

Examining the Structure of Text-Based Interventions in Large Language Models

Sweta Karlekar, Gabriel Chuang, Claudia Shi

Columbia Computer Science

Overview

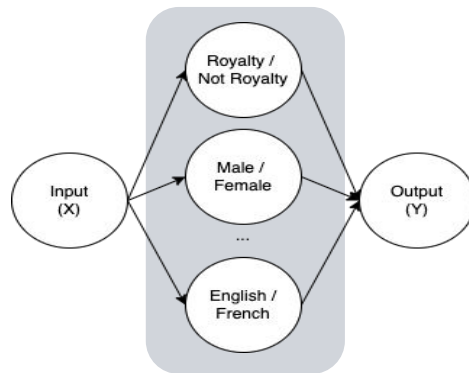
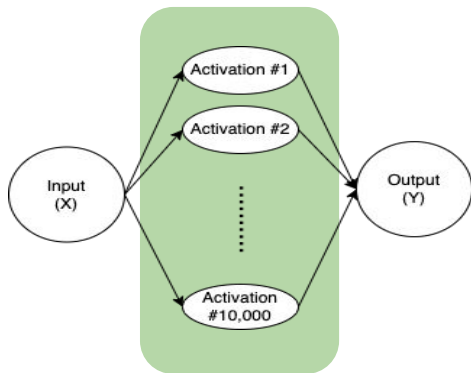
- Background & Related Work
 - ◆ Causal inference in LLMs
 - ◆ Mechanistic interpretability
 - ◆ Activation steering
 - ◆ Linear representation hypothesis
- Proposal
- Preliminary Results
- Future Work

Background & Related Work

A Causal Representation Learning Problem

Large Language Models

Causal DAGs



- Nodes in a neural network represent causal relationships amongst inputs and outputs
- Therefore, we can perform any arbitrary intervention in LLMs by turning certain nodes off and thus cutting off all downstream causal effects

A Causal Representation Learning Problem



- Nodes in a neural network represent causal relationships amongst inputs and outputs
- Technically there's no confounding! We know all the causes (AKA parents) of a given node in a network
- Therefore, we can perform any arbitrary intervention in LLMs by turning certain nodes off and thus cutting off all downstream causal effects

Challenge: Human-interpretable causal inference at this level is very difficult! It's not possible to understand the mechanisms of billions of neuron individually. The causal problem here is to understand what's the **right level of abstraction**, how to **define the “intervention”**, and understand **where to intervene**.

Mechanistic Interpretability

- Area of research that seeks to understand the neural mechanisms that enable specific behaviors in LLMs by leveraging **causality-based methods**.^[3]
- Importantly, mechanistic interpretability focuses on reverse-engineering model components into **human-understandable** algorithms^[4]

Intervene on an LLM by adjusting the activation values or inputs and examine the output

[3] Palit, V., Pandey, R., Arora, A., & Liang, P. P. (2023). Towards vision-language mechanistic interpretability: A causal tracing tool for blip. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 2856-2861).

[4] Conmy, A., Mavor-Parker, A., Lynch, A., Heimersheim, S., & Garriga-Alonso, A. (2023). Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36, 16318-16352.

Mechanistic Interpretability

- Area of research that seeks to understand the neural mechanisms that enable specific behaviors in LLMs by leveraging **causality-based methods**.^[3]
- Importantly, mechanistic interpretability focuses on reverse-engineering model components into **human-understandable** algorithms^[4]

Intervene on an LLM by adjusting the activation values or inputs and examine the output

- What are examples of LLM interventions?
 - ◆ Interchange interventions / activation patching
 - ◆ Jittering perturbations
 - ◆ Activation steering

Mechanistic Interpretability

- Area of research that seeks to understand the neural mechanisms that enable specific behaviors in LLMs by leveraging **causality-based methods**.^[3]
- Importantly, mechanistic interpretability focuses on reverse-engineering model components into **human-understandable** algorithms^[4]

Intervene on an LLM by adjusting the activation values or inputs and examine the output

- What are examples of LLM interventions?
 - ◆ **Interchange interventions / activation patching** — replacing specific dimensions of the activation vectors with new values, usually learned from relevant examples
 - ◆ Jittering perturbations
 - ◆ Activation steering

Mechanistic Interpretability

- Area of research that seeks to understand the neural mechanisms that enable specific behaviors in LLMs by leveraging **causality-based methods**.^[3]
- Importantly, mechanistic interpretability focuses on reverse-engineering model components into **human-understandable** algorithms^[4]

Intervene on an LLM by adjusting the activation values or inputs and examine the output

- What are examples of LLM interventions?
 - ◆ Interchange interventions / activation patching
 - ◆ **Jittering perturbations** — add noise/variations to inputs or activations to measure robustness or sensitivity, can be used in conjunction with other interventions
 - ◆ Activation steering

[3] Palit, V., Pandey, R., Arora, A., & Liang, P. P. (2023). Towards vision-language mechanistic interpretability: A causal tracing tool for blip. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 2856-2861).

[4] Conmy, A., Mavor-Parker, A., Lynch, A., Heimersheim, S., & Garriga-Alonso, A. (2023). Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36, 16318-16352.

Mechanistic Interpretability

- Area of research that seeks to understand the neural mechanisms that enable specific behaviors in LLMs by leveraging **causality-based methods**.^[3]
- Importantly, mechanistic interpretability focuses on reverse-engineering model components into **human-understandable** algorithms^[4]

Intervene on an LLM by adjusting the activation values or inputs and examine the output

- What are examples of LLM interventions?
 - ◆ Interchange interventions / activation patching
 - ◆ Jittering perturbations
 - ◆ **Activation steering** — adjusting the existing values of the activation vector with new values to steer model generation towards a specific target

[3] Palit, V., Pandey, R., Arora, A., & Liang, P. P. (2023). Towards vision-language mechanistic interpretability: A causal tracing tool for blip. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 2856-2861).

[4] Conmy, A., Mavor-Parker, A., Lynch, A., Heimersheim, S., & Garriga-Alonso, A. (2023). Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36, 16318-16352.

Residual Stream

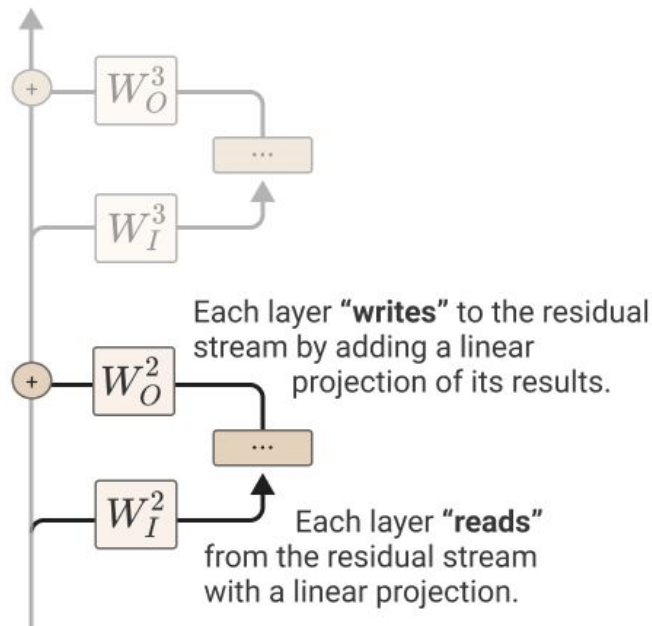
- Transformer-based models are comprised of blocks of attention layers followed by multi-layer perceptron (MLP) layers^[1]
- Attention heads can be understood as independent operations, each outputting results (aka activation values) that are added into the **residual stream**^[2]
- Residual stream is **high-dimensional vector space** made up of the sum of the output of all previous layers and the original embedding, can be thought of as a “communication channel”^[2]
- LLMs operate by extracting information from a common “residual stream” shared by all attention and MLP blocks^[2]
 - ◆ In other words, every layer reads from the residual stream through **linear projection** and performs another **linear projection** to the output before **adding** to “write” its output back into the residual stream ^[2]

^[1]Turner, A., Thiergart, L., Udell, D., Leech, G., Mini, U., & MacDiarmid, M. (2023). Activation addition: Steering language models without optimization. *arXiv preprint arXiv:2308.10248*.

^[2]Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., ... & Olah, C. (2021). A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1, 1.

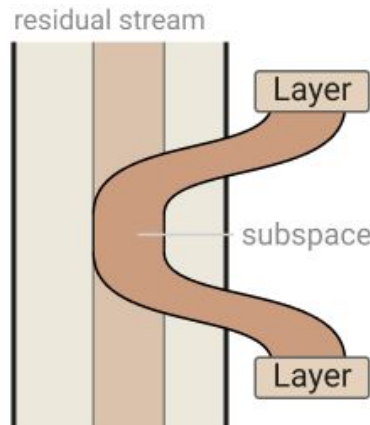
Residual Stream

The residual stream is modified by a sequence of MLP and attention layers “reading from” and “writing to” it with linear operations.



- Concept originates from the idea of skip connections or residual connections in deep neural networks
- We can think of attention heads as **independently adding** to the residual stream^[2]

The residual stream is high dimensional, and can be divided into different subspaces.

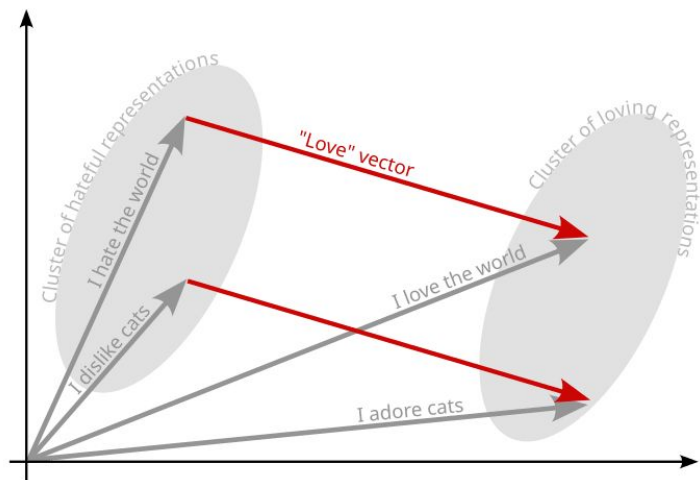


Layers can interact by writing to and reading from the same or overlapping subspaces. If they write to and read from disjoint subspaces, they won't interact. Typically the spaces only partially overlap.

[1]Turner, A., Thiergart, L., Udell, D., Leech, G., Mini, U., & MacDiarmid, M. (2023). Activation addition: Steering language models without optimization. *arXiv preprint arXiv:2308.10248*.
[2]Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., ... & Olah, C. (2021). A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1, 1.

Activation Steering

- Method to control or guide the behavior of the model by adjusting the internal states or activations of the model
- Goal: add some vector to the internal model activations at a given layer to influence the output in a specific way



Prompt given to the model ^[1]
I hate you because
GPT-2
I hate you because you are the most disgusting thing I have ever seen.
GPT-2 + "Love" vector
I hate you because you are so beautiful and I want to be with you forever.

The Linear Representation Hypothesis

Hypothesis:

High-level concepts are represented as directions in representation space.

The Linear Representation Hypothesis

Hypothesis:

High-level concepts are represented as directions in representation space.

A concept is “any factor of variation that can be changed in isolation”.

- English $\rightarrow$ French
- Male $\rightarrow$ Female

Concepts induce sets of “counterfactual pairs”:

e.g., { (“king”, “queen”), (“actor”, “actress”), ... } for Male $\rightarrow$ Female

Counterfactual pairs can even be sentences:

e.g. (“I love dogs”, “I hate dogs”) for Love $\rightarrow$ Hate

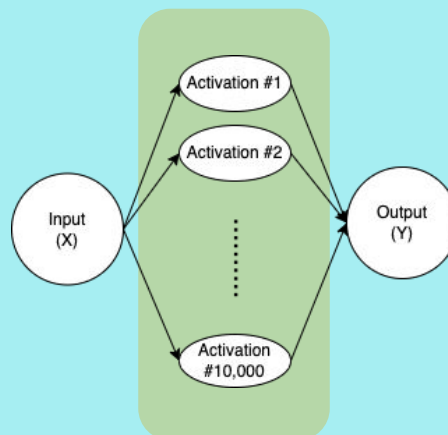
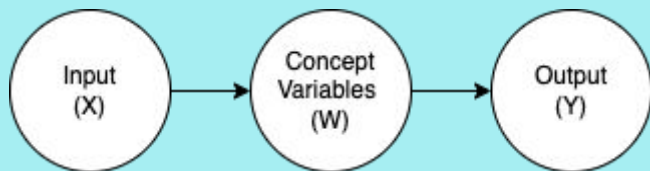
The Linear Representation Hypothesis

Hypothesis:

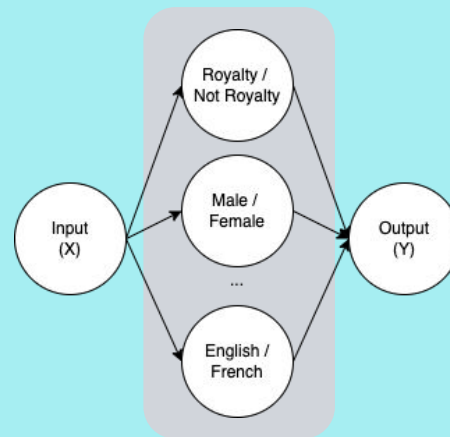
High-level concepts are represented as directions in representation space.

We consider *Concept Variables*:

- caused by the input
- causally affect the output



observable
and intervenable

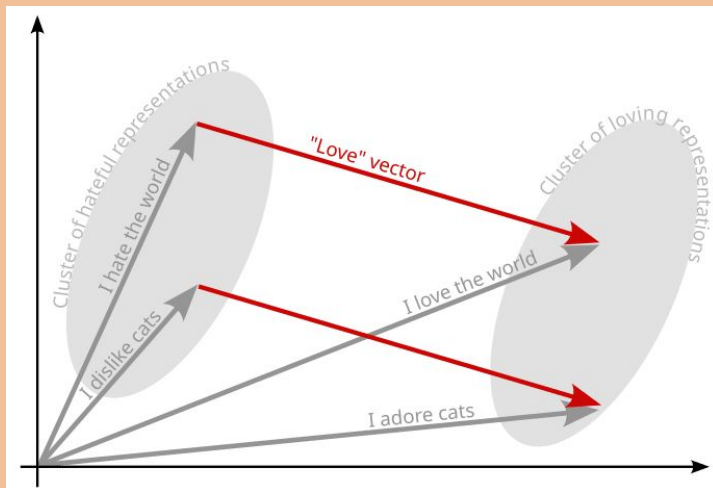


latent
abstraction

The Linear Representation Hypothesis

Hypothesis:

High-level concepts are represented as directions in representation space.



The “direction” for a concept is (hypothesized to be) the same across a broad class of inputs.

The notion of a “direction” lends itself to nice geometric properties.

The Linear Representation Hypothesis

Hypothesis:

High-level concepts are represented as directions in representation space.

Part of the hypothesis is that “orthogonal” concepts have orthogonal directions.

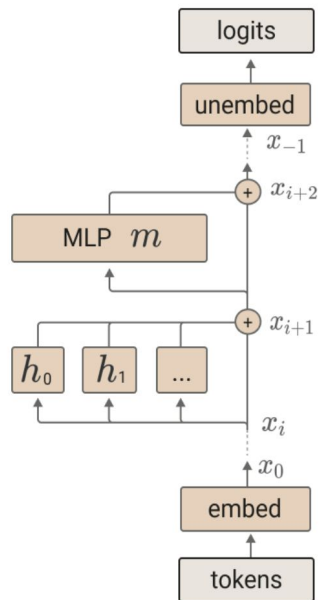
Concepts are **causally separable** if they can be varied “freely and in isolation”

- Male/Female and English/French  (king, queen, roi, reine)
- English/Russian and English/French 
- Present/Past and Singular/Plural 

The Linear Representation Hypothesis

Hypothesis:

High-level concepts are represented as directions in representation space.



Q: What space should we consider?

unembedding space?

the residual stream?

embedding space?

it turns out you can
unify them; we'll
think about the
residual stream.

The Linear Representation Hypothesis

Hypothesis:

High-level concepts are represented as directions in representation space.

Bad news: directions (might) not be enough.

Concept representations are only identified up to an affine transformation*, and affine transformations don't preserve directions, inner products, orthogonality, etc.

Hypothesis (ver. 2):

High-level concepts are represented linearly in representation space.

* this is because the representations only affect the objective used to train the model thru softmax, and so there's a class of affine-related representations that all have equivalent post-softmax values.

The Linear Representation Hypothesis

Hypothesis:

High-level concepts are represented linearly in representation space.

Q: What does it mean for a concept to be “represented linearly”?

A: many different notions:

Steering vectors: $\exists z$ such that
 $Y(A(x)) = \text{“king”} \Rightarrow Y(A(x) + z) = \text{“queen”}$

Subspace:
for all ctf-pairs (x, x^{cf}) , $A(x) - A(x^{\text{cf}})$ lie in
a low-dimensional subspace

Measurement: given activations A where
 $Y(A) \in \{\text{“king”}, \text{“queen”}\}$, a linear
classifier can tell which one it is given A .

All of these represent concepts as
an affine transformation + a direction!

The Linear Representation Hypothesis

Hypothesis ver. 1:

High-level concepts are represented as directions in representation space.

Hypothesis ver. 2:

High-level concepts are represented linearly in representation space.

A **concept** is a factor of variation that can be changed in isolation. (e.g., Male → Female)

- Concepts induce **counterfactual pairs**. e.g., {(king, queen), (actor, actress)}.

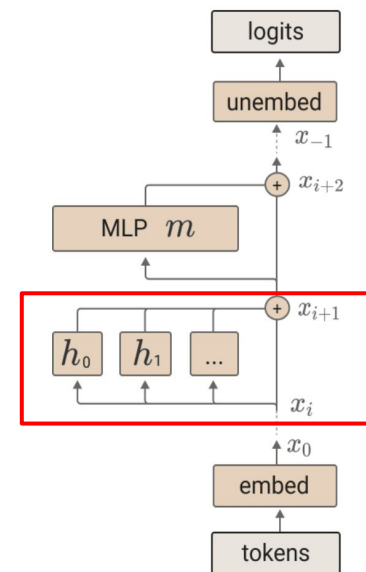
We consider the **residual stream** as our representation space.

We may hope that concepts are represented as **directions** (a la steering vectors).

Otherwise, several notions of a “linear representation” represent concepts as **affine transforms**.

Why might we believe in linear representations?

- Transformers already “split” the residual stream into a bunch of subspaces, which different attention heads use to communicate with each other
- Attention heads:
 - Are independent and additive
 - Copy information between tokens’ residual streams
 - Typically reading and writing to different subspaces
 - The “information copy” part is linear
 - (while the choice of tokens is nonlinear)



Activation Steering Related Works

- Subramani et al. (2022) & Hernandez et al. (2023) employ steering vectors that are added to the forward pass of GPT-2 with the goal of modifying generation; the former found steering vectors like “love”, “hate”, etc. while the latter used steering vectors for fact-editing
- Merullo et al. (2023) observed the linearity of transformer representations; were able to find a steering vector for country capitals; the vector added to the residuals to convert Poland to Warsaw could be used to transform China into Beijing
- Elhage et al. (2022) showed that Othello-GPT can be intervened on with linear activation vectors that represented to, in essence, represent “a black piece is here and not a white piece”.

Activation Steering Related Works

→ Turner et al. (2023)

- ◆ Found multiple linear steering vectors including “love” (attention layer 6), “intent to praise” (attention layer 6), “conspiracy” (attention layer 23), “want to die” (attention layer 10), “anger” (attention layer 20), and some odd ones like “talking about weddings” (attention layer 20), “Dragons live in Berkeley” (attention layer 15), and “the Eiffel tower is in Rome” (attention layer 24)
- ◆ Found that adding embedding vectors isn’t as effective as adding steering vectors

→ Park et al. (2023) found language concepts that allow translation (English⇒French, French⇒German, French⇒Spanish, and German⇒Spanish)

Related Works Summary

In summary, prior work has:

- Shown that it is reasonable to understand concepts as linearly representable in LLMs within the residual space
- Hand-picked concepts and manually found linear steering vectors for them
- Activation steering can, for examples shown, be an interpretable and useful intervention in LLMs

Related Works Summary

In summary, prior work has:

- Shown that it is reasonable to understand concepts as linearly representable in LLMs within the residual space
- Hand-picked concepts and manually found linear steering vectors for them
- Activation steering can, for examples shown, be an interpretable and useful intervention in LLMs

However, there has been no systematic or comprehensive **empirical** review about whether or not **most** concepts can actually be presented linearly. Further, there has been no exploration about whether this linear representation works for **all forms of concepts** (i.e. Do linear steering vectors only exist for commonly used concepts? What about concepts of different parts of speech?)

Proposal & Preliminary Results

Overview

Treatment: $C \in \{0, 1\}$ (whether the concept is present vs. its counterfactual)

Potential Outcomes: $A(C = 0)$
 $A(C = 1)$ (activation values at some layer of interest)

Question: What is the structure of the relationship between the potential outcomes?

Overview

Treatment: $C \in \{0, 1\}$ (whether the concept is present vs. its counterfactual)

Potential Outcomes: $A(C = 0)$
 $A(C = 1)$ (activation values at some layer of interest)

Question: What is the structure of the relationship between the potential outcomes?

In normal real-world scenarios, we don't have access to the counterfactuals. In order to find the treatment effect we have to make certain assumptions about the structure of the causal relationship between the potential outcomes (e.g. linearity).

But in this case, we do have access to the counterfactuals! So we can investigate the best structure to define the relationship between counterfactuals.

Overview

Treatment: $C \in \{0, 1\}$ (whether the concept is present vs. its counterfactual)

Potential Outcomes: $A(C = 0)$
 $A(C = 1)$ (activation values at some layer of interest)

Question: What is the structure of the relationship between the potential outcomes?

$$A(1) = A(0) + \delta$$

constant
(hypothesis v.1)

$$A(1) = w^T A(0) + \delta$$

linear
(hypothesis v.2)

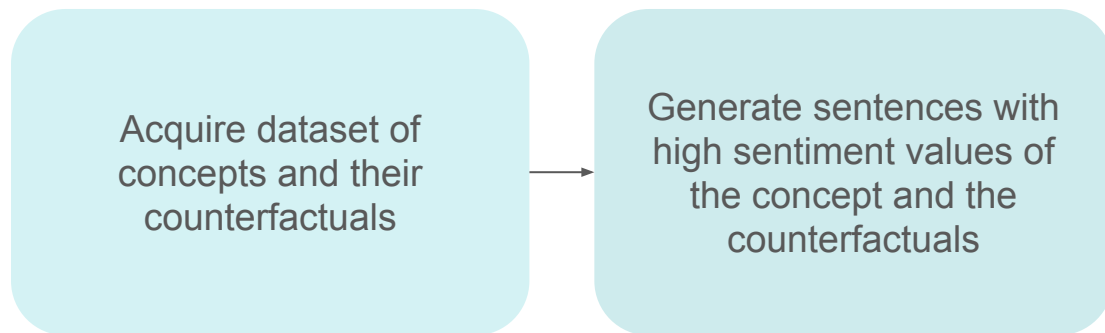
$$A(1) = f(A(0)) \text{ where } f(\cdot) \text{ is some nonlinear function}$$

nonlinear

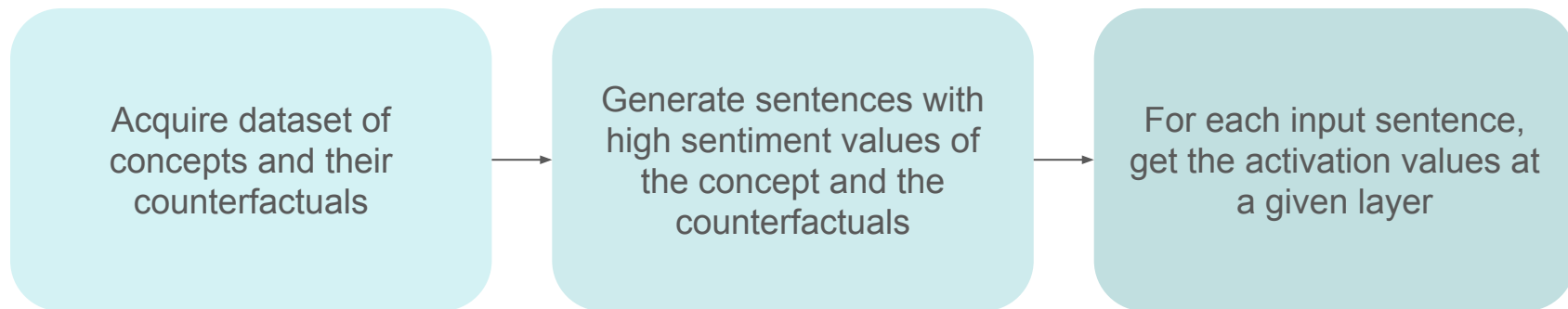
Overview

Acquire dataset of
concepts and their
counterfactuals

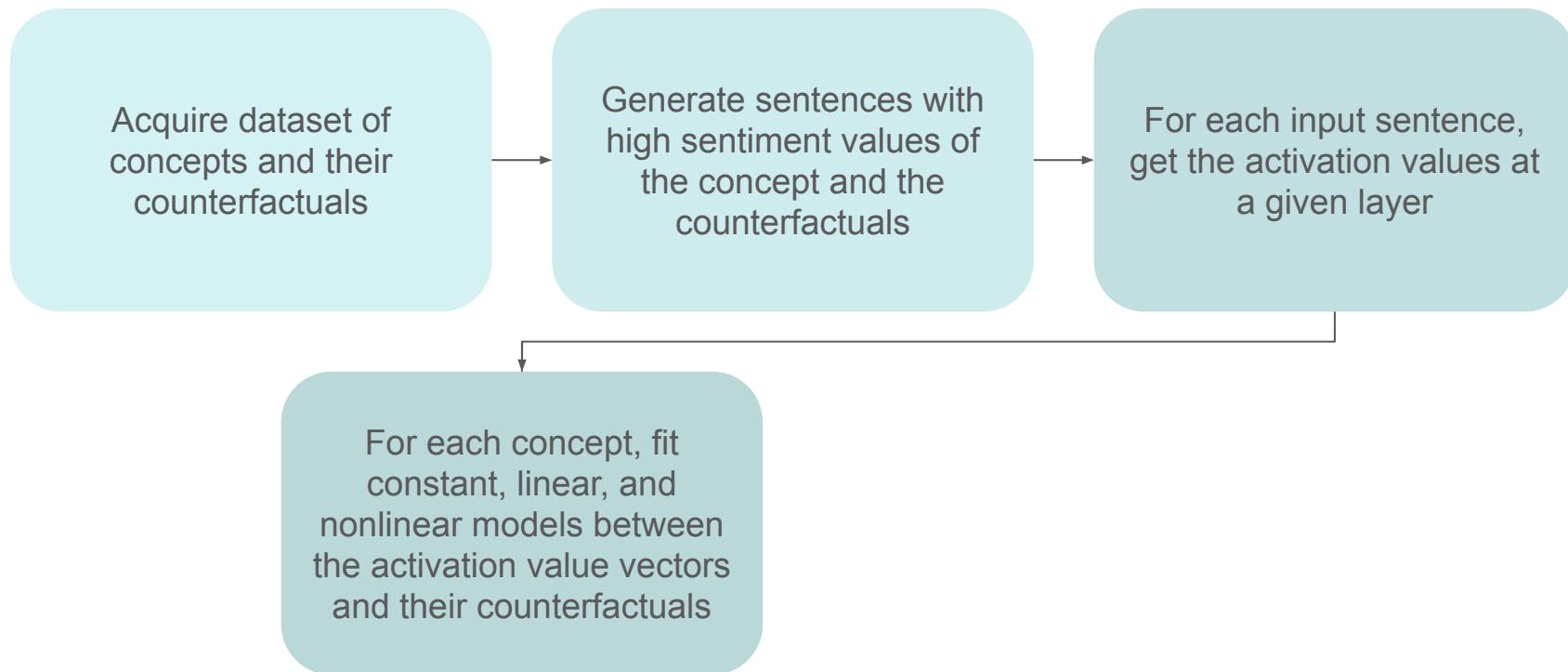
Overview



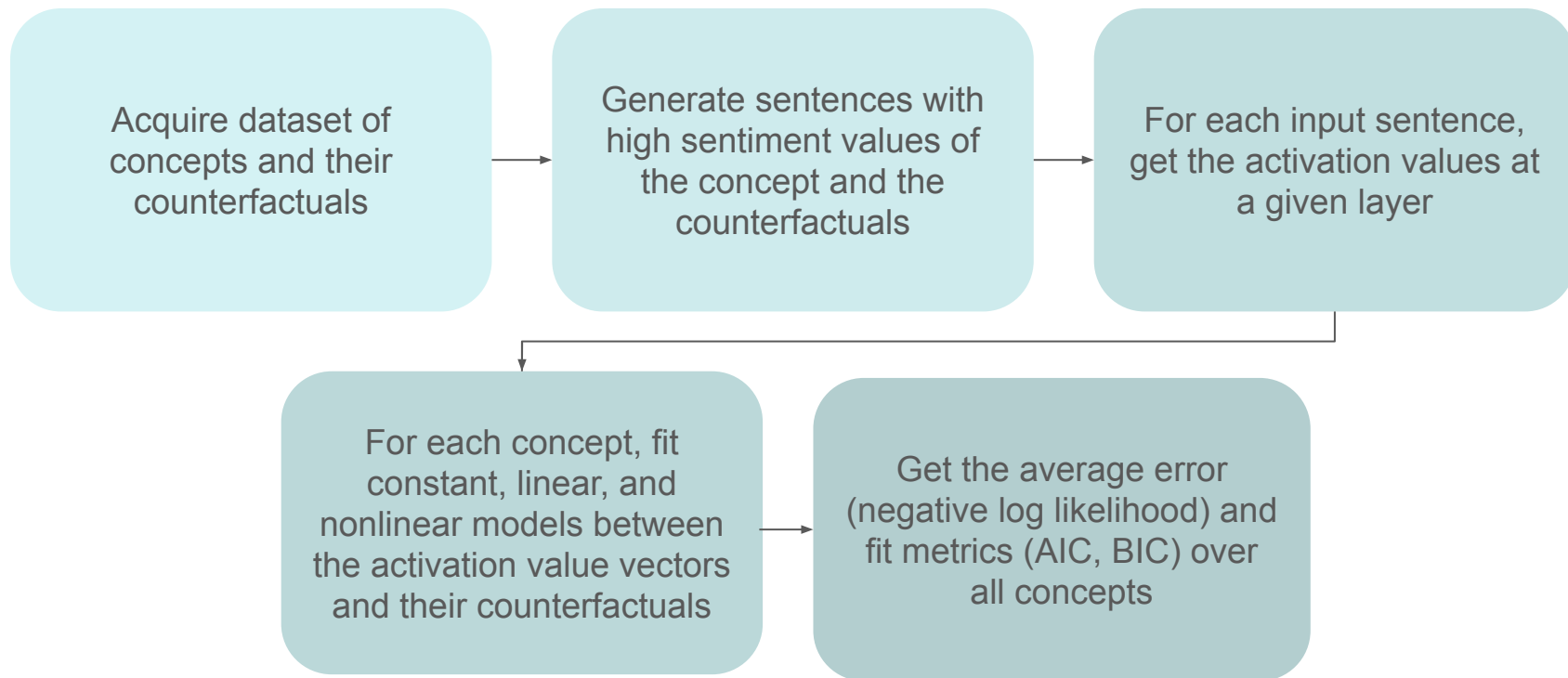
Overview



Overview



Overview



Dataset

- Start with WordNet Antonyms dataset^[5]
 - ◆ ~11k pairs of words in English (nouns, verbs, adverbs, adjectives) and their counterfactuals
 - E.g. “decode” ⇔ “encode” & “blab” ⇔ “keep quiet”
- Use GPT-4 to generate sentences for each concept, then use GPT-4 again to translate the sentences to their counterfactuals

Term: **wide**

1. The view from the top of the mountain was breathtaking, embracing the wide expanse of the landscape below.
2. In his wide-eyed innocence, he looked at the world with wonder and curiosity.
3. Her wide knowledge on the subject impressed everyone at the conference.

Counterfactual: **narrow**

1. The view from the cave entrance was limited, showing only a narrow segment of the landscape below.
2. In his narrow-minded suspicion, he looked at the world with doubt and caution.
3. Her narrow expertise on the subject hardly impressed anyone at the conference.

Getting Activation Values

→ Use the `baukit` python package

```
from baukit import Trace

# get the constant steering vector
layer_id = 5
module = model.layers[layer_id]
with Trace(module) as ret:
    _ = model("...wide-eyed innocence...")
    act_of_input = ret.output
```

Regression

$$A(1) = A(0) + \delta$$

constant
(hypothesis v.1)

$$A(1) = w^T A(0) + \delta$$

linear
(hypothesis v.2)

$$A(1) = f(A(0)) \text{ where } f(\cdot) \text{ is some nonlinear function}$$

nonlinear

For each of these, we:

1. Fit on the data
2. Compute the residuals and log likelihood on held-out data
3. Compute AIC/BIC on held-out data

Akaike and Bayesian Information Criteria

The AIC and BIC are criteria for model selection that take into account overfitting.

- Rewards high log likelihood
- Penalizes having a large number of parameters

L = log likelihood , k = # parameters, n = sample size:

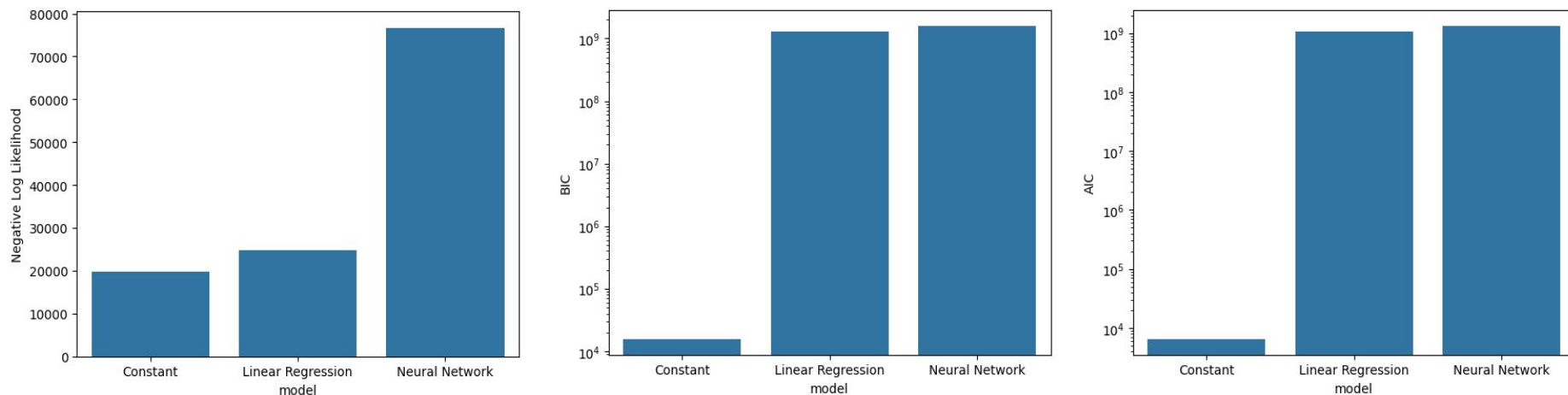
$$AIC = 2k - 2L$$

$$BIC = k \ln(n) - 2L$$

In both cases, lower values indicate better quality.

Preliminary results

Based on a small number of initial concepts and counterfactual pairs.



Having a constant steering vector seems the most parsimonious.

Planned Work

- Continue experimentation with a larger number of examples
- Stratify results by various characteristics of the concepts
 - ◆ Frequency of use (i.e. are only the most common concepts able to be presented linearly?)
 - ◆ POS of concept (i.e. are nouns more likely to have a linear representation than adverbs?)
 - ◆ ...and more!

Planned Work

→ Try other function classes (scale-free constant, sparse linear, polynomial)

$$A(1) = A(0) + \delta$$

constant
(hypothesis v.1)

$$A(1) = w^T A(0) + \delta$$

linear
(hypothesis v.2)

$$A(1) = f(A(0)) \text{ where } f(\cdot) \text{ is some nonlinear function}$$

nonlinear

$$A(1) = A(0) + \alpha_i \delta$$

scale-free
constant

$$A(1) = w^T A(0) + \delta$$

w sparse
sparse linear

$$A(1) = \text{poly}(A(0), w)$$

polynomial

Questions?

Appendix

Recap of LLM Architecture

In the original Vaswani *et al.* paper on transformers [1], the output of an attention layer is described by stacking the result vectors $r^{h_1}, r^{h_2}, \dots$, and then multiplying by an output matrix W_O^H . Let's split W_O^H into equal size blocks for each head $[W_O^{h_1}, W_O^{h_2} \dots]$. Then we observe that:

$$W_O^H \begin{bmatrix} r^{h_1} \\ r^{h_2} \\ \dots \end{bmatrix} = [W_O^{h_1}, W_O^{h_2}, \dots] \cdot \begin{bmatrix} r^{h_1} \\ r^{h_2} \\ \dots \end{bmatrix} = \sum_i W_O^{h_i} r^{h_i}$$

Revealing it to be equivalent to running heads independently, multiplying each by its own output matrix, and adding them into the residual stream. The concatenate definition is often preferred because it produces a larger and more compute efficient matrix multiply. But for understanding transformers theoretically, we prefer to think of them as independently additive.

Causal Inner Product

- An inner product with the property that concepts that can vary freely of each other are represented as orthogonal vectors

