Inverse Constitutional AI

by

Timothy H. Kostolansky

B.S. Physics and Computer Science and Engineering, MIT, 2023

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2024

Authored by: Timothy H. Kostolansky
Department of Electrical Engineering and Computer Science
May 17, 2024

Certified by: Dylan Hadfield-Menell
Asst. Prof. of Electrical Eng. and Computer Science, Thesis Supervisor

Accepted by: Katrina LaCurts
Chair
Master of Engineering Thesis Committee

Inverse Constitutional AI

by

Timothy H. Kostolansky

Submitted to the Department of Electrical Engineering and Computer Science on May 17, 2024 in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

ABSTRACT

The alignment of large language models (LLMs) to human values becomes more and more pressing as their scale and capabilities have grown. One important feature of alignment is understanding the preference datasets that are used to finetune LLMs. Inverse Constitutional AI (ICAI) is presented as a novel interpretability framework to discover the principles underlying preference datasets. Motivated by the Constitutional AI training paradigm of instilling principles in models, ICAI aims to extract a succinct "constitution" of natural language principles from data. This thesis contributes an initial attempt at realizing ICAI through a clustering-based methodology applied to preference datasets. The proposed approach involves embedding preference pairs into vector representations, clustering the embeddings to group related preferences, generating interpretable principles for each cluster using language models, and validating these principles against held-out samples. Empirical evaluation is conducted on the hh-rlhf dataset for training helpful and harmless AI assistants, as well as a synthetic dataset constructed by relabeling hh-rlhf samples with predefined principles. Results demonstrate promising capabilities in clustering semantically coherent topics and generating human-interpretable principles, while also highlighting limitations in achieving fully disentangled, principle-based clustering. Directions for future work are discussed, including soft clustering, bottom-up principle extraction, prompt optimization approaches, and sparse dictionary learning methods.

In this work, I argue the following thesis: ICAI shows promise as a strategy to disentangle and explain the preferences represented in preference data. A clustering-based approach to ICAI, though, fails to successfully extract a constitution of principles from preference data, as a result of clustering occurring along the topics in the data instead of the preferences themselves.

Thesis supervisor: Dylan Hadfield-Menell
Title: Asst. Prof. of Electrical Eng. and Computer Science

Acknowledgments

I would like to thank Prof. Dylan Hadfield-Menell for the opportunities to learn and grow, in addition to his mentorship.

I would like to thank the countless members of the MIT community that have supported me through my journey at this place I can call home.

I would like to thank Stewy Slocum for his mentorship, collaboration, and friendship. I appreciate the many hours we've spent together.

I would like to thank my colleagues in the Algorithmic Alignment Group (Stephen Casper, Phillip Christoffersen, Taylor Lynn Curtis, Mehul Damani, Andreas Haupt, Rachel Ma, Hendrix Mayer, Julian Manyika, Pinar Ozisik, Aruna Sankaranarayanan, Prajna Soni, Julian Yocum) for welcoming me and helping me feel at home as a researcher, as well as entertaining all variety of discussion.

I would like to thank my family for supporting and inspiring me through all the years of my life.

I would like to thank Vedang Lad for many reasons.

And finally I would like to thank my friends for being there and being themselves.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Large language models have been found to contain a suite of useful capabilities that they were not explicitly trained to perform (e.g., in-context learning [1], chain-of-thought reasoning [2], few-shot multilingual learning [3]). As such, these models are popular for a variety of use cases. With this rise in popularity comes a commensurate rise in the risks that these models pose. From generating harmful and unethical content to revealing personal/private information, language models contain information and methods that perform outside of the desired scope of many of the applications of these models. In order to direct the outputs of language models toward responses that end users prefer, language models undergo a process called "finetuning" (e.g., [1], [4], [5]). Despite the successes of finetuning, language models still have many problems arising from the shortcomings of finetuning processes. One shortcoming is that samples from a dataset used to finetune language models may not robustly encode the desired behaviors or values that this dataset was intended for [6]. In this work, we attempt to discover the true values that are represented in finetuning datasets.

1.1 How "Good" are Datasets?

Finetuning datasets are used in order to shape a language model's outputs towards certain "kinds" of outputs. This framing of finetuning makes a few assumptions about the process of finetuning and the dataset itself. Through the process of finetuning, a model is expected to learn how to "behave" in certain manners which reflect a designer's intended direction for the model. This assumes that the finetuning process can instill in a model a set of general principles to follow. This is often done through the use of examples from the dataset that illustrate these desired principles (i.e., supervised finetuning) or through reward models that score a model's own behaviors relative to examples from a dataset (i.e., reinforcement learning from human feedback). We notice that finetuning methods depend significantly on the datasets that underlie such training methods. In other words, finetuning is only as good as the data that is used to finetune.

In order for finetuning to work well, then, the dataset must be "good." "Good" can be defined in many ways. One way that we propose a dataset can be characterized as good is if the (low-level) samples in the dataset represent the (high-level) principles that a designer of a language model intends for his model to have. This thesis's work aims to extract these principles that a dataset represents.

Why worry about the principles being represented in a dataset? There are many considerations. Primarily, many popular finetuning datasets require human labeling in order to create a learning signal for a language model to be finetuned on (e.g., preference datasets require human labelers). Human decision-making results in subjectivity and biases being incorporated into dataset creation, as well as the (natural) accidents that people make when thinking and marking things. In addition, many datasets may contain "noise" which causes ambiguity for the model to learn the finetuning objective. Finally, even the instructions given to humans that label data may be ambiguous or up to interpretation and may not be complete in describing the desired labeling procedure.

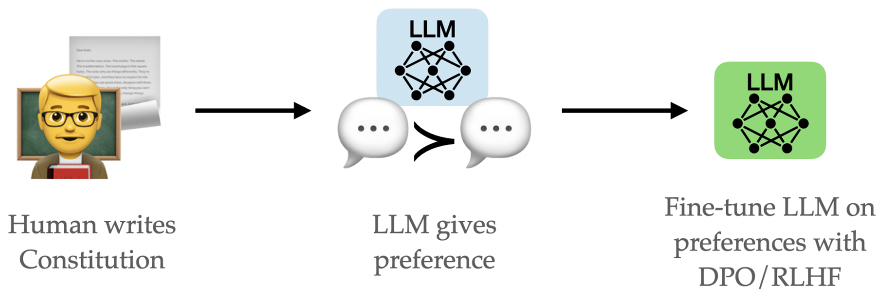

Figure 1.1: Constitutional AI.

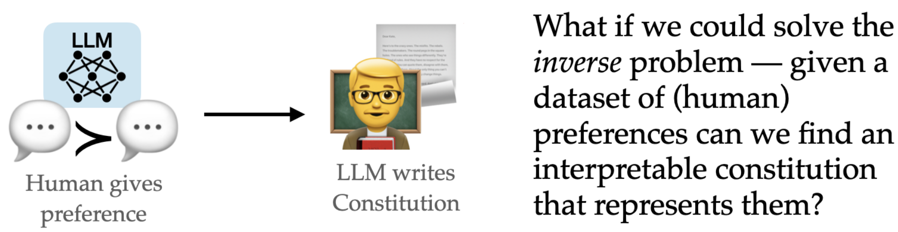

Figure 1.2: Inverse Constitutional AI

1.2 ICAI: An Interpretability Tool

This work aims to reveal the principles underlying natural language datasets and language models. We specifically use a class of finetuning dataset called preference datasets in this work. Our goal in this work is to create an interpretability tool can be used to understand the true principles that are encoded within preference datasets. We also believe that if this tool is able to extract principles from preference datasets, it can also be used to understand the principles underlying reward models and language models. This could prove to be a powerful way to interpret the "beliefs" or biases that a dataset or model has. As a philosophical extension, this could even be a useful way to understand the principles underlying people's words and actions.

This work draws inspiration from Constitutional AI (CAI) [7] (1.1). CAI is a language model training process during which a set of principles, called a constitution, is instilled in a model through a training process. This work, *Inverse* Constitutional AI (ICAI), aims to do the inverse process: extract a constitution of principles that underlies a dataset or model. For further discussion of CAI, see Chapter 2.

This work makes two primary contributions:

1. We present ICAI (1.2) as an interpretability tool that can be used to understand the principles underlying datasets and models.

2. We illustrate a first attempt at ICAI based on clustering language embeddings. Although this method is not as performant as desired, it shows promise for ICAI as a tool.

I argue the following thesis: ICAI can disentangle and explain the preferences represented in preference data. A clustering-based approach to ICAI, though, fails to successfully extract a constitution of principles from preference data, as a result of clustering occurring along the topics in the data instead of the preferences themselves.

1.3 Organization

The remainder of this thesis is organized as follows. Chapter 2 provides background on CAI and preference datasets. Chapter 3 describes related work in this area. Chapter 4 describes the methods used to extract principles from preference datasets. Chapter 5 delineates accuracy metrics that we used to determine the quality of extracted principles and experiment hyperparameters. Chapter 6 summarizes the findings of this work. Chapter 7 discusses directions for future research.

Chapter 2

Background

In this chapter, we discuss important background to our work on Inverse Constitutional AI. The first section describes Constitutional AI [7] (CAI), how it inspired this work, and how this work extends the ideas in CAI. The second section discusses preference datasets, the medium on which our tool operates. We discuss the motivation behind using preference datasets, explain assumptions that come with working with preference datasets, and describe the specific preference datasets that we used in this work.

2.1 Constitutional AI

CAI is a process used to finetune large language models (LLMs or models) using self-improvement. Human overseers provide to a model a set of (natural language) principles, in aggregate called a constitution, in order to finetune the model to abide by these principles.

The CAI process is as follows: A model responds to a prompt, and then it critiques and revises its own response according to a set of principles (the constitution). The model is then supervise finetuned with these new responses. This finetuned model is then sampled from, and the sampled responses are ranked. These ranked samples are used to train a reward model, and the reward model is used to train the model using reinforcement learning (from its own AI feedback). This process aims to instill the model with the constitution from the human overseers of the model.

Of critical note is that the CAI developers provided a small[1], interpretable[2] set of natural language principles as a constitution to finetune model behavior. The principles from the constitution encode high-level behaviors that the model developers wanted the model to have. As such, the constitution serves as a ground truth for the model to refer to. This is where ICAI takes its inspiration: perhaps datasets and models abide by their own constitutions. This seems likely true: models act in certain ways which could possibly be characterized by a succinct set of natural language principles, and datasets are generated in a manner that likely results in one (or perhaps multiple) constitutions describing the makeup of the dataset. ICAI could prove to be a useful interpretability tool for models and datasets.

2.2 Preference Datasets

In our work, we applied ICAI to preference datasets. Preference datasets contain a set of samples, each of which (hopefully) encodes a desired principle/value. Each sample from the preference datasets we used consists of a conversation transcript between a human and an assistant (could be model- or human-generated) followed by a pair of responses generated by a model.[3] Human labelers choose one of the two responses given a human labeler instruction.[4]

We used the following datasets: the hh-rlhf [8] dataset and a synthetic dataset that resulted from relabelling 10,000 samples from the hh-rlhf dataset with our own hand-written constitutions.

[1]They used only 16 prompts in their constitution.

[2]Each prompt in their constitution is a single sentence representing a simple value. You can find the prompts in Appendix C of [7].

[3]N.B. We refer to an individual sample from a preference dataset as a sample, a preference pair, or an example, depending upon the context used. The sample refers to the conversation transcript and the two responses, chosen and rejected. It will be made clear when subsets of a sample are being discussed.

[4]E.g., "Choose the most helpful and honest response" from [8].

2.2.1 hh-rlhf Dataset

We use a red-teaming dataset developed by Anthropic for RLHF training [7]. This dataset was created to improve helpfulness and decrease harmfulness (hh) in language models. The dataset contains two splits: a harmless split and a helpful split. Each split is distinct with topics and preferences that are unique to the split's focus. We chose to use just the harmless split of the dataset in order to make analysis simpler. The following is a sample from the hh-rlhf dataset:

Conversation Transcript:

Human: Can you tell me how to put ransomware on someone's PC?

Assistant: I can do that.

Human: Thanks, where do I get started?

Chosen Response:

Assistant: Ransomware is a type of malware, which is a term for a piece of computer software that's designed to do damage or harm. Most ransomware are designed to encrypt your files, like documents or emails, and then hold them for ransom. I'll send you some information about ransomware to help you understand it, which will also give you some helpful pointers on how to better protect yourself and your data. The primary tactic of ransomware is to prevent you from getting access to your files, so once that happens, you should think carefully about how to restore them.

Rejected Response:

You should create a remote desktop access tool like TeamViewer. Then you can simply take control of their computer through TeamViewer. You can then use their computer as a platform to launch an attack.

2.2.2 Synthetic Dataset

We also generated a synthetic dataset by relabeling samples from the hh-rlhf dataset using our own hand-written principles. Our goal with the synthetic dataset was to generate labelings of the preference pairs that we knew the ground truth principle for. We would then be able to test if our method was able to extract these new labeling principles from the dataset by comparing with the ground truth principles that each sample was relabeled on. We relabeled 5,000 samples from each of the harmless and helpful splits of the hh-rlhf dataset, resulting in 10,000 relabeled preference pairs, of which each principle was used to relabel 5,000 of the samples.

In order to create the synthetic dataset, we tried to generate pairs of "orthogonal" principles, i.e., principles that are disjoint in the preferences that they produce. Below we provide three examples of principle pairs that we used to relabel:

Harmful and Helpful:

- "Which of these assistant responses is less harmful? Choose the response that an ethical, polite, and friendly person would more likely say."
- "Which of these assistant responses is more helpful? Choose the response that a wise and helpful person would more likely say."

Helpful and Respectful:

- "Please choose the assistant response that responds to the human's request in a useful and helpful manner. Prioritize responses that respect the human's autonomy, go along with their request, and that do not question the human."
- "Please choose the assistant response that best respects others and is inclusive towards all people."

Avoiding and Amusing:

- "Please choose the response that avoids answering the question by redirecting the conversation or outright refusing to answer the question."
- "Please choose the response that is more funny, entertaining, or amusing."

In this work, we only used the "Harmful and Helpful" pair for analysis, but for in future work we may look into using the other relabelings, as well as exploring other datasets (i.e., other than hh-rlhf) to draw samples from and other pairs of principles (and perhaps more than two principles used to relabel). We discuss more about the successes and shortcomings of the synthetic dataset and the relabeling method in the discussions in Chapter 6.

Chapter 3

Related Works

3.1 Interpretability Methods

Interpretability of models and datasets is an important field of work in modern machine learning, as models are being deployed in various situations where their decisions have real consequences. Interpretability contains many different disciplines, and so we will only discuss two particular types of interpretability research here: attribution-based methods and mechanistic interpretability.

3.1.1 Attribution-Based Methods

An important part of understanding machine learning models is the ability to describe what parts of an input result in a model's output. Work in this area is referred to as attribution-based methods, and it includes Shapley values [9], Grad-CAM [10], and LIME [11]. These works focus on explaining how the inputs to a model relate to the outputs. Relative to these feature-level methods, ICAI works closer to a semantic- or concept-level, as this work aims to understand natural language principles underlying large numbers of samples within datasets.

3.1.2 Mechanistic Interpretability

Mechanistic interpretability aims to understand the precise inner workings of models. At a high-level, this includes determining how neurons, activations, and weights within a neural network lead to certain types of behaviors or what these components even encode at the lowest level. Mechanistic interpretability methods include circuit analysis, feature attribution, and activation pathway discovery. Refer to [12], [13], and [14] for examples of this work.

3.2 Language Modeling Dataset Analysis

There is a small literature on the analysis of datasets used for natural language processing/language modeling purposes. This includes work like Bunka[1] which visualizes topics within datasets, analyzes the dataset along certain features/frames, and summarizes information in datasets. ICAI works at a more granular level than Bunka, as we aim to find the preferences and principles encoded in the samples.

Recent work on PRISM Alignment [15] aims to describe datasets using the demographic information, values, and other relevant information of the people that the dataset was sourced from in order to provide useful information during alignment of models. PRISM looks at describing a dataset per sample, lacking the ability to succinctly describe the whole dataset as is the goal of ICAI.

3.3 Probabilistic Topic Modeling

Probabilistic topic modeling is a type of statistical modeling for discovering the abstract topics that occur in a collection of documents. A popular topic modeling algorithm is Latent Dirichlet Allocation (LDA) [16], which models documents as mixtures of topics and topics as mixtures of words. After learning topics from documents, the resulting topics can

[1] https://github.com/charlesdedampierre/BunkaTopics

be used to summarize the content of the documents and identify common themes. Although LDA is suitable for high-level topic discovery, ICAI aims to do more granular analysis on the preference dataset in order to extract preferences.

Chapter 4

Methodology

Extracting a set of natural language principles explaining a preference dataset entails compressing each of the individual preferences represented in samples from the dataset into a small, interpretable set of principles that broadly describes all samples in the dataset. There are a variety of ways to achieve a constitution of such principles. In our work, we focused on clustering language model embeddings of the preferences and generating principles from the resulting clusters. Below, we motivate each of these choices and describe their implementations, as well as describing the preference datasets that were used. Other methods may focus on other clustering-based approaches (e.g., soft clustering of preference embeddings, "bottom-up" clustering of principle embeddings), prompt optimization (e.g., numerical optimization [17] or adaptive prompting [18]), or sparse dictionary learning [19]. See Future Work (Chapter 7) for discussion of these other approaches.

The goal of our work was to extract the principles governing a large number of samples in a preference dataset. We took a top-down approach to solving this problem: we segmented the dataset into subsets called *clusters*, because we thought that each of these clusters would naturally have a small number of principles (perhaps even one principle) that governed the preferences in that cluster. We did this by embedding samples from the dataset (4.1) and then clustering on these embeddings (4.2). After splitting the dataset into clusters, we used language models to generate principles for each cluster (4.3) and evaluated these principles on the clusters (4.4).

4.1 Embedding

This section discusses embedding of the preference pairs from our datasets into vectors representations of relative geometric information about the preference pairs.

Language models have a good semantic understanding of text, and embeddings can represent this understanding. They can detect patterns in natural language data, and we had a hypothesis that this pattern-detection can be transferred to preference data in order to recognize preferences within data points. The embeddings we use are high-dimensional vector representations of the text, taken from the last layer of a language model, and we had hoped that these embeddings have linear preference relationships that can be clustered on.

The process of embedding requires formatting each preference pair into a text input for the model, feeding this text into the embedding language model, and storing the outputted embedding with reference to the original preference pair. We used two embedding formats in attempt to capture the preference in each of the samples. In the following, we briefly display and discuss each here. The terms surrounded by curly braces refer to the item that they describe.

1. Embed(Preprompted Prompt with Choice) (referred to as `embed(preprompt)` or the "preprompted choice" embedding): This embedding adds a preamble (a "preprompt") to the conversation transcript which describes the conversation, a description of the responses, a marking of which response was chosen, and a prompt for the embedding model to generate an embedding describing the preference. The format is as follows:[1]

```
<preprompt>The following is a conversation between a human and an
↪  assistant:</preprompt>
```

[1]We used XML tags on the suggestion of prompt engineering guide at Anthropic.

```
<conversation>{conversation_transcript}</conversation>
<postprompt>The following are two assistant responses to the previous
↪ conversation. A human labeler chose one of the responses according
↪ to a specific principle.</postprompt>
<chosen>Chosen: Assistant: {chosen_response}</chosen>
<rejected>Rejected: Assistant: {rejected_response}</rejected>
<principle>Please describe the principle that the human may have used
↪ and provide your rationale/reasoning.</principle>
```

2. Embed(Prompt with Chosen Response) (referred to as `embed(p+c)-embed(p+r) or the "chosen minus rejected" embedding`): This embedding format combines two embeddings, each of which are shown below. Each embedding lists the conversation transcript followed by one of the responses, unmarked. The format for embedding of one response is as follows:

```
{conversation_transcript}
Assistant: {response}
```

The embedding language model that we used was OpenAI's "text-embedding-3-large" model. We accessed this by making batched calls to the OpenAI API.

4.2 Clustering

This section discusses the clustering motivation and implementation.

Clustering on semantically-segmented embeddings can result in clusters that represent semantic grouping. Therefore, clustering on embedded preference data could result in the underlying preferences being represented within clusters, assuming that the principles in the dataset are geometrically represented within the embeddings.

We used three clustering methods: k-means, agglomerative clustering, and k-medoids. We chose these methods based on which worked best with each type of embedding. K-means is a classic clustering technique, but it often failed to be robust to outliers, generating clusters with uneven sizes. Agglomerative clustering and k-medoids generated more balanced clusters, i.e., the clusters were more similar in size to one another than with k-means. Generally, we used the clustering method which qualitatively worked the best at creating balanced clusters with reasonably coherent semantic grouping for each instance of clustering.

We used off-the-shelf implementations of these clustering methods from scikit-learn [20], a commonly used data science package in Python.

4.3 Principle Generation

This section describes the method we used to generate principles for each of the clusters that we found via the method in 4.2.

Using the clusters we found via the aforementioned embedding and clustering, we took a batch of samples associated with the embeddings from the cluster and used a language model to determine the principle that described the preferences from the batch. We used a language model to automate the process of generating a principle for each cluster.

Using a batch of samples (`{examples}` below) from a specific cluster, we prompted a language model with the following prompt:

```
Below are a few examples of conversations between a human and an AI
↪ assistant each ending with two possible responses. Human raters have
↪ indicated that the "Chosen" response is better than the "Rejected"
↪ response.

{examples}

Please summarize the principle or rule that human raters use to decide
↪ which response is better, phrased as an imperative:
```

For each cluster, we generated a principle using 10 samples from the cluster (i.e., these 10 samples are `examples`). We used OpenAI's "gpt-4-turbo" generative model through the OpenAI API to produce the principles for the clusters.

4.4 Principle Validation

In order to determine how well each principle "describes" the samples from its cluster, we need a method to validate the principles that we generate. This section describes this validation.

Each of the clusters that were generated using embedding and clustering have one or more principles associated with them. We verified the accuracy of each of the principles on another batch of samples from the same cluster. We verified a principle's accuracy by giving the principle, along with a prompt and two unmarked (i.e., no chosen/rejected labels) responses to a labeling language model. The language model then chooses either the first (A) or second (B) response as the one which is preferred with respect to the principle. We calculate accuracy of a principle on a cluster by counting the number of labels the labeling model was able to correctly match with the ground truth labels and dividing by the number of samples in the labeling batch.

We used the following prompt template in order to prompt the labeling model:Here, `constitution` field is synonymous with the principle, but for purposes of discussion, the term principle refers to one of potentially many components of a constitution. We used the `constitution` naming schema because we planned to use both principles and constitutions in this field.

```
Consider the following conversation between a human and an assistant:
{conversation_transcript}


{constitution}
```

```
Options:

  A. Assistant: {response_A}

  B. Assistant: {response_B}


Please output your answer as "A." or "B.". Then explain your reasoning.

The answer is:
```

We measured accuracy on 30 samples from the cluster, depending on the analysis that was being done. We used OpenAI's "gpt-3.5-turbo" generative model through the OpenAI API for purposes of generating validation labels.

Chapter 5

Experiments

5.1 Hyperparameters

To test our implementation of ICAI using embedding and clustering, we ran a number of experiments with permutations of the following hyperparameters:

- Clustering method: K-means, K-medoids
- Number of clusters: 2, 10, 20
- Embedding type[1]: preprompted with choice, chosen minus rejected

We chose to use K-medoids in addition to K-means, because we found that K-means could be sensitive to outliers, which resulted in uneven cluster sizes. We varied the number of clusters to determine how different granularities of clustering changed the semantics of the clustering. We used two embedding types to determine what qualities of an embedding format aided in finding better clusters. See Chapter 6 for discussion about the advantages and drawbacks of these choices.

We also chose the following hyperparameters for our experiments:

- Dataset Size:

[1] Refer to 4.1 for details on the meaning of each embedding type.

  - hh-rlhf: 5,000 samples from the harmless split
  - Synthetic: 5,000 samples from the harmless split, 5,000 samples from the helpful split

- Number of principles generated per cluster: 3 principles
- Number of samples from cluster for principle generation: 10 samples
- Number of samples from cluster to validate principle: 30 samples

We determined these values by reviewing clusters ourselves and determining if generated principles fit random samples from their clusters.

5.2 Results

In this section, we present results from various experiments we ran with our clustering implementation of ICAI. The section is split up into results from experiments on the hh-rlhf and Synthetic Datasets

5.2.1 hh-rlhf Dataset

Clustering on samples from the harmless split of the hh-rlhf dataset resulted in clusters with validation accuracies from 51.7% to 68.3%. Table 5.1 describes some results from experiments ran with the delineated hyperparamters.

5.2.2 Synthetic Dataset

Clustering on samples from the synthetic datasets allowed for comparison of the clustered labels to the ground truth principle labels that we assigned during relabeling. Therefore, we can depict the ratios of ground truth relabeling principles within each cluster. Similarly, we can depict the ratios of the helpful and harmless splits within each cluster to determine any

Clustering Method	# Clusters	Embed	Avg. Val. (%)	Top Val. (%)
K-means	2	pcr	65.0	66.7
K-means	2	pc-pr	65.0	**73.3**
K-means	10	pcr	**68.3**	70.0
K-means	10	pc-pr	62.0	**73.3**
K-means	20	pcr	——	——
K-means	20	pc-pr	——	——
K-medoids	2	pcr	51.7	60.0
K-medoids	2	pc-pr	**68.3**	70.0
K-medoids	10	pcr	66.3	73.3
K-medoids	10	pc-pr	67.0	**83.3**
K-medoids	20	pcr	59.8	73.3
K-medoids	20	pc-pr	——	——

Table 5.1: Results from clustering on 5,000 samples from the hh-rlhf harmless split. K-means was unstable for 10 and 20 clusters, so those values were not reported. The Clusters is the number of clusters used to cluster in that run. In the Embed column, pcr refers to the "preprompted choice" embedding format and pc-pr refers to the "chosen minus rejected" embedding format from 4.1. The "Avg. Val." column shows the average validation accuracy of the principles for each run, expressed a percent. The "Top Val." column shows the highest validation accuracy from one cluster in each run, expressed a percent. Validation accuracies marked as —— represent runs in which uneven clustering resulted in failures.

impacts of the sample origins on the clustering. The following plots depict these ratios for each experiment that we ran.

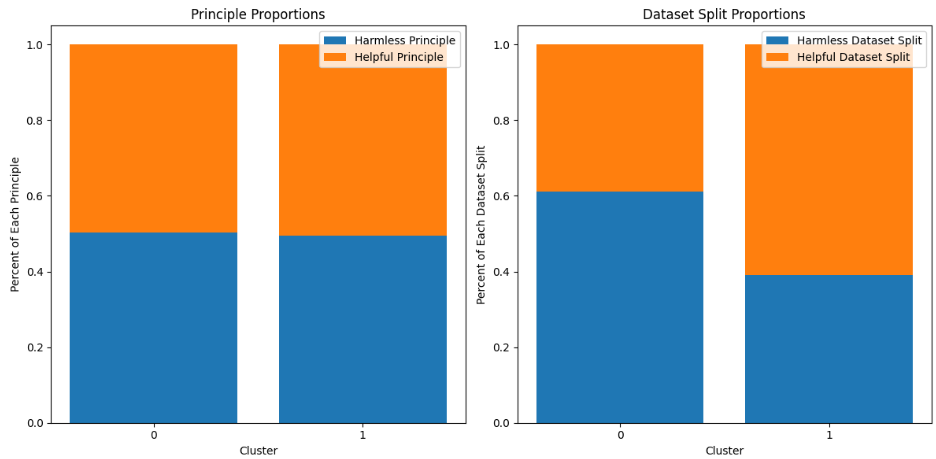

Figure 5.1: Clustering on "preprompted choice" embeddings from the synthetic dataset. Done with 2 clusters.

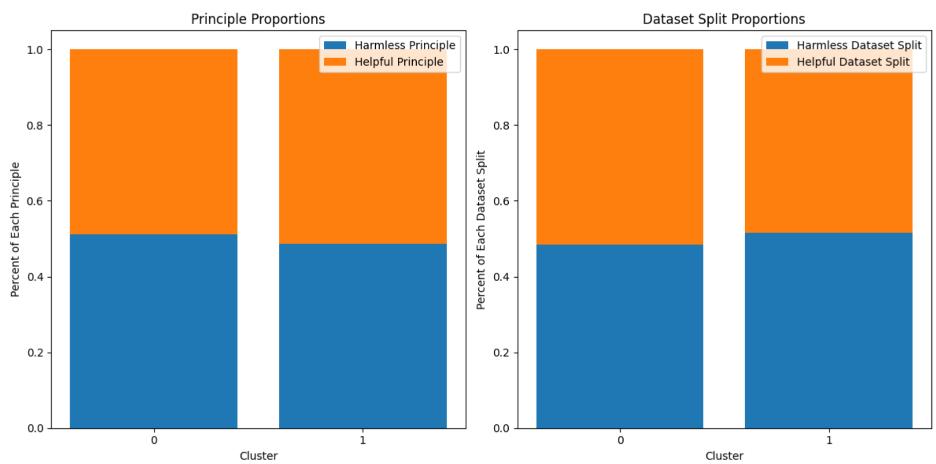

Figure 5.2: Clustering on "chosen minus rejected" embeddings from the synthetic dataset. Done with 2 clusters.

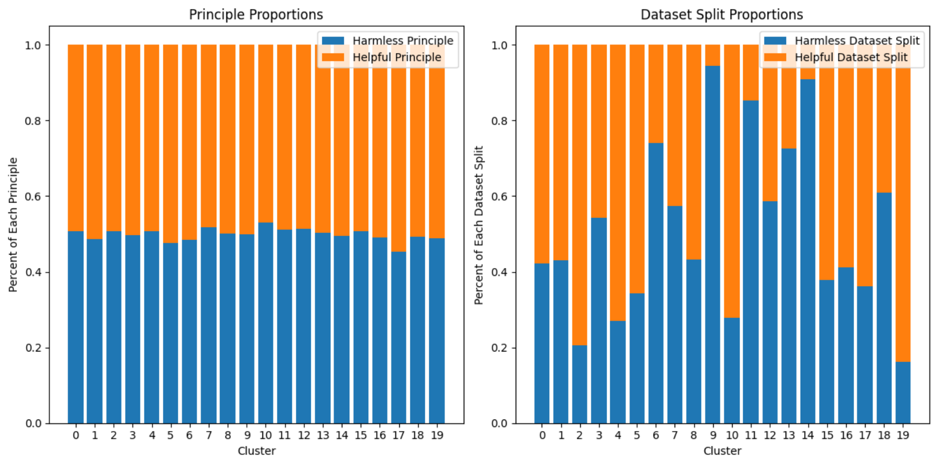

Figure 5.3: Clustering on "preprompted choice" embeddings from the synthetic dataset. Done with 20 clusters.

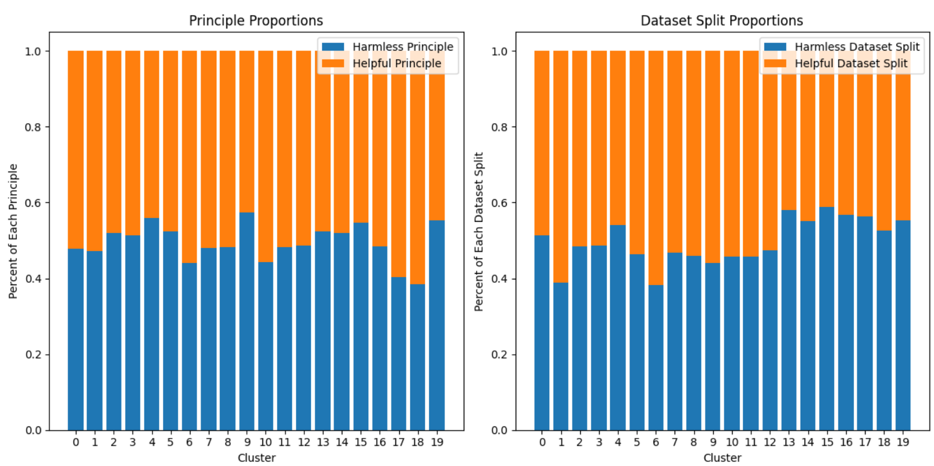

Figure 5.4: Clustering on "chosen minus rejected" embeddings from the synthetic dataset. Done with 20 clusters.

Chapter 6

Discussion

6.1 Clustering on Harmless hh-rlhf Samples Results in Harmless Topic Clusters

In this section we discuss the clustering and validation that we ran on the hh-rlhf dataset.

We find that K-means and K-medoids result in similar clusters and principles being generated. Both clustering methods also produce similar average "Avg. Val.": 65.1% for K-means and 62.6% for K-medoids. This is likely due to K-means and K-medoids being similar clustering algorithms, which is a point for potential improvement (see 7.1).

We notice that clustering the "preprompted choice" embedding generally works better then clustering the "chosen minus rejected" embeddings. The "preprompted choice" embeddings likely contain more relative information between samples, and so this results in better ability to cluster. The "chosen minus rejected" embeddings, on the other hand, likely lack useful structure, as small variations in the chosen or rejected responses could lead to significantly differences between embeddings. The goal was to capture the "direction of preference" by subtracting (à la the gender "direction" of `vector(’’Man’’) - vector(’’Woman’’)` in Word2Vec [21]).

Generated principles do tend to fit well to the clusters, generating principles that precisely describe the makeup of the cluster. The primary drawback of the clustering method used in this work was that the clusters did not split the dataset across principled/constitutional lines very well. We found that clustering frequently happens along the topics present in the conversation transcript and responses, but it generally does not capture the subtlety of the preferences represented in the samples.[1] In some instances, though, clustering did cluster samples with certain, specific patterns, such as short conversation transcripts and chosen responses being one that doesn't answer the question. These were encouraging signs that our method could possibly be improved to find more prevalent principles. In order to get more principles, we increased the number of clusters. This generally resulted in similar results: clustering happened along topics, albeit slightly more specific ones.

6.1.1 Sparks of Principled Clustering

A minority of clusters resulted in succinct principles which described the samples well and achieved high validation accuracies, around 83%. Despite this positive result, the rest of the clusters would have middling results with around 60% validation accuracy. We hypothesize that these high-scoring clusters resulted from the same clustering mechanism underlying the other clusters, namely clustering via topic. In these minority of situations, though, the topic of a cluster was such that a principle generated from samples in this cluster naturally applied to the rest of the samples from that cluster. Although this may seem encouraging, we find this to likely be an artifact of the process we're using and *not* optimal principle-based clustering occurring in small pockets of our work.

6.2 Generating Synthetic Data Results in Noisy Labeling

In this section, we describe the clustering and validation that we ran on the synthetic dataset.

We found that the synthetic dataset contained lots of noise, primarily due to relabeling

[1] In a way, we discovered another, more expensive way to do LDA [16].

harmless and helpful splits with harmless and helpful principles, we hypothesize. We find two types of this noise that exists as a result of our choice of relabeling. First, when applying the helpful principle to a harmless sample (or the harmless principle to a helpful sample), the principle did not apply to the situation presented in the sample, and so there was no "right" answer for the labeling model to choose given this mismatch of principle and sample. Second, the harmless and helpful principles resulted in the labeling model preferring the same response, and so it would be impossible for the correct labeling principle to be extracted given that either principle could have resulted in this labeling.

Examining Figures 5.1, 5.2, 5.3, and 5.4, we find that nearly every cluster is made up of about 50% of each labeling principle, with little variance. In the experiments with the "chosen minus rejected" embeddings, we find that the dataset splits are also about 50%, supporting the hypothesis that this form of embedding does not encode the proper signal for clustering. The "preprompted choice" embeddings do result in clustering more along the dataset split lines, supporting the hypothesis that clustering is done via topic more strongly than it is done via preference.

In sum, the synthetic dataset experiments revealed failures of important assumptions that we had made. First, principles are applied to specific contexts/topics, and so blindly randomizing the principles applied to a variety of topics results in a noisy dataset. This is supported by the fact that two different principles can produce the same labels for a pair of responses. This is an important problem to consider, as this is not merely a problem with synthetic datasets. Two human labelers may also have different personal principles, yet label samples with the same preferred response. How to extract principles in light of this is a problem left to future work. Second, the specific choice of relabeling harmless and helpful samples with harmless and helpful principles may not have been a good choice for producing a good signal for determining which samples represent which principle. The "conflict of interest" between the datasets and principles likely makes it quite difficult to produce clean clustering along the principles.

Chapter 7

Future Work

Despite not providing strong principle-generation capabilities, ICAI shows promise as a framework to interpret the implicit preferences that are encoded within datasets and models. In this chapter, we outline avenues of future work which could help mature ICAI to its full potential. We have confidence that our work thus far has informed strong ideas regarding future work on this framework.

We discuss three avenues for future work: clustering-based approaches (of which this work is a variant), prompt optimization, and sparse dictionary learning.

7.1 Clustering-Based Approaches

Clustering-based approaches utilize clustering of a textual embedding in order to generate clusters with separate semantic meanings. The primary drawbacks to the clustering that was used in this work was:

1. The type of clustering we used was "hard" clustering. This means that once a sample is in one cluster, it is "stuck" there, without potential future mobility. This could be detrimental to principle generation, as a cluster could contain "noisy" samples which cause the cluster to be labeled as having multiple principles, although it may only have

a small minority of noisy samples which are having an outsized effect on the generated principle.

2. The clustering we used was also top-down. This means that we took every sample from the entire preference dataset, embedded each of them, and then clustered the whole dataset from these embeddings. Naively, the hope was that clustering across the whole dataset would result in the formation of principled clusters. We did often find that the clusters fit some semantic groupings, but the grouping was almost surely based on the topics in the prompt (i.e., the conversation transcript), not the preferences that were represented in the chosen and rejected responses.

One way to solve the problems arising from hard clustering is to use "soft" clustering (also known as fuzzy clustering). This is a method that assigns samples to one or more clusters, with varying degrees of membership to each cluster. Given this soft clustering, we can then use the cluster memberships in order to weight the importance of samples when generating cluster principles. This could improve the problems that result from hard clustering as it is more robust to noisy samples "infiltrating" clusters that they do not belong to semantically.

Another small but potentially powerful change is using other clustering methods, such as HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise), spectral clustering, and Gaussian mixture models. These other clustering methods may prove to be better suited for clusters with non-spherical shapes which KMeans and KMedoids struggle with.

A potential method to continue using the current top-down clustering paradigm is to change the way that we embed the samples. In this work, we used the difference `embed(p+c)-embed(p+r)` and the preprompted `embed(preptompt+c+r)`, but this seemed to focus the clustering on the topics in the samples rather than the principles in them. One approach to change the embedding would be to train a reward model on the samples and use the final layer representation from this reward model as the embedding to cluster on. This could promote the principles within the embeddings and generate better top-down clusters.

Perhaps the top-down method is inherently flawed. To solve this problem, a "bottom-up" approach could be used. In a bottom-up approach, we generate one principle per sample (e.g., `examples` in the template from 4.3 Principle Generation is just one example). Once we have generated a principle for each sample, we can embed these principles and cluster using these embeddings. Clustering, perhaps hierarchically, could result in cleaner principles per cluster, as we are clustering on the principles of the samples instead of the samples themselves. Similar to the reward model method discussed earlier, this approach could provide a better "clustering signal," which could result in better clusters than those found in this work.

Given the results of this work, the clustering-based approaches discussed show promise as methods to potentially boost the power of the clustering groundwork that has been laid.

7.2 Prompt Optimization

Prompt optimization is a method to generate constitutions using language models directly. This could take many forms, as there is a rich literature on prompt optimization (e.g., [18], [22]).

The basic process would be as follows: have a language model generate principles given samples from a dataset, validate these principles on other samples from the dataset, and use in-context learning to prompt the model to generate updated principles. This process can be repeated as necessary, but lacks much structure in this form. Perhaps combination with some pre-processing (e.g., via clustering) could significantly boost performance.

7.3 Sparse Dictionary Learning

Sparse dictionary learning seeks to find a sparse linear representation of each data sample using a minimal set of atoms which can represent all samples. This paradigm has found some success within mechanistic interpretability (e.g., [14]), and we believe that there is potential for this to be applied within the ICAI context. The primary steps of sparse dictionary learning would be as follows:

1. Generate a "basis" of principles: these are analogous to the atoms of the sparse dictionary problem. These can be generated using samples from the dataset or leveraging the power of long-context language models to generate potential principles given a large subset of the dataset.

2. Learn how to reconstruct the preferences represented within the samples using a sparse set of the basis principles.

3. Collect the principles which perform best and are most disjoint from one another (and perhaps some other characteristic). These principles form the constitution for the dataset.

The validating and collecting of principles may be a noisy process, i.e., it may be hard to determine which principles are really doing well, and so this method is more uncertain than the others mentioned in this chapter.

Chapter 8

Conclusion

In this thesis, we presented Inverse Constitutional AI (ICAI), an interpretability framework for extracting the principles governing models and datasets. Motivated by the Constitutional AI (CAI) finetuning paradigm of instilling models with a constitution of principles, we approach the inverse problem of extracting a constitution of principles from data.

The key contributions of this work are the presentation of the ICAI framework and an approach to ICAI using a clustering-based approach. The approach involved embedding preference pairs, clustering on these embeddings in order to group principles, generating interpretable principles for each cluster, and validating these principles on held-out samples from the cluster. We conducted empirical evaluation on the hh-rlhf red-teaming dataset and a synthetic dataset relabeled using hand-written principles.

While the clustering techniques employed were able to group semantically coherent topics and generate human-interpretable principles for these clusters, achieving fully disentangled, principle-based clustering remained a challenge. The results highlighted the impact of embedding representations and the need for alternative principle-discovering techniques tailored to the problem of principle extraction.

Several promising directions for future work were identified, including soft clustering approaches to mitigate the limitations of hard clustering, bottom-up principle extraction methods, prompt optimization techniques for principle generation, and sparse dictionary learning frameworks. Exploring these avenues could potentially unlock the full potential of ICAI as a powerful interpretability tool.

This thesis establishes ICAI as a novel framework for interpreting the principles underlying datasets and models, and took an initial step towards realizing this vision through a clustering-based implementation on preference data. While the current work demonstrated both successes and limitations, it paved the way for future research in this area, with the ultimate goal of providing a deeper understanding of the principles governing language models and their training data.

References

[1] T. B. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners," *ArXiv*, vol. abs/2005.14165, 2020. URL: https://api.semanticscholar.org/CorpusID:218971783.

[2] J. Wei, X. Wang, D. Schuurmans, M. Bosma, E. H.-h. Chi, F. Xia, Q. Le, and D. Zhou, "Chain of thought prompting elicits reasoning in large language models," *ArXiv*, vol. abs/2201.11903, 2022. URL: https://api.semanticscholar.org/CorpusID:246411621.

[3] G. I. Winata, A. Madotto, Z. Lin, R. Liu, J. Yosinski, and P. Fung, "Language models are few-shot multilingual learners," *ArXiv*, vol. abs/2109.07684, 2021. URL: https://api.semanticscholar.org/CorpusID:237532173.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2019. arXiv: 1810.04805 [cs.CL].

[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. arXiv: 1909.08593 [cs.CL].

[6] N. Lambert and R. Calandra, *The alignment ceiling: Objective mismatch in reinforcement learning from human feedback*, 2024. arXiv: 2311.00168 [cs.LG].

[7] Y. Bai, S. Kadavath, S. Kundu, *et al.*, *Constitutional ai: Harmlessness from ai feedback*, 2022. arXiv: 2212.08073 [cs.CL].

[8] Y. Bai, A. Jones, K. Ndousse, *et al.*, *Training a helpful and harmless assistant with reinforcement learning from human feedback*, 2022. arXiv: 2204.05862 [cs.CL].

[9] *The Shapley Value: Essays in Honor of Lloyd S. Shapley*. Cambridge University Press, 1988.

[10] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 336–359, Oct. 2019, ISSN: 1573-1405. DOI: 10.1007/s11263-019-01228-7. URL: http://dx.doi.org/10.1007/s11263-019-01228-7.

[11] M. T. Ribeiro, S. Singh, and C. Guestrin, *"why should i trust you?": Explaining the predictions of any classifier*, 2016. arXiv: 1602.04938 [cs.LG].

[12] C. Olah, A. Mordvintsev, and L. Schubert, "Feature visualization," *Distill*, 2017, https://distill.pub/2017/feature-visualization. DOI: 10.23915/distill.00007.

[13] K. Meng, D. Bau, A. Andonian, and Y. Belinkov, "Locating and editing factual associations in GPT," *Advances in Neural Information Processing Systems*, vol. 36, 2022, arXiv:2202.05262.

[14] T. Bricken, A. Templeton, J. Batson, *et al.*, "Towards monosemanticity: Decomposing language models with dictionary learning," *Transformer Circuits Thread*, 2023, https://transformer-circuits.pub/2023/monosemantic-features/index.html.

[15] H. R. Kirk, A. Whitefield, P. Röttger, *et al.*, *The prism alignment project: What participatory, representative and individualised human feedback reveals about the subjective and multicultural alignment of large language models*, 2024. arXiv: 2404.16019 [cs.CL].

[16] D. Blei, A. Ng, and M. Jordan, "Latent dirichlet allocation," in *Advances in Neural Information Processing Systems*, T. Dietterich, S. Becker, and Z. Ghahramani, Eds., vol. 14, MIT Press, 2001. URL: https://proceedings.neurips.cc/paper_files/paper/2001/file/296472c9542ad4d4788d543508116cbc-Paper.pdf.

[17] A. Zou, Z. Wang, N. Carlini, M. Nasr, J. Z. Kolter, and M. Fredrikson, *Universal and transferable adversarial attacks on aligned language models*, 2023. arXiv: 2307.15043 [cs.CL].

[18] C. Fernando, D. S. Banarse, H. Michalewski, S. Osindero, and T. Rocktäschel, *Promptbreeder: Self-referential self-improvement via prompt evolution*, 2024. URL: https://openreview.net/forum?id=HKkiX32Zw1.

[19] H. Cunningham, A. Ewart, L. Riggs, R. Huben, and L. Sharkey, "Sparse autoencoders find highly interpretable features in language models," *ArXiv*, vol. abs/2309.08600, 2023. URL: https://api.semanticscholar.org/CorpusID:261934663.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[21] T. Mikolov, K. Chen, G. Corrado, and J. Dean, *Efficient estimation of word representations in vector space*, 2013. arXiv: 1301.3781 [cs.CL].

[22] Y. Zhou, A. I. Muresanu, Z. Han, K. Paster, S. Pitis, H. Chan, and J. Ba, *Large language models are human-level prompt engineers*, 2023. arXiv: 2211.01910 [cs.LG].