, labelers and user population. Meaning that deployment carries risks of bias, misalignment with under-represented groups, or unintended consequences.

Even positive use-cases may have side-effects. It is possible that persuasive text may increase usage time, which may not be good for the user's well-being.

Theres a need for ongoing monitoring and governance, to ensure models are used responsibly, that their alignment target is revisited as deployment contexts change.

It is emphasized that fine-tuning with human preferences is not sufficient alone and that other mechanisms like data filtering, monitoring, refusing harmful instructions are also necessary.