School of Engineering and Computer Science
Te Kura Mātai Pūkaha, Pūrorohiko

PO Box 600
Wellington
New Zealand

Tel: +64 4 463 5341
Internet: office@ecs.vuw.ac.nz

Project title not yet defined

James Thompson

Supervisor: Ali Knott

November 27, 2025

Submitted in partial fulfilment of the requirements for
Master of Artificial Intelligence.

Abstract

Also undecided at this time and will be filled in later.

Chapter 1

Introduction

In this section you should include a very brief introduction to the problem to the problem and the project.

Your project proposal should cover the following points:

- the engineering problem that you are going to solve;
- how you plan to solve your problem;
- how you intend to evaluate your solution; and
- any resource requirements for your project such as software, hardware or other resources that will be needed in the course of the project.

I am going to cite [2]
Your proposal should be not more than than 3 pages long.

1.1 The Problem

In this section you should give a brief description of the problem itself. You want to briefly explain the problem, why it is important to solve the problem and define your project aims. After reading this section, the reader should understand why it is a problem, believe that it is important to solve and have a clear idea of the aims of your project.

When describing the aims of the project, you should avoid vague, unmeasurable words like 'analyse', 'investigate', 'describe', and use specific, measurable words like 'implement', 'demonstrate', 'show', 'prove'.

For example:

Good The aim of this project is to implement and evaluate a management system for network switches;

is much better than:

Bad The aim of this project is to investigate management systems for network switches.

In the second case there is no idea of how much work is involved, and you will never know whether you have finished. You and your supervisor (and the markers of your project) may have very different ideas about what such an 'investigation' involves. Of course, it is possible that the task you set yourself is not achievable, but if you are clear from the outset this is less likely, and will more easily be corrected.

Chapter 2

Background

2.1 Large Language Models

A basic introduction to large language models that are transformer based. Including simple description of self attention, along with training process of pre-training and next token prediction.

2.2 Alignment of Large Language Models

2.2.1 What is alignment?

Introduction to the conecpt of alignment in large language models, why it is important, and common methods to achieve it.

2.2.2 Reinforcement Learning from Human Feedback

Having a model that is able to predict the next token is powerful however it doesn't guarantee that the model will behave in a manner that is aligned with the user preferences. Reinforcement Learning from Human Feedback (RLHF) [9] is a method that aims to align large language models to human preferences by using human feedback in the form of response rankings to guide the training process. This method was popularized by its use in training InstructGPT as giving it much better performance on following instructions compared to GPT-3 [3, 9].

How does RLHF work?

RLHF is separated into three distinct stages. Firstly is supervised fine-tuning (SFT) where a pre-trained language model is fine-tuned on a dataset of prompts and human written 'expected' responses to those prompts. The second step is collecting multiple responses from the SFT model to a set of prompts and having human labelers rank these responses from best to worst. This ranked data is then used to train a preference model (PM) that can predict which of two responses is better aligned to human preferences. Finally the PM is used as a reward signal in the [11] RL algorithm to further fine-tune the SFT model and create the final RLHF model.

The dataset of prompts are taken from the data distributions that you want to align towards. In the case of [9] the prompts were taken from playground use of an early version of InstructGPT. These prompts are intended to be representative of the types of queries and instructions that users would give to the model in practice. The human labelers that provide

the human feedback part is a small set (40) of contractors that are hired to both provide desired responses to the prompts as well as rank the responses generated by the SFT model.

Relevance

The idea of using human feedback to help train machine learning models is powerful, although it is not new [4]. The specific method of RLHF was adopted widely [13, 1, 12] yet was replaced with more effective methods to utilize human feedback such as Direct Preference Optimization (DPO) [10]. It does however form the basis for an alignment method that allows us to train models that directly align to human preferences rather than just next token prediction. It is one of the first key methods that created the post training methods for large language models.

2.2.3 Direct Preference Optimization

The idea presented in [9] introduces the concept of using human feedback to train large language models to be aligned to human preferences. However the process of RLHF is complex and requires training multiple models (SFT, PM, RLHF). Direct preference optimization (DPO) [10] is a method that simplifies this process by removing the need for reinforcement learning altogether. Instead DPO reformulates the preference modelling step to instead create a preference model implicitly from the model itself.

How does DPO work?

DPO starts wtih the same preferences data as RLHF, a set of prompts with responses ranked by humans. Like RLHF it starts with a pre-trained language model and is supervised fine-tuned on the prompts and highest ranked responses to create an SFT model. Instead of training a separate preference model, DPO uses the SFT model itself to create the DPO update rule:

$$\nabla_\theta \mathcal{L}_{DPO}(\theta) = -\mathbb{E}_{(x,y^+,y^-)\sim D}\left[\sigma(\hat{r}_\theta(x,y^-) - \hat{r}_\theta(x,y^+))\left[\nabla_\theta \log \pi_\theta(y^+|x) - \nabla_\theta \log \pi_\theta(y^-|x)\right]\right]$$

Where $\hat{r}_\theta(x,y) = \log \pi_\theta(y|x) - \log \pi_{SFT}(y|x)$ is the reward function defined as the log difference between the current model and the SFT model, σ is the sigmoid function, and (x, y^+, y^-) are the prompt, preferred response, and dis-preferred response respectively. [10] found that the first weighting term is important and can be interpreted as increasing the weight of the update when the reward estimate is wrong. The second term is there to increase likelihood of generated preferred responses and decrease the likelihood of dis-preferred responses. This update can be calculated with only 4 forward passes through the network per preference pair, making it much more efficient than RLHF [1].

Relevance

Aligning large language models to human preferences is a crucial step in ensuring that these models are useful and safe for real-world applications. DPO provides a more efficient and straight forward process. It is the most comonly used public methold for post training. However there are still more modern approaches that improve on alignment quality or efficiency such as [5, 7]. Due to its wide spread adoption DPO represents a strong baseline for alignment.

[1] RLHF will need to do many forward passes of the network to train the preference model, then it will still have to do 3 forward passes for each datapoint when it comes to the RL step.

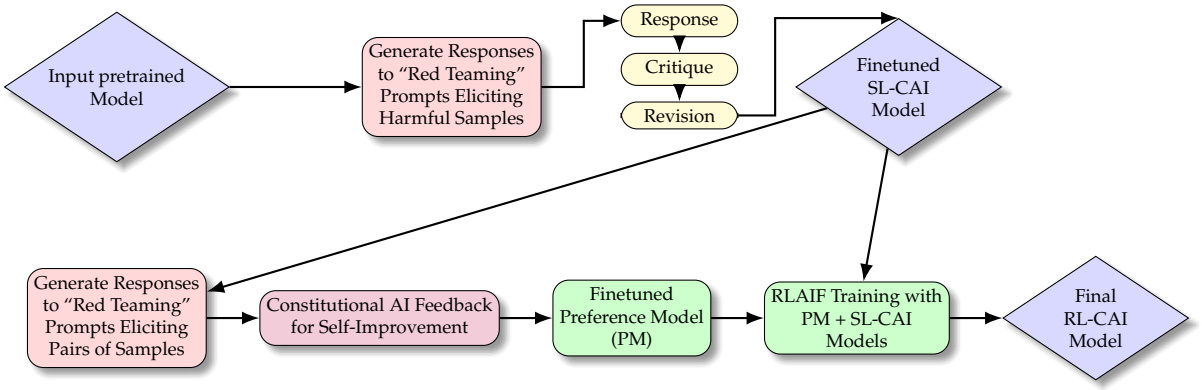

Figure 2.1: Overview of Constitutional AI process, adapted from [2]

2.2.4 Constitutional AI

What is Constitutional AI?

Constitutional AI (CAI) is a method of post-training a large language model to be aligned to set of principles that are outlined in simple natural language document called a constitution [2]. Importantly it works without the need for human labelling of data common in other alignment methods such as RLHF and DPO [9, 10]. INstead of human labelling CAI uses the model itself to generate feedback on its outputs based on the principles outlined in the constitution, this feedback is then used to further train the model to align it to the constitution.

How does Constitutional AI work?

CAI is a process that takes in a constitution, a pre-trained language model, and a dataset of prompts and outputs a model that is aligned to the constitution. The process consists of three main steps: self generative and supervised fine-tuning (SFT), preference modelling (PM), and reinforcement learning with human feedback (RLHF) [2]. The constitution was generated in an adhoc manner by the authors of [2] and consists of a set of principles that aim to make the model's outputs more helpful, honest, and harmless.

Supervised learning on revised responses

The first stage of the process involves taking the pre-trained model and using it to generate responses to a set of prompts. These prompts are designed to elicit harmful behavior. As the pre-trained model is designed to be helpful it will likely generate harmful responses to these prompts. These responses are then critiqued by the model itself using principles outlined in the constitution, only a subset of which are used for each response. The critique is then used to revise the original response to make it more aligned to the constitution. This process of generating responses, critiquing them, and revising them is done iteratively to create a dataset of revised responses. This dataset is then used to fine-tune the pre-trained model using self-supervised learning to create a model called the SL-CAI model.

Reinforcement learning with AI feedback

The next step involves using the SL-CAI model to generate pairs of responses to a set of prompts (overlapping with before or not). Then using the constitution the SL-CAI model is used to provide feedback on which of the two responses is better aligned to the constitution.

This feedback is then used to fine-tune a preference model (PM) that can predict which of two responses is better aligned to the constitution. Finally the PM and SL-CAI models are used together to do reinforcement learning in style of RFHL [9] to generate the final model RL-CAI. This final model is now aligned to the constitution without any human labelling of data. In the setup used in [2] the authors did in fact use some labels from human labelers which were used to provide the helpfulness signal in the preference model training, however in principle this step could of been done with AI.

Relevance

In the context of the time that this paper was to address the key dilemma of helpfulness vs harmlessness in large language models. The goal being to make a model that is both helpful and harmless, without sacrificing one for the other. Further research has expanded the horizons to be concerned with more than just harmfulness vs helpfulness trade off. Along with this there are concerns that methods like CAI and RLHF are simply superficially aligning models to avoid harmful outputs rather than deeply aligning to the values we want [6, 8]. As hinted at by the authors in [2], there is room for improvement in the CAI method to make it focus on more than just harmlessness vs helpfulness. This can be done by modifying the constitutions to include more diverse principles as well as modifying the red teaming prompts to target more diverse scenarios. In this way we can see how elements of CAI can be built upon by generating these diverse constitutions and red teaming prompts in a representative manner to create models aligned to a representative set of values.

2.3 Representing Human Values

Some some introduction to the problems of representing human values. How does one understand the values of a group and such. Furthermore relate it to how this ties into designing and building AI systems.

Chapter 3

Project Proposal

3. Proposed Solution

In this section you will explain how solve the problem, that is, how you intend to carry the project out. At this early stage you need to be both clear about what you are going to do and flexible enough to adapt to changing circumstances. Making an early plan will not prevent you from running into trouble, but it will help you identify possible problems early. For example, if you intended to run an experiment in HCI, you might realise early on that there would be problems gathering sufficient data to get reliable results, and that you should re-design your experiment.

Part of the planning process involves producing a timetable for when the work is actually going to be done.

Each part of the project should produce some output. For example you might plan on spending two weeks on background reading: the output of this will be a bibliography, and a possibly a literature survey for your report. Indeed, if you take the advice given above about having specific, measurable goals, you should describe this part of your project as:

Good Produce bibliography (est: 2 weeks)

rather than

Bad Background reading (est: 2 weeks)

Note that the methodology you outline here is dependent upon the type of project and engineering area. You must talk to your supervisor about this.

4. Evaluating your Solution

In this section you will explain how you will evaluate your solution once you have built it. The method of evaluation will be domain specific. Your supervisor should provide guidance as to what is an appropriate form of evaluation. For example, user testing for a HCI project or performance measurement for a Network Engineering project.

5. Resource Requirements

In this section you will detail any resource requirements such as hardware, software or access to subjects.

Bibliography

[1] Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, W., Han, Y., Huang, F., Hui, B., Ji, L., Li, M., Lin, J., Lin, R., Liu, D., Liu, G., Lu, C., Lu, K., Ma, J., Men, R., Ren, X., Ren, X., Tan, C., Tan, S., Tu, J., Wang, P., Wang, S., Wang, W., Wu, S., Xu, B., Xu, J., Yang, A., Yang, H., Yang, J., Yang, S., Yao, Y., Yu, B., Yuan, H., Yuan, Z., Zhang, J., Zhang, X., Zhang, Y., Zhang, Z., Zhou, C., Zhou, J., Zhou, X., and Zhu, T. Qwen Technical Report, Sept. 2023.

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

[3] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language Models are Few-Shot Learners, July 2020.

[4] Christiano, P., Leike, J., Brown, T. B., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences, June 2017.

[5] Ethayarajh, K., Xu, W., Muennighoff, N., Jurafsky, D., and Kiela, D. KTO: Model Alignment as Prospect Theoretic Optimization, Nov. 2024.

[6] Greenblatt, R., Denison, C., Wright, B., Roger, F., MacDiarmid, M., Marks, S., Treutlein, J., Belonax, T., Chen, J., Duvenaud, D., Khan, A., Michael, J., Mindermann, S., Perez, E., Petrini, L., Uesato, J., Kaplan, J., Shlegeris, B., Bowman, S. R., and Hubinger, E. Alignment faking in large language models, Dec. 2024.

[7] Hong, J., Lee, N., and Thorne, J. ORPO: Monolithic Preference Optimization without Reference Model, Mar. 2024.

[8] Hubinger, E., Denison, C., Mu, J., Lambert, M., Tong, M., MacDiarmid, M., Lanham, T., Ziegler, D. M., Maxwell, T., Cheng, N., Jermyn, A., Askell, A., Radhakrishnan, A., Anil, C., Duvenaud, D., Ganguli, D., Barez, F., Clark,

J., NDOUSSE, K., SACHAN, K., SELLITTO, M., SHARMA, M., DASSARMA, N., GROSSE, R., KRAVEC, S., BAI, Y., WITTEN, Z., FAVARO, M., BRAUNER, J., KARNOFSKY, H., CHRISTIANO, P., BOWMAN, S. R., GRAHAM, L., KAPLAN, J., MINDERMANN, S., GREENBLATT, R., SHLEGERIS, B., SCHIEFER, N., AND PEREZ, E. Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training, Jan. 2024.

[9] OUYANG, L., WU, J., JIANG, X., ALMEIDA, D., WAINWRIGHT, C. L., MISHKIN, P., ZHANG, C., AGARWAL, S., SLAMA, K., RAY, A., SCHULMAN, J., HILTON, J., KELTON, F., MILLER, L., SIMENS, M., ASKELL, A., WELINDER, P., CHRISTIANO, P., LEIKE, J., AND LOWE, R. Training language models to follow instructions with human feedback, Mar. 2022.

[10] RAFAILOV, R., SHARMA, A., MITCHELL, E., ERMON, S., MANNING, C. D., AND FINN, C. Direct Preference Optimization: Your Language Model is Secretly a Reward Model, July 2024.

[11] SCHULMAN, J., WOLSKI, F., DHARIWAL, P., RADFORD, A., AND KLIMOV, O. Proximal Policy Optimization Algorithms, Aug. 2017.

[12] SHAO, Z., WANG, P., ZHU, Q., XU, R., SONG, J., BI, X., ZHANG, H., ZHANG, M., LI, Y. K., WU, Y., AND GUO, D. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models, Apr. 2024.

[13] TEAM, G., ANIL, R., BORGEAUD, S., ALAYRAC, J.-B., YU, J., SORICUT, R., SCHALKWYK, J., DAI, A. M., HAUTH, A., MILLICAN, K., ET AL. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805* (2023).