Using LLMs to Understand LLMs (and other things)

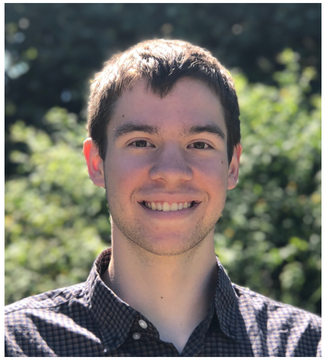
Jacob Steinhardt

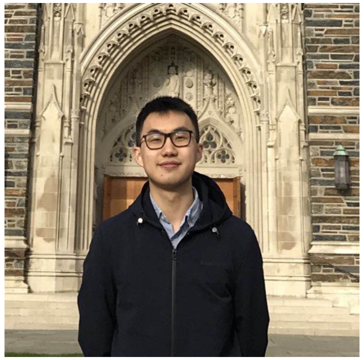
BAIR LLM Workshop
October 20, 2023

Motivation

Rapid proliferation of ML models; ever more capable and complex

How can understanding keep up, especially given emergent behavior?


Erik Jones


Ruiqi Zhong


Yossi Gandelsman

Key idea: use LLMs to understand LLMs

- As models get better, our understanding does as well

Understanding as Statistical Learning

Many forms of understanding reduce to statistics:

- Given data about a model's behavior, identify patterns
- Explain the important sources of variation in the training set
- Actively generate inputs that elicit problematic behavior

Understanding as Statistical Learning

Many forms of understanding reduce to statistics:

- Given data about a model's behavior, identify patterns
- Explain the important sources of variation in the training set
- Actively generate inputs that elicit problematic behavior

If we can get LLMs to "do statistics", we can tackle these problems!

The Statistics Pipeline

Look at some initial data (p_{train})

Form hypothesis h

Formalize h quantitatively

Test h on new data (p_{test})

- Held-out set, OOD data, or actively collected

The Statistics Pipeline

Look at some initial data (p_{train})

Form hypothesis h

Formalize h quantitatively

Test h on new data (p_{test})

- Held-out set, OOD data, or actively collected

Will automate each step with LLMs

The Statistics Pipeline

Look at some initial data (p_{train})

Form hypothesis h

Formalize h quantitatively

Test h on new data (p_{test})

- Held-out set, OOD data, or actively collected

Will automate each step with LLMs

Key difference: h will be a natural language string!

Case Study: Finding Failures in CLIP

CLIP: encoder that embeds both images and text

Backbone of many other models

Amazing results, but simple failures remain

Case Study: Finding Failures in CLIP

CLIP: encoder that embeds both images and text

Backbone of many other models

MidJourney 5.1 | DALL-E (New Bing) | Stable Diffusion XL | Stable Diffusion 2.1

"an empty glass" *"a family of five members"* *"a man descending a mountain"* *"there is no star in the night sky"*

"a runner is about to sprint" *"the soccer player throws the ball"* *"a woman proposing to a man"* *"a box with only a few chocolates"*

Amazing results, but simple failures remain

All failures above found automatically by LLMs!

Tong, Jones, Steinhardt (2023), "Mass-Producing Failures of Multimodal Systems"

Finding Failures Automatically

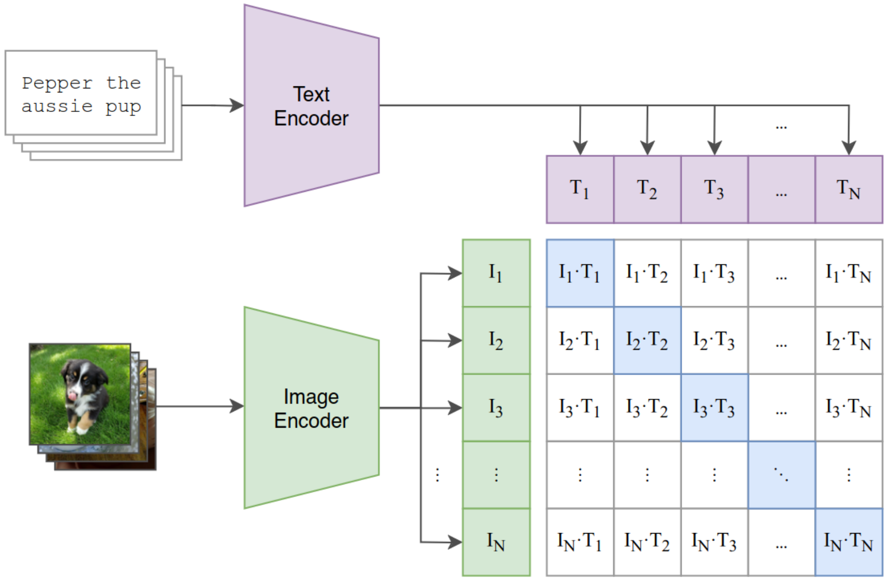
Key ideas:

- Find "hash collisions" in the CLIP encoder
- Categorize into coherent patterns
- Test patterns by generating new examples
- Check generalization to new domains, downstream tasks

Related work: Perez et al. (2022), Eyuboglu et al. (2022), Sheng et al. (2019)
Bolukbasi et al. (2016), Wallace et al. (2019), Ettinger (2020)

Initial Data: Hash Collisions

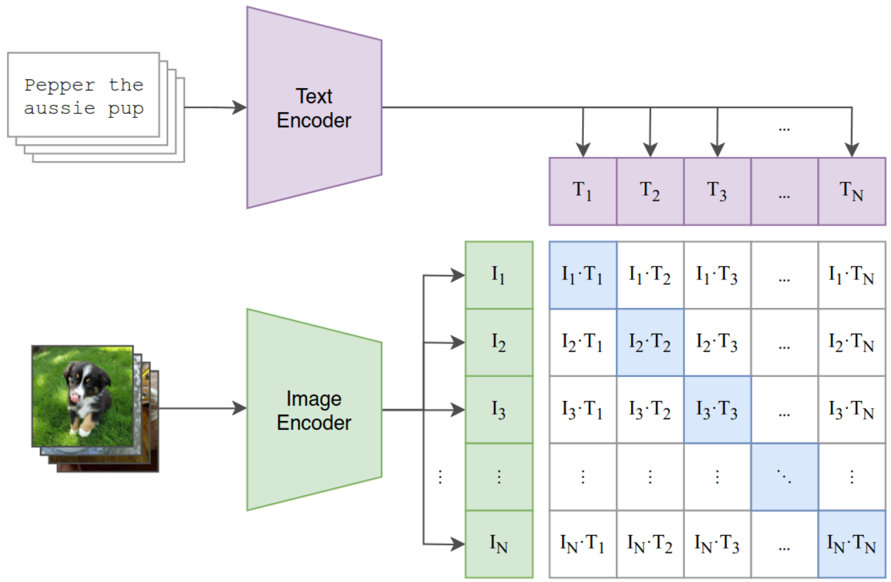
Background: CLIP embeds either image I, or text t



If t is a description of I, they should have similar embeddings

Initial Data: Hash Collisions

Background: CLIP embeds either image I, or text t



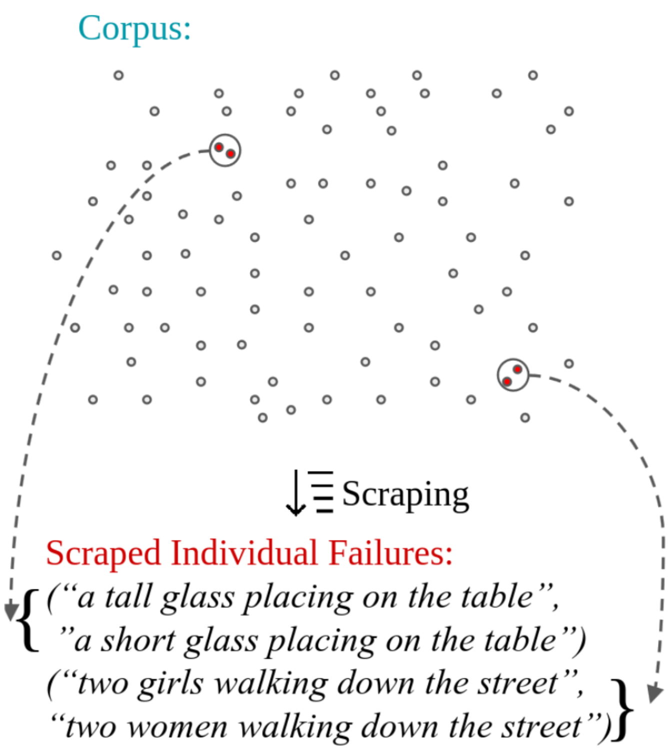
If t is a description of I, they should have similar embeddings

Therefore, if t, t' describe different images but have same embedding, at least one is "wrong"

Initial Data: Hash Collisions

Collect initial corpus of text inputs $t_1, \ldots, t_n$

Embed them under both CLIP and DistillRoBERTA (text-only model)



Find all pairs (t_i, t_j) with high CLIP and low RoBERTA similarity

Generating Hypotheses: Prompting GPT-4

Individual failures are text inputs, can feed them to GPT-4:

Prompt:

I will provide a series of data for you to remember. Subsequently, I will ask you some questions to test your performance! Here are some pairs of prompts for you to memorize.
[
the cat chases the dog, the dog chases the cat
a sky with one balloon, a sky with two balloons
...(k Failure Instances)
]
I'm trying to find failures with an embedding model. The above are some pairs of sentences that it encodes very similarly, even though they're conveying different concepts. Using these specific examples, are there any general types of failures you notice the embedding is making, or any common features that the embedding fails to encode? Try to give failures that are specific enough that someone could reliably produce examples that the embedding would encode similarly, even though it shouldn't. Please try to give as many general failures as possible. Please focus on differences that are important visually, as these embeddings are later used to generate images, or videos. In your failure modes, please explain clearly why the failure would lead to problems for future tasks related to visual generation.Please summarize as many as you can and stick to the examples.

Generating Hypotheses: Prompting GPT-4

1. Negation: Embedding models may not correctly capture the negative context in a sentence, leading to similarities between sentences with and without negation. This can result in incorrect visual representations, as the presence or absence of an action is significant in image or video generation.

2. Temporal differences: Embedding models might not differentiate between events happening in the past, present, or future. This failure can impact visual generation tasks by incorrectly representing the timing of events in generated images or videos.

3. Quantifiers: Embedding models may fail to distinguish between sentences that use quantifiers like "few," "some," or "many." This can lead to inaccuracies in the number of objects depicted in generated images or videos.

4. Semantic Role Ambiguity (Bag-Of-Words): The models might struggle to differentiate when the semantic roles are flipped. This failure can result in visual generation tasks depicting incorrect actions or object interactions.

5. Absence Vs Presence: Embedding models may not be able to distinguish between the presence or absence of certain objects. This can lead to visual generation tasks inaccurately including or excluding objects in the scene.

6. Homonyms: The models might not be able to differentiate between sentences with

Empirically GPT-4 uses consistent list format, so can automatically parse out individual hypotheses

Formalizing a Hypothesis

Have list of hypotheses $h_1, \ldots, h_k$ as natural language descriptions

How to test if h is good?

Formalizing a Hypothesis

Have list of hypotheses $h_1, \ldots, h_k$ as natural language descriptions

How to test if h is good?

Take-away

h is a good hypothesis if it can be used to generate new failures

Formalizing a Hypothesis

Have list of hypotheses $h_1, \ldots, h_k$ as natural language descriptions

How to test if h is good?

Take-away

h is a good hypothesis if it can be used to generate new failures

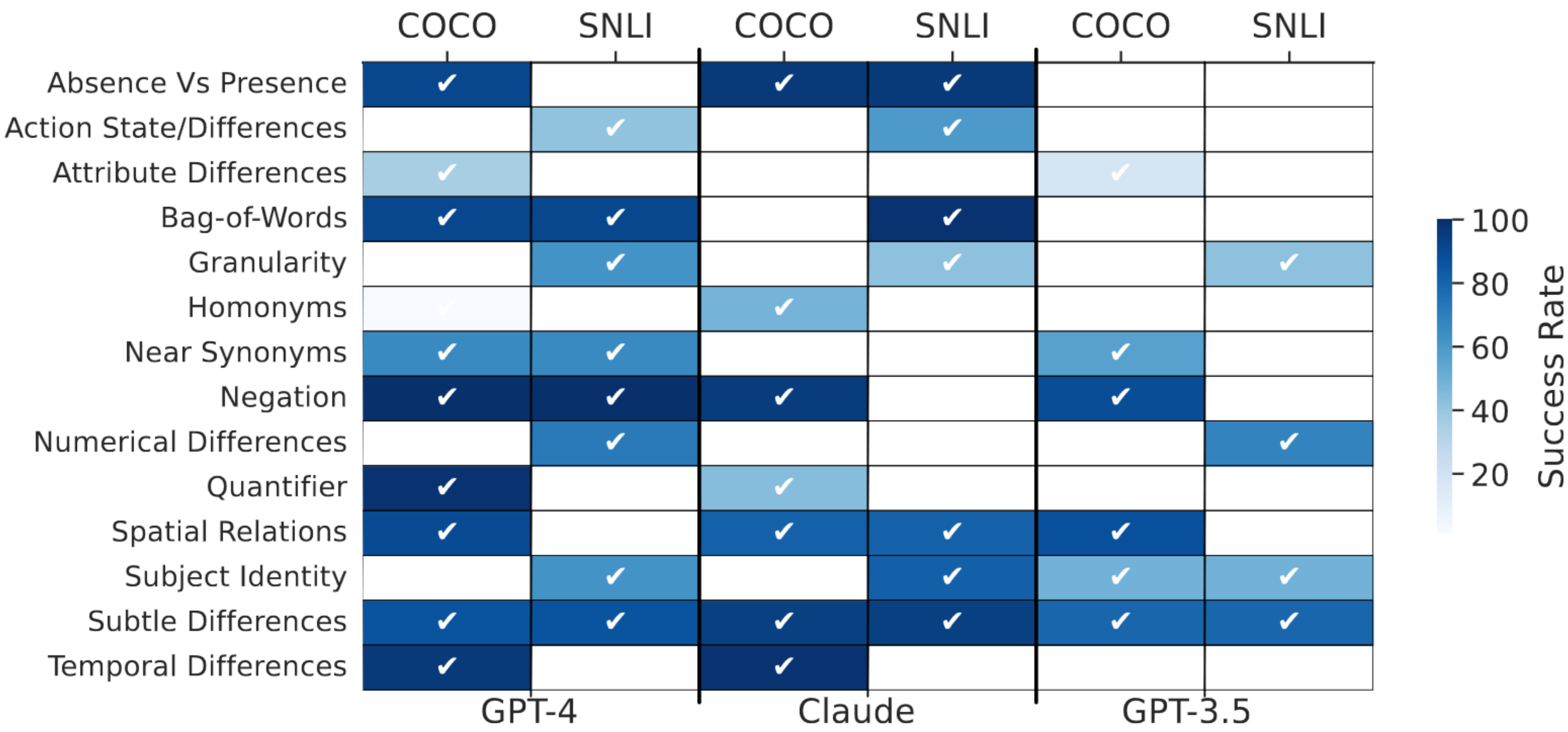
Prompt an LLM to generate new failures with h as context:

Prompt:

Write down 41 additional pairs of prompts that an embedding model with the following failure mode might encode similarly, even though they would correspond to different images if used as captions. Use the following format:
("prompt1", "prompt2"),
("prompt1", "prompt2"),
You will be evaluated on how well you actually perform. Your sentence structure and length can be creative; extrapolate based on the failure mode you've summarized. Be both creative and cautious.
Failure Mode:
[Systematic Failure (with full description)]

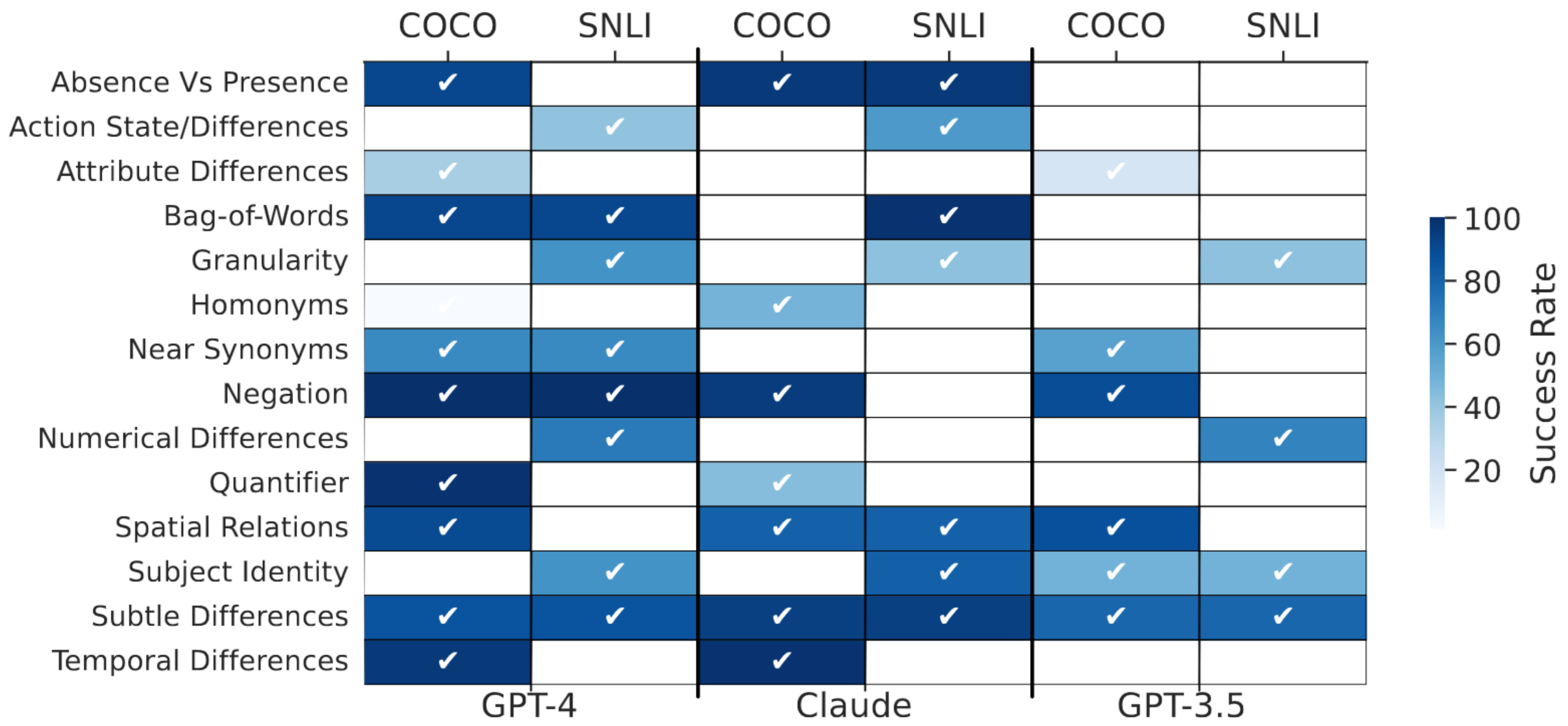
Testing on New Data: Hash Collisions

Fraction of new examples in each category that are hash collisions:



Testing on New Data: Hash Collisions

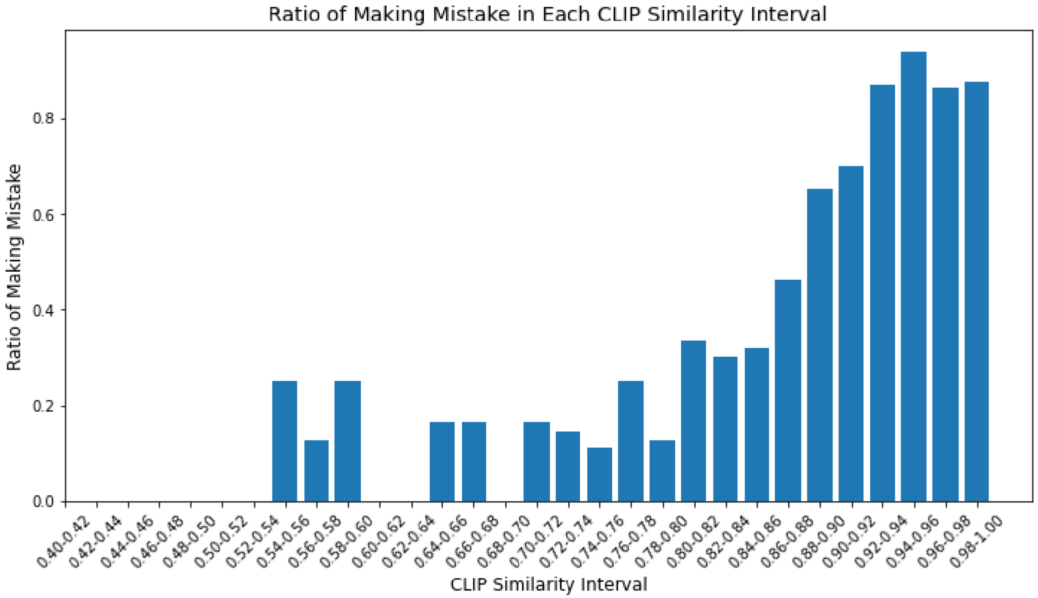
Fraction of new examples in each category that are hash collisions:



Larger models find more categories + describe them more effectively

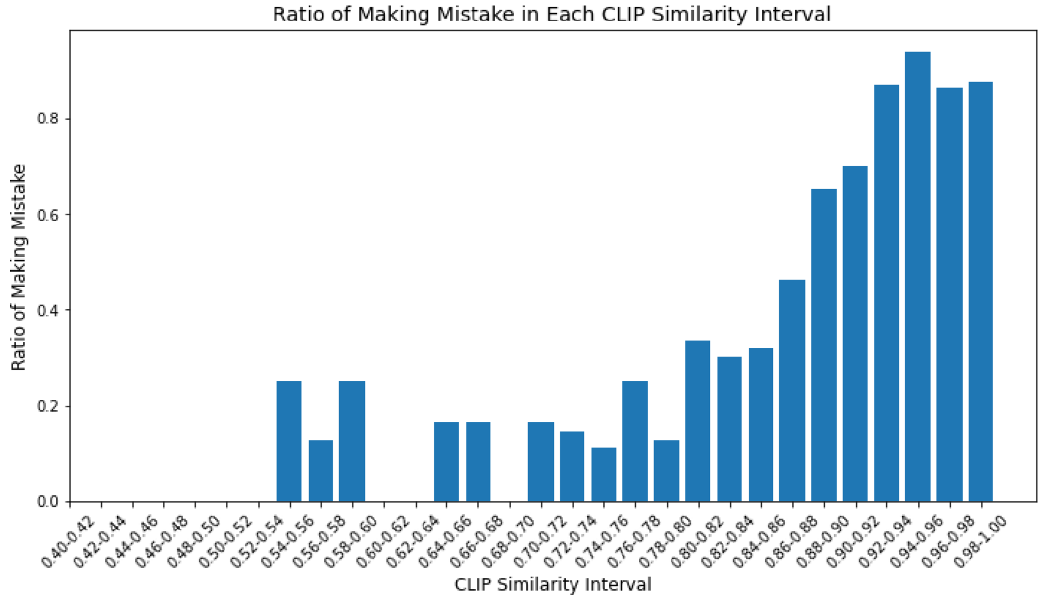
Testing on New Data: Human Evaluation

Hash collisions lead to images that humans say are wrong:



Testing on New Data: Human Evaluation

Hash collisions lead to images that humans say are wrong:



Data-driven descriptions help significantly

- Generate failures 80% of time, compared to 20% with baseline

Testing on New Data: Active Steering

Prompt GPT-4 to generate failures relevant to self-driving:

Stable Diffusion 2.1

“the car is on the right side of the lane”

DALL-E (New Bing)

“this is not a green light”

Shap-E

“a yield sign”

VideoFusion

“a car stops for red light”

Summary

Initial data: scrape hash collisions from text dataset

CLIP, DistillRobERTA

Generate hypothesis: prompt GPT-4

GPT-4

Formalize hypothesis: success rate generating new failures

GPT-4, CLIP

New data: actively generate examples in new domain

GPT-4

Statistical Modeling with Natural Language Parameters

Classifying with Natural Language Predicates

Task: given text datasets D_1 and D_2, find difference between them

Difference should be a natural language string h

Isomorphic to binary classification, but where function is described in natural language

Zhong et al. (2022), "Describing Differences between Text Distributions"
Zhong et al. (2023), "Goal-Driven Discovery via Language Descriptions"

Related: Andreas et al. (2017), Honovitch et al. (2022), Bills et al. (2023)
Hernandez et al. (2021), Zhu et al. (2022)

Example (Easy)

D_1

- Ma mère m’a emmené à l’hôpital.
- J'ai 10 $. Je dépense 3 $ sur un livre.
- Le gouvernement n'a pas réussi à localiser les suspects.

D_2

- My mom and I were best friends.
- Lucy and Peter co-authored a paper.
- I called her to explain why I did badly on the test.

Example (Easy)

D_1

- Ma mère m'a emmené à l'hôpital.
- J'ai 10 $. Je dépense 3 $ sur un livre.
- Le gouvernement n'a pas réussi à localiser les suspects.

D_2

- My mom and I were best friends.
- Lucy and Peter co-authored a paper.
- I called her to explain why I did badly on the test.

h = "D_1 contains more French sentences compared to D_2"

Use Cases

Example uses cases (separating distributions D_1 and D_2):

- The test distribution involves more formal writing than the training distribution.

Use Cases

Example uses cases (separating distributions D_1 and D_2):

- The test distribution involves more formal writing than the training distribution.
- The positive class contains more URLs than the negative class.

Use Cases

Example uses cases (separating distributions D_1 and D_2):

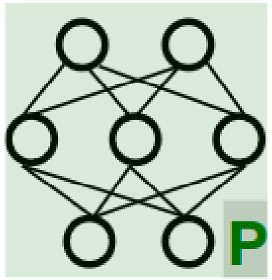
- The test distribution involves more formal writing than the training distribution.
- The positive class contains more URLs than the negative class.
- GPT-3's mistakes contain positive or uplifting language more often than TK-11B's mistakes.

Use Cases

Example uses cases (separating distributions D_1 and D_2):

- The test distribution involves more formal writing than the training distribution.
- The positive class contains more URLs than the negative class.
- GPT-3's mistakes contain positive or uplifting language more often than TK-11B's mistakes.
- Public opinion from this year is more optimistic about the pandemic than last year.

Using LLMs



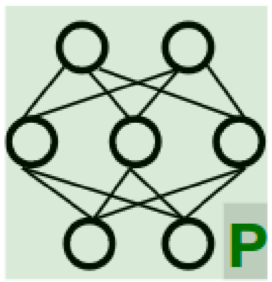
Hypothesis Proposer

A: Really curious about how the story would end!
A: Can't wait to see the next chapter.
A: Wow this was foreshadowed back in Chapter 1...

B: Still a lot of mysteries unsolved : (
B: The ending is so abrupt.
B: Wasted the tension it has built

Compared to group B, each sentence from group A.........

Using LLMs



Hypothesis Proposer

A: Really curious about how the story would end!
A: Can't wait to see the next chapter.
A: Wow this was foreshadowed back in Chapter 1…

B: Still a lot of mysteries unsolved : (
B: The ending is so abrupt.
B: Wasted the tension it has built

Compared to group B, each sentence from group A………

GPT-3 samples completions

- **is more positive**
- **contains the word "chapter"**
- **is longer**
- **...**

Candidate Hypotheses

Statistics Pipeline

Look at initial data

Form hypothesis h

Prompt GPT-n w/ examples from D_1, D_2

Ask how they are different

Statistics Pipeline

Look at initial data	Prompt GPT-n w/ examples from D_1, D_2
Form hypothesis h	Ask how they are different
Formalize h quantitatively	???
Test h on new data	???

Statistics Pipeline

Look at initial data	Prompt GPT-n w/ examples from D_1, D_2
Form hypothesis h	Ask how they are different
Formalize h quantitatively	???
Test h on new data	???

How can we quantitatively formalize h?

Statistics Pipeline

Look at initial data	Prompt GPT-n w/ examples from D_1, D_2
Form hypothesis h	Ask how they are different
Formalize h quantitatively	???
Test h on new data	???

How can we quantitatively formalize h?

Take-away

A good hypothesis helps tell D_1 and D_2 apart.

Natural Language Predicates

Example: $h =$ involves more formal writing

Interpret as two-argument predicate:

- For sentences x_1, x_2, $h(x_1, x_2) \in \{0, 1\}$ is the truth value of

  "x_1" involves more formal writing than "x_2"

Natural Language Predicates

Example: $h =$ involves more formal writing

Interpret as two-argument predicate:

- For sentences x_1, x_2, $h(x_1, x_2) \in \{0, 1\}$ is the truth value of

  "x_1" involves more formal writing than "x_2"

h is a correct hypothesis about D_1 vs. D_2 if

$$\mathbb{E}_{x_1 \sim D_1, x_2 \sim D_2}[h(x_1, x_2)] \ll 0.5$$

Natural Language Predicates

Example: $h =$ involves more formal writing

Interpret as two-argument predicate:

- For sentences x_1, x_2, $h(x_1, x_2) \in \{0, 1\}$ is the truth value of
  "x_1" involves more formal writing than "x_2"

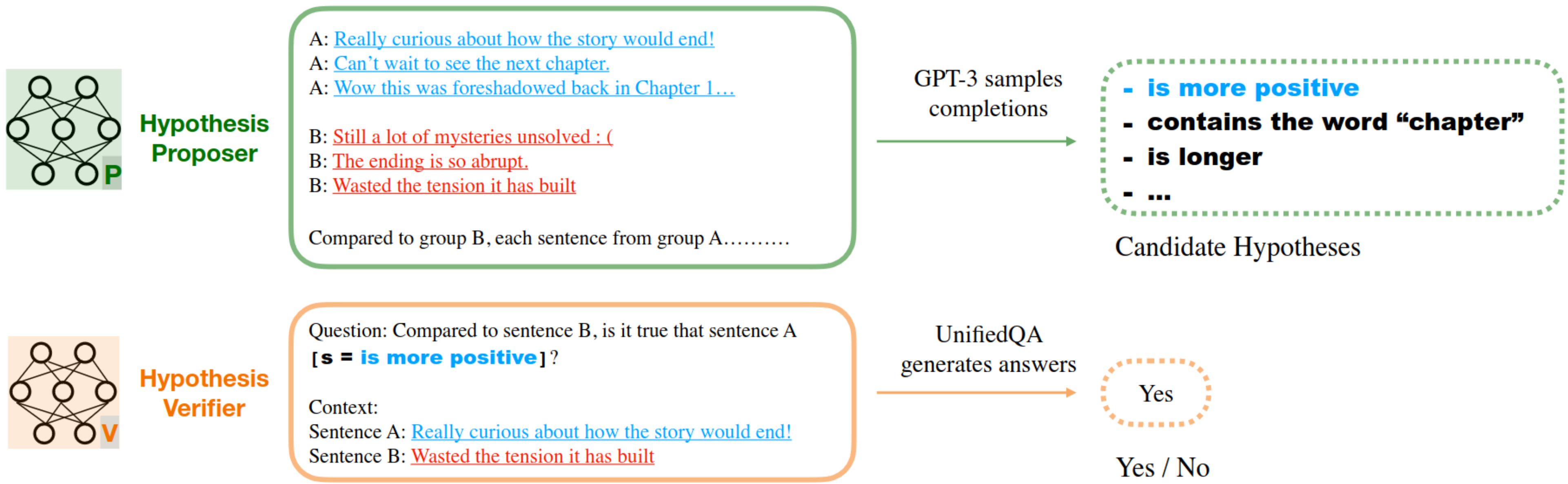
h is a correct hypothesis about D_1 vs. D_2 if

$$\mathbb{E}_{x_1 \sim D_1, x_2 \sim D_2}[h(x_1, x_2)] \ll 0.5$$

How to implement $h(\cdot, \cdot)$? Humans, or query a LLM

- LLMs reduce cost by 1000x ($0.07/hypothesis with gpt-3.5-turbo)

Overall System



Proposer: sees $\sim$30 examples (context window)

Verifier: can see thousands of examples

Can also steer proposer based on use case!

Use Cases: Understanding ML and Beyond

Finding spurious cues:

- Subjectivity analysis dataset: is a quote from a film review
- MNLI dataset: has a negative verb
- Spam classification: has a high number of hyperlinks

Use Cases: Understanding ML and Beyond

Finding spurious cues:

- Subjectivity analysis dataset: is a quote from a film review
- MNLI dataset: has a negative verb
- Spam classification: has a high number of hyperlinks

Automated error analysis: GPT-3 Curie vs Tk-11B

- Curie errs on language that is positive or uplifting

Use Cases: Understanding ML and Beyond

Finding spurious cues:

- Subjectivity analysis dataset: is a quote from a film review
- MNLI dataset: has a negative verb
- Spam classification: has a high number of hyperlinks

Automated error analysis: GPT-3 Curie vs Tk-11B

- Curie errs on language that is positive or uplifting

Other applications: 675 total use cases across several domains

Domain	Example Datasets	How the Corpus Pairs are Generated	
		Corpus A	Corpus B
		87 **Business** problems	
Commercial Reviews	Airline reviews	1st-class passenger reviews	Economy passenger reviews
	Product Reviews	Reviews that give 10 stars	Reviews that give 0 star
Finance	YC startups	Successful startup descriptions	Failed startup descriptions
	News Headlines	Top headlines when S&P rises	Top headlines when S&P falls
		278 **Social Sciences** problems	
Politics	Administration policy	Admin policy from Trump	Admin policy from Obama
News	Reuters headlines	Headlines from 2014	Headlines from 2015
Language	Craiglist Negotiations	Dialogue from successes	Dialogue from failures
	Diplomacy Dialogues	Lies	Honest statements
Sociology	Happy moments	Self-reported happy moments from females	Self-reported happy moments from males
	Rate My Professor	Reviews of female lecturers	Reviews of male lecturers
		169 **Humanities** problems	
Arts	Music lyrics	Drake rap lyrics	Kanye rap lyrics
Education	Student essays	Essays that received full score	Essays with only partial credit
		10 **Health** problems	
Health	Doctor's note	Patients diagnosed with pneumonia	Patients not diagnosed with pneumonia
		131 **Machine Learning** problems	
Machine Learning	NLI — distribution shift	Samples from SNLI	Samples from MNLI
	QQP — spurious correlation	Individual questions with label "paraphrase"	Individual questions with label "non-paraphrase"
	LM's output	Generations from one LM	Generations from another LM
	inputs — error analysis	Inputs where one model is correct	Inputs where one model is wrong
	WikiText — clustering	Samples from one cluster	Samples not from a cluster

Summary

Initial data: text distributions D_1 and D_2

Generate hypothesis: prompt GPT-3

GPT-3 (fine-tuned), Text-Davinci-003 (prompting)

Formalize hypothesis: success rate distinguishing samples

UnifiedQA (fine-tuned)

New data: test on held-out samples from D_1, D_2

Extension: Exponential Families

Given natural language predicate h, define

$$[[h]] : x \mapsto \{0, 1\}$$

as the truth value of h on input x

Then can define exponential family:

$$p(x \mid \vec{w}, \vec{h}) \propto \exp(w_1[[h_1]](x) + \cdots + w_k[[h_k]](x))$$

Use this as basis of more complex models (topic modeling, low-rank factorization, clustering, ...)

Zhong et al. (2023), in preparation

Application: Multimodal Clustering

Query: I'm classifying dogs vs. elephants. I want to understand how their backgrounds are different.

Zhong et al. (2023), in preparation

Coda: Labeling Activation Vectors

Find text-backed “principle components” for each attention head:

“Interpreting CLIP’s Image Representation via Text-Based Decomposition”
Gandelsman, Efros, Steinhardt (2023)

Related: Hernandez et al. (2022), Oikarinen and Weng (2023)

Coda: Labeling Activation Vectors

Find text-backed "principle components" for each attention head:

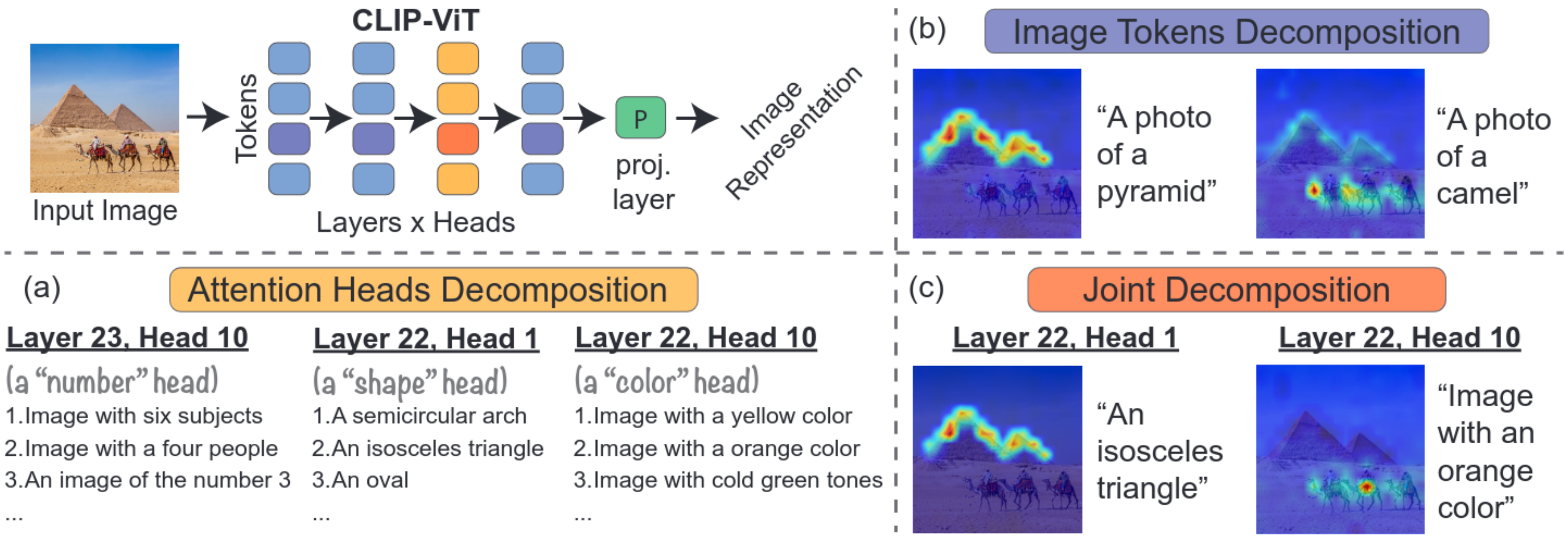
Automatically generates thousands of descriptions

Can be used for model repair

"Interpreting CLIP's Image Representation via Text-Based Decomposition"
Gandelsman, Efros, Steinhardt (2023)

Related: Hernandez et al. (2022), Oikarinen and Weng (2023)

Coda: Labeling Activation Vectors



"Interpreting CLIP's Image Representation via Text-Based Decomposition"
Gandelsman, Efros, Steinhardt (2023)