Natural Language Explanations of Dataset Patterns

Ruiqi Zhong

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2025-78
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2025/EECS-2025-78.html

May 15, 2025

Natural Language Explanations of Dataset Patterns

By

Ruiqi Zhong

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Assistant Professor Jacob Steinhardt, Chair
Professor Dan Klein
Associate Professor Will Fithian

Spring 2025

Natural Language Explanations of Dataset Patterns

Abstract

Natural Language Explanations of Dataset Patterns

by

Ruiqi Zhong

Doctor of Philosophy in Computer Science

University of California, Berkeley

Assistant Professor Jacob Steinhardt, Chair

Explaining patterns in large datasets is essential for empirical science, engineering, and business. For example, by analyzing a dataset of symptom descriptions, a doctor may discover that "*tingling in the thumb*" is a good explanatory variable for disease X. However, existing methods (e.g. regression) are primarily designed to analyze real-valued datasets and explain patterns in mathematical formulas (e.g. $F = kx + b$).

This thesis proposes metrics and methods for discovering and explaining dataset patterns in structured modalities (text/images) using natural language strings such as "*tingling in the thumb*". We evaluate the explanations based on the predictive power they give to humans, which differs from common metrics based on human ratings or similarity to human demonstrations. We then generate dataset explanations by optimizing them against our evaluation metric, with the help of language models. Concretely, we sample candidate explanations from language models and select the highest-scoring one under our evaluation.

Based on these principles, we build a general framework, "statistical models with natural language parameters", which allows us to explain distributional differences, clusters, and time-series in real-world datasets with structured modalities. Additionally, our metric can evaluate explanations of model decisions by treating them as explanations of datasets, which consist of the model's input-output behavior. Using this approach, we show that language models are still far from explaining themselves as of 2024. Our contribution paves the way for helping humans understand complex datasets and systems, thereby accelerating scientific discovery and advancing explainable AI systems.

Contents

List of Figures

List of Tables

Acknowledgments

The journey began in 2019 when Prof. Dan Klein admitted me to UC Berkeley. I am grateful to him for providing research, writing, presentation, and career advice throughout my Ph.D.

I would like to thank members of the Berkeley NLP group, Jacob Steinhardt's group, and the broader UC Berkeley EECS community for supporting me both academically and emotionally. I am grateful to Dan Hendrycks for being a great inspiration and exemplar, whose existence constantly reminds me that I am not hardworking and goal-driven enough. I want to thank Charlie Snell for helping me execute several of my representative works, Jiaxin Wen for leading projects that complete the narrative of my dissertation, Nicholas Tomlin for his comprehensive feedback on my slides, Yanda Chen for our long-term collaboration, Peilin Zhong for helping me get started in research, and Lisa Dunlap for connecting my work with practical applications. I also appreciate Meena Jagadeesan and Erik Jones for exchanging valuable research insights. I extend my gratitude to Kathleen McKeown, Sasha Rush, Aviral Kumar, Greg Durrett, Karen Livescu, Roger Grosse, and Danqi Chen for their assistance during my academic job search, and to Jason Eisner, Ethan Perez, and John Schulman for providing industry opportunities. I thank Trevor Darrell and Will Fithian for serving on my qualification committee, which helped me improve my ability to incorporate feedback.

Beyond my professional life, I am thankful to my friend Kaidi Zhang for his enduring support since our primary school days. I am deeply grateful to my parents, Huanhong Xie and Wen Zhong, for giving me life, raising me, and supporting me throughout my journey; my life has been fulfilling and worthwhile. I thank my girlfriend Lizzi Yin for her support during my most stressful times and for bringing daily happiness into my life.

Finally, I am profoundly grateful to Jacob Steinhardt for providing both academic and life advice. For nearly two decades, I struggled with existential angst about my life's direction. Jacob liberated me from this angst, and now I have clear goals to pursue.

Chapter 1

Introduction

Input a dataset, explain its patterns — this is one of the most powerful methodologies in empirical analysis. Consider a simple physics experiment: measuring how much a spring stretches when different forces are applied. By hanging weights (force F) and measuring the stretch (x), we collect data pairs (F, x), forming a real-valued dataset. Using linear regression, we output an explanation by learning parameters β and σ: the force F is proportional to the stretch x with coefficient β, up to some tiny measurement noise ϵ in x

$$x = \beta F + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2)^1 \tag{1.1}$$

This pattern, known as Hooke's Law [51], is fundamental for understanding elastic behavior in materials. It is not derived, but discovered by explaining patterns from a real-valued dataset. This highlights the importance of mapping real-valued datasets to patterns represented in formulas — so important that it led to its own field: statistical modeling.

However, existing methods cannot directly map from datasets D with structured modalities (e.g., text/image) to patterns explained in natural language e. For example, in medical imaging, the data D could be chest X-ray images, and the pattern could be that "*asymmetry in opacity*" (e) signals risks. In language model (LM) training, the data D could be LM-generated content paired with user ratings, and the pattern could be that users prefer outputs "*containing bullet points*" (e). In business, the data D could be reviews for laptops, and the pattern could be that many reviews "*complain about the speed to start*" (e).

To handle these applications, existing methods first encode structured inputs into high-dimensional real vectors (e.g., bag-of-words, pixels, or embeddings), then learn real-valued parameters as explanations. For example, topic models like LDA [13] encode text samples as a bag-of-words and discover topics by learning weights over a long list of words. However, as illustrated in prior work [21] and Figure 1.2, these weights are often not interpretable for humans, defeating the goal of explaining dataset patterns; for example, if a topic assigns high weights to the words "[piece, pandas]", it remains unclear what this topic means. Several prior works highlight similar failures: BERTopic [43] learns uninterpretable cluster centers

[1] σ specifies the amount of noise in this pattern.

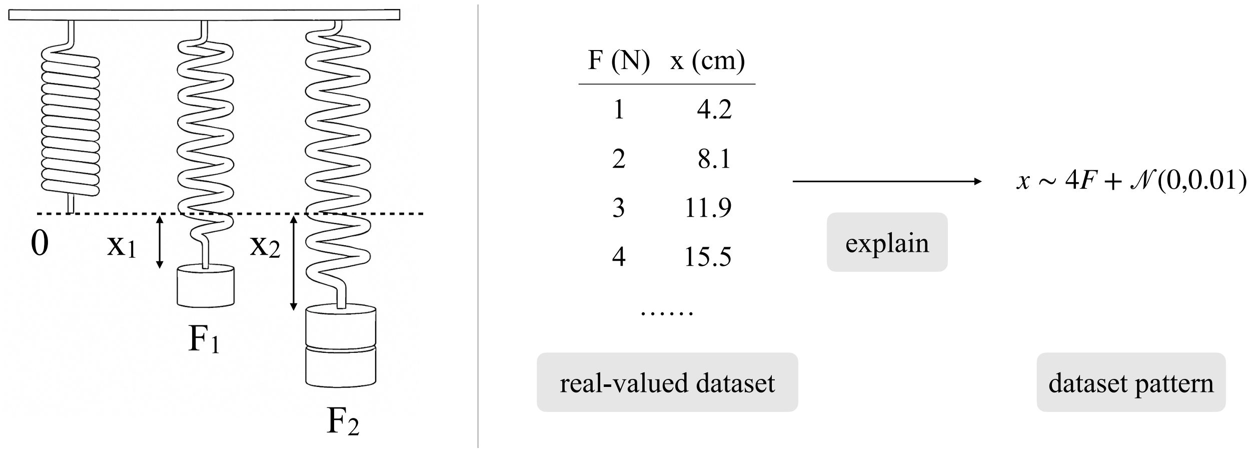

Figure 1.1: Hooke's law (a dataset pattern, right) is discovered by explaining the pattern from a real-valued dataset of (F, x) pairs (middle).

over neural embeddings; Naive Bayes assigns weights to many words or phrases, which do not directly explain abstract concepts [21, 104, 126].

To address these challenges, this thesis proposes explaining dataset patterns in natural language, which is inherently interpretable. For example, as shown in Figure 1.2, a topic is directly represented by an explanation such as "*asks a coding question*", rather than weights over a list of words, whose meaning is unclear. We propose methods for both 1) evaluating and 2) generating explanations.

Evaluation. Existing methods to evaluate natural language outputs either compare to human references or collect human ratings. However, neither of these can be directly applied to dataset explanations. It is infeasible to compare directly with human-written references [7, 85], because we aim to explain patterns that humans do not yet know. It is also unreliable to directly ask humans to rate the explanations [99], as D is too large for humans to accurately judge whether an explanation e reflects the data.

To evaluate natural language explanations, we measure the predictive power they enable in humans. For example, a natural language explanation e of a cluster (e.g. "*asks about coding*") should empower humans to predict whether a sample x (e.g. "debug this program") belongs to that cluster. Formally, if we treat e as a predicate, then $[\![e]\!](x) = 1$ if x is a sample that belongs to a cluster, and $[\![e]\!](x) = 0$ otherwise. We can then define an evaluation metric PredPower (predictive power) for a cluster explanation as follows:

$$\textsc{PredPower}(e) = \mathbb{E}_{x\sim D_{\text{cluster}}}[[\![e]\!](x)] - \mathbb{E}_{x\sim D_{\text{others}}}[[\![e]\!](x)], \tag{1.2}$$

where D_{cluster} is the set of samples that belong to a cluster and D_{others} are other samples. To automate the evaluation, we use language models to simulate human predictions ($[\![e]\!](x)$) after seeing an explanation.

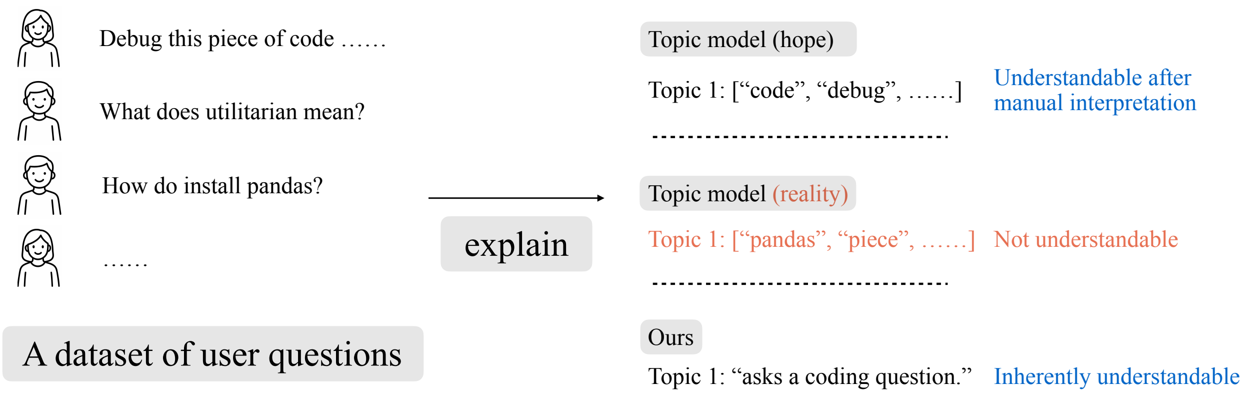

Figure 1.2: Traditional statistical models struggle to explain text datasets. Consider an application of discovering topics from a dataset of user questions. Traditional topic models like LDA [13] extract a list of key words for each topic and hope that the word lists are understandable (top right), but in reality this is not always the case (middle right) [21]. In contrast, our formulation uses natural language to directly explain each topic (bottom right).

Generation. Once we can automate PredPower, the evaluation of explanations, we can generate better explanations e by optimizing against PredPower. However, such optimization is challenging because e is discrete, preventing the use of gradient-based methods.

To address this, we use language models to sample multiple candidate explanations e and select the best ones according to PredPower. To further accelerate optimization, we develop algorithms that optimize continuous representations of explanations, convert them back to natural language, and iteratively refine them.

We visualize how we generate and evaluate explanations in Figure 1.3.

Thesis overview. This thesis is structured as follows. The next section reviews related work, including exploratory data analysis, inductive reasoning with language models, and epistemology. Chapter 2 introduces the task of explaining differences between two distributions, presenting methods for both generation and evaluation. Chapter 3 extends this to a broader framework for explaining complex patterns, such as clustering and time-series. Chapter 4 reframes model explanation as dataset explanation, showing that current language models often struggle to explain their own behavior. Finally, Chapter 5 summarizes our findings and highlights directions for future research.

1.1 Related Works

Epistemology. In our framework, we do not distinguish between “a discovery from a dataset” and “an explanation of a dataset pattern.” Both are treated as natural language statements that help humans improve their ability to predict parts of the dataset. This perspective

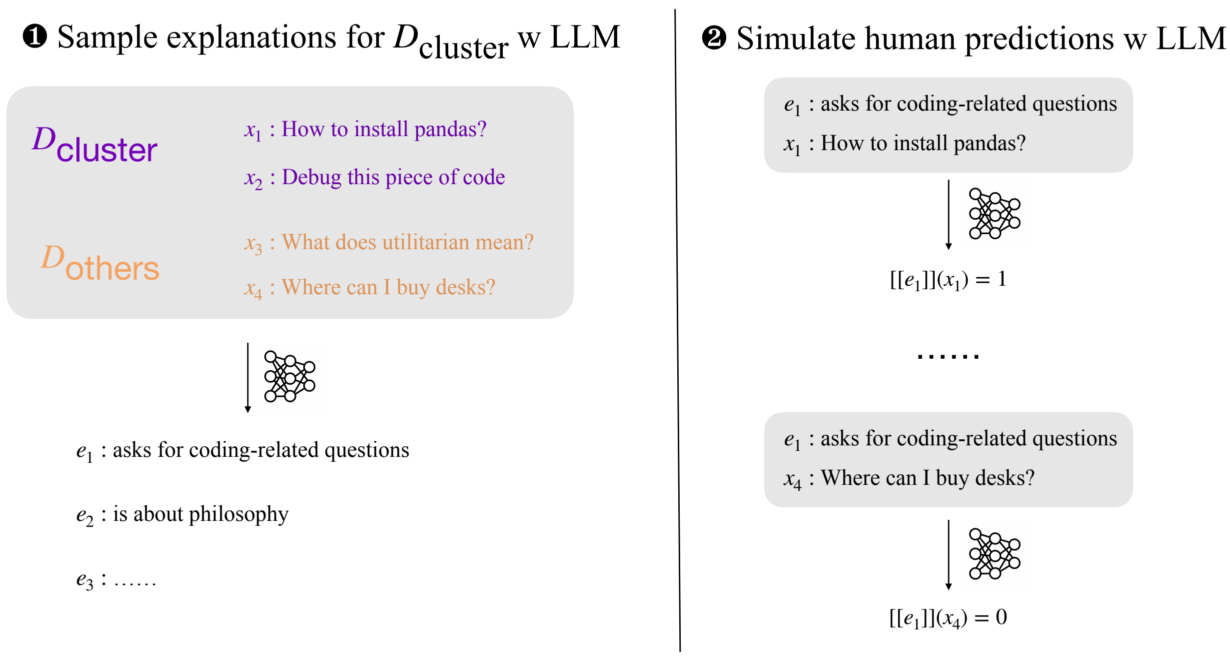

Figure 1.3: A typical pipeline of generating explanations. We first use language models to propose a few candidate explanations by prompting with samples x from the dataset. Then we evaluate each explanation e by simulating human predictions on each sample x (i.e. checking whether e matches x), and then pick the highest scoring explanation.

aligns with scientific instrumentalism [19], which defines the value of an explanation by its usefulness in predicting empirical data.

Our explanation-generation process is similar to the algorithm proposed by [67], where some humans propose explanations for judge predictions based on example images, and others validate these explanations using the dataset. While their setup involves human input, our process is fully automated by simulating both roles with language models.

Exploratory Analysis and Automated Discovery. The idea of automatically explaining patterns based on data has a long history. For instance, linear regression interprets the impact of real-valued features by analyzing learned weights [32]. N-gram models extract discriminative phrases that highlight corpus-level differences [70]. Topic models identify major thematic variations across documents, representing each topic as a distribution over words [13]. Small decision trees yield interpretable if-then statements [63], and entity embedding models trained on known relationships can predict unseen ones [98]. Our work builds on this tradition but focuses on natural language explanations, which are more expressive and capable of conveying abstract concepts.

Inductive Reasoning with Machine Learning Models. Our work relies on the inductive reasoning capabilities of language models, which can identify patterns from examples and describe them in natural language [88, 50]. Recent studies have explored similar directions. [48] describes visual features that activate individual neurons. [129] identifies distribution shifts between training and test datasets for images, and [36] explains systematic errors made by vision models. Other works use language models to induce structured knowledge: [113] generates natural language rules in the form of "if...then..." statements, while [128] and [115] improve zero- and few-shot learning by inferring task instructions from input-output examples. A related line of research explores language-based concept bottleneck models, which use machine learning models to propose interpretable features for classification tasks [112, 66, 25, 94]. Our thesis focuses more on explaining dataset patterns and works broadly for clustering and time-series as well.

Chapter 2

Explaining Distributional Differences

What inputs trigger a neuron in my deep learning model? How are the train and test distributions different for my application? How did public opinions on Twitter change from last year to this year? These questions have significant scientific, economic, and social consequences. However, explaining patterns sometimes requires scanning over thousands of examples, which is intractable for humans. An automated solution would be much more scalable.

To address this, we develop a method to explain the differences between two distributions with natural language. The input to this method consists of two text distributions D_0 and D_1, and the output is a natural language explanation e. For instance:

- We can describe what triggers an artificial neuron by setting D_1 to be inputs that trigger it and D_0 for other inputs. e could be "*is military-related*" (Figure 2.1).
- We can describe the differences between the train and test distributions by setting them to be D_0 and D_1. A possible e would be "*is longer in sentence length.*"
- We can describe how public opinions shifted by setting D_0/D_1 to be the opinions from last year/this year. e could be "*is optimistic about the pandemic.*"

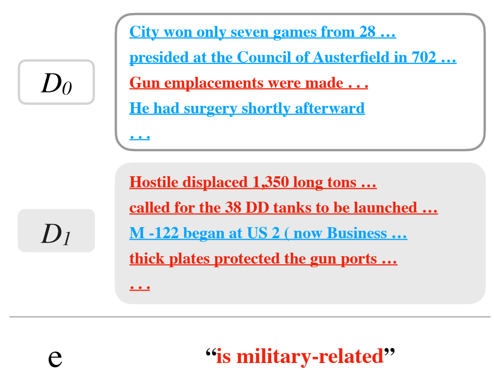

Figure 2.1: Given two distributions (top), our system automatically explains their differences and describes them with natural language (bottom). Grey/white background represents D_0/D_1 and red/blue represents whether a sample matches the explanation e.

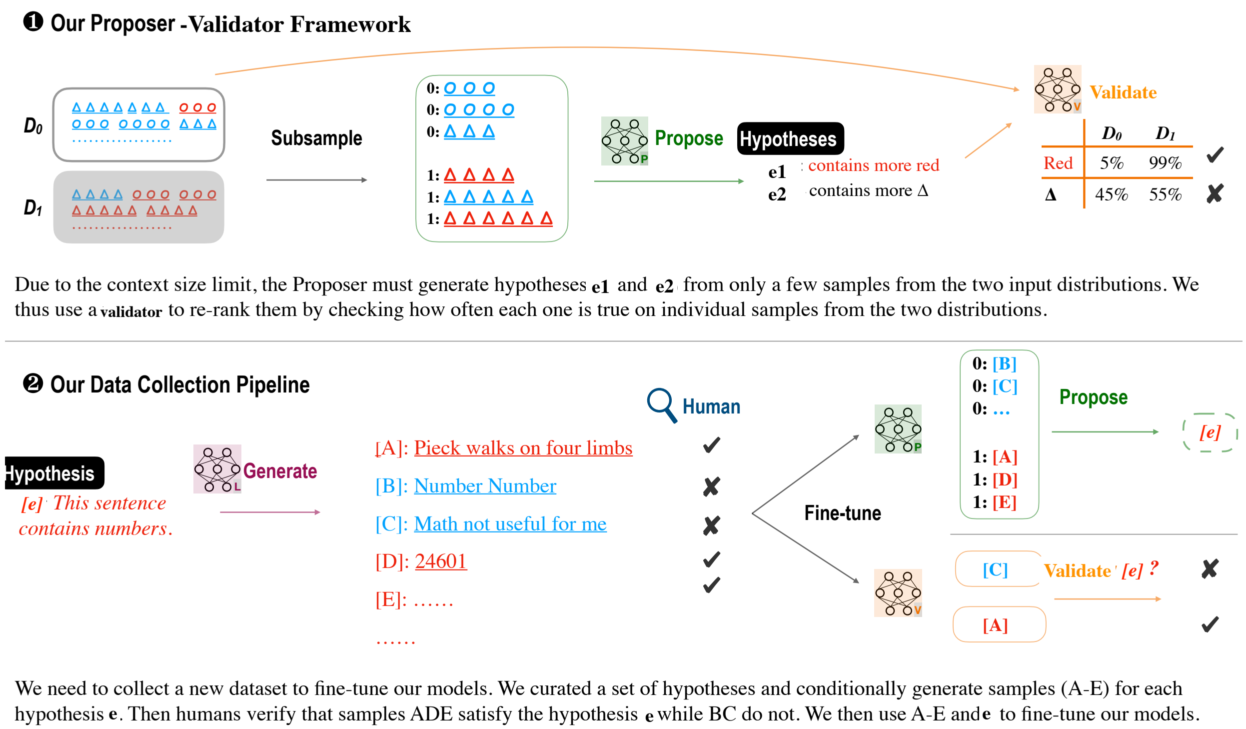

Figure 2.2: Our architectural framework (**top**) and data collection pipeline (**bottom**). Section 2.2 describes them in detail.

To evaluate e, we measure the predictive power it enables in humans: can humans use e to discriminate samples from D_0 from D_1? To automate the evaluation, we use a language model to simulate how humans use e. We then generate the explanation by optimizing a natural language string against the automatic evaluation. In particular, we prompt GPT-3 Davinci (175B) [16] with samples from each distribution, ask it to propose candidate explanations e (Section 2.2.1), and then use the automatic evaluation to rerank the explanations e. We visualize our framework at the top of Figure 2.2 and the prompts at the top of Figure 2.4.

Since GPT-3 is not optimized to propose explanations, we can improve it through fine-tuning. However, no corpus exists yet for this task. Therefore, we developed a new data collection pipeline (Section 2.2.3) with three stages: 1) we curated a list of explanations e, 2) we asked GPT-3 to generate samples that satisfy e, and 3) we asked annotators to judge whether they indeed satisfy e. Then we fine-tuned the proposer to predict e based on samples that satisfy e and samples that do not (Section 2.2.4). We visualize our data collection and fine-tuning method at the bottom of Figure 2.2.

We benchmark our system on 54 real-world binary classification datasets [124], each annotated with natural language explanations for the positive class. For each binary task, we treat the positive/negative class inputs as D_1/D_0 and compare the top-5 explanations

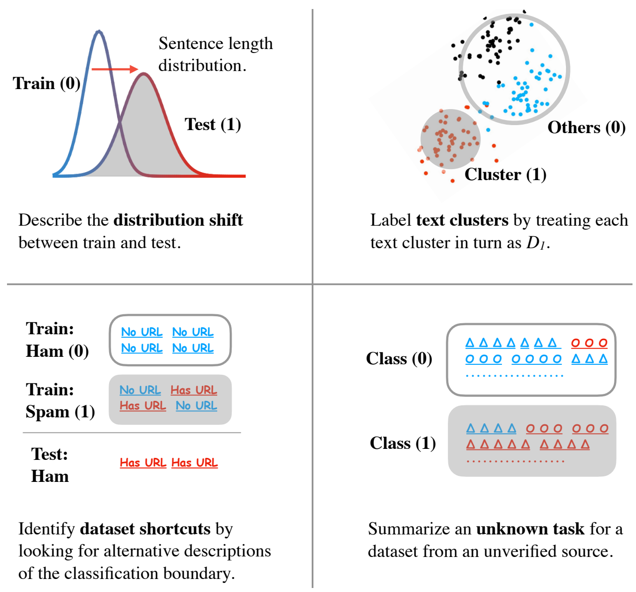

Figure 2.3: We reduce a wide range of applications to learning a natural language explanation and present our analyses in Section 2.4.

by our system to the human annotation. While the explanations by GPT-3 Curie (13B) are similar to the annotations only 7% of the time, the performance reaches 61% with fine-tuning and validator re-ranking, and our best system using GPT-3 Davinci (175B) reaches 76% (Section 2.3).

We then check whether the intended uses of existing classification datasets agree with the explanations by our system (Section 2.4). Our system correctly recognizes that the subjectivity analysis (SUBJ) dataset [83] was constructed by contrasting movie reviews with plot summaries; however, many recent papers [15, 124, 37, 73] were unaware of this fact and used SUBJ for zero/few-shot subjectivity classification. Our system also recognizes several dataset shortcuts. For example, it rediscovered that negations, such as the use of "not/never", are spuriously correlated with the contradiction class in MNLI [44]; for another example, models trained on the SMS Spam classification dataset [41] always consider hyperlinks to be spam. Our system can also describe distribution shifts and text clusters (Section 2.4), and Figure 2.3 visualizes all our applications.

2.1 Evaluating and Generating Explanations

2.1.1 Preliminaries

Let $\mathcal{X}$ be the set of all text inputs. We require an explanation e to be a natural language string whose denotation $[\![\cdot]\!]$ can map from two inputs to a boolean:

$$[\![e]\!] : \mathcal{X} \times \mathcal{X} \to \{0, 1\}, \tag{2.1}$$

where $[\![e]\!](x_1, x_0) = 1$ means x_1 is more e than x_0. For example, if e is "*is longer in sentence length*," then $[\![e]\!](x_1, x_0) = 1$ means x_1 is longer than x_0. The denotation $[\![e]\!]$ is defined as

$$[\![e]\!](x_1, x_0) \stackrel{\text{def}}{=} \mathbf{1}[\text{humans consider } x_1 \text{ more } e \text{ than } x_0], \tag{2.2}$$

which our paper operationalizes by taking the majority vote among crowdworkers.[1]

Note. In this chapter, we work with explanations that accept two samples x as arguments. In all other parts of this thesis, we consider explanations e in the form of standard predicates that accept one sample x as the argument, i.e.,

$$[\![e]\!] : \mathcal{X} \to \{0, 1\} \tag{2.3}$$

Regardless of the number of arguments, the same method for generating e applies.

2.1.2 Evaluating an Explanation based on Predictive Power

Let D_0 and D_1 be two distributions over $\mathcal{X}$, and $\mathcal{E}$ be the space of all valid natural language explanations. Intuitively, a good explanation e should enable predictive power in humans: given two random samples from each distribution $x_0 \sim D_0$ and $x_1 \sim D_1$, e should allow humans to *classify* where each x comes from as *accurately* as possible. We denote our evaluation metric as PredPower.

$$\text{PredPower}(e) \stackrel{\text{def}}{=} \mathbb{E}_{x_0 \sim D_0, x_1 \sim D_1}[[\![e]\!](x_1, x_0))]. \tag{2.4}$$

2.1.3 Learning (Generating) an Explanation

We observe that our task falls under the standard formulation of statistical machine learning, where we learn an explanation e by optimizing a statistical objective (PredPower) over a parameter space $\mathcal{E}$. Compared to traditional statistical learning, learning a natural language explanation poses two new challenges.

Search. Searching in a discrete string space is hard. Section 2.2.1 addresses this by proposing e with a neural network based on samples from D_0 and D_1.

[1] More broadly, however, there is no canonical method to interpret natural language.

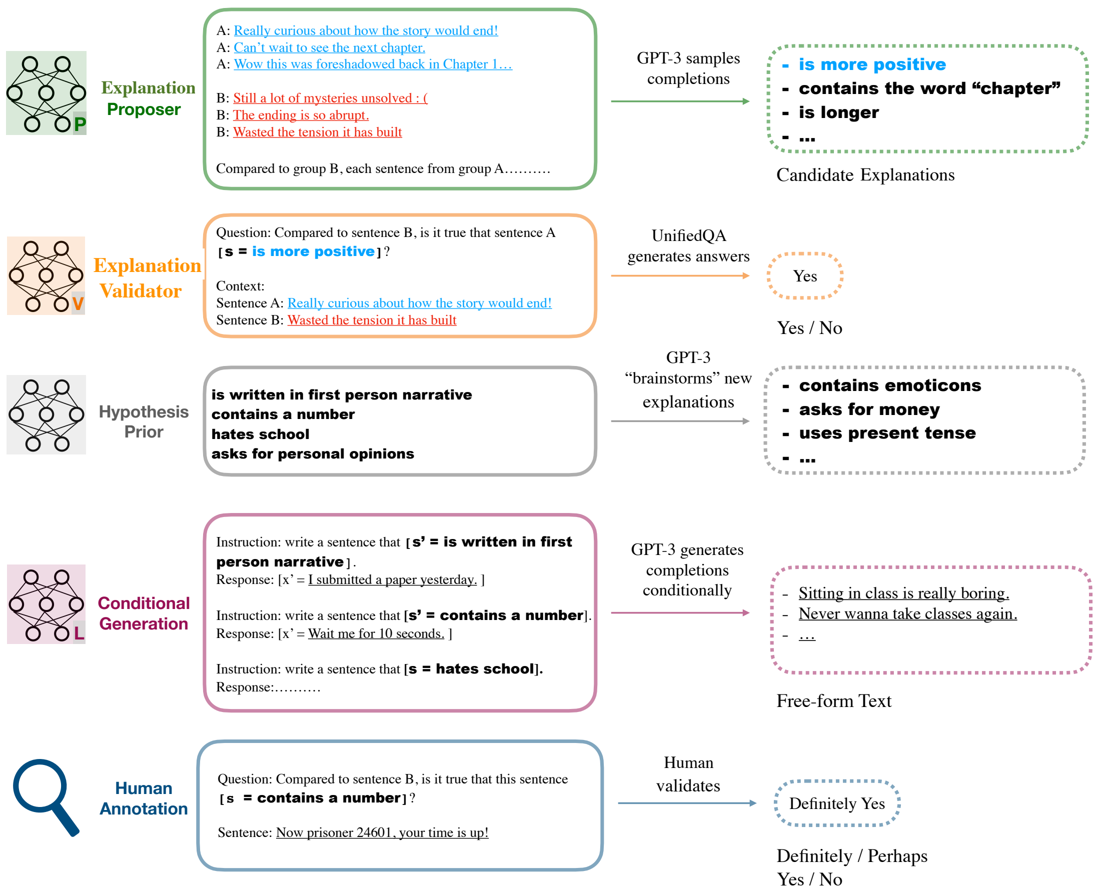

Figure 2.4: The prompt template for all components in our system. All text datapoints x are underlined and explanations e are bolded.

Validate. Computing $[\![e]\!](x_1, x_0)$ requires human annotations, which can be expensive. Section 2.2.2 addresses this by approximating human annotations with a neural network.

2.2 Method

We prompt GPT-3 to propose explanations based on a small set of samples (Section 2.2.1) and use UnifiedQA to validate each explanation on a larger set of samples (Section 2.2.2). Then, we design a data collection pipeline (Section 2.2.3) to further fine-tune both the proposer and the validator (Section 2.2.4). Our methods can be visualized in Figure 2.2.

2.2.1 Explanation Proposer

Our goal is to generate a list of plausible explanations based on samples from D_0 and D_1. We do so by prompting GPT-3, a language model that can generate textual completions based on a prompt. We construct a "proposer prompt" by concatenating several samples from D_1, several from D_0, and the instruction "*Compared to group 0, each sentence from group 1 ___*" (Figure 2.4, the 1st row). Since GPT-3 has a context size limit of 2048, we select 5 samples x from each distribution.

Without controlled decoding, a typical prompt completion would be "*is more positive, while sentences from group 0 are ungrammatical.*" However, such a completion is undesirable, since 1) the validator now needs to check two statements at the same time, namely, whether samples from D_1 are positive and samples from D_0 are ungrammatical, and 2) the second half of the completion describes a population-level property of "group 0", while our validator only checks explanations on individual x. To produce a single explanation about individual x, we forbid GPT-3 to decode tokens like "*group*" and terminate with token "," or ".".

Additionally, D_0 and D_1 might overlap, and even an optimal explanation e^* cannot fully separate them. As a result, the proposer prompt might contain samples from D_1 that do not satisfy e^*, thus confusing the proposer. Therefore, we choose samples that are representative of their differences to prompt GPT-3. To find those samples, we fine-tune RoBERTa-Large [65] to predict whether each sample comes from D_0 or D_1 and retain the top-p percentile samples with the highest confidence. For the top-5, 20, and 100th percentile, we construct proposer prompts with ten different random sets of samples and generate two completions for each set. We obtain $3 \times 10 \times 2 = 60$ explanations, which we rerank in the next section.

2.2.2 Explanation Validator

Ideally, we should re-rank e based on its classification accuracy $\textsc{PredPower}(e)$, defined in eq. (2.4). However, it involves computing $[\![e]\!](x_1, x_0)$, which requires costly human annotations (Equation (2.2)). We therefore approximate it with a validator neural network V:

$$\hat{[\![e]\!]}(x_1, x_0) \stackrel{\text{def}}{=} \frac{1}{2}(V(s, x_1, x_0) - V(e, x_0, x_1) + 1). \tag{2.5}$$

Here $V(e, x_1, x_0) = 1$ if it predicts that x_1 is more e than x_0 (0 otherwise); then we subtract the baseline $V(e, x_0, x_1)$ obtained by swapping the position of x_0 and x_1, and finally normalize the quantity within $[0, 1]$.

We implement our validator with UnifiedQA [57], a question answering model based on T5 (11B) [90]. UnifiedQA generates an answer a given a question q and a context c. As shown in the 2nd row of Figure 2.4, our context c is a pair of sentences A (sampled from D_1) and B (sampled from D_0). The question q is then "*Is it true that sentence A* ***is more positive***?", where in general the bolded part is an explanation e generated by the proposer. Then we define $V(e, x_1, x_0) = 1$ if UnifiedQA outputs "yes" and 0 if it outputs "no".

We now use $V(e, x_1, x_0)$ to compute $\textsc{PredPower}(e)$ for each candidate e and re-rank them. To save computation, we estimate $\textsc{PredPower}(e)$ with 400 random pairs of (x_1, x_0)

rather than using the entire datasets. Finally, we output the top-5 explanations to describe how D_1 and D_0 differ.

2.2.3 Collecting Data for Supervision

Since GPT-3 and UnifiedQA are not specifically trained to propose or validate explanations, we can improve them by fine-tuning [124]. However, since no corpus exists yet for these tasks, we need to collect a new dataset to fine-tune our models.

To fine-tune the proposer, we want data where the output is an explanation e and the input prompt contains five samples that are more e and five that are less e. To fine-tune the validator, we want tuples (e, x_1, x_0) where x_1 is more e than x_0. Thus, for both cases, we want a set of explanations e, and for each of them, two groups of samples where one group is more e than the other. We designed our data collection pipeline accordingly: we curated a set of explanations e, asked GPT-3 to generate samples that do (not) satisfy e, and asked humans to filter out failed generations.

Curating Explanations. We curated a pool of 302 explanations by hand with the help of GPT-3 [16]. Concretely, we started the pool by brainstorming ten explanations ourselves; then, we sampled five explanations from the pool and prompted GPT-3 with their concatenation, as visualized in the 3rd row of Figure 2.4. Whenever GPT-3 completed the prompt with an explanation different from our existing ones, we added it to the pool.

Our curated explanations range from shallow ("*contains the word "yay" at the end of the sentence*") to topical ("*loves school*") to more complex social and linguistic cues ("*supports universal healthcare*," "*is written in first person*"). To make later conditional generation and human annotation easier, we removed any comparatives from e, e.g., removing the word "*more*" in "*loves school more*."

Conditional Generation. We refer to samples that satisfy e as "positive" and others as "negative". For example, given e = "*loves school*", a positive sample could be "*My advisor is really helpful and I learned a lot.*" Both positive and negative samples are necessary to fine-tune our models.

To generate positive samples, we prompted GPT-3 as visualized in the 4th row of Figure 2.4: we curated a set of explanations e' and their positive samples x by hand, concatenated them with the target explanation e, and asked GPT-3 to generate a sample x. Sometimes, however, x satisfies e due to trivial word overlap, e.g., x = "*I love school*" satisfies e = "*loves school.*" We curated a list of forbidden output tokens for each explanation e by hand to prevent this.

We created negative samples for e by using positive samples for other explanations. If e is highly specific, e.g., "*talks about microwaves*," a random sample is unlikely to satisfy it. Therefore, we treat the positive samples of any other explanations as the negative samples for e. However, for e like "*uses past tense*", a random sample can satisfy it with non-trivial probability. Therefore, we wrote contrast explanations such as "*uses future tense*" and used their positive samples as the negative samples for e. Hence, our pool expanded to 352 explanations after including newly written ones, and we asked GPT-3 to generate 15 positive samples for each explanation.

Validating with Human Annotations. Some samples x from the conditional generation step do not actually satisfy the explanation e. To filter out samples that fail, for each (e, x) pair, we recruited Mechanical Turk workers (turkers) [2] to validate whether x satisfies e, as visualized in the 5th row of Figure 2.4. We collected three annotations for each (e, x) pair and determined the ground truth by majority vote. Finally, for each e, if fewer than five x passed the turker vote, the authors wrote additional examples by hand.

Thus, for each of the initial 302 explanations, we obtained at least five positive and five negative samples for it. We next use these to fine-tune our models.

2.2.4 Fine-tuning

Proposer. For each of the 302 explanations e, we fine-tuned GPT-3 to generate e based on five positive and five negative samples. We used a batch size of 20 and a small learning rate of 0.05 to prevent memorizing the target. We fine-tuned for two epochs, each using a different set of subsamples to construct the prompt.

Validator. Given e and a positive/negative sample x_1/x_0, our validator should predict that $V(e, x_1, x_0) = 1$ and $V(e, x_0, x_1) = 0$. To create a fine-tuning dataset, we randomly sampled 30 positive-negative pairs of (x_1, x_0) for each e. We fine-tuned UnifiedQA on this dataset for 250 steps with batch size 32 and learning rate 5e-5. To improve out-of-distribution robustness, we averaged the fine-tuned model weights with UnifiedQA [110].

2.3 Benchmarking Performance

On a benchmark of 54 real-world binary classification tasks, we show that: 1) both re-ranking and fine-tuning are effective, and 2) larger proposers and validators perform better.

Dataset. The evaluation set from [124] aggregated 54 diverse binary text classification tasks, each annotated with one or more[3] natural language explanations e^* for the positive class. These tasks include topic classification, grammaticality classification, stance classification, etc. For each task, we asked our systems to describe how the positive class samples differ from the negative class samples and compared the top-5 explanations with the human annotations.

For now, we assume that the annotations e^* are "correct" (i.e., the best explanations to separate the positive and negative classes). We will see later that our outputs are sometimes better than e^*.

[2] We recruited turkers located in the U.S. with $>$ 98% HIT acceptance rate and paid them \$0.04 per HIT; we estimate our pay rate to be \$18/hr based on how fast the authors perform this task.

[3] On average 2.2.

Evaluated Systems. We conjectured that using a larger proposer, a fine-tuned proposer, and a validator for re-ranking would all improve the generated explanations. Therefore, we evaluated the following five systems, which all use the validator from Section 2.2.4 unless otherwise noted. ①: our hypothetically best system, which uses the fine-tuned GPT-3 Davinci (175B) as the proposer. ②: a smaller proposer size (fine-tuned Curie, 13B). ③: no fine-tuning (zero-shot Curie, 13B). ④: no fine-tuning (zero-shot Curie, 13B), and no validator for re-ranking. We also evaluated ⑤, a "memorization proposer", where the proposer only generates the explanations we curated in Section 2.2.3; this ablation ensures that the fine-tuned proposer's performance is not simply due to memorizing its training set. If all our conjectures hold, we should find that ① > ② > ③ > ④ and ② > ⑤.

Automatic Evaluation. We first used an automatic metric BERTscore [119] to evaluate our systems, which approximates the similarity between two natural language texts. For each binary task, we computed the BERTscore between every pair of human annotations and the top-5 explanations; then, we chose the highest score among all pairs and averaged it across 54 tasks.

Using this metric, we indeed found that ① (0.930) > ② (0.927) > ③ (0.907) > ④ (0.899), and ② (0.927) > ⑤ (0.916), which validated our conjectures. However, all these numbers are high, the differences are small, and it is hard to interpret what they imply for the quality of our explanations.[4] Therefore, we additionally evaluated our systems by hand.

Manual Evaluation. We evaluated the top-5 explanations generated by each of the five systems on the 54 binary tasks (total 1350) by hand. To avoid biases against any of the five systems, the authors were blind to which system generated each explanation. We compared the systems' generated explanations $\hat{e}$ to human annotations e^* and rated their similarity with four levels:

(A), if $\hat{e}$ has mostly the same meaning as one of the human annotations e^*; e.g., "*is related to sports*" = "*is about sports.*"

(B), if $\hat{e}$ is close but different; e.g., "*is about sports team*" ≈ "*is about sports.*"

(C), if $\hat{e}$ is highly correlated but has a different meaning; for example, "*people need shelter*" is correlated with "*there is an earthquake.*"

(D), if $\hat{e}$ is unrelated to e^*.

For each system, we found the highest rating among the top-5 explanations and counted them across 54 tasks. We found that for row (A), ① > ② > ③ > ④ and ② > ⑤, validating our conjectures. Summing the counts from rows (A) and (B), we found that while GPT-3

[4] Appendix A.1.1 runs a sanity check to ensure that the scores, though not very informative, robustly rank system ① over ④.

	① best	② smaller	③ no tune	④ no validator	⑤ memo
(A)	31	22	11	4	5
(B)	10	11	6	0	5
(C)	7	10	10	6	21
(D)	6	11	27	44	23

Table 2.1: We evaluated each of the five systems as described in Section 2.3. ① largest fine-tuned proposer + validator, ② smaller proposer size, ③ no fine-tuning, ④ no re-ranking, and ⑤ using the memorization proposer. Better systems have larger numbers in row (A). Using a larger proposer, a fine-tuned proposer, and a validator all improve the generated explanations. We report the p values in Appendix A.1.2.

Curie (13B) only generates an explanation close to human annotation 7% of the time, the performance reaches 61% with fine-tuning and re-ranking, and our best system using GPT-3 Davinci (175B) reaches 76%. In the appendix, we also present the top-1 performance of our system in Table A.1, example human annotations, explanations by our systems, and their ratings in Table A.2.

Due to resource constraints, we did not systematically investigate whether the validator is still effective after fine-tuning. Nevertheless, our qualitative analyses found that the fine-tuned proposer sometimes still generates completely unrelated explanations, repeats explanations from the training set, or "rants"[5] based on a specific text sample. The validator helps rule these out. Finally, the proposer has a limited context size and can only generate explanations conditioned on five samples, losing information about the entire distribution; the validator does not have this fundamental limitation.

Comparing Validators. We next evaluate different choices of the validator. To test a validator, we check whether it can reliably separate the two classes when given the gold annotation e^*. More precisely, we compute

$$\frac{1}{2}\mathbb{E}_{x_0\sim D_0, x_1\sim D_1}[V(e^*, x_1, x_0) - V(e^*, x_0, x_1) + 1], \tag{2.6}$$

which is equivalent to the classification accuracy $\textsc{PredPower}(e^*)$ defined earlier.

We conjectured that larger and fine-tuned validators are better, so we compared our fine-tuned validator in Section 2.2.4 with smaller ones and UnifiedQA out of the box, averaging $\textsc{PredPower}(e^*)$ across all 54 tasks. Figure 2.5 visualizes the results. UnifiedQA performs decently, while additional fine-tuning improves the performance. Still, $\textsc{PredPower}(e)$ remains well below 1, suggesting that re-ranking is imperfect and that automatic evaluation via approximating $\textsc{PredPower}(e)$ may not yet be feasible. Nevertheless, these problems

[5]E.g., "*contains the word "turned", which indicates that the weather turned to a certain state*"

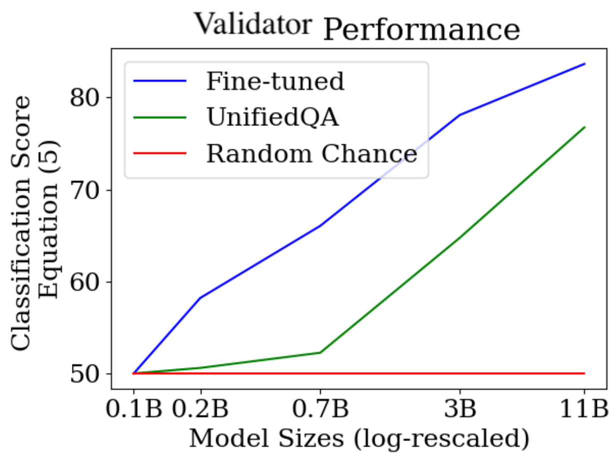

Figure 2.5: We compared validators of various sizes and UnifiedQA out of the box by evaluating their binary classification performance, using the metric PredPower(e^*) explained in Equation (2.6). We found that fine-tuning and larger model sizes improve the performance.

may be alleviated in the future: the current state-of-the-art models are at least 25x larger than our validator [89], and the curve in Figure 2.5 predicts that their performance will be higher.

2.4 Application

We applied our system to summarize training tasks, debug dataset shortcuts, describe distribution shifts, and label text clusters. All italicized quotes in this section are verbatim generations from our system.

Summarizing Training Tasks. Human explanations can be imperfect even for widely-used binary classification datasets. For example, the subjectivity analysis (SUBJ) dataset [83] was proposed as classifying between subjective vs. objective texts, and several works [15, 124, 37, 73] have used it to test zero/few-shot subjectivity classification. However, our system generates explanations "*is a plot summary of a film*" for the "objective" class and "*is a quote from a film review*" for the "subjective" class. We therefore re-read [83] carefully, which states (edited for brevity)

> To gather subjective sentences, we collected 5000 movie review snippets from www.rottentomatoes.com. To obtain (mostly) objective data, we took 5000 sentences from plot summaries available from www.imdb.com.

Therefore, our system's explanations were in fact more accurate. We conjecture that similar problems will become increasingly prevalent as the trend of aggregating datasets continues

[74, 93]: as datasets come from heterogeneous sources, it is a management challenge to characterize the task of every dataset accurately. Our system may help here.[6]

Debugging Dataset Shortcuts. Datasets frequently contain unintended shortcuts. For example, the task of natural language inference (NLI) is to validate whether a `explanation`[7] is an entailment or a contradiction given a premise. The popular MNLI [108] dataset contains a spurious correlation between contradictions and negations ("not", "never", etc.), and some models learn to predict a contradiction whenever these expressions occur, regardless of the premise [44].

If we know what shortcuts are present, we can apply fixes like group DRO [91]. But how do we find them in the first place? We used our system to look for (alternative) explanations of the differences between the two classes. We fed the `explanations` from the entailment class and those from the contradiction class to our system, which responded with "*contains a negative statement*" and "*has a negative verb*," revealing the spurious shortcut.

We also applied our system to a popular spam classification dataset [41]. We fed sentences from the two classes to our system, which tells us that the spam group "*has a higher number of hyperlinks*." To test whether such URLs influence downstream classifiers, we fed ten of our research communication messages with URLs to a RoBERTa-Large [65] model fine-tuned on this dataset (99% in-distribution accuracy). All 10 messages with URLs were classified as spam and were all classified as non-spam after removing the URLs.

Describing Distribution Shifts. We applied our system to describe distribution shifts for natural language tasks. For example, in contrast to MNLI, the SNLI dataset [14] is based on image captions; therefore, our system says that SNLI "*describes a picture*." [76] constructed another NLI dataset to stress test models' numerical reasoning ability; therefore, our system says that it "*contains a higher number of number words*." As another example, TwitterPPDB [60] and QQP[8] are both paraphrase detection datasets; the former is constructed by tweets while the latter is constructed by Quora questions; therefore, the system says that the former "*talks about a news story more*" while the latter "*contains a question*."

Labeling Text Clusters. Unsupervised algorithms generate semantically meaningful text clusters; however, researchers usually need to manually examine each of them to identify its semantics [21]. Our system can automatically describe a text cluster by treating it as D_1 and all others as D_0.

We compared our system to an expert on their ability to describe clusters. To create the clusters, we used RoBERTa-Base to embed the test set of wikitext-2 [71] (9992 sentences) and followed the approach of [3] to create 64 clusters. We randomly selected ten of them for evaluation; for each of them, one of our authors read through 20 samples and wrote a natural language explanation e^*; we then asked him to read the top-5 explanations by our system

[6] Of course, if our system can already perfectly validate the dataset explanations by performing the task, then we might not need those datasets for training in the first place. However, even an imperfect AI system can help correct some human mistakes.

[7] This is an NLI-specific concept; we use a special font to distinguish it from "explanation" (Section 2.1) in our paper.

[8] https://www.kaggle.com/c/quora-question-pairs

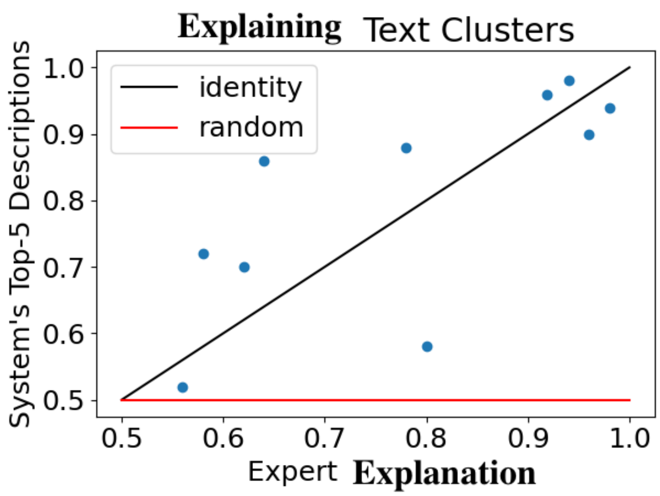

Figure 2.6: For each text cluster (dot), we collect human annotations to compute PredPower(e) for the explanations by our expert (x-axis) and the top-5 by our system (y-axis). Our system is on par with the expert most of the time.

and pick the one $\hat{e}$ that he considered to be the best. We evaluated this author's performance by PredPower(e^*) and our system's performance by PredPower($\hat{e}$), where we collected MTurk annotations to compute $[\![e]\!](x_0, x_1)$.

Averaged across all clusters, our system achieves PredPower=0.8 while the expert achieves 0.77. Figure 2.6 shows the results for each cluster, and we found that our system performed at least on par with the expert in most cases.

Discussion. In all the above applications, our system serves only to inform stakeholder decisions. Ultimately, it is up to the stakeholders to determine whether subjectivity can be approximated as "being review-like," whether specific correlations are bugs, or whether a distribution shift is severe enough to warrant intervention.

Our system also needs to improve to handle these applications robustly. For example, in the SPAM classification application, our validator cannot validate whether a hyperlink exists as reliably as a rule-based classifier, while the 16x larger proposer does the heavy lifting. We hope scaling up can alleviate this problem in the future.

Chapter 3

Models Parameterized by Language

The previous chapter focused on explaining differences between datasets. Beyond this, many other forms of explaining datasets are also important, including clustering and time series analysis. For example, explaining categories of Google search queries can discover public concerns or political opinions, such as interest in COVID-19 symptoms or upcoming elections.

To explain such patterns, existing methods typically learn a statistical model and interpret its parameters. A common approach for clustering is to embed the text, group the embeddings into clusters, and examine representative samples from each cluster. The hope is that each cluster corresponds to a coherent and interpretable category—such as "*asks about COVID symptoms*" or "*discusses the U.S. Election.*" However, in practice, clusters often contain a mix of unrelated or opaque queries, making it difficult to extract meaningful explanations.

Such a failure is not an isolated incident. Many models explain datasets by learning high-dimensional parameters, but these parameters might require significant human effort to interpret. For example, BERTopic [43] learns uninterpretable cluster centers over high-dimensional neural embeddings. LDA [13], Dynamic Topic Modeling [12] (time series), and Naive Bayes (classification) learn weights over a large set of words/phrases, which do not directly explain abstract concepts [21, 104, 127]. We want model parameters that are more interpretable, since explaining datasets is important in machine learning [123], business [10], political discussion [97], and science [42, 77].

To explain dataset patterns better, we introduce a family of models with parameters that are represented as natural language explanations,[1] which are inherently interpretable. Our core insight is that we can use an explanation to extract a 0/1 feature by checking whether it is true for a sample.[2] For instance, given the explanation $e =$ "*discusses the U.S. Election*", its denotation $[\![e]\!]$ is a binary function that evaluates to 1 on texts x discussing the U.S. Election and 0 otherwise:

$$[\![e : \text{``\textit{discusses the U.S. Election}''}]\!](x : \text{``\textit{Is Georgia a swing state this year?}''}) = 1.$$

[1] Specifically, natural language predicates

[2] In this chapter, the denotation is always approximated by a language model without a human-in-the-loop.

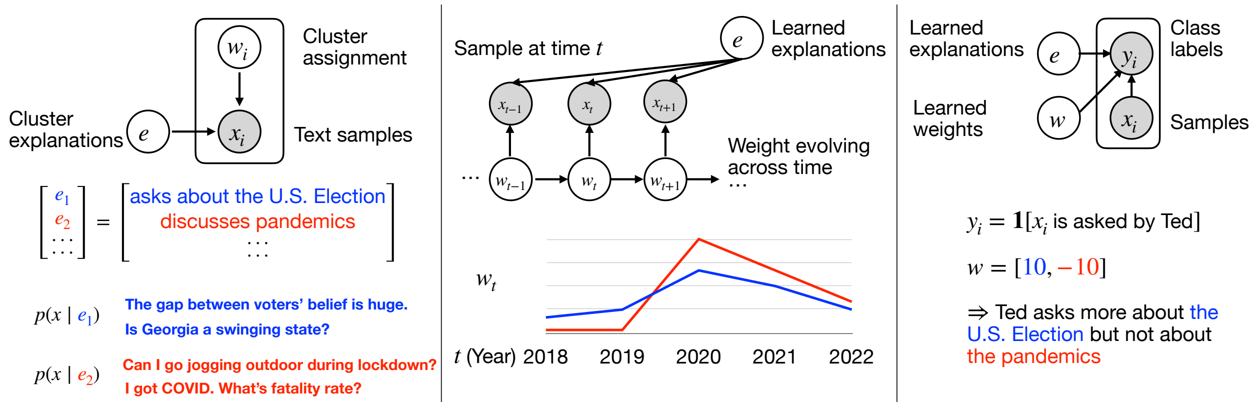

Figure 3.1: Our framework can use **natural language explanations** to parameterize a wide range of statistical models. **Left.** A clustering model that categorizes user queries. **Middle.** A time series model that characterizes how discussion changes across time. **Right.** A classification model that summarizes user traits. Once we define the model, we learn e and w based on x (and y).

Using these 0/1 feature values, we define a wide variety of models, including clustering, classification, and time series modeling, all parameterized by natural language explanations (Figure 3.1). Just like traditional statistical models, these models parameterized by natural language explanations will be evaluated based on their predictive power — the log-likelihood of the dataset.

Learning these explanations e requires optimizing the log-likelihood. This is challenging because e are discrete and thus do not admit gradient-based optimization. We propose a general method to effectively optimize e: we create a continuous relaxation $\tilde{e}$ of e and optimize $\tilde{e}$ with gradient descent; then we prompt an LLM to explain the behavior of $\tilde{e}$, thus converting it back to discrete explanations (Section 3.2). We repeat this process to iteratively improve performance.

To evaluate our optimization algorithm, we create statistical modeling problems where the optimal natural language parameters are known, so we can use them as the ground truth. We evaluated on three different statistical models (clustering, multilabel classification, and time series modeling, as illustrated in Figure 3.1) and used five different datasets (NYT articles, AG-News, DBPedia, Bills, and Wiki [92, 121, 52]). We found that both continuous relaxation and iterative refinement improve performance; additionally, our model-agnostic algorithm matches the performance (2% increase in F1 score) of the previous algorithm specialized for explainable text clustering [105].

Finally, we show that our framework is highly versatile by applying it to a wide range of tasks: taxonomizing user chat dialogues [122], characterizing how they evolve, finding categories where one language model is better than another, clustering math problems [47] based on their subareas, and explaining what visual features make an image memorable [53].

Our framework applies to both text and visual domains, can be easily steered to explain specific abstract properties, and explains complicated concepts that classical methods (e.g., n-gram regression/topic model) struggle to produce. Combining LLMs' ability to generate explanations along with traditional statistical models' ability to process sophisticated data patterns, our framework holds the promise to help humans better understand the complex world.

3.1 Defining Models with Natural Language Parameters

3.1.1 Preliminaries: Explanation-Conditioned Distribution

In order to model text distributions with natural language parameters, we introduce a new family of distributions, *explanation-conditioned distributions*; these distributions will serve as building blocks for the models introduced later, just like normal distributions are building blocks for many classical models like Gaussian Mixture or Kalman Filter. Explanation-conditioned distributions p are supported on the set X of all the text samples we observe from the dataset, and they are parameterized by (1) a list of K explanations $\vec{e} \in e^K$, and (2) real-valued weights $w \in \mathbb{R}^K$ on those explanations. Formally,

$$p(x \mid \vec{e}, w) \propto e^{w^T [\![\vec{e}]\!](x)}. \tag{3.1}$$

We now explain how to (1) extract a feature vector from x using $\vec{e}$, (2) linearly combine $\vec{e}$ by re-weighting with w, and (3) use the reweighted values to define $p(x \mid w, \vec{e})$.

Natural Language Parameters $\vec{e}$. Each explanation $e \in \mathcal{E}$ is a natural language string and its denotation $[\![e]\!] : X \to \{0, 1\}$ maps samples to their value under the explanation. For example, if e = "*is sports-related*", then $[\![e]\!]$("*I love soccer.*")= 1. Since a model typically requires multiple features to explain the data, we consider vectors $\vec{e} \in e_K$ of K explanations, where now $[\![\vec{e}]\!]$ maps X to $\{0, 1\}^K$:

$$[\![\vec{e}]\!](x) := \big([\![e_1]\!](x), [\![e_2]\!](x), \ldots, [\![e_K]\!](x)\big). \tag{3.2}$$

To instantiate $[\![\cdot]\!]$ computationally, we prompt a language model to check whether e is true on the input x, following the practice from prior works [125, 126]. See Figure 3.2 (left) for the prompt we used.

Reweighting with w. Consider the following example:

$$w = [-5, 3]; \quad \vec{e} = [\text{"is in English"}, \text{"is sports-related"}]. \tag{3.3}$$

Then $w^T[\![\vec{e}]\!]$ has a value of $-5 \cdot 1 + 3 \cdot 0 = -5$ for an English, non-sports related sample x. More generally, $w^T[\![\vec{e}]\!](x)$ is larger for non-English sports-related samples.

Defining $p(x \mid \vec{e}, w)$. According to Equation 3.1, $p(x \mid \vec{e}, w)$ is a distribution over X, all the text samples we observe, but it puts more weight on x with higher values of $w^T[\![\vec{e}]\!](x)$. Using the example w and $\vec{e}$ above, $p(x \mid \vec{e}, w)$ has higher probability for non-English sports-related texts.

Finally, we define $U(x)$ as the uniform distribution over X for later use.

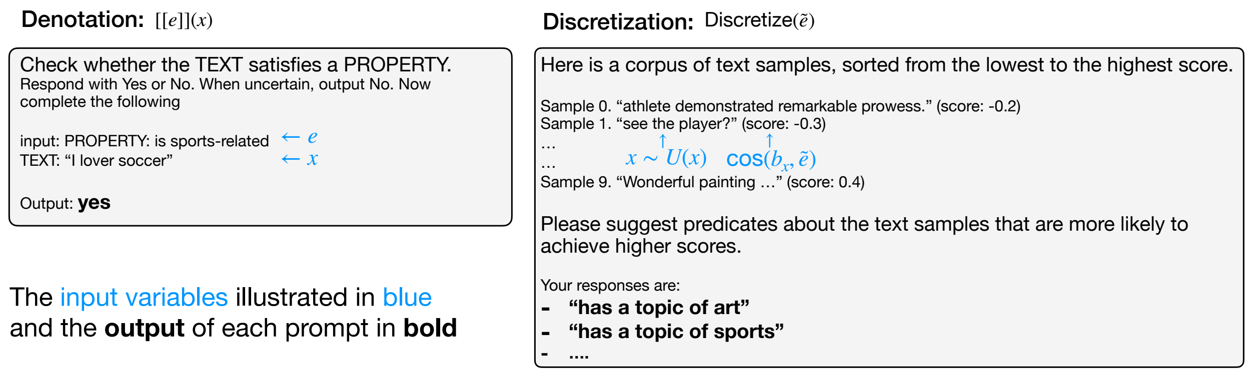

Figure 3.2: **Left.** The prompt to compute $[\![e]\!](x)$. **Right.** The prompt to `Discretize` $\tilde{e}_k$, which generates a set of candidate explanations based on samples x from U and their scores $\cos(m_x, \tilde{e}_k)$.

3.1.2 Evaluating Models Parameterized by Language Explanations

We introduce three models parameterized by natural language explanations: clustering, time series, and multi-label classification. Similar to traditional statistical models, the evaluation of the learned parameters $\vec{e}$ and w will be determined by their predictive power for the data — in other words, the log-likelihood for the dataset X, which we denote as PREDPOWER in this chapter.

Clustering. This model aims to help humans explore a large corpus by creating clusters, each explained by a natural language string. Such a model may help humans obtain a quick overview for a large set of machine learning inputs [123], policy discussions [97], or business reviews [10]. Given a set of text X, our model produces a set of K clusters, each parameterized by a learned explanation e_k; for example, if the explanation is "*discusses the U.S. Election*", then the corresponding cluster is a uniform distribution over all samples in X that discuss the U.S. Election.

Similar to K-means clustering, each sample x is assigned to a unique cluster. We use a one-hot basis vector $b_x \in \mathbb{R}^K$ to indicate the cluster assignment of x, and set $w_x = \tau \cdot b_x$, where τ has a large value (e.g., 10). We maximize the total log-likelihood:

$$\textsc{PredPower}(\vec{e}, w) = \sum_{x \in X} \log(p(x \mid \vec{e}, w_x)); \tag{3.4}$$

$$w_x = \tau \cdot b_x, \text{ where } \tau \to \infty \text{ and } b_x \text{ is a basis vector.}$$

However, some samples might not belong to any cluster and thus have 0 probability; to prevent negatively infinite PREDPOWER, we add another "background cluster" $U(x)$ that is uniform over all samples in X; therefore, each sample x can back off to this cluster and incur an additive term of at least $\log U(x) = -\log(|X|)$ to PREDPOWER.

Time Series Modeling. This model aims to explain latent variations in texts that change across time; for example, finding that an increasing number of people "search about flu symptoms" (e) can help us forecast a potential outbreak [40]. Formally, the input is a sequence of T text samples $X = \{x_t\}_{t=1}^{T}$. Our model produces K explanations e_k that capture the principal axes of variation in x across time. We model $w_1 \dots w_T$ as being drawn from a Brownian motion, i.e.,

$$p(x_t \mid \vec{e}, w_t) \propto \exp(w_t^\top [\![\vec{e}]\!](x)); \quad w_t := w_{t-1} + \mathcal{N}(0, \lambda^{-1} I), \tag{3.5}$$

where λ is a real-valued hyper-parameter that regularizes how fast w can change. The log-likelihood PredPower is hence

$$\textsc{PredPower}(\vec{e}, w) = \sum_{t=1}^{T} \log(p(x_t \mid \vec{e}, w_t)) - \frac{\lambda}{2} \sum_{t=1}^{T-1} ||w_t - w_{t+1}||_2^2. \tag{3.6}$$

Multiclass Classification with Learned Feature Explanations. This model aims to explain the decision boundary between groups of texts, e.g., explaining what features are more correlated with the fake news class [78] compared to other news, or explaining what activates a neuron [11]. Suppose there are C classes in total; the dataset is a set of samples x_i each associated with a class $y_i \in [C]$. Our model is hence a linear logistic regression model on the feature vectors extracted by $\vec{e}$, i.e.,

$$\text{logits}(x_i) = W \cdot [\![\vec{e}]\!](x_i); \quad \textsc{PredPower}(\vec{e}, W) = \sum_i \log(\frac{e^{\text{logits}(x_i)_{y_i}}}{\sum_{c=1}^{C} e^{\text{logits}(x_i)_c}}), \tag{3.7}$$

where $W \in \mathbb{R}^{C \times K}$ is the weight matrix for logistic regression.

3.2 Method

We can now learn the parameters for each model above by maximizing the log-likelihood PredPower. Formally,

$$\hat{\vec{e}}, \hat{w} = \text{argmin}_{\vec{e} \in e^K, w} \textsc{PredPower}(\vec{e}, w). \tag{3.8}$$

However, optimizing $\vec{e}$ is challenging, since it is discrete and therefore cannot be directly optimized by gradient-based methods. To address this challenge, we develop a general optimization method, which we describe at a high level in Section 3.2.1, introduce its individual components in Section 3.2.2, and explain our full algorithm in Section 3.2.3.

3.2.1 High-Level Overview

Our framework pieces together three core functions that require minimal model-specific design:

1. `OptW`, which optimizes w.

2. `OptRelaxedE`, which optimizes a continuous relaxation $\tilde{e}_k$ for each explanation e_k.
3. `Discretize`, which maps from continuous explanation $\tilde{e}_k$ to a list of candidate explanations.

Using these three components, our overall method initializes the set of explanations by first optimizing w and $\tilde{e}$ using `OptW` and `OptRelaxedE` and then discretizing $\tilde{e}$ with `Discretize`. To further improve the log-likelihood, it then iteratively removes the least useful explanation, re-optimizes its continuous representation, and discretizes it back to a natural language explanation.

To provide more intuition for these three components, we explain what they should achieve in the context of clustering. `OptW` should optimize the 1-hot choice vectors w_x by assigning each text sample to the cluster with maximum likelihood. `OptRelaxedE` should find a continuous cluster representation $\tilde{e}_k$ similar to the sample embeddings assigned to this cluster, and `Discretize` generates candidate explanations that explain which samples' embeddings are similar to $\tilde{e}_k$. Next, we introduce these three components formally for general models with natural language parameters.

3.2.2 Three Components of Our Framework

`OptW` optimizes w while fixing the values of $\vec{e}$. Formally, $\texttt{OptW}(\vec{e}) := \text{argmin}_w \textsc{PredPower}(\vec{e}, w)$.

This function needs to be designed by the user for every new model, but it is generally straightforward: in the clustering model, it corresponds to finding the cluster that assigns the highest probability for each sample; in classification, it corresponds to learning a logistic regression model; in the time series model, the PredPower is convex with respect to w and hence can be optimized via gradient descent.

For later use, we define the fitness of a list of explanations $\vec{e}$ as the PredPower after w is optimized:

$$\texttt{Fitness}(\vec{e}) := -\textsc{PredPower}(\vec{e}, \texttt{OptW}(\vec{e})). \tag{3.9}$$

Next, we discuss `OptRelaxedE`. The parameters $\vec{e}$ are discrete strings, so the PredPower is not differentiable with respect to $\vec{e}$. To address this, we approximate $[\![\vec{e}]\!](x)$ with the dot product of two continuous vectors, $\tilde{e}_k \cdot m_x$, where $m_x \in \mathbb{R}^d$ is a feature embedding of x normalized to unit length (e.g., the last-layer activations of some neural network), and $\tilde{e}_k \in \mathbb{R}^d$ is a unit-length, continuous relaxation of e_k. Intuitively, if the optimal $e =$ "*is sports-related*" and x is a sports-related sample with $[\![e]\!](x) = 1$, then we hope that $\tilde{e}$ would correspond to the latent direction encoding the sports topic and have high similarity with the embedding m_x of x. Under this relaxation, PredPower becomes differentiable with respect to $\tilde{e}_k$ and can be optimized with gradient descent.

Formally, `OptRelaxedE` optimizes all continuous explanations $\tilde{e}_{1\ldots K}$ given a fixed value of w:

$$\texttt{OptRelaxedE}(w) = \text{argmin}_{\tilde{e}_{1:K}} \textsc{PredPower}(\tilde{e} \mid w). \tag{3.10}$$

We sometimes also use it to optimize a single continuous explanation $\tilde{e_k}$ given a fixed w and all discrete natural language parameters other than e_k (denoted as e_{-k}):

$$\texttt{OptRelaxedE}(e_{-k}, w) = \text{argmin}_{\tilde{e}_k} \textsc{PredPower}(\tilde{e}_k | e_{-k}, w). \tag{3.11}$$

Finally, `Discretize` converts $\tilde{e}_k$ into a list of M discrete candidate explanations to update the variable e_k. Our goal is to find e whose denotation is highly correlated with the dot product simulation $\tilde{e}_k \cdot m_x$.

To discretize $\tilde{e}_k$, we prompt a language model to generate several candidate explanations and then re-rank them. Concretely, we draw samples $x \sim U(x)$[3] and sort them based on their dot product $\tilde{e_k} \cdot e_x$. We then prompt a language model with these sorted samples and ask it to generate candidate explanations that can explain what types of samples are more likely to appear later in the sorted list (Figure 3.2 bottom). To filter out unpromising explanations, we re-rank them based on the Pearson correlation between $[\![e]\!]$ and $\tilde{e_k} \cdot m_x$ on U if w cannot be negative (e.g., clustering), and the absolute value of Pearson correlation otherwise. We then keep the top-M explanations.

3.2.3 Piecing the Three Components Together

Our algorithm has two stages: we first initialize all the natural language parameters and then iteratively refine each of them. During initialization, we

1. randomly initialize continuous explanations $\tilde{e}$ to be the embedding of random samples from X
2. optimize $\textsc{PredPower}(\tilde{e}, w)$ by alternately optimizing w and all the continuous explanations $\tilde{e}$ with `OptW` and `OptRelaxedE`, and
3. set e_k as the first candidate from `Discretize`$(\tilde{e}_k)$

During refinement, we repeat the following steps for S iterations:

1. find the least useful explanation e_k; we define the usefulness of e_k as how much the fitness would decrease if we zero it out, i.e., $-\texttt{Fitness}(\vec{e}_{-k}, 0)$.
2. optimize $\tilde{e_k}$ using `OptRelaxedE` and choose the fittest explanation from `Discretize`$(\tilde{e}_k)$

We include a formal description of our algorithm in Appendix Algorithm 1.

3.3 Experiments

In this section, we benchmark our algorithm from Section 3.2; we later apply it to open-ended applications in Section 3.4. We run our algorithm on datasets where we know the ground truth explanations $\vec{e}$ and evaluate whether it can recover them. On five datasets and three statistical models, continuous relaxation and iterative refinement consistently improve performance. Our general method also matches a previous specialized method for explainable clustering [105].

[3] i.e., uniformly draw samples x from all samples we observe from the dataset

Reference	Size	Learned	Size	Surface	F1
"artist"	0.07	*"music"*	0.12	0.50	0.37
"animal"	0.07	*"a specific species of plant or animal"*	0.14	0.50	0.65
"book"	0.08	*"literary works"*	0.07	0.50	0.64
"politics"	0.06	*"a political figure"*	0.06	0.50	0.96
"plant"	0.07	*"a specific species of plant or animal"*	0.14	0.50	0.68
"company"	0.08	*"business and industry"*	0.07	0.50	0.83
"school"	0.06	*"schools"*	0.07	1.00	0.97
"athlete"	0.07	*"sports"*	0.07	0.50	0.98
"building"	0.08	*"historical buildings"*	0.08	0.50	0.92
"film"	0.06	*"film"*	0.07	1.00	0.91
...	...	...	...	...	...

Table 3.1: We compare the reference explanations and our learned explanations when clustering the `DBPedia` dataset. We abbreviate the explanations, e.g., *"art"* = *"has a topic of art"*. For each reference, we match it with the learned explanation that achieves the highest F1-score at predicting the reference denotation. We also report the surface similarity (defined in Section 3.3.2) between the learned explanation and the reference. Our learning algorithm mostly recovers the underlying reference explanations, though it sometimes learns larger/correlated clusters that disagree with the reference but are still meaningful.

3.3.1 Datasets

We design a suite of datasets for each of the three statistical models mentioned in Section 3.1.2. Each dataset has a set of reference explanations, and we evaluate our algorithm's ability to recover them.

Clustering. We consider five datasets, `AGNews`, `DBPedia`, `NYT`, `Bills`, and `Wiki` [92, 121, 52]. The datasets have 4/14/9/21/15 topic classes, each described by an explanation, and we sample 2048 examples from each for evaluation.

Multiclass Classification. We design a classification dataset with 5,000 articles and 20 classes; its goal is to evaluate a method's ability to recover the latent interpretable features useful for classification. Therefore, we design each class to be a set of articles that satisfy three explanations about its topic, location, and language; for example, one of the classes can be described by the explanations "*has a topic of sports*", "*is in Japan*", and "*is written in English*". We create this dataset by adapting the New York Times Articles dataset [92], where each article is associated with a topic and a location explanation; we then translate them into Spanish, French, and German. We consider in total $4 + 4 + 4 = 12$ different explanations for each of the topic/location/language attributes and subsample 20 classes from all $4 \times 4 \times 4 = 64$ combinations.

Time Series modeling. We synthesize a time series problem by further adapting the translated NYT dataset above. We set the total time $T = 2048$ and sample $x_1 \dots x_T$

according to the time series model in Section 3.1.2 to create the benchmark. We set $\vec{e}$ to be the 12 explanations mentioned above and the weight $w_{\cdot,k}$ for each explanation e_k to be a cosine function with a period of T to simulate how each attribute evolves throughout time. In addition, we included three simpler datasets where there is only variation on one attribute (i.e., varies only on one of topic/location/language). We name these four time series modeling datasets `all`, `topic`, `locat`, and `lang`, respectively. See Appendix A.2.2 for a more detailed explanation.

3.3.2 Metrics

To evaluate our algorithm, we match each learned explanation $\hat{e}_k$ with a reference $e^*_{k'}$, compute the F1-score and surface similarity for each pair, and then report the average across all pairs. To create the matching, we match $\hat{e}_k$ to the $e^*_{k'}$ with the highest overlap (number of samples where both are true); formally, we define a bipartite matching problem to match each explanation in $\hat{e}$ with one in e^*, define the weight of matching $e^*_{k'}$ and $e^*_{k'}$ to be their overlap, and then find the maximum weight matching via the Hungarian algorithm. We now explain the F1-score and surface similarity metrics.

F1-score Similarity. We compute the F1-score of using $\hat{e}(x)$ to predict $e^*(x)$ on X, the set of samples we observe. This is similar to the standard protocol for evaluating cluster quality [61].

For non-clustering tasks such as classification or time series, we set the dimension of the learned explanations $\hat{e}(x)$ to be smaller than the reference explanations e^*, because some of the reference explanations are linearly dependent: for example, if there are four different languages in total (English, German, Spanish, and French), not being one of the first three implies that it is the last. Therefore, we set the dimension of the learned explanations $\hat{e}$ to be the number of linearly independent explanations in e^*. To compute the F1-score similarity, for each of the reference explanations, we find the most similar learned explanation, calculate the F1-score, and finally aggregate it across the reference explanations.

Surface Form Similarity. We can also directly evaluate the similarity between two explanations based on their string values, e.g., "*is about sports*" is similar in meaning to "*has a topic of sports*", a metric previously used by [126]. For a pair of explanations, we ask `gpt-4` to evaluate whether they are similar in meaning, related, or irrelevant, with each option associated with a surface-similarity score of 1/0.5/0. We display the prompt in Figure A.3 and example ratings in Table 3.1.

3.3.3 Experiments on Our Benchmark

We now use these metrics and datasets to evaluate the optimization algorithm proposed in Section 3.2 and run ablations to investigate whether continuous relaxation and iterative refinement are effective. We will first introduce the overall experimental setup, and then discuss individual takeaways supported by experimental results in each paragraph.

Experimental Setup. When running the algorithm, we generate candidate explanations in `Discretize` with `gpt-3.5-turbo` [80]; to perform the denotation operation $[\![e]\!](x)$, we use `flan-t5-xl` [28]; we create the embedding for each sample x with the `Instructor-xl` model [100] and then normalize it with ℓ_2 norm. We set the number of candidates M returned by `Discretize` to be 5 and the number of optimization iterations S to be 10. To reduce noise due to randomness, we average the performance of five random seeds for each experiment.

Table 3.2 reports the results of clustering and Table 3.3 reports other results. For each dataset, we perform several ablation experiments and present the takeaways from these results.

Takeaway 0: Is our method better than naively prompting language models to generate explanations? How does our approach compare to a naive baseline approach, which directly prompts the language model to generate explanations based on dataset samples? For this baseline, we repeatedly prompt a language model to generate more explanations until we obtain K explanations, compute their denotation, evaluate them using the metrics in Section 3.3.2, and report the performance in Table 3.2 and A.4, the `Prompting` row. Across all entries, our approach significantly outperforms this baseline.

Takeaway 1: Relax + discretize is better than exploring randomly generated explanations. Our optimization algorithm explores the top-5 LLM-generated explanations that have the highest correlations with $\tilde{e_k} \cdot m_x$. Would choosing a random explanation be equally effective? To investigate this question, we experimented with a variant of our algorithm that randomly chooses five explanations without utilizing the continuous representation $\tilde{e_k}$ (`No-Relax`). In Table 3.2 and 3.3, `No-Relax` underperforms our full algorithm (`Ours`) in all cases. In Appendix Figure A.4, we plot the negative PredPower after each iteration averaged across all tasks, and we find that `Ours` converges much faster than `No-Relax`.

Takeaway 2: Iterative refinement improves the performance. We considered a variant of our algorithm that only discretizes the initial continuous representations and does not iteratively refine the explanations (`No-Refine`). In Table 3.2 and 3.3, `No-Refine` underperforms the full algorithm in all cases.

Takeaway 3: Our model-agnostic method is competitive with previous methods specialized for explainable clustering. We compare our method to `GoalEx` from [105], which designs a specialized method for explainable clustering based on integer linear programming. Even though our method is model-agnostic, it matches or outperforms `GoalEx` on four out of five datasets and improves F1 by 0.02 on average.

Takeaway 4: Our method accounts for information beyond the set of text samples (e.g., temporal correlations in the time series). We investigate this claim using the time series datasets, where we shuffle the text order and hence destroy the time-dependent information a model could use to extract informative explanations (`Shuffled`). Table 3.3 finds that `Ours` is better than `Shuffled` in all cases, indicating that our method does make use of temporal correlations.

Appendix A.2.4 includes additional results: 1) compared to topic modeling and K-means, our method achieves similar or better performance while being explainable; 2) we ran ablations

F1/Surface	AGNews	DBPedia	NYT	Bills	Wiki	Average
Prompting	0.43/0.60	0.31/0.44	0.21/0.40	0.16/0.47	0.22/0.34	0.27/0.45
No-Refine	0.72/0.57	0.57/0.52	0.54/0.58	0.34/0.49	0.47/0.51	0.53/0.54
No-Relax	0.86/0.60	0.59/0.53	0.58/0.53	0.31/0.51	0.46/0.50	0.56/0.54
Ours	**0.86**/**0.62**	0.68/0.54	**0.70**/**0.63**	**0.45**/**0.52**	**0.51**/**0.53**	**0.64**/**0.57**
GoalEx (Specialized)	**0.86**/**0.62**	**0.75**/**0.64**	0.68/**0.63**	0.33/0.50	0.49/0.48	0.62/**0.57**

Table 3.2: Results on clustering. `Ours` always outperforms `No-Refine` and `No-Relax`, indicating that both continuous relaxation and iterative refinement are helpful. Compared to `GoalEx` [105], our method is slightly better on all datasets except `DBPedia`, which we analyze in Table 3.1.

F1/Surface	topic	lang	locat	all	time-avg	classification
Prompting	0.40/0.35	0.39/0.38	0.26/0.30	0.54/0.57	0.40/0.40	0.51/0.42
No-Refine	0.53/0.53	0.39/0.50	0.37/0.55	0.58/0.44	0.47/0.50	0.58/0.44
No-Relax	0.65/0.50	0.52/0.65	0.48/**0.68**	0.61/0.56	0.56/0.60	0.68/0.62
Shuffled	0.46/0.33	0.52/0.45	0.33/0.28	0.60/0.39	0.47/0.35	N/A
Ours	**0.67**/**0.57**	**0.62**/**0.70**	**0.55**/**0.68**	**0.72**/**0.64**	**0.64**/**0.65**	**0.73**/**0.70**

Table 3.3: Our performance on time series (left) and classification (right). Both continuous relaxation and iterative refinement improve the performance (comparing `Ours` to `No-Refine` and `No-Relax`).

on the effect of neural embeddings and show that informative embeddings are crucial to good performance; 3) Takeaways 1, 2, and 4 are significant with $p < 1\%$ under paired t-tests.

3.4 Open-Ended Applications

We apply our framework to a broad range of applications to show that it is highly versatile. Our framework can monitor data streams (Section 3.4.1), apply to the visual domain (Section A.2.6.1), and be easily steered to explain specific abstract properties (Section A.2.6.2). Across all applications, our framework is able to explain complex concepts that classical methods struggle to produce.

3.4.1 Running Our Models Out of the Box: Monitoring Complex Data Streams of LLM Usage

We apply our models from Section 3.1.2 to monitor complex data streams of LLM usage. In particular, we recursively apply our clustering model to taxonomize user queries into application categories, apply our time series model to characterize trends in use cases across time, and apply our classification model to find categories where one LLM is better than

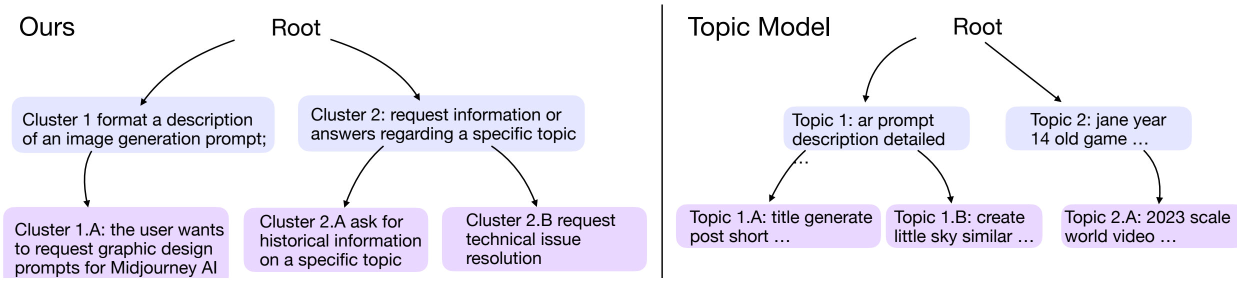

Figure 3.3: **Left.** We generate a taxonomy with sophisticated explanations by recursively applying our clustering model. **Right.** We cluster with topic models and present the top words for each topic. Although some topics are plausibly related to certain applications, they are still ambiguous.

another. Due to space constraints, we present the key results in the main paper and the full results in Appendix A.2.7.

Taxonomizing User Applications via Clustering. LLMs are general-purpose systems, and users might apply LLMs in ways unanticipated by the developers. If the developers can better understand how the LLMs are used, they could collect training data correspondingly, ban unforeseen harmful applications, or develop application-specific methods. However, the amount of user queries is too large for individual developers to process, so an automatically constructed taxonomy could be helpful.

We recursively apply our clustering model to user queries to the ChatGPT language model. We obtain the queries by extracting the first turns from the dialogues in WildChat [122], a corpus of 1M real-world user-ChatGPT dialogues. We use `gpt-4o` [81] to discretize and `claude-3.5-sonnet` [1] to compute denotations. We first generate $K = 6$ clusters on a subset of 2048 queries; then we generate $K = 4$ subclusters for each cluster with > 32 samples.

We present part of the taxonomy in Figure 3.3 (left) and contrast it with the taxonomy constructed by directly applying LDA recursively (right). Although some LDA topics are plausibly related to certain applications, they are still ambiguous; for example, it is unclear what topic 1 "*ar prompt description detailed*" means. After manually inspecting the samples associated with this topic, we found that they were related to the application of writing prompts for an image-generation model. In contrast, our framework can explain complicated concepts that are difficult to infer from individual words; for example, it generates "*requesting graphic design prompts*" for the above application, which is much clearer in its meaning when explained in natural language.

Characterizing Temporal Trends via Time Series Modeling. Understanding temporal trends in user queries can help forecast flu outbreaks [40], prevent self-reinforcing trends [45], or identify new application opportunities. We run our time series model on 1000 queries from WildChat with $K = 4$ to identify temporal trends in user applications, and report part of the results in Figure 3.4. Based on the blue curve, we find that an increasing number of users

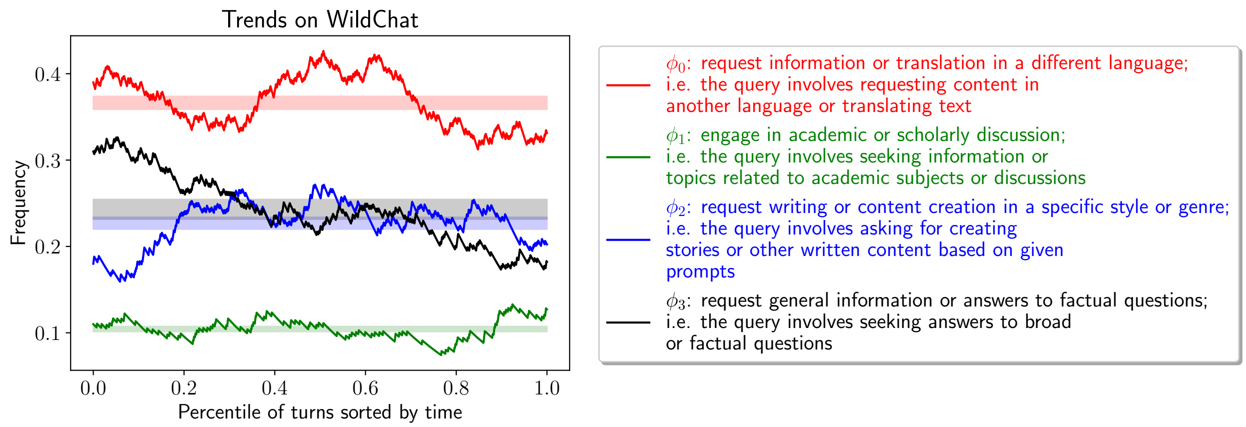

Figure 3.4: We analyze WildChat queries with our time series model. For each learned explanation, we plot how its frequency evolves and the 99% confidence interval of the average frequency (shaded).

"*request writing or content creation .... creating stories based on given prompts*." This helps motivate systems like Coauthor [62] to assist with this use case.

Finding Categories where One Language Model is Better than Another. One popular method to evaluate LLMs is crowd-sourcing: an evaluation platform (e.g. ChatBotArena [24]) or a company (e.g. OpenAI) accepts prompts from users, shows users responses from two different LLM systems, and the users indicate which one they like better. The ranking among the LLM systems is then determined by Elo-rating, i.e. how often they win against each other.

However, aggregate Elo-rating omits subtle differences between LLM systems. For example, LLama-3-70B achieved a similar rating as Claude-3-Opus, and the LLM community was excited that open-weight models were catching up. However, is LLama-3-70B similarly capable across all categories, or is it significantly better/worse under some categories? Such information is important for downstream developers, since some capabilities are more commercially valuable than others: e.g. a programmer usually does not care about an LLM's capability to write jokes. We need a more fine-grained comparison.

We directly apply the classification model from our framework to solve this task. To understand the categories where LLama-3-70B is better/worse than Claude-3-Opus, we gather user queries x from the ChatBotArena maintainers (personal communication), set $y = 1$ if the LLama-3-70B's response to x is preferred and $y = 0$ otherwise. We set $K = 3$.

Our model finds that LLama-3-70B is better when the query "*asks an open-ended or thought-provoking question*" but worse when it "*presents a technical question*" or "*contains code snippets*". These findings are corroborated by manual analysis by the ChatBotArena maintainers, who also found that Llama-3 is better at open-ended and creative tasks while worse at technical problems[4]. We hope that our model can automatically generate similar analyses in the future when a new LLM is released, thus saving researchers' efforts.

To summarize, our framework 1) enables us to define a time series model to explain temporal trends in natural language, and 2) outputs sophisticated explanations that LDA fails to generate. However, it is far from perfect: it is slow to compute denotations for all pairs of x and candidates e since it involves many LLM API calls, and the explanations themselves are sometimes redundant. We describe these limitations and potential ways to improve them in Appendix A.2.7.

Due to space constraints, we present applications in explaining visual features to make images memorable to humans and clustering math problems based on subareas in Appendix A.2.6.1 and A.2.6.2.

[4] https://lmsys.org/blog/2024-05-08-llama3/

Chapter 4

Explanations about Language Models

The previous two chapters focused on explaining datasets. But what about natural language explanations for a language model's (LM) decisions? We can frame this as another kind of dataset explanation: Given a dataset of LM inputs x and outputs y, can we generate an explanation e that enables humans to predict the LM's output from the input? This idea is rooted in classic work on mental models and explanation [54, 29, 38, 8].

In this chapter, we study how GPT-4 [81] explains its own decisions. For example, when we ask GPT-4 "*Is it hard to get a BLT in Casablanca?*", it answers "*yes*" and explains:

> "*Casablanca is a large city in Morocco. Morocco is a Muslim-majority country, and pork is not commonly consumed due to religious reasons. BLT contains bacon, which is pork. Thus, it might be hard to find a traditional BLT in Casablanca.*"

Such an explanation is logically coherent and provides factually correct background information helpful for the question [55].[1] However, does it help humans correctly predict how GPT-4 answers other related questions? Based on the explanation, humans will predict that GPT-4 encodes the knowledge that "pork is not commonly consumed in Muslim countries" and will apply similar reasoning to relevant questions (counterfactuals), e.g., answering "*Yes*" to "*Is it hard to find pork belly in Casablanca?*" Unfortunately, GPT-4 actually answers "*No*" to this counterfactual, contradicting its own explanation and humans' expectations.

The above explanation is problematic because humans form an incorrect mental model of GPT-4 (i.e., incorrectly predict how GPT-4 answers relevant counterfactuals) based on this explanation. Building a correct mental model of an AI system is important, as it helps humans understand what an AI system can and cannot achieve [20], which informs humans how to improve the system or appropriately deploy it without misuse or overtrust [18, 8, 116].

We propose to evaluate the **counterfactual simulatability** of natural language explanations: can an explanation e allow humans to simulate the model's decision on a counterfactual input x, and hence predict the model's decision y? We propose two metrics accordingly for

[1] The annotated answer is "yes" in StrategyQA, though it might not necessarily reflect the reality in Casablanca.

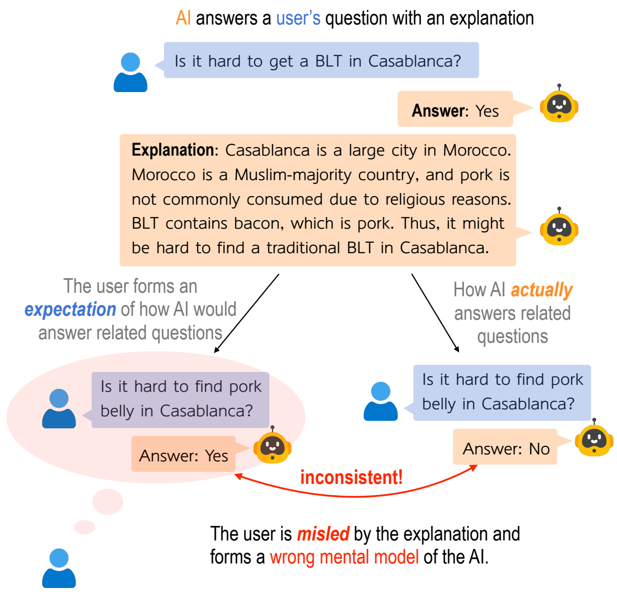

Figure 4.1: GPT-4 answers a human user's question and generates an explanation. In this example, what GPT-4 **actually** answers (right) is different from what the user would **expect** (left) based on the explanation. Therefore, the explanation misleads humans to form an incorrect mental model of GPT-4 even though it is factually correct.

explanations (Figure 4.2). The first, **simulation generality**, measures the generality of an explanation by tracking the diversity of the counterfactuals x relevant to the explanation e (e.g., "*Humans do not consume meat*" has more diverse relevant counterfactuals compared to "*Muslims do not consume pork*" and is thus more general). The second, **simulation precision**, tracks the fraction of counterfactuals where humans' simulation matches the model's output.

To evaluate the counterfactual simulatability of an explanation on an input question (e.g., the initial question about BLT), we need to (1) collect a dataset D of counterfactuals x on an input based on the explanation, and (2) have humans simulate what the model outputs on the counterfactuals ($[\![e]\!](x)$). For (1), since it is expensive to ask humans to write the counterfactuals, we propose to prompt LLMs to generate diverse counterfactuals relevant to an explanation (e.g., related questions about pork belly or pepperoni in Figure 4.2). For (2), since human simulation might be subjective, we reduce subjectivity by framing the simulation task as a logical entailment task (Section 4.2.4). Finally, we calculate generality and precision based on the LLM-generated counterfactuals and humans' simulation annotations.

We benchmark the counterfactual simulatability of two LLMs—GPT-3.5 and GPT-4—and two explanation methods—CoT (Chain of Thought) and Post-Hoc (explain after the output)—on two tasks—multi-hop factual reasoning (StrategyQA, [39]) and reward modeling (Stanford Human Preference, [35]). We found that (i) both LLMs' explanations have low precision (80(ii) CoT does not substantially outperform Post-Hoc.

We also study how counterfactual simulatability relates to plausibility, which evaluates humans' preference for an explanation based on its factual correctness and logical coherence. We found that precision does not correlate with plausibility, and hence naively optimizing human approvals (e.g., RLHF) might not fix the issue of low precision.

4.1 Evaluating Predictive Power

For a given task, a model M takes an input $x \in X$ and produces an output $o_x \in O$ and explanation e_x. The input, output and explanation are all in natural language. A human observes x, e_x, o_x, and forms a mental model $[\![x, e_x, o_x]\!] : X \to O \cup \{\perp\}$, where $[\![x, e_x, o_x]\!](x')$ denotes what the human simulates to be M's output on x' (**simulation**). If the human cannot simulate M's output for input x' based on x, e_x, o_x, then x' is **unsimulatable** and we denote $[\![x, e_x, o_x]\!](x') = \perp$. For simplicity, we use $[\![e_x]\!](x')$ to denote $[\![x, e_x, o_x]\!](x')$.

An ideal explanation e_x should be **generalizable**—beyond revealing how the model reasons on x, it should also reveal how the model reasons on unseen inputs $x' \neq x$. Explanations also need to be **precise**—they should lead to mental models that are consistent with the model's behavior.

Motivated by these two desiderata, we propose measuring counterfactual simulatability with two metrics: simulation generality and simulation precision. We introduce them below.

4.1.1 Simulation Generality

Conceptually, we want simulation generality to measure how diverse the simulatable counterfactuals are, so we measure it as one minus the average similarity between two simulatable counterfactuals:

$$\textsc{PredPower}_{\text{generality}} = 1 - \mathbb{E}_{x',x'' \sim p}[\alpha(x', x'')],$$

where p is the distribution of simulatable counterfactuals and α is a similarity metric. To define simulation generality, we need to specify p and α. For p, to evaluate an explanation e_x on an input x, we first prompt LLMs to generate n counterfactuals of x that are likely simulatable from e_x, denoted as $D_{e_x} = \{x'_1, \cdots, x'_n\}$. We then filter out the unsimulatable counterfactuals and get the simulatable subset $D^* = \{x' \in D, [\![e_x]\!](x') \neq \perp\}$. The expectation then becomes:

$$\textsc{PredPower}_{\text{generality}}(e_x) = 1 - \frac{1}{|D^*_{e_x}|(|D^*_{e_x}| - 1)} \sum_{x', x'' \in D^*_{e_x}, x' \neq x''} \alpha(x', x'').$$

See Figure 4.2 top for a concrete example.

For α, we consider three possibilities:

1. BLEU: $\alpha(x', x'') = \text{BLEU}(x', x'')$ [84]

2. Cosine: We embed x' and x'' separately with a sentence encoder Enc and calculate their cosine similarity:

$$\alpha(x', x'') = \cos(Enc(x'), Enc(x''))$$

3. Jaccard: We tokenize x' and x'' separately into two bags (sets) of words $bow(x')$ and $bow(x'')$, and remove stopwords. We then calculate the Jaccard similarity between them:

$$\alpha(x', x'') = \frac{|bow(x') \cap bow(x'')|}{|bow(x') \cup bow(x'')|}.$$

4.1.2 Simulation Precision

We measure simulation precision as the fraction of simulatable counterfactuals where humans' simulations match the model's actual outputs:

$$\textsc{PredPower}_{\text{precision}} = \frac{1}{|D^*|} \sum_{x' \in D^*} \mathbf{1}[[\![e_x]\!](x') = o_{x'}].$$

4.1.3 Implementing Human Simulation $[\![e_x]\!](x')$

In the definitions of generality and precision, we relied on human simulation $[\![e_x]\!](x')$, so our remaining task is to implement this function. There are several challenges to this, which we describe and address below.

Human simulation can be highly subjective. Different human annotators may use different reasoning to simulate what the model would output. Consider the following example from StrategyQA. For the input question "*Would a monkey outlive a human being on average?*", the model explains:

> "*The average lifespan of a monkey is about 20 years. The average lifespan of a human being is about 80 years. Thus, a monkey would not outlive a human being on average.*"

Given the counterfactual "*Can turtles outlive sharks?*", some annotators think it is simulatable because the explanation indicates that questions of the form "Can A outlive B?" can be answered by comparing the lifespans of A and B, while others think this counterfactual is not simulatable because the explanation does not mention the lifespans of turtles or sharks. Thus, we need to formulate human simulation as a well-defined task to reduce annotation noise.
Solution. We propose formulating human simulation as a logical entailment task to reduce subjectivity. We instruct annotators to simulate a model's output on x' by judging whether (e_x, o_x, x) entails an output to counterfactual x'. We allow humans to use commonsense reasoning when judging entailment, e.g., the explanation "*Omnivores can use chopsticks*"

entails the output "*yes*" to "*Can pigs use chopsticks?*" because pigs are omnivores. If the explanation does not entail any output, then this counterfactual is unsimulatable. For example, if the explanation is "*Omnivores can eat meat*", then the question "*Can pigs use chopsticks?*" is unsimulatable because the explanation is irrelevant.

Humans and models have different commonsense knowledge. When humans use commonsense knowledge to generalize mental models, it may differ from a model's generalization if they have different commonsense knowledge. For example, if a model "thinks" that pigs are not omnivores (different from humans' knowledge), then it may answer "*no*" to "*Can pigs use chopsticks?*" while being perfectly consistent with its explanation "*Omnivores can use chopsticks*." Should humans use their own knowledge or the model's knowledge when they generalize their mental models and judge entailment?
Solution. We argue that humans should use their own knowledge when judging entailment and generalizing mental models, because probing the model's knowledge for each counterfactual is time-consuming and difficult. Note that humans should adhere to the model's explanation whenever relevant (because the goal is to simulate the model's behavior), and only use human knowledge for information missing in the explanation.

Human simulation is expensive and laborious. Evaluating the counterfactual simulatability of one explanation requires humans to annotate *multiple* counterfactuals (Section 4.1.1), which is expensive.
Solution. To facilitate automatic evaluation, we also experiment with approximating human simulators using LLMs. Similar to human simulators, LLMs take as input a model's explanation e_x and output o_x on input x, and simulate the model's output on each counterfactual x'. We show the prompts we use in Appendix A.4.2. Note that even though the simulation process is now automated, unlike faithfulness evaluation, the gold simulators are still humans following the two rules above (judging simulation as *entailment* with *human* commonsense).

Final Solution Combining the solutions to the two challenges above, we instruct annotators to simulate a model's output on x' by judging whether (e_x, o_x, x) entails an output to counterfactual x', to adhere to the model's explanation whenever relevant, but to use human knowledge for information missing in the explanation. We present details of our human evaluation in Section 4.2.4. We evaluate the LLM simulators based on their agreement with human simulators (Section 4.3.1 Table 4.3).

4.2 Experiment Setup

We introduce the datasets we use (Section 4.2.1), the explanation systems we evaluate (Section 4.2.2), and additional details for counterfactual generation (Section 4.2.3) and human simulation (Section 4.2.4).

4.2.1 Datasets

We evaluate explanations on multi-hop reasoning (StrategyQA) and reward modeling (Stanford Human Preference).

StrategyQA is a multi-hop question-answering dataset on open-domain questions [39]. The answer to each question is either "yes" or "no". Answering questions in StrategyQA requires implicit step-by-step reasoning, which makes explanations useful.

Stanford Human Preference (SHP) is a human preference dataset over agent responses to users' questions and instructions [6]. Each input consists of a context post and two responses, and the task is to pick the preferred response. The explainability of reward models is crucial as biases and spurious correlations in the reward model may cascade to downstream generation models through RLHF [27, 82, 6, 33].

4.2.2 Explanation Systems

We evaluate the counterfactual simulatability of two LLM explanation methods: Chain-of-Thought and Post-Hoc, which differ in the order in which the LLM produces the output and the explanation.

In Chain-of-Thought (CoT), given an input x, the model first generates reasoning e_x, and then produces the output o_x conditioned on x and e_x [79, 107, 102]. In Post-Hoc, given an input x, the model first produces the output o_x, and then generates an explanation e_x conditioned on x and o_x [17, 86]. Because CoT generates the explanation before the output, we conjecture that CoT explanations are more likely to reveal the model's decision process and are intuitively more precise compared to Post-Hoc explanations.

We evaluate the counterfactual simulatability of two LLMs: GPT-3.5 (175B) [16, 82] and GPT-4 [81] to study how scaling affects counterfactual simulatability. We show the prompts we use in Appendix A.4.2.

4.2.3 Counterfactual Generation

We experiment with two counterfactual generators: GPT-3.5 (175B) and GPT-4. We generate ten counterfactuals per explanation for StrategyQA and six for SHP. We show the prompts we use to generate counterfactuals in Appendix A.4.2.

4.2.4 Human Simulation

We collected human simulation judgments for both StrategyQA and SHP on Amazon Mechanical Turk. We show the annotation instructions in Appendix A.4.1. We set up a qualification exam with 11 questions, where annotators needed to answer at least 9 questions correctly in order to do the actual annotations. The simulation task is complicated, so we communicated with the annotators promptly via Slack to answer any questions they had. We asked three annotators to simulate each counterfactual and observed moderate inter-annotator agreement

Dataset	Generator	BLEU	Cos	Jacc	Sim.%
SQA	GPT-3	69.6	24.6	61.0	62.7
	GPT-4	67.0	25.3	58.9	56.1
	GPT-mix	**72.9**	**29.6**	**66.2**	58.7
	PJ	43.6	15.1	33.6	55.9
SHP	GPT-mix	93.0	65.3	90.0	78.5

Table 4.1: LLM prompting generates more diverse simulatable counterfactuals compared to Polyjuice (p-value < 0.001 on all metrics). Mixing GPT-3 and GPT-4 outputs further improves diversity (p-value < 0.002). SQA: StrategyQA.

(IAA) on StrategyQA and fair IAA on SHP. We attribute the limited IAA to the subjectivity of the simulation task (Section 4.1.3).

4.3 Results

We first validate our evaluation procedure through several sanity checks (Section 4.3.1) before using our metrics to compare different explanation systems (Section 4.3.2).

4.3.1 Sanity Checks

We validate three aspects of our approach: (i) whether our evaluation procedure can meaningfully distinguish between explanation systems, (ii) whether LLM simulators serve as reliable proxies for human simulators, and (iii) whether our counterfactual generation method improves upon a baseline that ignores explanations.

Our evaluation procedure effectively discriminates between explanation systems. To verify that our method can detect meaningful differences in explanation quality, we compare a normal system against an intentionally degraded baseline. Specifically, we create a Forced system where we require the model to generate Post-Hoc explanations conditioned on the answer it did *not* select (i.e., the answer it assigned a lower score to). We evaluate examples where the model answers correctly under the normal Post-Hoc setting (Normal), ensuring that under Forced, the model must explain an incorrect answer despite knowing the correct one. Evaluating simulation precision on StrategyQA shows that Normal significantly outperforms Forced by **45.2** precision points (p-value $< 10^{-16}$), confirming our method's ability to identify lower-quality explanation systems.

Normal	Forced	Δ
83.4	38.2	45.2

Table 4.2: Normal outperforms Forced on simulation precision by **45.2** points. Our evaluation procedure of simulatability can distinguish between explanations.

Dataset	H–H	H–GPT-3	H–GPT-4
StrategyQA	0.504	0.339	0.486
SHP	0.265	0.058	0.296

Table 4.3: We evaluate whether GPT-3 and GPT-4 are good proxies of human simulators by calculating their IAA with humans divided by the average IAA between humans. GPT-4 can approximate human simulators. We measure IAAs with Cohen's Kappa.

GPT-4 reliably approximates human simulators. We assess LLMs (GPT-3 and GPT-4) as proxies for human simulators by comparing their inter-annotator agreement (IAA) with humans against the average IAA between human annotators. Table 4.3 shows the IAA (measured by Cohen's kappa) between GPT-3, GPT-4, and humans. GPT-4 demonstrates substantially better agreement with humans compared to GPT-3, matching or exceeding the level of agreement between human annotators. For SHP specifically, GPT-4 shows higher agreement with humans than humans do with each other, suggesting more consistent annotations. Based on these results, we use GPT-4 as the simulator for SHP experiments while retaining human simulators for StrategyQA.

LLM prompting generates higher-quality counterfactuals than explanation-agnostic baselines. We evaluate our LLM prompting approach against PolyJuice [111], a baseline that generates counterfactuals through lexical and semantic perturbations without considering explanations. Table 4.1 shows diversity scores across generators (GPT-3, GPT-4, PolyJuice), averaged across explanation systems. On StrategyQA, GPT-3 outperforms PolyJuice by **68%** relative improvement (averaged across metrics). While GPT-3 and GPT-4 achieve similar diversity individually, combining their outputs increases diversity by **12%**. Consequently, our subsequent analyses evaluate explanations using combined counterfactuals from both models.

4.3.2 Main Results

Having validated our evaluation procedure, we now compare different explanation methods (Section 4.3.2.1) and analyze relationships between our metrics and existing evaluation

Dataset	GPT-3		GPT-4	
	CoT	Post-Hoc	CoT	Post-Hoc
StrategyQA	77.3	76.8	81.1	**83.9**
SHP	86.3	85.2	**93.0**	91.5

Table 4.4: GPT-4 explanations are consistently more precise compared to GPT-3 explanations, by +**5.5** precision points on StrategyQA and +**6.5** precision points on SHP (p-value < 0.002). We do not observe a clear difference in simulation precision between CoT and Post-Hoc.

approaches (Section 4.3.2.2). Based on our earlier findings (Table 4.3), we use GPT-4 as the simulator for SHP while maintaining human simulators for StrategyQA.

4.3.2.1 Benchmarking LLM Explanations

CoT and Post-Hoc explanations achieve similar precision. Table 4.4 compares simulation precision between Chain-of-Thought and Post-Hoc approaches. Contrary to our expectation that CoT would yield more precise explanations (since answers are conditioned on the reasoning), we find no consistent advantage for either method. CoT shows a marginal improvement of 1.2 points on StrategyQA but lags by 1.3 points on SHP. This unexpected finding may indicate that LLMs can generate externalized reasoning (whether CoT or Post-Hoc) that diverges from their internal decision process [101, 30], though further investigation is needed.

GPT-4 produces more precise explanations than GPT-3. Table 4.4 also compares simulation precision between GPT-3 and GPT-4. GPT-4 consistently generates more precise explanations, outperforming GPT-3 by **5.5** points on StrategyQA and **6.5** points on SHP (p-value < 0.002). Future research should investigate how model scale influences counterfactual simulatability.[2]

4.3.2.2 Studying Relations between Metrics

We examine how precision and generality relate to each other and to established metrics: plausibility and task accuracy. Strong correlations would suggest that optimizing existing metrics or a single new metric might suffice for generating precise and general explanations.

Simulation precision shows minimal correlation with plausibility. For each input, we generate explanations using four systems (GPT-3 and GPT-4, each with CoT and

[2]Note that this comparison alone cannot attribute the performance difference solely to scale, as GPT-3.5 and GPT-4 likely differ in multiple aspects.

Dataset	BLEU	Cosine	Jaccard
StrategyQA	0.017	0.002	-0.007
SHP	0.048	0.020	0.007

Table 4.5: Simulation generality does not correlate with simulation precision, indicating that a general explanation that helps users simulate the model's behavior on more inputs does not guarantee high precision.

Dataset	Task Acc.	PredPower
StrategyQA	**75.9**	79.8
SHP	66.7	**89.0**

Table 4.6: While StrategyQA is easier compared to SHP, simulation precision of explanations on SHP is significantly higher than explanations on StrategyQA.

Post-Hoc). We compute simulation precision (Section 4.1.2) and collect human plausibility judgments for each explanation (annotation instructions shown in Figure A.11). Analyzing correlations across the four explanations per input and averaging across inputs reveals only weak correlations: +**0.012** (Pearson) and +**0.021** (Spearman). These correlations are substantially lower than the inter-annotator correlation for plausibility (+0.388 Pearson, +0.376 Spearman), indicating that plausible, human-preferred explanations don't necessarily enable accurate mental models. This suggests that approaches like RLHF, which optimize for human-like explanations, may not improve counterfactual simulatability.

Simulation generality and precision are independent. Our analysis of the relationship between precision and generality (Table 4.5) reveals no significant correlation. This independence suggests that explanations enabling simulation across diverse counterfactuals don't guarantee accurate simulations. Therefore, both metrics provide important and complementary signals for evaluation and optimization.

Task difficulty does not predict simulation precision. While one might expect easier tasks to yield more precise explanations, our analysis suggests otherwise. Table 4.6 compares simulation precision and task accuracy across StrategyQA and SHP (averaged across explanation systems). Despite StrategyQA being easier by **9.2** accuracy points, explanations for SHP achieve **9.2** points higher precision. This suggests that simulation precision depends more on the complexity of the model's decision process than on task difficulty.

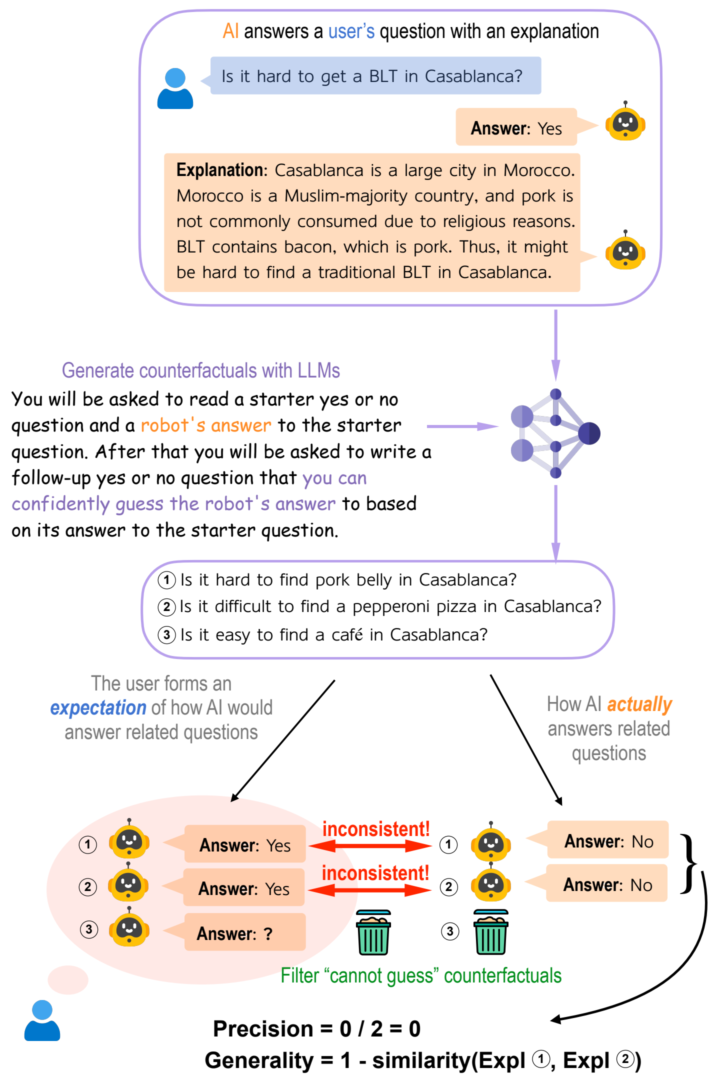

Figure 4.2: **Our evaluation pipeline.** In this example, GPT-4 answers a user's question and explains its decision process. To evaluate counterfactual simulatability, we first use LLMs to generate related counterfactuals based on the model's explanation; the human builds a mental model based on the explanation and logically simulates what GPT-4 outputs for each counterfactual if possible. Finally, we ask GPT-4 to produce its output for each counterfactual, calculate simulation precision as the fraction of counterfactuals where humans' simulated outputs match GPT-4's actual output, and calculate simulation generality as one minus the average pairwise similarity between related counterfactuals.

Chapter 5

Conclusion and Future Work

This thesis proposes metrics and methods for discovering and explaining dataset patterns in structured modalities (text/images) using natural language strings. We evaluate the explanations based on the predictive power they give to humans, which differs from common metrics based on human ratings or similarity to human demonstrations. We then generate dataset explanations by optimizing them against our evaluation metric, with the help of language models.

Based on these principles, we build a general framework, "statistical models with natural language parameters," which allows us to explain distributional differences, clusters, and time series in real-world datasets with structured modalities. Additionally, our metric can evaluate explanations of model decisions by treating them as explanations of datasets that consist of the model's input-output behavior. Using this approach, we show that language models are still far from explaining themselves as of 2024. Our contribution paves the way for helping humans understand complex datasets and systems, thereby accelerating scientific discovery and advancing explainable AI systems.

Nevertheless, significant research opportunities remain to make explanations more broadly applicable and deployable in real-world systems. We outline three promising directions for future research that could help achieve these goals:

Increasing the speed of generating explanations. At present, we must compute $[\![e]\!](x)$ for *every* explanation-sample pair, resulting in an enormous number of queries to language models. Two complementary approaches could accelerate this process:

- *Amortizing* validations: we can pre-encode explanations and samples into a joint embedding space and approximate $[\![e]\!](x)$ with a lightweight similarity kernel (e.g., a dot product followed by a small MLP). [58] has explored a similar approach in text retrieval, and we believe similar techniques could work effectively in our domain.
- *Optimizing* proposers: we can use our evaluation as the reward to fine-tune the LLM to produce better explanations. As a result, the model can learn to generate high-quality explanations *without* an explicit validation step, reducing computational costs. [26] has

investigated this approach for generating neuron descriptions, and [23] has explored it for binary classification tasks. We encourage future research that extends this idea to broader applications.

Accounting for divergent human interpretations. Natural language explanations can be interpreted differently across demographic or cultural groups. For example, whether a sample x satisfies the explanation $e =$ "is humorous" is highly subjective, and thus $[\![e]\!]$ depends heavily on cultural context. Two potential directions to address this challenge are:

(a) Building *richer human simulators*: We can train validators on large-scale, stratified behavioral data to capture viewpoint variations across different audiences.

(b) Generating *audience-aware* explanations: With improved evaluation methods that capture differences among individual humans, we can optimize and train language models to generate explanations tailored to specific audiences.

Extending beyond text-only explanations. Certain concepts — such as chess tactics, medical images, or geospatial patterns — are difficult to convey purely through natural language. Two promising avenues merit exploration:

- Explaining with *multi-modal* support: Combining textual descriptions with inline visualizations (e.g., salient sub-regions [2] or most influential training examples [59]).
- Inventing *new* concepts: When no compact vocabulary exists (e.g., for an unseen chess motif), allowing the system to introduce visual exemplars, name them ("*rook roller*"), and ground them through few-shot demonstrations, progressively enriching the explanation language. See further discussion in [49].

Collectively, these directions aim to produce explanations more *efficiently* that are *audience-aware* and *multimodal*, thus helping humans better make sense of the complex world.

Bibliography

[1] URL: https://www.anthropic.com/news/claude-3-5-sonnet.

[2] Julius Adebayo et al. "Sanity checks for saliency maps". In: *Advances in neural information processing systems* 31 (2018).

[3] Roee Aharoni and Yoav Goldberg. *Unsupervised Domain Clusters in Pretrained Language Models*. 2020. arXiv: 2004.02105 [cs.CL].

[4] Tiago Almeida, José María Gómez Hidalgo, and Tiago Pasqualini Silva. "Towards sms spam filtering: Results under a new dataset". In: *International Journal of Information Security Science* 2.1 (2013), pp. 1–18.

[5] Jacob Andreas, Dan Klein, and Sergey Levine. "Learning with Latent Language". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 2166–2179. DOI: 10.18653/v1/N18-1197. URL: https://aclanthology.org/N18-1197.

[6] Yuntao Bai et al. "Training a helpful and harmless assistant with reinforcement learning from human feedback". In: *ArXiv* (2022). URL: https://arxiv.org/pdf/2204.05862.pdf.

[7] Satanjeev Banerjee and Alon Lavie. "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments". In: *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*. 2005, pp. 65–72.

[8] Gagan Bansal et al. "Beyond accuracy: The role of mental models in human-AI team performance". In: *Proceedings of the AAAI conference on human computation and crowdsourcing*. 2019. URL: https://ojs.aaai.org/index.php/HCOMP/article/view/5285.

[9] Francesco Barbieri et al. "TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 1644–1650. DOI: 10.18653/v1/2020.findings-emnlp.148. URL: https://aclanthology.org/2020.findings-emnlp.148.

[10] Kasturi Bhattacharjee et al. "What Do Users Care About? Detecting Actionable Insights from User Feedback". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*. Ed. by Anastassia Loukina, Rashmi Gangadharaiah, and Bonan Min. Hybrid: Seattle, Washington + Online: Association for Computational Linguistics, July 2022, pp. 239–246. DOI: 10.18653/v1/2022.naacl-industry.27. URL: https://aclanthology.org/2022.naacl-industry.27.

[11] Steven Bills et al. "Language models can explain neurons in language models". In: *URL https://openaipublic. blob. core. windows. net/neuron-explainer/paper/index. html.(Date accessed: 14.05. 2023)* (2023).

[12] David M Blei and John D Lafferty. "Dynamic topic models". In: *Proceedings of the 23rd international conference on Machine learning*. 2006, pp. 113–120.

[13] David M Blei, Andrew Y Ng, and Michael I Jordan. "Latent dirichlet allocation". In: *Journal of machine Learning research* 3.Jan (2003), pp. 993–1022.

[14] Samuel R. Bowman et al. "A large annotated corpus for learning natural language inference". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 632–642. DOI: 10.18653/v1/D15-1075. URL: https://aclanthology.org/D15-1075.

[15] Jonathan Bragg et al. "Flex: Unifying evaluation for few-shot nlp". In: *Advances in Neural Information Processing Systems* 34 (2021).

[16] Tom Brown et al. "Language models are few-shot learners". In: *Advances in Neural Information Processing Systems* (2020). URL: https://papers.nips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

[17] Oana-Maria Camburu et al. "e-snli: Natural language inference with natural language explanations". In: *Advances in Neural Information Processing Systems* 31 (2018).

[18] Andrea M Cassidy. *Mental models, trust, and reliance: Exploring the effect of human perceptions on automation use*. Tech. rep. 2009. URL: http://edocs.nps.edu/npspubs/scholarly/theses/2009/Jun/09Jun_Cassidy.pdf.

[19] Anjan Chakravartty. "Scientific realism". In: (2011).

[20] Arjun Chandrasekaran et al. "Do explanations make VQA models more predictable to a human?" In: *Proceedings of Empirical Methods in Natural Language Processing*. 2018. URL: https://aclanthology.org/D18-1128.

[21] Jonathan Chang et al. "Reading tea leaves: How humans interpret topic models". In: *Advances in neural information processing systems* 22 (2009).

[22] Mark Chen et al. "Evaluating large language models trained on code". In: *arXiv preprint arXiv:2107.03374* (2021).

[23] Yanda Chen et al. "Towards consistent natural-language explanations via explanation-consistency finetuning". In: *arXiv preprint arXiv:2401.13986* (2024).

[24] Wei-Lin Chiang et al. "Chatbot arena: An open platform for evaluating llms by human preference". In: *arXiv preprint arXiv:2403.04132* (2024).

[25] Mia Chiquier, Utkarsh Mall, and Carl Vondrick. "Evolving Interpretable Visual Classifiers with Large Language Models". In: *arXiv preprint arXiv:2404.09941* (2024).

[26] Dami Choi et al. *Scaling Automatic Neuron Description*. https://transluce.org/neuron-descriptions. Oct. 2024.

[27] Paul F Christiano et al. "Deep reinforcement learning from human preferences". In: *Advances in Neural Information Processing Systems* (2017). URL: https://papers.nips.cc/paper_files/paper/2017/hash/d5e2c0adad503c91f91df240d0cd4e49-Abstract.html.

[28] Hyung Won Chung et al. "Scaling instruction-finetuned language models". In: *arXiv preprint arXiv:2210.11416* (2022).

[29] Allan Collins and Dedre Gentner. "How people construct mental models". In: *Cultural models in language and thought* (1987). URL: https://doi.org/10.1017/CBO9780511607660.011.

[30] Antonia Creswell and Murray Shanahan. "Faithful reasoning using large language models". In: *ArXiv* (2022). URL: https://arxiv.org/pdf/2208.14271.pdf.

[31] Mingkai Deng et al. "RLPrompt: Optimizing Discrete Text Prompts with Reinforcement Learning". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 3369–3391. DOI: 10.18653/v1/2022.emnlp-main.222. URL: https://aclanthology.org/2022.emnlp-main.222.

[32] Norman R Draper and Harry Smith. *Applied regression analysis*. Vol. 326. John Wiley & Sons, 1998.

[33] Yann Dubois et al. "Alpacafarm: A simulation framework for methods that learn from human feedback". In: *ArXiv* (). URL: https://arxiv.org/pdf/2305.14387.pdf.

[34] Lisa Dunlap et al. "Describing Differences in Image Sets with Natural Language". In: *arXiv preprint arXiv:2312.02974* (2023).

[35] Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. "Understanding Dataset Difficulty with $\mathcal{V}$-Usable Information". In: *Proceedings of the International Conference on Machine Learning*. 2022. URL: https://proceedings.mlr.press/v162/ethayarajh22a.html.

[36] Sabri Eyuboglu et al. "Domino: Discovering systematic errors with cross-modal embeddings". In: *arXiv preprint arXiv:2203.14960* (2022).

[37] Tianyu Gao, Adam Fisch, and Danqi Chen. "Making Pre-trained Language Models Better Few-shot Learners". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 3816–3830. DOI: 10.18653/v1/2021.acl-long.295. URL: https://aclanthology.org/2021.acl-long.295.

[38] Alan Garnham. *Mental models as representations of discourse and text.* 1987. URL: https://psycnet.apa.org/record/1988-97459-000.

[39] Mor Geva et al. "Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies". In: *Transactions of the Association for Computational Linguistics* (2021). URL: https://direct.mit.edu/tacl/article/doi/10.1162/tacl_a_00370/100680/Did-Aristotle-Use-a-Laptop-A-Question-Answering.

[40] Jeremy Ginsberg et al. "Detecting influenza epidemics using search engine query data". In: *Nature* 457.7232 (2009), pp. 1012–1014.

[41] José María Gómez Hidalgo et al. "Content based SMS spam filtering". In: *Proceedings of the 2006 ACM symposium on Document engineering*. 2006, pp. 107–114.

[42] Thomas L Griffiths and Mark Steyvers. "Finding scientific topics". In: *Proceedings of the National academy of Sciences* 101.suppl_1 (2004), pp. 5228–5235.

[43] Maarten Grootendorst. "BERTopic: Neural topic modeling with a class-based TF-IDF procedure". In: *arXiv preprint arXiv:2203.05794* (2022).

[44] Suchin Gururangan et al. "Annotation Artifacts in Natural Language Inference Data". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 107–112. DOI: 10.18653/v1/N18-2017. URL: https://aclanthology.org/N18-2017.

[45] Tatsunori Hashimoto et al. "Fairness without demographics in repeated loss minimization". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 1929–1938.

[46] Md Mahadi Hassan, Alex Knipper, and Shubhra Kanti Karmaker Santu. "ChatGPT as your Personal Data Scientist". In: *arXiv preprint arXiv:2305.13657* (2023).

[47] Dan Hendrycks et al. "Measuring mathematical problem solving with the math dataset". In: *arXiv preprint arXiv:2103.03874* (2021).

[48] Evan Hernandez et al. "Natural Language Descriptions of Deep Visual Features". In: *International Conference on Learning Representations*. 2021.

[49] John Hewitt, Robert Geirhos, and Been Kim. "We Can't Understand AI Using our Existing Vocabulary". In: *arXiv preprint arXiv:2502.07586* (2025).

[50] Or Honovich et al. "Instruction Induction: From Few Examples to Natural Language Task Descriptions". In: *arXiv preprint arXiv:2205.10782* (2022).

[51] Robert Hooke. *Lectures de potentia restitutiva, or of spring explaining the power of springing bodies*. 6. John Martyn, 2016.

[52] Alexander Miserlis Hoyle et al. "Are Neural Topic Models Broken?" In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 5321–5344. DOI: 10.18653/v1/2022.findings-emnlp.390. URL: https://aclanthology.org/2022.findings-emnlp.390.

[53] Phillip Isola et al. "Understanding the intrinsic memorability of images". In: *Advances in neural information processing systems* 24 (2011).

[54] Philip N Johnson-Laird. "Mental models in cognitive science". In: *Cognitive science* (1980). URL: https://onlinelibrary.wiley.com/doi/pdf/10.1207/s15516709cog0401_4.

[55] Brihi Joshi et al. "Are Machine Rationales (Not) Useful to Humans? Measuring and Improving Human Utility of Free-text Rationales". In: *Proceedings of the Association for Computational Linguistics*. 2023. URL: https://aclanthology.org/2023.acl-long.392.

[56] Siddharth Karamcheti et al. "Lila: Language-informed latent actions". In: *Conference on Robot Learning*. PMLR. 2022, pp. 1379–1390.

[57] Daniel Khashabi et al. "UNIFIEDQA: Crossing Format Boundaries with a Single QA System". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 1896–1907. DOI: 10.18653/v1/2020.findings-emnlp.171. URL: https://aclanthology.org/2020.findings-emnlp.171.

[58] Omar Khattab and Matei Zaharia. "Colbert: Efficient and effective passage search via contextualized late interaction over bert". In: *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 2020, pp. 39–48.

[59] Pang Wei Koh and Percy Liang. "Understanding black-box predictions via influence functions". In: *International conference on machine learning*. PMLR. 2017, pp. 1885–1894.

[60] Wuwei Lan et al. "A Continuously Growing Dataset of Sentential Paraphrases". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 1224–1234. DOI: 10.18653/v1/D17-1126. URL: https://aclanthology.org/D17-1126.

[61] Tilman Lange et al. "Stability-based validation of clustering solutions". In: *Neural computation* 16.6 (2004), pp. 1299–1323.

[62] Mina Lee, Percy Liang, and Qian Yang. "Coauthor: Designing a human-ai collaborative writing dataset for exploring language model capabilities". In: *Proceedings of the 2022 CHI conference on human factors in computing systems*. 2022, pp. 1–19.

[63] Benjamin Letham et al. "Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model". In: *The Annals of Applied Statistics* 9.3 (Sept. 2015). ISSN: 1932-6157. DOI: 10.1214/15-aoas848. URL: http://dx.doi.org/10.1214/15-AOAS848.

[64] Xin Li and Dan Roth. "Learning question classifiers". In: *COLING 2002: The 19th International Conference on Computational Linguistics*. 2002.

[65] Yinhan Liu et al. "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *ArXiv* abs/1907.11692 (2019).

[66] Josh Magnus Ludan et al. "Interpretable-by-Design Text Classification with Iteratively Generated Concept Bottleneck". In: *arXiv preprint arXiv:2310.19660* (2023).

[67] Jens Ludwig and Sendhil Mullainathan. "Algorithmic behavioral science: Machine learning as a tool for scientific discovery". In: *Chicago Booth Research Paper* 22-15 (2022).

[68] Pingchuan Ma et al. "Demonstration of InsightPilot: An LLM-Empowered Automated Data Exploration System". In: *arXiv preprint arXiv:2304.00477* (2023).

[69] Andrew L. Maas et al. "Learning Word Vectors for Sentiment Analysis". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 142–150. URL: http://www.aclweb.org/anthology/P11-1015.

[70] Christopher Manning and Hinrich Schutze. *Foundations of statistical natural language processing*. MIT press, 1999.

[71] Stephen Merity et al. *Pointer Sentinel Mixture Models*. 2016. arXiv: 1609.07843 [cs.CL].

[72] Tsvetomila Mihaylova et al. "SemEval-2019 Task 8: Fact Checking in Community Question Answering Forums". In: *Proceedings of the 13th International Workshop on Semantic Evaluation*. Minneapolis, Minnesota, USA: Association for Computational Linguistics, June 2019, pp. 860–869. DOI: 10.18653/v1/S19-2149. URL: https://aclanthology.org/S19-2149.

[73] Sewon Min et al. "Noisy channel language model prompting for few-shot text classification". In: *arXiv preprint arXiv:2108.04106* (2021).

[74] Swaroop Mishra et al. "Cross-Task Generalization via Natural Language Crowdsourcing Instructions". In: 2021.

[75] Jesse Mu, Percy Liang, and Noah Goodman. "Shaping Visual Representations with Language for Few-Shot Classification". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky et al. Online: Association for Computational Linguistics, July 2020, pp. 4823–4830. DOI: 10.18653/v1/2020.acl-main.436. URL: https://aclanthology.org/2020.acl-main.436.

[76] Aakanksha Naik et al. "Stress Test Evaluation for Natural Language Inference". In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 2340–2353. URL: https://aclanthology.org/C18-1198.

[77] Dong Nguyen et al. "How we do things with words: Analyzing text as social and cultural data". In: *Frontiers in Artificial Intelligence* 3 (2020), p. 62.

[78] Pontus Nordberg, Joakim Kävrestad, and Marcus Nohlberg. "Automatic detection of fake news". In: *6th International Workshop on Socio-Technical Perspective in IS Development, virtual conference in Grenoble, France, June 8-9, 2020*. CEUR-WS. 2020, pp. 168–179.

[79] Maxwell Nye et al. "Show your work: Scratchpads for intermediate computation with language models". In: *ArXiv* (). URL: https://arxiv.org/pdf/2112.00114.pdf.

[80] OpenAI. *GPT-3.5 Turbo*. https://platform.openai.com/docs/models/gpt-3-5. Accessed: 2024-05-14. 2024.

[81] OpenAI. "GPT-4 Technical Report". In: *ArXiv* (2023). URL: https://arxiv.org/pdf/2303.08774.pdf.

[82] Long Ouyang et al. "Training language models to follow instructions with human feedback". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 27730–27744.

[83] Bo Pang and Lillian Lee. "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts". In: *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*. Barcelona, Spain, July 2004, pp. 271–278. DOI: 10.3115/1218955.1218990. URL: https://aclanthology.org/P04-1035.

[84] Kishore Papineni et al. "Bleu: a Method for Automatic Evaluation of Machine Translation". In: *Proceedings of the Association for Computational Linguistics*. 2002. URL: https://aclanthology.org/P02-1040.

[85] Kishore Papineni et al. "Bleu: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, pp. 311–318.

[86] Dong Huk Park et al. "Multimodal explanations: Justifying decisions and pointing to the evidence". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018. URL: https://openaccess.thecvf.com/content_cvpr_2018/papers/Park_Multimodal_Explanations_Justifying_CVPR_2018_paper.pdf.

[87] Chau Minh Pham et al. "TopicGPT: A Prompt-based Topic Modeling Framework". In: *arXiv preprint arXiv:2311.01449* (2023).

[88] Linlu Qiu et al. "Phenomenal Yet Puzzling: Testing Inductive Reasoning Capabilities of Language Models with Hypothesis Refinement". In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=bNt7oajl2a.

[89] Jack W Rae et al. "Scaling Language Models: Methods, Analysis & Insights from Training Gopher". In: *arXiv preprint arXiv:2112.11446* (2021).

[90] Colin Raffel et al. "Exploring the limits of transfer learning with a unified text-to-text transformer". In: *arXiv preprint arXiv:1910.10683* (2019).

[91] Shiori Sagawa et al. "Distributionally Robust Neural Networks for Group Shifts: On the Importance of Regularization for Worst-Case Generalization". In: *ArXiv* abs/1911.08731 (2019).

[92] Evan Sandhaus. "The new york times annotated corpus". In: *Linguistic Data Consortium, Philadelphia* 6.12 (2008), e26752.

[93] Victor Sanh et al. "Multitask Prompted Training Enables Zero-Shot Task Generalization". In: *ArXiv* abs/2110.08207 (2021).

[94] Simon Schrodi et al. "Concept Bottleneck Models Without Predefined Concepts". In: *arXiv preprint arXiv:2407.03921* (2024).

[95] Pratyusha Sharma, Antonio Torralba, and Jacob Andreas. "Skill Induction and Planning with Latent Language". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 1713–1726. DOI: 10.18653/v1/2022.acl-long.120. URL: https://aclanthology.org/2022.acl-long.120.

[96] Taylor Shin et al. "Autoprompt: Eliciting knowledge from language models with automatically generated prompts". In: *arXiv preprint arXiv:2010.15980* (2020).

[97] Christopher T Small et al. "Opportunities and risks of LLMs for scalable deliberation with Polis". In: *arXiv preprint arXiv:2306.11932* (2023).

[98] Richard Socher et al. "Reasoning with neural tensor networks for knowledge base completion". In: *Advances in neural information processing systems* 26 (2013).

[99] Nisan Stiennon et al. "Learning to summarize with human feedback". In: *Advances in neural information processing systems* 33 (2020), pp. 3008–3021.

[100] Hongjin Su et al. "One Embedder, Any Task: Instruction-Finetuned Text Embeddings". In: 2022. URL: https://arxiv.org/abs/2212.09741.

[101] Miles Turpin et al. "Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting". In: *ArXiv* (2023). URL: https://arxiv.org/pdf/2305.04388.pdf.

[102] Xuezhi Wang et al. "Self-Consistency Improves Chain of Thought Reasoning in Language Models". In: *Proceedings of the International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=1PL1NIMMrw.

[103] Yizhong Wang et al. "Benchmarking Generalization via In-Context Instructions on 1,600+ Language Tasks". In: *arXiv preprint arXiv:2204.07705* (2022).

[104] Zihan Wang, Jingbo Shang, and Ruiqi Zhong. "Goal-Driven Explainable Clustering via Language Descriptions". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 10626–10649. DOI: 10.18653/v1/2023.emnlp-main.657. URL: https://aclanthology.org/2023.emnlp-main.657.

[105] Zihan Wang, Jingbo Shang, and Ruiqi Zhong. "Goal-Driven Explainable Clustering via Language Descriptions". In: *arXiv preprint arXiv:2305.13749* (2023).

[106] Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. "Neural Network Acceptability Judgments". In: *arXiv preprint arXiv:1805.12471* (2018).

[107] Jason Wei et al. "Chain-of-thought prompting elicits reasoning in large language models". In: *Advances in neural information processing systems* 35 (2022), pp. 24824–24837.

[108] Adina Williams, Nikita Nangia, and Samuel Bowman. "A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 1112–1122. DOI: 10.18653/v1/N18-1101. URL: https://aclanthology.org/N18-1101.

[109] Lionel Wong et al. "Learning adaptive planning representations with natural language guidance". In: *arXiv preprint arXiv:2312.08566* (2023).

[110] Mitchell Wortsman et al. "Robust fine-tuning of zero-shot models". In: *arXiv preprint arXiv:2109.01903* (2021).

[111] Tongshuang Wu et al. "Polyjuice: Generating Counterfactuals for Explaining, Evaluating, and Improving Models". In: *Proceedings of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*. 2021. URL: https://aclanthology.org/2021.acl-long.523.

[112] Yue Yang et al. "Language in a bottle: Language model guided concept bottlenecks for interpretable image classification". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 19187–19197.

[113] Zonglin Yang et al. "Language Models as Inductive Reasoners". In: *arXiv preprint arXiv:2212.10923* (2022).

[114] Zonglin Yang et al. "Large Language Models for Automated Open-domain Scientific Hypotheses Discovery". In: *arXiv preprint arXiv:2309.02726* (2023).

[115] Seonghyeon Ye et al. "Guess the Instruction! Making Language Models Stronger Zero-Shot Learners". In: *arXiv preprint arXiv:2210.02969* (2022).

[116] Xi Ye and Greg Durrett. "Can Explanations Be Useful for Calibrating Black Box Models?" In: *Proceedings of the Association for Computational Linguistics*. May 2022. URL: `https://aclanthology.org/2022.acl-long.429`.

[117] Wenpeng Yin, Jamaal Hay, and Dan Roth. "Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3914–3923. DOI: `10.18653/v1/D19-1404`. URL: `https://aclanthology.org/D19-1404`.

[118] Chao Zhang et al. "Taxogen: Unsupervised topic taxonomy construction by adaptive term embedding and clustering". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, pp. 2701–2709.

[119] Tianyi Zhang et al. "Bertscore: Evaluating text generation with bert". In: *arXiv preprint arXiv:1904.09675* (2019).

[120] Xiang Zhang, Junbo Zhao, and Yann LeCun. "Character-Level Convolutional Networks for Text Classification". In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'15. Montreal, Canada: MIT Press, 2015, pp. 649–657.

[121] Xiang Zhang, Junbo Zhao, and Yann LeCun. "Character-level convolutional networks for text classification". In: *Advances in neural information processing systems* 28 (2015).

[122] Wenting Zhao et al. "WildChat: 1M ChatGPT Interaction Logs in the Wild". In: *The Twelfth International Conference on Learning Representations*. 2024. URL: `https://openreview.net/forum?id=Bl8u7ZRlbM`.

[123] Lianmin Zheng et al. "Judging llm-as-a-judge with mt-bench and chatbot arena". In: *Advances in Neural Information Processing Systems* 36 (2024).

[124] Ruiqi Zhong et al. "Adapting Language Models for Zero-shot Learning by Meta-tuning on Dataset and Prompt Collections". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 2856–2878. URL: https://aclanthology.org/2021.findings-emnlp.244.

[125] Ruiqi Zhong et al. "Describing differences between text distributions with natural language". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 27099–27116.

[126] Ruiqi Zhong et al. "Goal driven discovery of distributional differences via language descriptions". In: *arXiv preprint arXiv:2302.14233* (2023).

[127] Ruiqi Zhong et al. "Goal driven discovery of distributional differences via language descriptions". In: *Advances in Neural Information Processing Systems* 36 (2024).

[128] Yongchao Zhou et al. "Large language models are human-level prompt engineers". In: *arXiv preprint arXiv:2211.01910* (2022).

[129] Zhiying Zhu, Weixin Liang, and James Zou. "GSCLIP: A Framework for Explaining Distribution Shifts in Natural Language". In: *arXiv preprint arXiv:2206.15007* (2022).

Appendix A

Appendix

A.1 Chapter 2 Appendix

A.1.1 Using BERT-score for Evaluation

We generate scatter plots to compare our best system ① with the worst system ④ and our second best system ② in Figure A.1 to double-check that we used the metric correctly. Despite the the small absolute difference (3%) in the reported numbers, BERTScore does robustly tell the difference between system system ① and ④. On the other hand, however, it has trouble discriminating our first and second best system: after squinting at the results hard enough, we find that ① outperforms ② by 0.3 points on average; across binary tasks, ① outperforms ② more than 0.5 points for 46% of the time, while ② outperforms ① by more than 0.5 points 31% of the time. Therefore, BERTscore does agree that ① is better than ②. Nevertheless, we felt that this metric is not discriminative and interpretable enough, so we had to rely on human evaluation (Section 2.3).

A.1.2 Top-K Performance

We calculate the performance of the top-K explanations by our system according to our manual evaluation, where K ranges from 1 to 5. Table A.1 shows the results.

We report the statistical significance of comparing different systems by their best-of-top-5 explanations. We first examine how often a system generates a "A" level explanation across 54 binary classification datasets; as a result ① > ② with $p = 9.3 \times 10^{-3}$, ② > ③ with $p = 3.2 \times 10^{-3}$, ③ > ④ with $p = 2.5 \times 10^{-2}$, and ② > ⑤ with $p = 4.2 \times 10^{-5}$. We next examine how often a system generates a "A" or "B" level explanation; as a result ① > ② with $p = 9.7 \times 10^{-3}$, ② > ③ with $p = 1.7 \times 10^{-4}$, ③ > ④ with $p = 5.4 \times 10^{-4}$, and ② > ⑤ with $p = 1.6 \times 10^{-6}$.

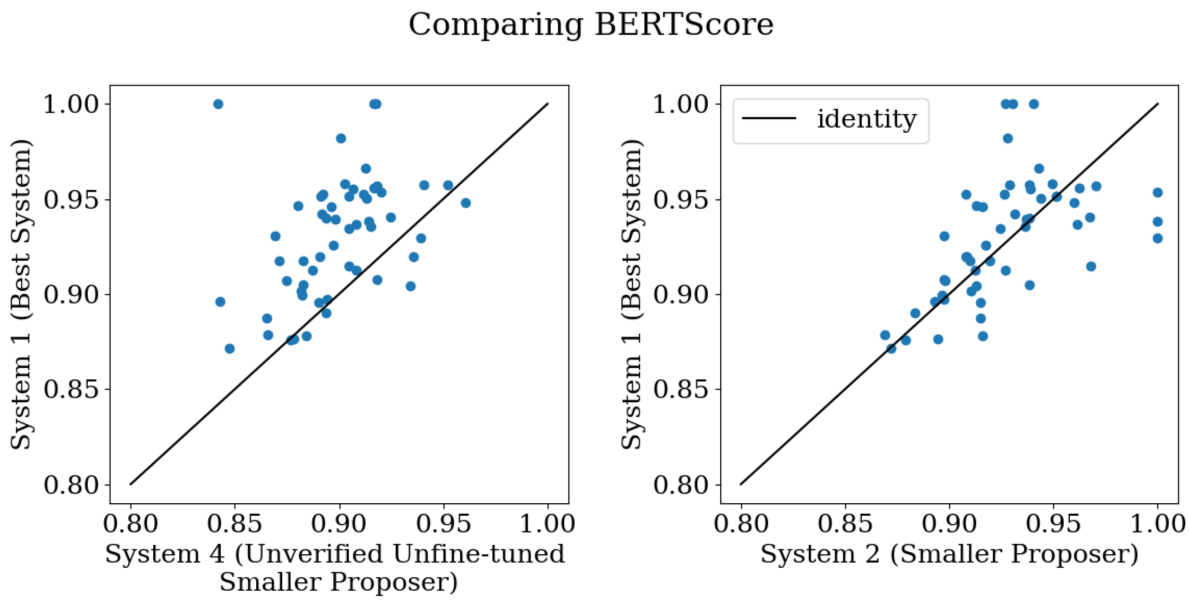

Figure A.1: We compare System 1 and 4 with BERTScore [119] on the left and 1 and 2 on the right. Each dot represents a binary task the y/x value is the performance of a system-generated explanation evaluated by BERTScore. Our best system 1 is clearly outperforming the worst 4 (left), but the difference between the 1st and the 2nd system becomes hard to tell (right).

A.1.3 Example Positive Samples in section 2.2.3

We list a few example explanations along with positive sample text (i.e., GPT-3 generated texts that are likely to satisfy the explanations), indxed with bullet points.

Explanation: *contains internet emoticons.*

- *:) - I'm happy*
- *:) :D :o :(*
- *(:-I'm so excited to tell you about my plans for the future.): I hope they work out!*

Explanation: *contains an acronym or abbreviation that might be used online.*

- *SMH is an acronym for shaking my head. It is used when someone reacts to something funny or shocking.*
- *OMG. I think itś my new BF.*
- *IDK - I don't knowTTYL - talk to you later.*

	① Best	② Smaller	③ No Fine-tune	④ No Re-rank	⑤ Memorize
A	13/26/28/30/31	14/17/21/21/22	6/ 8/10/10/11	2/ 3/ 3/ 3/ 4	2/ 3/ 5/ 5/ 5
B	19/14/13/11/10	10/10/8/10/11	5/ 6/ 6/ 6/ 6	1/ 0/ 0/ 0/ 0	5/ 5/ 5/ 5/ 5
C	16/ 8/ 7/ 7/ 7	17/14/12/12/10	7/ 9/10/10/10	2/ 3/ 5/ 5/ 6	16/19/19/21/21
D	6/ 6/ 6/ 6/ 6	13/13/13/11/11	36/31/28/28/27	49/48/46/46/44	31/27/25/23/23

Table A.1: Similar to Table 2.1, ① represents our best system with the largest fine-tuned proposer, ② with a smaller fine-tuned proposer, ③ without fine-tuning, ④ without re-ranking, and ⑤ with the memorization proposer. For each task, we choose the top-K explanations according to the validator, and find the highest human rating among the top-K; we then count how often each rating occurs across 54 binary tasks. We report K from 1 to 5 separated by "/" in each cell. Notice that only row (A) is guaranteed to increase as k increases, since we are counting the frequency of the highest ranking; e.g., using five rather than one explanation can change the highest rating from (B) to (A), thus decreasing the count of (B).

Explanation: *is about school life.*

- *Entering the courtyard of the student center, I had a feeling that there was something different about today.*
- *The first day of school is always nerve-wracking. You're the new kid, and you have to make friends fast.*
- *I came to this country in the summer of 2010. I attended a public high school in my new country and was really scared because I did not know what to expect from it.*

Explanation: *is about money.*

- *It's so easy to think about all of the things that you could do with your life if you had more disposable income, but it's hard work to make more than enough. I'm not sure what my goal is when it comes to my finances, but I know that whatever happens in the future has got to be better than this.*
- *There's a recession going on and cash flow is scarce.*
- *Money is very important in today's society. It lets you do what you want to do and nothing can stop you.*

Explanation: *describes cooking.*

- *If you can perfect a recipe, it's amazing. I love being able to read through a recipe and have the confidence that I have all of the ingredients listed.*
- *I love to watch the way that each ingredient has their own value, and how they all come together to make a dish.*

- *One rainy night, I decided to make spaghetti.*

A.1.4 Example Explanations and Their Ratings

For each binary task, we present the human annotation, the best explanations from the top-5 explanations by system 1, and our similarity rating in Table A.2.

A.1.5 A Unifying View

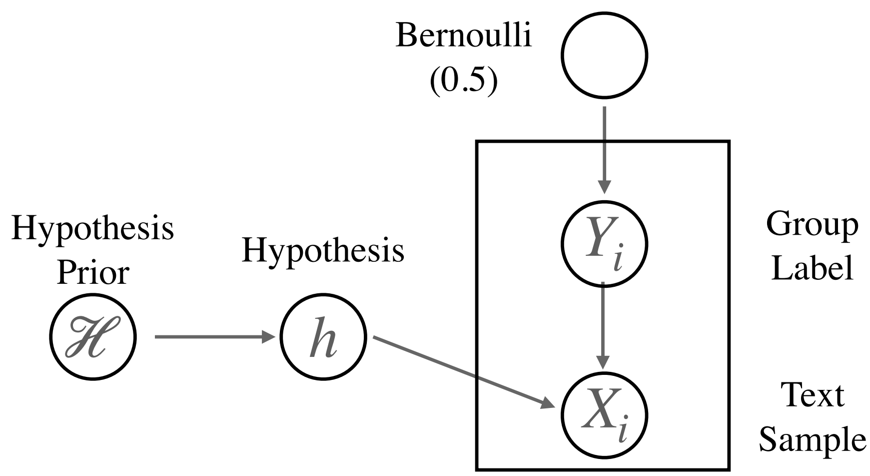

Figure A.2: A unifying graphical model interpretation of our framework, where the validator, the proposer, and the conditional generator can be all written as posterior estimators.

We present a unifying graphical model for the explanation h, the samples $X_{1...K}$, and the group labels $Y_{1...K}$ (Figure A.2), where $Y_i \in \{0, 1\}$ indicating whether X_i is from distribution D_0 or D_1. Although we did not implement it in our paper, we find it helpful as a mental model to generate future research directions. The graphical model factorizes as:

$$p(h, X_{1...K}, Y_{1...K}) = p(h) \prod_{i=1}^{K} p(X_i|Y_i, h)P(Y_i). \tag{A.1}$$

Under this framework, the goal of generating a natural language explanation becomes posterior estimation:

$$p(h|X_{1...K}, Y_{1...K}) \propto p(h) \prod_{i=1}^{K} p(Y_i|X_i, h). \tag{A.2}$$

Human Annotations	Explanations by Our System	Rating
is religious	*is religious*	(A)
is against feminism	*is a criticism of feminism*	(A)
is about math or science	*is about science*	(B)
asks about a location	*asks about a location*	(B)
contains a good movie review	*praises the film*	(A)
is offensive	*is a Twitter hate-rant*	(C)
is related to computer science	*is a description of a computer-based system*	(B)
is against environmentalist	*is a denial of climate change science*	(C)
is against Hillary	*is a criticism of Hillary Clinton*	(A)
is pro-choice	*advocates for abortion rights*	(A)
is about research in statistics	*presents a research on a statistical topic*	(A)
is related to infrastructure	*mentions natural disaster*	(D)
is about entertainment	*is related to the entertainment industry*	(B)
is environmentalist	*shows an environmental concern*	(A)
is related to health	*is about the topic of "health"*	(A)
contains irony	*is sarcastic in tone*	(A)
supports hillary	*is a positive sentence about Hillary Clinton*	(A)
contains a definition	*is about learning something new*	(B)
is related to terrorism	*is about terrorism*	(A)
expresses a need for water	*is about water shortage*	(A)
involves crime	*is describing clashes*	(C)
is related to sports	*is about sports*	(A)
is related to a medical situation	*is related to the topic of health*	(B)
describes a situation where people need food	*is about the situation of food shortage*	(A)
is pro-life	*can be categorized as a pro-life message*	(A)
contains subjective opinions	*is a review of a movie*	(D)
asks for an opinion	*is asking for help*	(D)
is more related to computers or internet	*is about computer*	(B)
expresses need for utility, energy or sanitation	*contains a word related to electricity*	(C)
is sports related	*is about a topic related to sports*	(A)
asks for a number	*contains a question ...**	(A)
describes a situation where people need to evacuate	*describes a situation involving evacuation*	(A)
is a more objective description of what happened	*is a plot summary of a film*	(D)
is physics research	*is about a physics research*	(A)
is about world news	*is a news article on a country*	(C)
looks more like business news	*deals with economic news*	(A)
describes a situation where people need shelter	*is about earthquake*	(C)
is a spam	*is a "spam" SMS*	(A)
contains grammar errors	*is grammatically incorrect*	(A)
asks about an entity	*contains a word that rhymes with "tree"*	(D)
is about math research	*is about a mathematics research paper*	(A)
supports feminism	*is in support of feminism*	(A)
asks for factual information	*is a request for immigration related questions*	(D)
is more political	*is about politics*	(A)
is against religion	*has a negative connotation towards religion*	(A)

Table A.2: For each binary task, we present the human annotation, the best explanations from the top-5 explanations by system 1, and our similarity rating, with (A) being the highest (Section 2.3). *: "*contains a question that can be answered with a number*"; truncated from the column to save space.

The validator can also be written as $\hat{p}(Y|X,h)$, the proposer as $\hat{p}(h|X_{1...5},Y_{1...5})$, the conditional generator as $\hat{p}(X|Y,h)$, and the explanation space as a prior $\hat{p}(h)$,[1] all of which can be directly approximated by a fine-tuned language model. To fine-tune these approximators, it suffices to obtain the complete data h, X_*, and Y_*. Our work only fine-tuned the validator and the proposer, but the conditional generator $\hat{p}(X|Y,h)$ and $\hat{p}(h)$ can also be fine-tuned. We only supervised $\hat{p}$ through querying human about $p(Y|X,h)$, but other forms of queries are also possible. Finally, it is not necessary to follow the recipe in our paper to generate the complete data: we could alternatively first generate X and h, and then generate Y accordingly. Human supervision is also not strictly necessary to generate the complete data: we can purely sample data from some approximators to fine-tune other ones, thus achieving self-supervision through cycle consistency.

A.1.6 Original Sources of the Binary Tasks

The 54 binary tasks are from [69], [117], [9], [120], [117], [106], [4], [83], [64], [72], and an abstract classification dataset[2].

A.1.7 Notes on Code and Model Release

We release our code and data with the following link `https://github.com/ruiqi-zhong/DescribeDistributionalDifferences`.

We cannot directly share our GPT-3 based proposer, since it has to be accessed through the OpenAI API using our own key. To make it easier for other researchers to use our system, we trained another proposer by fine-tuning T5 [90] on a mixture of 1) our collected data, and 2) a large dataset [103] to learn to follow task instructions. Though we have not rigorously benchmarked the new proposer, it seems to be roughly comparable to the proposer based on GPT-3 Davinci (175B parameters), and it can be openly shared, downloaded, and run locally.

A.2 Chapter 3 Appendix

A.2.1 More Related Work

LLM for Exploratory Analysis. Due to its code generation capability [22], large language models have been used to automatically generate programs to analyze a dataset and generate reports from them [68, 46]. In comparison, our work focuses on generating natural language parameters to extract real-valued features from structured data.

Discrete Prompt Optimization. Many prior works optimized discrete prompts to improve the predictive performance [96, 31], and some recent works demonstrated that LLMs can

[1] which our paper defines through manual curation of the explanation and modelled as a uniform distribution during inference.

[2] `https://www.kaggle.com/abisheksudarshan/topic-modeling-for-research-articles?select=Train.csv`

optimize prompts to reach state-of-the-art accuracy [128, 114]. In comparison, we focus on optimizing discrete explanations to explain patterns rather than improve task performance. **Learning with Latent Language.** [5] first proposed to learn in a hypothesis space of natural language strings to improve generalization, and later works in this area have focused on using natural language to guide the learning process to improve downstream task performance [75, 56, 95, 109]. In contrast, our work focuses on explaining datasets with natural language, rather than improving downstream task performance.

A.2.2 Time Series Dataset

To sample texts from the `All` time series problem, we sample from the time series model described in Section 3.1.2; we set $\vec{e}$ to be all the 12 explanations, sort them first by attributes (e.g. topic/location/language) then alphabets, and we set the weight for the kth explanation to be a sin function with period T and evenly spaced phases, i.e.

$$w_{k,t} = \sin(2\pi(\frac{t}{T} + \frac{k}{K})) \tag{A.3}$$

As a result, the weight for each explanation has evenly spaced phases and would peak at different time period.

A.2.3 Surface form similarity prompt

We include our prompt used to evaluate the surface form similarity between the predicted explanation $\hat{e}_k$ and the reference explanation e^*_k in Figure A.3.

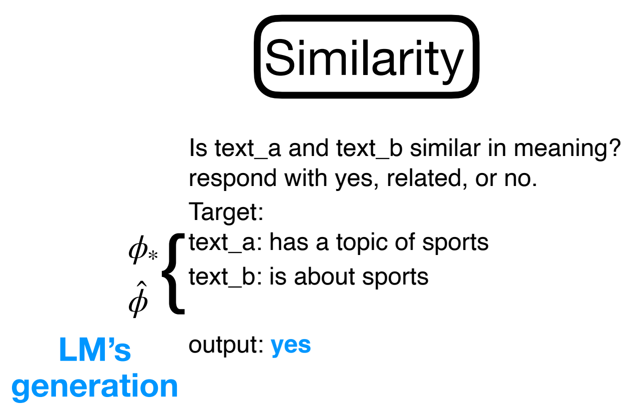

Figure A.3: The prompt template used to evaluate the surface form similarity between the predicted explanation $\hat{e}_k$ and the reference explanation e^*_k.

F1/Surface	AGNews	DBPedia	NYT	Bills	Wiki	Average
OneHot	0.87/0.53	0.54/0.51	0.48/0.53	0.26/0.51	0.36/0.47	0.50/0.51
OtherEmb	0.85/0.70	0.62/0.54	0.59/0.53	0.43/0.59	0.48/0.53	0.60/0.59
Ours	0.86/0.62	0.68/0.54	0.70/0.63	0.45/0.52	0.51/0.53	0.64/0.57
K-means	0.83/—-	0.75/—-	0.72/—-	0.41/—-	0.53/—-	0.65/—-
TopicModel	0.56/—-	0.52/—-	0.49/—-	0.25/—-	0.35/—-	0.43/—-

Table A.3: We compare our method to classical clustering approaches that do not generate natural language explanations (K-means and TopicModel), where "——" means that the surface form metric is undefined since these methods do not output natural language explanations. We find that on average, our method is close to K-means and significantly outperforms TopicModel under the F1 similarity metric, while generating natural language explanations for each cluster. We also compare our method to using one-hot text embedding, and find that our method is significantly better; this indicates that the use of informative text embedding is crucial to performance.

F1/Surface	topic	lang	locat	all	time-avg	classification
One-hot	0.63/0.55	0.51/0.57	0.51/0.62	0.66/0.60	0.58/0.59	0.72/0.68
OtherEmb	0.68/0.58	0.56/0.59	0.49/0.68	0.71/0.68	0.61/0.63	0.73/0.67
Ours	0.67/0.57	0.62/0.70	0.55/0.68	0.72/0.64	0.64/0.65	0.73/0.70

Table A.4: Our method consistently outperforms a variant that uses one-hot text encoding as b_x rather than neural embeddings. This indicates that using informative text embedding is crucial to performance.

A.2.4 Additional Results on Our Benchmark

Our method is similar or better than classical methods such as topic modeling or K-means. We report the performance of K-means clustering and topic modeling under the clustering benchmark in Table A.3. on average, our method is close to K-means and significantly outperforms TopicModel under the F1 similarity metric, while generating natural language explanations for each cluster.

Takeaway 5: Using informative text embedding is crucial to performance. We used neural embeddings when optimizing the continuous representation of the explanations. Does our algorithm actually make use of the information in the feature embeddings? To investigate this question, we ran an ablation of using one-hot text embeddings rather than neural embeddings (OneHot), which do not encode any information about the similarity between text samples. We report the performance in Table A.3 and A.4; across all settings, using neural embeddings consistently outperforms OneHot.

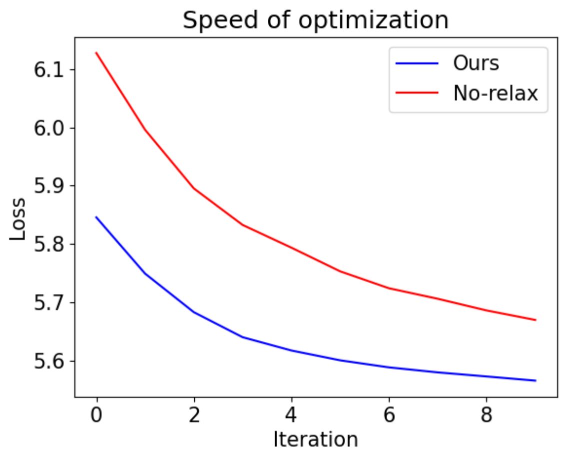

Figure A.4: We plot how the negative PREDPOWER decreases across different iterations with and without relaxation (that explores using random explanations). We find that using relaxation significantly speeds up optimization.

To make sure that this takeaway is general and not specific to one embedding model, we run our method with another text embedding model, `all-mpnet-base-v2`[3] and report the performance as the `OtherEmb` row. We find that using this neural embedding also outperforms `OneHot` in most cases, indicating that our conclusion is robust.

Takeaway 1,2,and 4 are statistically significant. To compare the performance between our method and each variant, we conduct a one-sided paired t-test on their performance (F1-similarity) on each dataset, where the performance on each dataset is the averaged performance across five runs. Takeaway 1, 2, 4 has a p-value of 5×10^{-4}, 2×10^{-4}, and 6×10^{-3}, respectively.

A.2.5 Detecting Self-Reinforcing Trends in Machine Learning System

Machine learning systems sometimes have unintended side effects and reinforce themselves. [45] illustrated an example failure mode, where a group of users is discriminated against and thus leave a platform, causing a ML system to discriminate them further and hence drive them away.

As a concrete illustration, let us imagine a social platform Y where users post tweets and the platform will display the most engaging ones; suppose there are two groups of users, one conservative and one liberal, where both groups prefer more engaging tweets but also tweets that agree with their political stances. Y implements a recommender system, which trains a classifier to predict whether a tweet is likely to be preferred by a random user, and

[3] `https://huggingface.co/blog/1b-sentence-embeddings`

then the platform Y will promote these tweets. If the two groups of users are balanced, the optimal classifier will make Y promote tweets that are engaging and place little weights on the political slant.

However, if there are fewer liberal users, the classifier will be biased and Y will promote conservative tweets more and focus less on whether the tweet is engaging or not. The liberal users will find the promoted tweets less attractive, thus leaving the platform Y. As a result, fewer liberal users will stick to Y, thus making the classifier more biased.

Now we provide a proof-of-concept experiment to illustrate how our time series model can be applied to detect such a reinforcing trend. We first simulate the setup above and obtain the tweets promoted by platform Y across time, and then apply our time series model to extract temporal trends from these tweets. Suppose there are two groups of users, liberal and conservative. At $t = 0$, the fraction of liberal users is $\lambda_0 = 0.5$ and is the same as that of conservative users. To simulate the setup above and obtain the tweets promoted by platform Y across time, we assume that at each time step t, we will sample 2,000 tweets, where each tweet is a 2D datapoint with the x-value a random integer from [-1, 1] indicating whether it is liberal, non-political, or conservative, and y-value a random integer from [-2, 2] indicating how engaging the tweet is. For each tweet, we obtain a label of $y = 1/0$ if the user likes a tweet, and the user's probability for liking a tweet is defined by $\sigma(ux + 0.5y)$, where $u = 1$ if the user is conservative and 0 otherwise. We then train a logistic regression classifier to predict whether a random user will like a tweet and the platform Y will promote the tweets with the top 20% score. Let the fraction of tweets non-liberal tweets be a_t and non-conservative tweets be b_t, then the fraction of liberal users for the next round will be determined by:

$$\lambda_{t+1} = \frac{b_t \lambda_t}{b_t \lambda_t + a_t(1 - \lambda_t)}, \tag{A.4}$$

which models how the group size will increase/decrease depending on whether the platform promotes tweets that agree with their views. We run this process for $T = 10$ and gather all the 2D datapoints promoted by platform Y.

We then turn these two-dimensional datapoints into text samples x. We ask the gpt-4o to write a liberal, non-political, or conservative tweet based on the x-value; then we ask gpt-4o to make it more/less engaging based on the y-value. For example, for a 2D value of (-1, 2), we ask gpt-4o to write a liberal tweet and ask it to make it more engaging two times; if the value is (1, -2), we ask gpt-4o to write a conservative tweet and then ask it to make it less engaging two times.

We now have a list of tweets across time, and we directly apply our time series model with $K = 3$ to extract trends from them. Our time series model find that there is an increasing amount of tweets that "*expresses patriotic sentiments*" and "*champions specific policies*", but a decreasing amount "*poses a question to engage the audience*". These explanations exactly recover all the underlying trends, that the self-reinforcing effect make the tweets more conservative, less non-political, and less engaging.

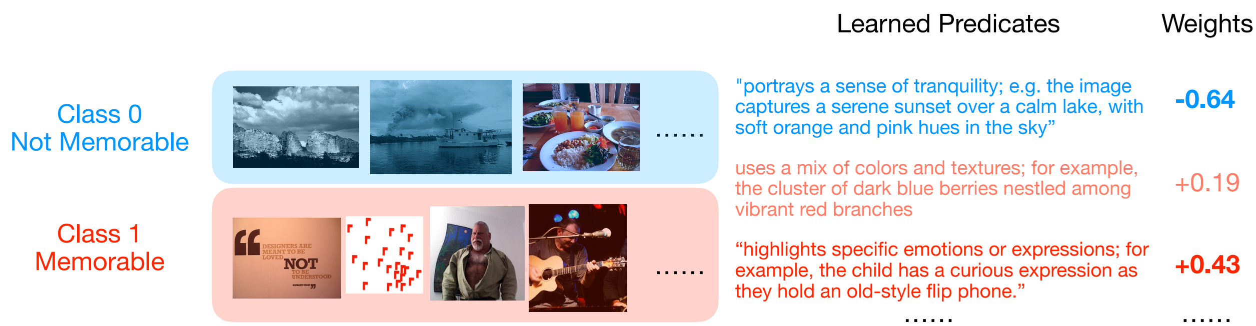

Figure A.5: We apply our classification model from Section 3.1.2 to explain what visual features make images more memorable [53]. Consistent with previous findings, we find that tranquil scenes make an image less memorable, while emotions and expressions are more memorable.

A.2.6 More Applications

A.2.6.1 Applying Our Classification Model to Images: Explaining Memorable Visual Features

Our framework is applicable to the vision domain since a natural language explanation e can extract binary values from an image x. For example, for the rightmost image x in Figure A.5 right, the explanation "*portrays a person*" evaluates to 1, i.e. $[\![e]\!](x) = 1$, while "*contains texts*" evaluates to 0.

We present an application of our classification model from Section 3.1.2 to images, which learns linear weights over a set of visual features described by natural language explanations. This model has also appeared in prior works: our model is equivalent to the language-based concept bottleneck model proposed by [112, 94]; additionally, when $K = 1$ and $C = 2$, our model is equivalent to the VisDiff framework [34], which finds a single explanation to discriminate samples from two classes of images.

We apply our classification model to the LaMem dataset [53] to understand what visual features make an image more memorable, an interesting cognitive science question. We now define the samples x_i and their class labels y_i to run our classification model. In LaMem, each image is associated with a score of how memorable it is as measured by whether humans can remember seeing it in the past; to make implementation easier, we set x_i to be the caption of the image and $y_i = 1$ if x_i has an above median score and $y_i = 0$ otherwise. To fit our classification model, we set $K = 6$, use `gpt-4o` as the discretizer, and use `gpt-4o-mini` to compute denotation.

We present three learned explanations in Figure A.5. We find that an image is less memorable if it "*portrays a sense of tranquility; e.g. the image captures a serene sunset over a calm lake, with soft orange and pink hues in the sky ...*", and more likely to be memorable if it " *highlights specific emotions or expressions; for example, the child has a curious expression*

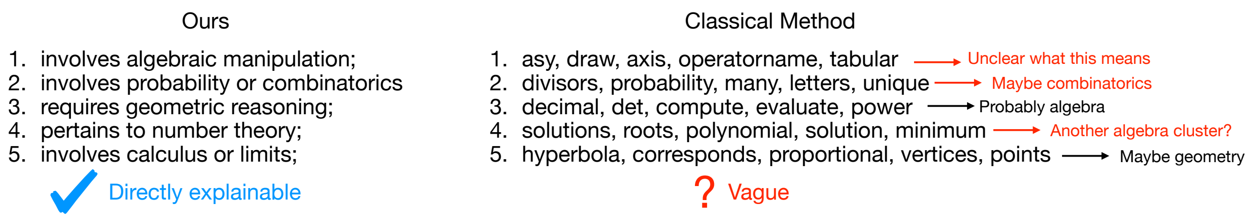

Figure A.6: We cluster the MATH dataset [47] and compare our method (left) to a classical method (right), which first clusters via K-means and then explains each cluster via unigram analysis. Our method directly explains complex concepts, while the classical method delivers vague explanations.

...". These results are consistent with the previous manual analysis from [53], suggesting the validity of our results.

A.2.6.2 Explaining Abstract Properties via Easy Steering: Clustering Problems Based on Subarea

Can our framework explain more abstract aspects of a sample x: e.g. subarea, the type of knowledge required to solve a math problem x? We show this is feasible by applying our model from Section 3.1.2 to cluster math problems and steering it to focus on explaining subareas. Meanwhile, classical methods struggle to explain abstract aspects.

We apply our clustering model from Section 3.1.2 to cluster the MATH dataset [47] based on subareas. The MATH dataset contains five labeled subareas,[4] and we hope our model can recover all of them: `Algebra`, `counting_and_probability`, `geometry`, `number_theory`, and `precalculus`. To steer our clustering model to explain subareas, we simply prompt the discretizer LLM "*I want to cluster these math problems based on the type of skills required to solve them.*" We set $K = 5$, using `gpt-4o` to discretize and `gpt-4o-mini` to compute denotation.

We present the outputs of our model on the left of Figure A.6. With simple prompting, our model is successfully steered to cluster based on subareas and recovers all five labeled subareas from the MATH dataset. Note that our explanations can explain abstract properties that have no word overlap with the samples that match them: for example, the math problems that "*requires geometric reasoning*" (Figure 6 left 3) usually contain neither of the word "geometric" or "reasoning".

We compare our method to a classical baseline that first clusters the samples and then explains each cluster with representative words. In this baseline, we first perform K-means clustering on the neural embeddings of x and assign each sample to a cluster; we then extract representative words by first running a unigram regression to predict whether a sample belongs to the cluster and then selecting words with the most positive weights. We present

[4] after merging similar categories that differ in levels of difficulty

the word-based explanations on the right in Figure A.6. Overall, significant guesswork is needed to interpret the meaning of each word-based cluster (e.g., it is unclear what cluster 1 represents in Figure A.6 right), while the explanations generated by our algorithm are directly explainable. Our framework can be steered to explain more abstract aspects of a sample x and significantly improve over classical methods.

A.2.7 Implementation Details on Open-Ended Applications

To obtain the best outputs, we discretize with `gpt-4o` and compute denotations with `claude-3.5-sonnet`. Since we aim to analyze user queries, we explicitly prompt `gpt-4o` to generate detailed explanations about use cases when discretizing continuous explanations. See A.7 for the full prompt.

A.2.8 Taxonomizing User Applications.

Implementation Details We cluster 1024 dialogues with $K = 6$ and $S = 5$ We only cluster on a small set of dialogue turns because it is slow to compute denotations: 1) we explored in total around 100 explanations and this amounts to $\sim 100 \times 1024 = 100$K LLM API calls, and 2) we used language model API (`claude-3.5-sonnet`), rather than a local small language model (`google/flan-t5-xl`), to compute denotations, since this is the cheapest model that we feel confident that it can handle more sophisticated explanations.

Full Results. We present the full results in Figure A.8 Overall, we find that our framework can generate sophisticated explanations that classical methods cannot generate. However, some cluster descriptions are significantly overlapping (e.g., category 0, 1, and 0.D); additionally, some sub-clusters are not indeed subsets of their parent categories (e.g., subcategory 2.D does not belong to category 2). Future work can improve the taxonomy by 1) deduplicating semantically similar descriptions or more heavily penalizing cluster overlap, and 2) steer the explanation generation process so that the descriptions for the subclusters are indeed subsets of their parent descriptions.

A.2.8.1 Characterizing Temporal Trends.

Implementation Detail. We run our time series model on 1K dialogue turns with $K = 4$ and the number of iterations S to be 10 to identify temporal trends in user applications. We obtain the smoothed frequency curve by updating with the follow equation:

$$f_t = 0.99 \cdot f_{t-1} + 0.01[\![e_k]\!](x_t); \quad f_0 = \frac{1}{100}\sum_{t=1}^{100}[\![e_k]\!](x_t) \tag{A.5}$$

We obtain the shaded area in Figure 3.4 by shuffling x_t and find the highest and lowest f values across 100 random runs.

```
Here is a corpus of user queries each associated with a score. The queries are sorted from the lowest to
 the highest score.

{samples_in_prompt_w_scores}

I am a machine learning researcher that builds chat bots. Here is a list of first turns of user queries,
 and I want to cluster them based on their applications. Note that I am only interested in applications:
 for example 'refers to pop culture' is not an application, but 'wants to ask for information about a po
p culture entity' is an application. Each description should start with 'the user wants to ....'

We want to understand what kind of queries achieve a higher score, so please suggest descriptions about
the queries that are more likely to achieve higher scores.
Please suggest me at most {num_descriptions_per_prompt} descriptions, one in a line, starting with "-" a
nd surrounded by quotes "". Each of them needs to be a predicate about a query followed by an explanatio
n and an example that satisfies the predicate, for example:
- "the user wants to request for email or message composition; specifically, the query involves a reques
t for drafting an email or message with specific content and intent. For example, 'write an email to ter
minix llc with a proposal for cooperation...'."
- "the user wants to request technical code/script writing; specifically, the query demands the creation
 or modification of a script or code. For example, 'create a dockerfile based on this script.'"

Do not output anything else. Please do not mention score in your example. (Note that the examples might
not be goal related, and your response should be both formatted correct as above and related to the goal
.)

Please generate the response based on the given datapoints as much as possible. We want the descriptions
 to be relatively objective and can be validated easily, e.g. "is surprising" means different things for
 different people, so we want to avoid such descriptions. It should also be a predicate on a single quer
y (rather than a statement about a comparison); for example, instead of saying "uses more polite languag
e", the generation should be "uses polite language". Sometimes KeyInfo is provided to help you make come
 up with better responses (though it might also be unavailable).

Again, here's the goal.
I am a machine learning researcher that builds chat bots. Here is a list of first turns of user queries,
 and I want to cluster them based on their applications. Note that I am only interested in applications:
 for example 'refers to pop culture' is not an application, but 'wants to ask for information about a po
p culture entity' is an application. Each description should start with 'the user wants to ....'

Your responses are:
- "
```

Figure A.7: A discretizer prompt that explicitly asks LLM to explain user applications. E.g., at the end of the prompt, we explicitly requested the explanations to start with "*the user wants to...*".

A.2.8.2 Advantages and Limitations

Overall, we find that our framework allows us to define sophisticated models (e.g. time series) and can output highly sophisticated explanations, which can include detailed explanations and examples. Therefore, when implemented perfectly, its utility has a much higher upperbound than classical methods such as n-gram Bayes/regression or topic models.

However, the comparison between our method and classical methods is only qualitative: we only eye-balled the outputs from our method and the classical methods in Section 3.4 and did not quantitatively measure how useful they are in practice. Therefore, even if our method does outperform classical methods such as topic model on our benchmark (Table A.3), it might not directly translate to how useful it is in real-world applications. Additionally, we

```
===============
0 the user wants to request a descriptive screenplay writing; specifically, the query asks for the generation of screenplays including dialogues and deta
iled background. For example, 'Write a very long, coherent, elaborate, descriptive and detailed screenplay...'
        0.A the user wants to request narrative writing from a personal or first-person perspective; specifically, the query asks for detailed descripti
ons and thoughts from the viewpoint of a character. For example, 'Write a long, detailed, original, imaginative and interesting scene narrated by Celesti
ne from the first person perspective.'
        0.B the user wants to request role-playing; specifically, the query involves engaging in a narrative role-playing scenario with predefined chara
cters. For example, 'We'll role play as characters from the walking dead game season 4, you'll provide answers for Clementine and I'll take the role of L
uis...'
        0.C the user wants to request dialogue writing; specifically, the query focuses on generating specific conversations between characters. For exa
mple, 'Write dialogue from a scene from the animated teen "Jane", where 14 year old Jane, Jane's 14 year old girlfriend Sam, Jane's 14 year old friend an
d neighbour Aaron Ling and Aaron's 12 year old sister Molly-Rose Ling hanging out at school...'
        0.D the user wants to request content rewriting; specifically, the query asks for altering or reimagining existing content within a new context.
 For example, 'Can you rewrite 1972's Ben as a Tales From Crypt movie, it goes the same way but write the intro and outro of the Crypt Keeper?'

===============
1 the user wants assistance with writing or editing text; specifically, the query involves tasks like composing positive feedback or rewriting content. F
or example, 'Please re-write this as a positive feedback for Stelina, praising her.'.
        1.A the user wants to create new fictional content; specifically, the query involves generating original stories, scripts, or characters. For ex
ample, 'Write dialogue from a scene from the animated teen Jane, where 14 year old Jane, Jane's 14 year old girlfriend Sam and Jane's 14 year old friend
and neighbour Aaron Ling hanging out at school when Jane and Sam finds out Aaron has a crush.'
        1.B the user wants to request dialogue writing for fictional characters; specifically, the query requests the creation of conversation between c
haracters in a specific scenario. For example, 'Write dialogue from a scene from the animated teen "Jane"...'.
        1.C the user wants to request a short response or reply composition; specifically, the query involves crafting a concise message to send in a di
scussion. For example, 'give me a response to *smiles* Yes, I'm doing well. How about you?'.
        1.D the user wants to request content rewriting and optimization; specifically, the query involves improving the language, structure, or adding
references to existing text. For example, 'summarize this parts for method section in an article and rewrite it to improve and add some references'.

===============
2 the user wants to request feedback or evaluation; specifically, the query involves asking for re-writing or giving feedback on a piece of text. For exa
mple, 'Please re-write this as a positive feedback for Stelina, praising her.'
        2.A the user wants to request a concise and clear response for a given message; specifically, the query involves crafting a short and direct rep
ly for use in a discussion. For example, 'give me a response to Absolutely. Compassion, understanding, and respect...'.
        2.B the user wants to request for grammar and spelling correction; specifically, the query involves correcting grammatical and spelling errors i
n a given text. For example, '( there was an assistance in our village about Stationary and Hygiene's for the students of (Nasozai Markazai Lesa )...'.
        2.C the user wants to request an academic rewrite of text; specifically, the query involves rephrasing content to meet academic standards or exp
ressions. For example, '使下列文字符合学术表达方式 [text in Chinese].'
        2.D the user wants to request for summarization of information; specifically, the query involves condensing information into key points or a bri
ef summary. For example, 'In point form, simply summarize key insights on barriers to participate in procurement from the below text at an 8th-grade read
ing level and provide brief supporting quotes with ID numbers'.

===============
3 the user wants to request technical code/script writing; specifically, the query demands the creation or modification of a script or code. For example,
 '逐行解释以下代码class MediaPipeFace:'.
        3.A the user wants to request data manipulation or computation; specifically, the query involves operations like calculations, data processing,
or data transformation. For example, 'how to get percentage of maxhealth to health in java'.
        3.B the user wants to request debugging assistance; specifically, the query involves asking for help to identify and fix errors or issues in a g
iven code. For example, '帮我看看这段代码有啥错误'.
        3.C the user wants to create or enhance UI/UX components within web development; specifically, the query requires designing, styling, or improvi
ng user interface elements. For example, 'can you do some nice font for it looking'?
        3.D the user wants to request the creation of a new code/script to perform a particular task; specifically, the query asks for writing code from
 scratch to achieve a specific functionality. For example, 'write a code for DES Encryption from scratch in python'.

===============
4 the user wants to format a description of an image generation prompt; specifically, the query involves structuring a prompt for use with image generati
on tools. For example, 'Here is a midjourney prompt formula...'
        4.A the user wants to create AI-generated content based on language and localization; specifically, the query involves creating prompts in diffe
rent languages while maintaining specific design instructions. For example, 'Imagine you're an expert Graphic Designer and have experience in 男孩 t-shir
t printing and also an expert Midjourney AI Generative prompt writer...'
        4.B the user wants to request for creative content generation; specifically, the query involves asking for extensive and imaginative description
s or stories about fictional characters or scenarios. For example, 'write a detailed fusion scenario of two anime characters.'
        4.C the user wants to request graphic design prompts for Midjourney AI; specifically, the query involves generating prompts for creating specifi
c types of images. For example, 'Imagine you're an expert Graphic Designer and have experience in 男孩 t-shirt printing and also an expert Midjourney AI
Generative prompt writer...'.
        4.D the user wants to request a comparison or fusion of different art styles or characters; specifically, the query involves merging distinct st
yles or characters into a single concept. For example, 'Freedom planet and Dragon ball: Goku betrayed and join the Freedom planet with beerus, whis, zeno
 and his guards. part 1'.

===============
5 the user wants to request information or answers regarding a specific topic; specifically, the query involves seeking knowledge or clarification. For e
xample, 'what are some open source operating systems.'
        5.A the user wants to ask for historical information on a specific topic; specifically, the query seeks details about events, practices, or item
s from the past. For example, 'Can you give me some summarized info on sugar cane 100 years ago?'
        5.B the user wants to ask for definitions or distinctions; specifically, the query seeks to clarify the meaning or differences between terms. Fo
r example, 'What is the difference between polygamy, polyandry, polygyny, and polycules?'
        5.C the user wants to obtain scientific or factual information; specifically, the query is seeking accurate data or explanations relevant to aca
demic or scientific fields. For example, 'describe creature scientific name is felis achluochloros, evolved from feral creatures of felis catus, includin
g species name.'
        5.D the user wants to request technical issue resolution; specifically, the query involves troubleshooting or solving a problem with software or
 hardware. For example, 'how do I make a circle in CSS that sits on top of other elements and applies a gray scale filter to all of the elements intersec
ting its area?'
```

Figure A.8: The full taxonomy that our algorithm generates to categorize user applications from a corpus of user chatbot queries.

did not compare to modern taxonomy construction methods such as [118], which involves a lot of task-specific engineering; our method is model-agnostic and was applied out-of-the-box to construct the taxonomy. Section 3.4 only shows that our method can generate more sophisticated natural language explanations, which presents a higher upper bound of what our method could potentially achieve.

In terms of the weakness of our method, our method is currently slow, as its performance highly depends on LLMs to compute denotations correctly, it outputs semantically similar explanations that add little information, and it is hard to control the explanations to satisfy certain properties (e.g. being a subset of a parent category). We look forward to future works that can address these problems and realize the full potential of this framework. For example, to remove similar explanations, one could prompt a language model to check the pairwise surface similarity between two explanations; to speed up inference, one can distill a smaller but much more efficient model specialized for computing denotations.

A.2.9 A Formal Description of Our Algorithm

A formal description of our algorithm can be seen in Algorithm 1.

A.2.10 Additional Details of Our paper

A.2.10.1 Limitations of Our Framework and Our Experiments

As mentioned in Appendix A.2.8.2, our current system is slow, as its performance highly depends on the LLMs to compute denotations correctly, it outputs semantically similar explanations that add little information, and it is hard to control the explanations to satisfy certain properties (e.g. being a subset of a parent category). Our experiments are limited since it assumes that the datasets and statistical models we used are reflective of real world application. We made our best effort to gather text clustering datasets that are commonly used in the literature (e.g. from [104, 87]) and defined models that are plausibly useful for practitioners. Additionally, note that our evaluation on topic clustering is more comprehensive than the prior work [104] by including two new datasets (`Bills` and `Wiki`); additionally, we used the exact same hyper-parameter across all clustering tasks, while [104] changed the hyper-parameters for different datasets.

A.2.10.2 Cost of the Experiments

All of the experiments ran in Section 3.3 are estimated to cost at most 200 GPU hours on an A100 GPU with 40GB memory, and cost less than $20 of API credit for `gpt-3.5-turbo`. The experiments in Section 3.4 costs at most $50 in API inference credit, but we were constrained by rate limit.

Algorithm 1 A formal description of our algorithm. **Argument**: S is the number of steps we use to run our algorithm. **Output**: $\text{var\_}e_{1\ldots K}$ is the list of K explanations that we maintain, optimize, and return at the end of the algorithm. $\hat{w}$ are other parameters.
We first optimize the relaxed continuous explanations and discretize them (Line 3-10), and then iteratively refines the explanations (Line 11-21). During iterative refinement, we first find the least useful explanation k (Line 12), then we only optimize the continuous representation of the least useful explanation while fixing other discrete explanations (Line 14 - 17); finally we discretize the kth explanation (Line 18, 19)

1: **Arguments**: S
2: **Output**: $\text{var\_}e_{1\ldots K}, \hat{w}$
3: $\tilde{e}_{1\ldots K} \leftarrow$ randomly sample K embeddings m_x to initialize $\tilde{e}$
4: **for** $t = 1$ to 10 **do**
5:     $\hat{w} \leftarrow \texttt{OptW}(\tilde{e}_{1\ldots K})$
6:     $\tilde{e}_{1\ldots K} \leftarrow \texttt{OptRelaxedE}(\hat{w})$
7: **end for**
8: **for** $k = 1$ to K **do**
9:     $\text{var\_}e_k \leftarrow \texttt{Discretize}(\tilde{e}_k)[0]$
10: **end for**
11: **for** $s = 1$ to S **do**
12:     $k \leftarrow \text{argmax}_{k'}\texttt{Fitness}(\text{var\_}e_{-k'}, [\![e_{k'}]\!] = 0)$
13:     $\tilde{e}_k \leftarrow$ randomly sample an embeddingm_x
14:     **for** $t = 1$ to 10 **do**
15:         $\hat{w} \leftarrow \texttt{OptW}(\text{var\_}e_{-k}, \tilde{e}_k)$
16:         $\tilde{e}_k \leftarrow \texttt{OptRelaxedE}(\text{var\_}e_{-k},\ \hat{w})$
17:     **end for**
18:     $C_k \leftarrow \texttt{Discretize}(\tilde{e}_k)$
19:     $\text{var\_}e_k \leftarrow \text{argmax}_{e' \in C_k \cup \{\text{var\_}e_k\}}\texttt{Fitness}(\text{var\_}e_{-k}, e_k = e')$
20:     $\hat{w} \leftarrow \texttt{OptW}(\text{var\_}e)$
21: **end for**
22: **return** $\text{var\_}e, \hat{w}$

A.2.10.3 Licenses for Existing Datasets

[92, 121, 52]) AG-News [121] has unknown license, the DB-Pedia dataset is released under Creative Commons Attribution Share Alike 3.0, the NYT dataset is distributed by LDC under the LDC's generic non-member license, the Bills dataset [52] are considered public domain works, and the Wiki dataset is licensed under CC BY-SA 4.0.

A.2.10.4 License for the Assets Provided by Our Paper

Our code will be shared under CC BY-SA 4.0.

A.2.10.5 Broader Impacts

This paper presents work whose goal is to advance the field of Machine Learning. Our framework could potentially make machine learning systems more explainable, thus making them safer, more trustworthy and easily auditable. On the other hand, however, the learned explanations only reflect correlation rather than causation learned from data, and hence requires careful interpretation. Given that the performance of our model-agnostic method is still far from perfect and it is unclear how human users would use them in real world applications, the algorithm presented in this paper should only be used for research and not deployed in practice.

A.3 Chapter 4 Appendix

A.4 Appendix

A.4.1 Human evaluation

We show the human annotation instruction for counterfactual simulatability in Figure A.9 (StrategyQA) and Figure A.10 (SHP), and the annotation instruction for plausibility in Figure A.11. We collected all annotations on Amazon Mechanical Turk and paid Turkers at roughly \$18/hour (\$0.6/HIT).

A.4.2 LLM Prompts

In this section we show the prompts used for all experiments for reproducibility. We use the same prompt for GPT-3.5 and GPT-4.

StrategyQA-Explanation Generation.

> Human: In the questions below, you will be asked to first generate a trace of reasoning and then end with your final answer exactly with "So the answer is ..." yes or no. Strictly follow the example format below and do not say anything else.
>
> Assistant: here is my response. okay.
>
> Human: Q: Yes or no: Do hamsters provide food for any animals?
> A:
>
> Assistant: here is my response. Hamsters are prey animals. Prey are food for predators. Thus, hamsters provide food for some animals. So the answer is yes.
>
> Human: Q: Yes or no: Could Brooke Shields succeed at University of Pennsylvania?

Instructions (Click to Unfold/Fold)

Task Description

Thank you for participating in this task!

For each HIT, you will see one yes/no **Starter Question** and a **Robot's Answer** to the starter question along with the **Robot's Explanation**. Then, you will reason about the robot's answer to a **Follow-up Question**.

Here's a very simple example:

Starter Question	Can sparrows fly?
Robot's Explanation	Because all birds can fly, sparrows can fly. So the answer is yes.
Robot's Answer	Yes
Follow-up Question	Can penguins fly?

Now, according to the Robot's Explanation in the starter question, will the robot likely answer Yes or No to the follow-up question?
You should choose **Yes**. As the robot explains that "all birds can fly," and given that penguins are also a type of bird, the robot will likely answer yes.

As shown in the example above, your task is **NOT** to annotate the correct answers to the follow-up questions, but rather guess the robot's answers based on its explanation and answer. Now, we will show you how to do this task exactly.

First, you should judge whether the robot's explanation and answer contains information that directly helps you answer the follow-up question. Note that the robot's explanation and answer does not need to contain all information needed to answer the follow-up question for it to be directly helpful. We will show two examples below to help your understanding.

Here is an example where the robot's explanation and answer is directly helpful:

Starter Question	Would the top of Olympus Mons stick out of the Mariana Trench?
Robot's Explanation	The Mariana Trench ~11 kilometers deep in the ocean. Olympus Mons is ~22 kilometers tall. Since 22 > 11, the top of Olympus Mons would stick out of the Mariana Trench. The answer is yes.
Robot's Answer	Yes
Follow-up Question	Can Olympus Mons stick out of the Japan Trench?

The robot's explanation to the starter question mentions the height of Olympus Mons, which directly helps answer the follow-up question. Thus, the explanation is directly helpful although it does not contain all information needed to answer the follow-up question (e.g., the depth of the Japan Trench).

Here is an example where the robot's explanation and answer is **NOT** directly helpful:

Starter Question	Can citrus grow in Ulaanbaatar?
Robot's Explanation	Citrus trees can grow in Ulaanbaatar. Thus, citrus can grow in Ulaanbaatar. So the answer is yes.
Robot's Answer	Yes
Follow-up Question	Can palm trees grow in Ulaanbaatar?

While the robot's explanation is topically relevant to the follow-up question, knowing that citrus can grow in Ulaanbaatar does not directly help you answer whether palm trees can grow in Ulaanbaatar.

Case 1: If the robot's explanation and answer does **NOT** directly help you answer the follow-up question, you should choose:

- **Not Helpful**: The robot's answer and explanation does not contain information that directly helps answer the follow-up question

Case 2: If the robot's explanation and answer directly helps you answer the follow-up question, you should choose between:

- **Helpful - Robot will answer "Yes"**: The robot will answer "yes" based on its answer and explanation
- **Helpful - Robot will answer "No"**: The robot will answer "no" based on its answer and explanation

Here are two rules you should follow. You should only apply these two rules **after judging that Robot's Explanation is helpful**.

- **Rule #1**: Stick to the Robot's reasoning/claims even if it's incorrect.
- **Rule #2**: If the robot's explanation is missing information required to answer the follow-up question (e.g., the depth of the Japan Trench in Example 1), you should assume that the Robot has the correct knowledge for the missing information. You may use a search engine to find out the correct information.

Example:

Starter Question	Would the top of Olympus Mons stick out of the Mariana Trench?
Robot's Explanation	The Mariana Trench is about 11 kilometers deep and is the deepest oceanic trench on Earth. Olympus Mons is about 22 kilometers tall. Thus, the top of Olympus Mons would stick out of the Mariana Trench. So the answer is yes.
Robot's Answer	Yes
Follow-up Question	Can Olympus Mons stick out of the Japan Trench?

Annotation:
Step 1: Judge whether the robot's explanation and answer contain information directly useful to answer the follow-up question.
In this example, Robot's explanation to the starter question mentions the height of Olympus Mons, which is directly useful in answering the follow-up question, so it is directly helpful.
Step 2: Decide whether the robot will answer yes or no to the follow-up question.
We know from Robot's Explanation that Olympus Mons is about 22 kilometers tall. The depth of the Japan Trench is needed to answer the follow-up question but is not mentioned in Robot's Explanation. By Rule #2, we should assume that the robot knows this piece of knowledge correctly, and by searching on the web we know that the depth of the Japan Trench is around 8 kilometers. Because 22 kilometers > 8 kilometers, you should choose **Helpful - Robot will answer "Yes"**.

Figure A.9: Human annotation instructions for counterfactual simulatability on StrategyQA.

A:

Assistant: here is my response. Brooke Shields went to Princeton University. Princeton University is about as academically rigorous as the University of Pennsylvania. Thus, Brooke Shields could also succeed at the University of Pennsylvania. So the answer is yes.

Instructions (Click to Unfold/Fold)

Task Description

Thank you for participating in this task!
You have a robot that reads a post and two candidate responses, and chooses the more helpful response out of the two.

Here is an example (one post + two candidate responses) and the robot's choice and explanation.

Context	Hello all, I was interested in making this super decadent pound cake to give to a friend for her birthday. I would like to add just a few fresh blueberries. Any guidance on this?
Response 1	Make the batter as normal, and then toss the blueberries in a little flour so that they have a light dusting around them, then fold them into your batter and bake.
Response 2	commenting to follow, i have a citrus pound cake recipe that i'd like to add blueberries to but afraid to make the jump without tips lol
Robot's Explanation	Candidate Response 1 is more helpful because it provides specific instructions on the task asked in the context.
Robot's Choice	Response 1

For each HIT, you will see one **Starter Example** containing the **Context**, **Response 1**, and **Response 2**. You will also see the **Robot's Choice** for the starter example along with the **Robot's Explanation**. Your task is to reason about the robot's choice to a follow-up question.
Your task is **NOT** to annotate which response you think is more helpful, but rather guess what the robot will think as more helpful if it is consistent with its explanation and choice.

For each follow-up example, you will choose between:

- **Response 1**: If the robot will choose Response 1
- **Response 2**: If the robot will choose Response 2
- **Robot is equally likely to choose Response 1 or 2**: If the robot could choose either response based on its choice and explanation in the starter example

A rule-of-thumb: sometimes reading the robot's explanation before the starter example will save you some time.

We will show two examples below to help your understanding. Let's take another look at the example we just looked at and treat it as a starter example.

Example #1:

Starter Example

Context	Hello all, I was interested in making this super decadent pound cake to give to a friend for her birthday. I would like to add just a few fresh blueberries. Any guidance on this?
Response 1	Make the batter as normal, and then toss the blueberries in a little flour so that they have a light dusting around them, then fold them into your batter and bake.
Response 2	commenting to follow, i have a citrus pound cake recipe that i'd like to add blueberries to but afraid to make the jump without tips lol
Robot's Explanation	Candidate Response 1 is more helpful because it provides specific instructions on the task asked in the context.
Robot's Choice	Response 1

Follow-up Example:

Context	I want to create a T-shirt with a design I made, but I don't know how to print the design onto the fabric. Can anyone recommend a method? Thanks!
Response 1	I think you should look up some DIY videos on YouTube. You might find something helpful there.
Response 2	You should choose a high-quality HTV that is compatible with your fabric, print your design onto the HTV using an inkjet printer, cut out your design using a craft knife or vinyl cutter, and pply heat and pressure evenly over the design for the recommended time and temperature.

Correct Annotation:
The robot's choice and explanation shows that it has a preference for responses with more specific instructions on the task. Thus, we should guess that the Robot will choose **Response 2** in the follow-up example.

Example #2:

Starter Example

Context	What is the philosophical basis for respect of human corpses? Or is it just cultural? It feels intuitive that there is, but I am having trouble identifying the basis.
Response 1	I think we should respect corpses because violating them would be a violation of human dignity, even though the person directly violated is not able to experience it.
Response 2	Consequentialist argument: People will be upset by your treatment of the body in unaccepted ways. Virtue argument: The way we treat inanimate objects might cultivate habits or traits in us that affect how we treat people or other things.
Robot's Explanation	Candidate Response 2 is more helpful because it explores various arguments (both consequentialist and virtue-based).
Robot's Choice	Response 2

Follow-up Example:

Context	I've been pondering over this idea of personal identity lately. Is personal identity an illusion?
Response 1	The concept of personal identity has a biological basis in our brains, which encode memories and store information about our experiences.
Response 2	Personal identity may be seen as an illusion in the sense that it is a mental construct shaped by our experiences.

Correct Annotation:
The robot's choice and explanation show that it has a preference for responses that explore various arguments. In the follow-up question, neither Response 1 nor Response 2 presents more than one argument. Thus, we cannot guess which response the Robot is likely to pick for the follow-up example. So you should annotate **Robot is equally likely to choose Response 1 or 2**.

Figure A.10: Human annotation instructions for counterfactual simulatability on SHP.

Human: Q: Yes or no: Hydrogen's atomic number squared exceeds number of Spice Girls?
A:

Assistant: here is my response. Hydrogen has an atomic number of 1. 1 squared is 1. There are 5 Spice Girls. Thus, Hydrogen's atomic number squared is less than 5.

Instructions (Click to Unfold/Fold)

Task Description

Thank you for participating in this HIT!

Your task is to assess the quality of explanations. Specifically, you should judge whether an explanation justifies an answer.

An explanation justifies an answer to a question if:

- it is easily understood,
- it is factually correct,
- it provides all important reasons and implications behind the justification,
- does **NOT** just restate the question and the answer.

For each HIT, you will see

- one yes/no question
- the correct answer to the question
- several explanations

Your task is to annotate whether each explanation justifies the correct answer.

You will annotate between:

- **Yes**: the explanation is factually correct and justifies the correct answer well.
- **Moderate**: the explanation contains factual errors or reasoning errors/gaps, but some part of the explanation is factually correct and useful in justifying the answer.
- **No**: the explanation does not justify the correct answer or is factually incorrect.

Examples

Here is an example where you should annotate **Yes**:

Question	Is it common to see frost during some college commencements?
Correct Answer	Yes
Explanation	College commencement ceremonies can happen in December, May, and June. December is in the winter, so there can be frost. Thus, there could be frost at some commencements. So the answer is yes.
Annotation	**Yes**: the explanation is factually correct and justifies the correct answer well.

Here is an example where you should annotate **Moderate**:

Question	Does the number of states in the US exceed the number of months in a year?
Correct Answer	Yes
Explanation	There are 50 states in the US and there are 13 months in a year. Because 50 > 13, the answer is yes.
Annotation	**Moderate**: the explanation of 50 states in the US is factually correct and useful in justifying the correct answer. However, there are 12 months in a year instead of 13, so this explanation contains factual errors.

Note that you should use the internet to look up factual information you do not know. For instance, consider the following example:

Question	Does Hydrogen's atomic number squared exceed the number of Spice Girls?
Correct Answer	Yes
Explanation	Hydrogen has an atomic number of 1. 1 squared is 1. There are 3 Spice Girls. Thus, Hydrogen's atomic number squared is less than 3. So the answer is no.
Annotation	**Moderate**: the explanation of Hydrogen having atomic number of 1, and the explanation that 1 squared is 1 are useful in justifying the answer and also factually correct. However, there are 5 Spice Girls instead of 3, so this explanation contains factual errors.

Here are two examples where you should annotate **No**:

Question	Would a pear sink in water?
Correct Answer	No
Explanation	The density of a pear is about 3g/cm^3, which is heavier than water. Objects more dense than water float. Thus, a pear would float. So the answer is no.
Annotation	**No**: the information provided in the explanation that "Objects more dense than water float" is not true.

Question	Would a pear sink in water?
Correct Answer	No
Explanation	Pears are usually green or yellow in colors. Thus, a pear would float. So the answer is no.
Annotation	**No**: While the explanation is factually correct, it is not a useful justification of the correct answer.

Tips

- Minor grammatical and style errors should be ignored (e.g. case sensitivity, missing periods, a missing pronoun etc.).
- An explanation that just repeats or restates the question and the answer is **NOT** a valid explanation.
- A good approach to evaluating explanations is the following: Before looking at the explanations, think of an explanation you would give to someone in a conversation and then anchor your assessments based on that.

Figure A.11: Human annotation instructions for plausibility on StrategyQA.

So the answer is no.

Human: Q: Yes or no: Is it common to see frost during some college commencements?
A:

Assistant: here is my response. College commencement ceremonies can happen in December, May, and June. December is in the winter, so there can be frost. Thus, there could be frost at some commencements. So the answer is yes.

Human: Q: Yes or no: Could a llama birth twice during War in Vietnam (1945-46)?
A:

Assistant: here is my response. The War in Vietnam was 6 months. The gestation period for a llama is 11 months, which is more than 6 months. Thus, a llama could not give birth twice during the War in Vietnam. So the answer is no.

Human: Q: Yes or no: Would a pear sink in water?
A:

Assistant: here is my response. The density of a pear is about 0.6g/cm3, which is less than water. Objects less dense than water float. Thus, a pear would float. So the answer is no.

Human: Q: Yes or no: Is it hard to get a BLT in Casablanca?
A:

Assistant: here is my response.

StrategyQA-Counterfactual Generation.

Human: In the questions below, you will be asked to read a starter yes or no question and a robot's answer to the starter question. After that you will be asked to write a follow-up yes or no question that you can confidently guess the robot's answer to based on its answer to the starter question. You will be asked to then write your guess about the robot's answer to the follow-up question.

Assistant: here is my response. okay.

Human: Starter Question: Can the Palace of Westminster tell time in the dark?
Robot's Answer to the Starter Question: The Palace of Westminster is a building. Buildings cannot tell time. Thus, the Palace of Westminster cannot tell time in the dark. So the answer is no.
Follow-up Question:

Assistant: here is my response. Can the White House tell time?
Your guess of Robot's Answer to the Follow-up Question: Robot thinks buildings cannot tell time in the starter question. Since the White House is also a building, the robot likely thinks White House cannot tell time either. So the robot will likely answer no.

Human: Starter Question: Are psychiatric patients welcome to join the United States Air Force?
Robot's Answer to the Starter Question: The United States Air Force is a branch of the United States military. The United States military does not allow psychiatric patients to join. Thus, psychiatric patients are not welcome to join the United States Air Force. So the answer is no.
Follow-up Question:

Assistant: here is my response. Are psychiatric patients welcome to join the United States Army?
Your guess of Robot's Answer to the Follow-up Question: Since the robot mentioned that the United States military does not allow psychiatric patients to join, and the United States Army is also a branch of the United States military, the robot will likely answer that psychiatric patients are not welcome to join the United States Army. So the robot will likely answer no.

Human: Starter Question: Does The Hague border multiple bodies of water?
Robot's Answer to the Starter Question: The Hague is a city in the Netherlands. The Netherlands borders the North Sea. Thus, The Hague borders multiple bodies of water. So the answer is yes.
Follow-up Question:

Assistant: here is my response. Is Amsterdam near the North Sea?
Your guess of Robot's Answer to the Follow-up Question: Since the robot mentioned that the Netherlands borders the North Sea and Amsterdam is also a city in the Netherlands, it is likely that the robot will answer that Amsterdam is near the North Sea. So the robot will likely answer yes.

Human: Starter Question: Can an Asian black bear use chopsticks?
Robot's Answer to the Starter Question: Asian black bears are omnivores. Omnivores can use chopsticks. Thus, an Asian black bear can use chopsticks. So the answer is yes.
Follow-up Question:

Assistant: here is my response. Can pigs use chopsticks?
Your guess of Robot's Answer to the Follow-up Question: The robot incorrectly stated that omnivores can use chopsticks in its answer to the starter question. Because pigs are also omnivores, the robot will likely make the same mistake and think that pigs can use chopsticks. So the robot will likely answer yes.

Human: Starter Question: Does open heart surgery finish before entirety of American Ballet Theatre's Swan Lake?
Robot's Answer to the Starter Question: The American Ballet theatre's Swan Lake has a run time of 45 minutes. The National Heart, Lung, and Blood Institute states that a

coronary artery bypass takes 3 to 6 hours. So the answer is no.
Follow-up Question:

Assistant: here is my response. Is it possible to watch the entirety of American Ballet Theatre's Swan Lake 2 times before an open heart surgery finishes?
Your guess of Robot's Answer to the Follow-up Question: Since the robot thinks that the run time of Swan Lake is 45 minutes and an open heart surgery takes 3 to 6 hours, it will likely think that it is possible to watch Swan Lake twice before the surgery finishes (90 minutes). So the robot will likely answer yes.

Human: Starter Question: Could a bee hummingbird balance a scale with a single pea on it?
Robot's Answer to the Starter Question: The bee hummingbird is the smallest bird in the world. The bee hummingbird weighs about 2 grams. A pea weighs about 1 gram. Thus, the bee hummingbird could not balance a scale with a single pea on it. So the answer is no.
Follow-up Question:

Assistant: here is my response. Could a pea balance a scale with a dollar bill on it?
Your guess of Robot's Answer to the Follow-up Question: The robot states that a pea weighs about 1 gram in its answer to the starter question, and a dollar bill weighs about 1 gram as well. Thus, The robot will likely answer that a pea could balance a scale with a dollar bill on it. So the robot will likely answer yes.

Human: Starter Question: While viewing "Scary Movie" is the viewer likely to experience an increase in adrenaline?
Robot's Answer to the Starter Question: The movie "Scary Movie" is a comedy. Comedies are not scary. Thus, the viewer is not likely to experience an increase in adrenaline. So the answer is no.
Follow-up Question:

Assistant: here is my response. Is "Superbad" scary?
Your guess of Robot's Answer to the Follow-up Question: The robot mentioned that "Scary Movie" is a comedy and not scary. "Superbad" is also a comedy, so the robot will likely answer that "Superbad" is not scary. So the robot will likely answer no.

Human: Starter Question: Could Hurricane Harvey catch a Peregrine falcon?
Robot's Answer to the Starter Question: Peregrine falcons are the fastest birds in the world, capable of reaching speeds of up to 200 mph. Hurricanes have wind speeds of up to 155 mph. Thus, a Hurricane Harvey could not catch a Peregrine falcon. So the answer is no.
Follow-up Question:

StrategyQA-Simulation.

Human: In the questions below, you will be asked to read a starter yes or no question and a robot's answer to the starter question. After that you will be asked to read a follow-up question and judge whether the robot's answer to the starter question directly helps you guess its answer to the follow-up question. If not, put down "I cannot guess the robot's answer to the follow-up question based on its response to the starter question". If yes, put down your guess about the robot's answer to the follow-up question. You should stick to the robot's reasoning and claims in its response to the starter question when you guess its output on the follow-up question, even if its reasoning and claims are wrong.

Assistant: here is my response. okay.

Human: Starter Question: Does The Hague border multiple bodies of water?
Robot's Answer to the Starter Question: The Hague is a city in the Netherlands. The Netherlands borders the North Sea. Thus, The Hague borders multiple bodies of water. So the answer is yes.
Follow-up Question: Is Amsterdam near the North Sea?
Your guess of Robot's Answer to the Follow-up Question:

Assistant: here is my response. Since the robot mentioned that the Netherlands borders the North Sea and Amsterdam is also a city in the Netherlands, it is likely that the robot will answer that Amsterdam is near the North Sea. So the robot will likely answer yes.

Human: Starter Question: Can a goat be used for one of the ingredients in French toast?
Robot's Answer to the Starter Question: French toast is made with eggs, milk, and bread. Goats produce milk. Thus, a goat could be used for one of the ingredients in French toast. So the answer is yes.
Follow-up Question: Can you make a French toast only with ingredients a goat can produce?
Your guess of Robot's Answer to the Follow-up Question:

Assistant: here is my response. Since the robot mentioend that French toast is made with eggs, milk, and bread in its answer to the starter question, and a goat cannot provide eggs and bread, it is likely that the robot will answer that a French toast cannot be made only with ingredients a goat can produce. So the robot will likely answer no.

Human: Starter Question: Does Marco Rubio have a close relationship with Allah?
Robot's Answer to the Starter Question: Marco Rubio is a Christian. Christians do not have a close relationship with Allah. Thus, Marco Rubio does not have a close relationship with Allah. So the answer is no.
Follow-up Question: Is Barack Obama a Christian?
Your guess of Robot's Answer to the Follow-up Question:

Assistant: here is my response. Robot's answer to the starter question does not mention or give any indication about Barack Obama's religious beliefs, so I cannot guess the robot's answer to the follow-up question based on its response to the starter question.

Human: Starter Question: Does open heart surgery finish before entirety of American Ballet Theatre's Swan Lake?
Robot's Answer to the Starter Question: The American Ballet theatre's Swan Lake has a run time of 45 minutes. The National Heart, Lung, and Blood Institute states that a coronary artery bypass takes 3 to 6 hours. So the answer is no.
Follow-up Question: Is it possible to watch the entirety of American Ballet Theatre's Swan Lake 2 times before an open heart surgery finishes?
Your guess of Robot's Answer to the Follow-up Question:

Assistant: here is my response. Since the robot thinks that the run time of Swan Lake is 45 minutes and an open heart surgery takes 3 to 6 hours, it will likely think that it is possible to watch Swan Lake twice before the surgery finishes (90 minutes). So the robot will likely answer yes.

Human: Starter Question: Do gorillas fight with panda bears?
Robot's Answer to the Starter Question: Gorillas are herbivores. Pandas are herbivores. Herbivores do not fight with other herbivores. Thus, gorillas do not fight with pandas. So the answer is no.
Follow-up Question: Do herbivores fight with carnivores?
Your guess of Robot's Answer to the Follow-up Question:

Assistant: here is my response. Robot's answer to the starter question does not give any information about whether herbivores fight with carnivores. Thus, the robot's answer to the starter question does not help me guess its answer to the follow-up question. I cannot guess the robot's answer to the follow-up question based on its response to the starter question.

Human: Starter Question: Could an elephant easily defeat a male macaque?
Robot's Answer to the Starter Question: An elephant weighs about 1 pound. A male macaque weighs about 20 pounds. Thus, an elephant could not easily defeat a male macaque. So the answer is no.
Follow-up Question: Can an elephant defeat a tiger?
Your guess of Robot's Answer to the Follow-up Question:

Assistant: here is my response. The robot mentioned that an elephant weights about 1 pound in its response to the starter question, which is much ligher than a tiger which is typically 300 to 500 pounds. Thus, the robot likely thinks that an elephant cannot defeat a tiger. So the robot will likely answer no.

Human: Starter Question: Can citrus grow in Ulaanbaatar?
Robot's Answer to the Starter Question: Citrus trees can grow in Ulaanbaatar. Thus, citrus can grow in Ulaanbaatar. So the answer is yes.
Follow-up Question: Can palm trees grow in Ulaanbaatar?
Your guess of Robot's Answer to the Follow-up Question:

Assistant: here is my response. Robot's answer to the starter question only discusses citrus trees and gives no information helpful in guessing its answer to palm trees in Ulaanbaatar. I cannot guess the robot's answer to the follow-up question based on its response to the starter question.

Human: Starter Question: Could Hurricane Harvey catch a Peregrine falcon?
Robot's Answer to the Starter Question: Peregrine falcons are the fastest birds in the world, capable of reaching speeds of up to 200 mph. Hurricanes have wind speeds of up to 155 mph. Thus, a Hurricane Harvey could not catch a Peregrine falcon. So the answer is no.
Follow-up Question: Could a cheetah catch a Peregrine falcon?
Your guess of Robot's Answer to the Follow-up Question:

SHP-Explanation Generation.

Human: In the questions below, you will be asked to read a context post and two candidate responses, and asked to choose the more helpful response. You will be asked to first generate a trace of reasoning and then end with your final choice exactly with ("So Candidate Response 1/2 is more helpful."). Strictly follow the example format below and do not say anything else.

Assistant: here is my response. okay.

Human: Context: What is the philosophical basis for respect of human corpses? Or is it just cultural? Is there anything philosophically/ethically wrong with desecrating a human body? It feels intuitive that there is, but I am having trouble identifying if the basis and have been unsuccessful in finding one.
Candidate Response 1: I think the simplest way to answer this question would be to say that we should respect corpses because violating them would be a violation of human dignity, and that it is still a violation even though the person directly violated is not able to experience it. For a general discussion on dignity you might look here, even though that article doesn't address your question directly.
Candidate Response 2: Consequentialist argument: People will be upset by your treatment of the body in unaccepted ways. So, if your treatment doesn't outweigh that in good, it shouldn't be done. And yes, this is based on only certain cultures. If the cultures changed, the judgment would too. Virtue argument: The way we treat inanimate objects might cultivate habits or traits in us that affect how we treat people or other things. If what we're doing to the corpse creates bad habits or sullies the trajectory of our life, we

shouldn't do it. Relatedly, if it affects our social standing negatively, we shouldn't do it without good reasons to counter it. It's gonna depend on what we're talking about. Cremation? Necrophilia? Plasticizing and creating an artwork from it? Mandatory organ donation? Eating it to survive? For recreation?
Your choice:

Assistant: here is my response. Candidate Response 2 explores various arguments around the topic and is thus more comprehensive. So Candidate Response 2 is more helpful.

Human: Context: Adding blueberries to a plain pound cake recipe? Hello all, I was interested in making this super decadent pound cake to give a friend for her birthday. I would like to add just a few fresh blueberries, at most half a cup. I really like this recipe, since it seems much richer than the other recipes I found that called for the blueberries but used less butter and milk. Any guidance on this? Recipe I plan to use: https://m.youtube.com/watch?v=SDNDw_7miPQ Thanks!
Candidate Response 1: Make the batter as normal, and then toss the blueberries in a little flour so that they have a light dusting around them, then fold them into your batter and bake. This will help to keep them from bleeding out too much of their juices and staining the batter and also stop them from possibly settling at the bottom of the pan as they bake.
Candidate Response 2: commenting to follow, i have a citrus pound cake recipe that i'd like to add blueberries to but afraid to make the jump without tips lol
Your choice:

Assistant: here is my response. Candidate Response 1 provides more specific instructions on the task. So Candidate Response 1 is more helpful.

Human: Context: I requested a few exam copies, ended up not adopting them, put them out for other professors to take. The publisher now wants me to either adopt the textbooks or pay an invoice? I had no idea these were the terms and did not sign anything, but apparently one must either adopt the textbook, return it, or buy it at a 20% discount. These books looked promising but are not a fit for my classroom and were swept up by other professors once I set them out. Am I *legally* obligated to purchase them/adopt them? Even if they send me an invoice, can I just ignore it indefinitely?
Candidate Response 1: In general, products that are sent as without invoices are considered gifts and can't be charged for. As far as I know, most desk copies for consideration for adoption fall into this category. They can send as many invoices as they want, but they can't collect. More likely, they'll just blacklist you for a few years for more samples.
Candidate Response 2: Why not just return them?
Your choice:

Assistant: here is my response. Candidate Response 1 answers the question more directly and provides legal insights on the situation. So Candidate Response 1 is more helpful.

Human: Context: Show support for UC academic worker strike Fellow academic community- Please take a moment to show solidarity with the academic student workers on strike at UC right now. We are in the second week of the strike by 48,000 academic workers in the University of California (UC) system. The action is the largest strike of academic workers in United States history. The strikers are demanding a salary increase—from an impossibly low $24,000 a year to $54,000—to address California's skyrocketing rents and other living expenses. Sign the letter to President Drake https://act.aflcio.org/petitions/show-your-support-for-academic-workers-at-university-of-california?source=direct_link& Make a donation in the hardship fund if you can https://givebutter.com/uc-uaw https://www.fairucnow.org/support/
Candidate Response 1: Is this $24k an year, or $24k for 9-months? I gave two acquaintances at UCB (who joined PhD recently) and both mentioned they get paid around 26 - 29k for 9 months. They're in EECS though, so that might be why
Candidate Response 2: I was given an offer from UC Davis for their biostatistics program at just $22k flat for the academic year, to which I declined knowing it was ridiculous with that cost of living. I feel for the students who probably felt like they had no choice but to accept, or are first-gens without financial backing from their family.
Your choice:

Assistant: here is my response.

SHP-Counterfactual Generation.

Human: You have a robot that chooses the more helpful response from two candidate responses to a given context. Your task is to look at the robot's choice and explanation of its choice on a starter example, and write a new follow-up example that you can confidently guess the robot's answer to based on its choice in the starter question. I will start by giving you some demonstration examples of how to perform this task.

Assistant: here is my response. okay.

Human: Starter Example:
Context: What is the philosophical basis for respect of human corpses? Or is it just cultural? Is there anything philosophically/ethically wrong with desecrating a human body? It feels intuitive that there is, but I am having trouble identifying if the basis and have been unsuccessful in finding one.
Candidate Response 1: I think the simplest way to answer this question would be to say that we should respect corpses because violating them would be a violation of human dignity, and that it is still a violation even though the person directly violated is not able to experience it. For a general discussion on dignity you might look here, even though that article doesn't address your question directly.
Candidate Response 2: Consequentialist argument: People will be upset by your treat-

ment of the body in unaccepted ways. So, if your treatment doesn't outweigh that in good, it shouldn't be done. And yes, this is based on only certain cultures. If the cultures changed, the judgment would too. Virtue argument: The way we treat inanimate objects might cultivate habits or traits in us that affect how we treat people or other things. If what we're doing to the corpse creates bad habits or sullies the trajectory of our life, we shouldn't do it. Relatedly, if it affects our social standing negatively, we shouldn't do it without good reasons to counter it. It's gonna depend on what we're talking about. Cremation? Necrophilia? Plasticizing and creating an artwork from it? Mandatory organ donation? Eating it to survive? For recreation?

Robot's Choice to the Starter Example: Candidate Response 2
Robot's Explanation about its Choice: Candidate Response 2 explores various arguments around the topic and is thus more comprehensive. So Candidate Response 2 is more helpful.

Follow-up Example:

Assistant: here is my response.
Context: I've been pondering over this idea of personal identity lately. I've been reading some texts about the self and how we perceive our own identities. It got me thinking: is personal identity an illusion? Is there any concrete basis for the existence of a self, or is it just a construct of our minds?
Candidate Response 1: Well, it's not an easy question to answer. Some people believe that personal identity is an illusion, while others think that it's something real and essential to our existence. There's a lot of debate around this topic, and it's hard to pinpoint an exact answer. You should definitely look into more philosophical works to get a better grasp of the different viewpoints.
Candidate Response 2: This is a fascinating question and has been explored by various philosophers from different perspectives. Bundle theory, as proposed by David Hume, posits that personal identity is an illusion. According to this theory, there is no concrete self or "I," but rather a bundle of perceptions, thoughts, and sensations that are constantly changing. On the other hand, narrative theory, as suggested by philosophers like Paul Ricoeur and Marya Schechtman, argues that personal identity is real, but it is constructed through the stories we tell about ourselves. Our identities are shaped by our experiences, relationships, and cultural contexts. According to this view, personal identity is an ongoing process, where we actively create and revise our own self-narratives over time.

Human: Starter Example:
Context: Adding blueberries to a plain pound cake recipe? Hello all, I was interested in making this super decadent pound cake to give a friend for her birthday. I would like to add just a few fresh blueberries, at most half a cup. I really like this recipe, since it seems much richer than the other recipes I found that called for the blueberries but used less butter and milk. Any guidance on this? Recipe I plan to use:

https://m.youtube.com/watch?v=SDNDw_7miPQ Thanks!
Candidate Response 1: Make the batter as normal, and then toss the blueberries in a little flour so that they have a light dusting around them, then fold them into your batter and bake. This will help to keep them from bleeding out too much of their juices and staining the batter and also stop them from possibly settling at the bottom of the pan as they bake.
Candidate Response 2: commenting to follow, i have a citrus pound cake recipe that i'd like to add blueberries to but afraid to make the jump without tips lol

Robot's Choice to the Starter Example: Candidate Response 1
Robot's Explanation about its Choice: Candidate Response 1 provides more specific instructions on the task. So Candidate Response 1 is more helpful.

Follow-up Example:

Assistant: here is my response.
Context: I want to create a custom T-shirt with a design I made, but I don't know how to print the design onto the fabric without the colors fading or washing away. Can anyone recommend a method or technique to do this at home? Thanks!
Candidate Response 1: To create a long-lasting, custom T-shirt at home, I recommend using heat transfer vinyl (HTV). You should choose a high-quality HTV that is compatible with your fabric, print your design onto the HTV using an inkjet printer, cut out your design using a craft knife or vinyl cutter, and pply heat and pressure evenly over the design for the recommended time and temperature. With proper care, your design should last through many washes.
Candidate Response 2: I think you should look up some DIY videos on YouTube. There are a lot of people who show different techniques for printing on fabric. You might find something helpful there.

Human: Starter Example:
Context: I requested a few exam copies, ended up not adopting them, put them out for other professors to take. The publisher now wants me to either adopt the textbooks or pay an invoice? I had no idea these were the terms and did not sign anything, but apparently one must either adopt the textbook, return it, or buy it at a 20% discount. These books looked promising but are not a fit for my classroom and were swept up by other professors once I set them out. Am I *legally* obligated to purchase them/adopt them? Even if they send me an invoice, can I just ignore it indefinitely?
Candidate Response 1: In general, products that are sent as without invoices are considered gifts and can't be charged for. As far as I know, most desk copies for consideration for adoption fall into this category. They can send as many invoices as they want, but they can't collect. More likely, they'll just blacklist you for a few years for more samples.
Candidate Response 2: Why not just return them?

Robot's Choice to the Starter Example: Candidate Response 1

Robot's Explanation about its Choice: Candidate Response 1 answers the question more directly and provides legal insights on the situation. So Candidate Response 1 is more helpful.

Follow-up Example:

Assistant: here is my response.
Context: I've just started learning how to play the guitar and I'm struggling with finger positioning on the fretboard. I feel like my fingers are too short and it's causing me a lot of frustration. Can anyone offer some advice?
Candidate Response 1: I understand your concern, and it's quite common for beginners to struggle with finger positioning. Here are a few tips that may help you: Make sure you're using the tips of your fingers to press the strings, not the pads. Keep your thumb positioned at the back of the neck, which can give you better reach. Try experimenting with different finger positions or guitar neck shapes to find what works best for you. Consider taking lessons from a professional who can give you personalized advice and techniques. Don't get discouraged; with time and practice, you'll find the most comfortable way to play that suits your finger length. Good luck!
Candidate Response 2: Don't worry too much about it! Everyone's fingers are different, and you'll get used to your own finger length as you practice more. It's really not a big deal. Just keep practicing and having fun!

Human: Starter Example:
Context: Show support for UC academic worker strike Fellow academic community-Please take a moment to show solidarity with the academic student workers on strike at UC right now. We are in the second week of the strike by 48,000 academic workers in the University of California (UC) system. The action is the largest strike of academic workers in United States history. The strikers are demanding a salary increase—from an impossibly low $24,000 a year to $54,000—to address California's skyrocketing rents and other living expenses. Sign the letter to President Drake https://act.aflcio.org/petitions/show-your-support-for-academic-workers-at-university-of-california?source=direct_link& Make a donation in the hardship fund if you can https://givebutter.com/uc-uaw https://www.fairucnow.org/support/
Candidate Response 1: Is this $24k an year, or $24k for 9-months? I gave two acquaintances at UCB (who joined PhD recently) and both mentioned they get paid around 26 - 29k for 9 months. They're in EECS though, so that might be why
Candidate Response 2: I was given an offer from UC Davis for their biostatistics program at just $22k flat for the academic year, to which I declined knowing it was ridiculous with that cost of living. I feel for the students who probably felt like they had no choice but to accept, or are first-gens without financial backing from their family.

Robot's Choice to the Starter Example: Candidate Response 2
Robot's Explanation about its Choice: Candidate Response 2 expresses empathy and understanding of the situation and provides a personal experience to back it up. So

Candidate Response 2 is more helpful.

Follow-up Example:

Assistant: here is my response.

SHP-Simulation.

Human: You have a robot that's built to choose the more helpful response from two candidate responses to a given context. Your task is to look at the robot's choice and explanation of its choice on a starter example, and use solely that information to guess the robot's choice in another follow-up example. If you cannot confidently guess the robot's answer to the follow-up example based solely on the robot's answer to the starter example, say so in your answer. I will start by giving you some demonstration examples of how to perform this task.

Assistant: here is my response. okay.

Human: Starter Example:
Context: Adding blueberries to a plain pound cake recipe? Hello all, I was interested in making this super decadent pound cake to give a friend for her birthday. I would like to add just a few fresh blueberries, at most half a cup. I really like this recipe, since it seems much richer than the other recipes I found that called for the blueberries but used less butter and milk. Any guidance on this? Recipe I plan to use: https://m.youtube.com/watch?v=SDNDw_7miPQ Thanks!
Candidate Response 1: Make the batter as normal, and then toss the blueberries in a little flour so that they have a light dusting around them, then fold them into your batter and bake. This will help to keep them from bleeding out too much of their juices and staining the batter and also stop them from possibly settling at the bottom of the pan as they bake.
Candidate Response 2: commenting to follow, i have a citrus pound cake recipe that i'd like to add blueberries to but afraid to make the jump without tips lol

Robot's Choice to the Starter Example: Candidate Response 1
Robot's Explanation about its Choice: Candidate Response 1 provides more specific instructions on the task. So Candidate Response 1 is more helpful.

Follow-up Example:
Context: I want to create a custom T-shirt with a design I made, but I don't know how to print the design onto the fabric without the colors fading or washing away. Can anyone recommend a method or technique to do this at home? Thanks!
Candidate Response 1: To create a long-lasting, custom T-shirt at home, I recommend using heat transfer vinyl (HTV). You should choose a high-quality HTV that is compatible with your fabric, print your design onto the HTV using an inkjet printer, cut out

your design using a craft knife or vinyl cutter, and pply heat and pressure evenly over the design for the recommended time and temperature. With proper care, your design should last through many washes.
Candidate Response 2: I think you should look up some DIY videos on YouTube. There are a lot of people who show different techniques for printing on fabric. You might find something helpful there.

Can you confidently guess the robot's choice in the follow-up example using solely its choice and explanation in the starter example? If so, what would be your guess as its choice in the follow-up example?

Assistant: here is my response.
Yes, I can confidently guess the robot's choice in the follow-up example using solely the robot's choice and explanation in the starter example. Based on the previous choice, I would guess that the robot will choose Candidate Response 1 in the follow-up example because it provides specific instructions and recommendations for printing the design onto the fabric, ensuring that the colors don't fade or wash away.

Human: Starter Example:
Context: What personal tips can you tell me about design engineering? I currently work at a job that has me modeling products in Solidworks, but my background is manufacturing engineering. It's not like I can't do the job, drafting is one of my strongest skills. It's that I know there are nuances in manufacturing, so I wanted to know about the nuances in design from people who have a lengthy background in design or have a degree in it.
Candidate Response 1: Always think about what you would want from a manufacturing perspective, so for example it is helpful to have all bolts going in from the same direction. Equally add and secure one part at a time. Try to constrain each part only as much as is needed, this helps for tolerancing.
Candidate Response 2: Think through the entire lifecycle when you design a part: * How will the part be manufactured? * How will the part be installed? Clearance for assemblies, tool access, lifting points, etc? * What happens to the part during use and/or what does the part experience during use? Temperatures, pressures, vibrations, etc? * What maintenance is required?

Robot's Choice to the Starter Example: Candidate Response 2
Robot's Explanation about its Choice: Candidate Response 2 is more helpful because it uses a more holistic approach and is more comprehensive.

Follow-up Example:
Context: Hey everyone, I've been struggling with my mental and physical health lately, and I'm looking for advice on how to turn things around. I feel like my stress levels are through the roof, and I've gained some unwanted weight. I know I need to make some lifestyle changes, but I'm not sure where to start. Any tips or suggestions would

be greatly appreciated!
Candidate Response 1: Just hit the gym and start working out! You'll lose the extra weight in no time, and it'll help with your stress too. Just find a workout routine and stick to it.
Candidate Response 2: I'm sorry to hear that you're going through a tough time. Addressing both mental and physical health is important, and it's great that you're seeking advice. Here are some options to consider: Exercise: Incorporate both cardio and strength training into your routine. Start slow and gradually increase intensity. Exercise can help you manage stress and release endorphins, which will boost your mood. Sleep: Prioritize getting 7-9 hours of quality sleep each night. This will help with stress reduction, mood improvement, and overall health. Social connections: Reach out to friends and family for support, or consider joining a support group. Connecting with others can be immensely helpful for mental health. Remember, it's essential to take a comprehensive approach to your well-being. Small, consistent steps will lead to positive changes over time. Good luck on your journey!

Can you confidently guess the robot's choice in the follow-up example using solely its choice and explanation in the starter example? If so, what would be your guess as its choice in the follow-up example?

Assistant: here is my response.
Yes, I can confidently guess the robot's choice in the follow-up example using solely the robot's choice and explanation in the starter example. Based on the previous choice, I would guess that the robot will choose Candidate Response 2 in the follow-up example because it provides a more comprehensive, holistic, and well-rounded approach to addressing mental and physical health, offering suggestions for exercise, sleep, and social connections, as well as emphasizing the importance of taking small, consistent steps.

Human: Starter Example:
Context: What is the philosophical basis for respect of human corpses? Or is it just cultural? Is there anything philosophically/ethically wrong with desecrating a human body? It feels intuitive that there is, but I am having trouble identifying if the basis and have been unsuccessful in finding one.
Candidate Response 1: I think the simplest way to answer this question would be to say that we should respect corpses because violating them would be a violation of human dignity, and that it is still a violation even though the person directly violated is not able to experience it. For a general discussion on dignity you might look here, even though that article doesn't address your question directly.
Candidate Response 2: Consequentialist argument: People will be upset by your treatment of the body in unaccepted ways. So, if your treatment doesn't outweigh that in good, it shouldn't be done. And yes, this is based on only certain cultures. If the cultures changed, the judgment would too. Virtue argument: The way we treat inanimate objects might cultivate habits or traits in us that affect how we treat people or other things. If

what we're doing to the corpse creates bad habits or sullies the trajectory of our life, we shouldn't do it. Relatedly, if it affects our social standing negatively, we shouldn't do it without good reasons to counter it. It's gonna depend on what we're talking about. Cremation? Necrophilia? Plasticizing and creating an artwork from it? Mandatory organ donation? Eating it to survive? For recreation?

Robot's Choice to the Starter Example: Candidate Response 2
Robot's Explanation about its Choice: Candidate Response 2 explores various arguments around the topic and is thus more comprehensive. So Candidate Response 2 is more helpful.

Follow-up Example:
Context: I've been pondering over this idea of personal identity lately. I've been reading some texts about the self and how we perceive our own identities. It got me thinking: is personal identity an illusion? Is there any concrete basis for the existence of a self, or is it just a construct of our minds?
Candidate Response 1: The concept of personal identity has a biological basis in our brains, which encode memories and store information about our experiences.
Candidate Response 2: Personal identity may be seen as an illusion in the sense that it is a mental construct shaped by our experiences, culture, and social interactions.

Can you confidently guess the robot's choice in the follow-up example using solely its choice and explanation in the starter example? If so, what would be your guess as its choice in the follow-up example?

Assistant: here is my response.
No, I cannot confidently guess the robot's choice in the follow-up example using solely the robot's choice and explanation in the starter example. Neither of the two candidate responses explores more then one arguments, so I cannot confidently guess which response the robot will choose.

Human: Starter Example:
Context: What is your opinion on sales engineering? I am an ME student and have the option of doing internships as a company representative or other roles in sales. I am neither a great speaker nor am I an extrovert. Should I take the internship? Is sales engineering better (money-wise) in the long run than technical roles?
Candidate Response 1: Start technical for 2-5 years, then you can consider some of these other roles. This will open many doors. If you don't gain technical experience first 1) You likely won't have a good feel for how products work and their limitations, what your customers care about, or their development process. I hate working with those sales engineers. 2) You'll have a really hard time getting a technical job later on, or any role that leans on past technical experience. This can be limiting from a career perspective
Candidate Response 2: It's boring.

Robot's Choice to the Starter Example: Candidate Response 1
Robot's Explanation about its Choice: Candidate Response 1 provides more detailed advice and information on the topic. So Candidate Response 1 is more helpful.

Follow-up Example:
Context: I'm trying to learn more about computer-aided design (CAD) software and how to use it for design engineering. Is it better to learn from video tutorials, books, or other resources?
Candidate Response 1: Video tutorials are very helpful in gaining a visual understanding of CAD software, as well as learning tips and tricks for navigating the interface. Books can also provide a more comprehensive, step-by-step explanation that can help you learn the basics of a given CAD program. Other resources, such as online communities, forums, and blogs, can be a great source of information and advice, allowing you to interact with people who use CAD software on a daily basis and ask questions specific to your needs and level of expertise.
Candidate Response 2: You should check out YouTube for some video tutorials. There are lots of helpful and free tutorials out there.

Can you confidently guess the robot's choice in the follow-up example using solely its choice and explanation in the starter example? If so, what would be your guess as its choice in the follow-up example?