SCULPT: Systematic Tuning of Long Prompts

Anonymous ACL submission

Abstract

As large language models become increasingly central to solving complex tasks, the challenge of optimizing long, unstructured prompts has become critical. Existing optimization techniques often struggle to effectively handle such prompts, leading to suboptimal performance. We introduce **SCULPT** (***S**ystematic **Tu**ning of **L**ong **P**rompts*), a novel framework that systematically refines long prompts by structuring them hierarchically and applying an iterative actor-critic mechanism. To enhance robustness and generalizability, SCULPT utilizes two complementary feedback mechanisms: *Preliminary Assessment*, which assesses the prompt's structure before execution, and *Error Assessment*, which diagnoses and addresses errors post-execution. By aggregating feedback from these mechanisms, SCULPT avoids overfitting and ensures consistent improvements in performance. Our experimental results demonstrate significant accuracy gains and enhanced robustness, particularly in handling erroneous and misaligned prompts. SCULPT consistently outperforms existing approaches, establishing itself as a scalable solution for optimizing long prompts across diverse and real-world tasks.

1 Introduction

Due to the emergence of large language models (LLMs), natural language processing (NLP) has achieved unprecedented advances, enabling models to perform complex language understanding and generation tasks with remarkable proficiency (Brown, 2020; Achiam et al., 2023; Bubeck et al., 2023; Abdin et al., 2024; Dubey et al., 2024). These models excel in generating human-like text, navigating nuanced conversations, and interpreting intricate instructions, transforming domains such as content creation, customer support, and data analysis.

As LLMs are increasingly employed for complex, real-world tasks, the challenge of designing effective prompts has grown substantially. These tasks often require processing extended sequences of instructions, detailed reasoning, and structured outputs, with prompts that can span thousands of tokens (Schnabel and Neville, 2024). Crafting such prompts is labor-intensive, requiring significant human effort, specialized expertise, and extensive trial-and-error iterations (Jiang et al., 2022; Zamfirescu-Pereira et al., 2023).

A key issue is the sensitivity of LLMs to prompt phrasing and structure. Even minor modifications such as reordering instructions, tweaking wording, or adjusting token length can lead to significantly different outputs (Jiang et al., 2020; Zhao et al., 2021; Reynolds and McDonell, 2021; Lu et al., 2022). This sensitivity makes prompt design prone to inconsistencies and necessitates substantial experimentation to find optimal phrasing.

Recent efforts to optimize prompts for black-box LLMs like APE (Zhou et al., 2022) and ProTeGi (Pryzant et al., 2023), OPRO (Yang et al., 2024) and APEX (Hsieh et al., 2024) have primarily focused on short prompts or few-shot scenarios. However, these methods are less effective for long prompts, which involve multiple instructions, examples, and complex dependencies. Short-prompt optimization methods often make isolated, local edits, failing to account for the broader context and relationships within longer prompts. This can lead to inconsistencies in structural flow, misalignment between instructions and examples, and incomplete refinement of the prompt's overall logic. Overcoming these limitations requires a more structured, context-aware approach that maintains coherence across diverse tasks.

Recognizing these limitations, we propose **SCULPT** (***S**ystematic **Tu**ning of **L**ong **P**rompts*), a novel approach designed to optimize unstructured, long prompts. Rather than treating prompts as flat text, our method restructures them into hierarchical, tree-like representations where headings, sub-

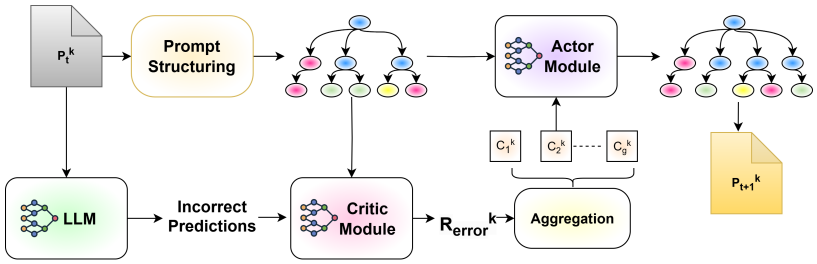


Figure 1: Overview of the SCULPT framework, detailing prompt refinement through Prompt Structuring, Critic, and Actor Modules. Incorrect model predictions generate Error Assessments, which are aggregated and refined by the Actor. For clarity, UCB-based prompt selection and Preliminary Assessment are omitted. $\mathcal{D}_{\text{val}}$ is the k-th candidate prompt at iteration t.

headings, instructions, and examples are organized as parent-child nodes. This structured format allows for systematic and controlled optimization while preserving the integrity of the original content.

At the core of our framework is an iterative critic-actor process for prompt refinement. The *Critic Module* takes the current structured prompt, along with example predictions and ground truth, to identify areas needing improvement. It generates reflections based on discrepancies between predictions and expected outcomes. The *Actor Module* then applies a diverse set of actions inspired by how human experts optimize prompts, ensuring systematic improvements in clarity, coherence, and overall effectiveness. Fig. 1 shows the overview of our proposed SCULPT framework.

To ensure robustness, we introduce two complementary feedback mechanisms: *Preliminary Assessment* and *Error Assessment*. *Preliminary Assessment* evaluates the overall structure and clarity of the prompt before task execution, providing broad recommendations for improvement. After execution, *Error Assessment* identifies specific nodes responsible for model errors. By aggregating multiple error-based reflections (Juneja et al., 2024), we prevent overfitting to isolated errors while capturing recurring patterns, ensuring that refinements are both generalizable and robust across diverse inputs. Our contributions can be summarized as follows: (1) We introduce SCULPT, a novel framework for optimizing long prompts through a hierarchical tree structure and actor-critic mechanism, enabling systematic and iterative refinements. (2) We demonstrate SCULPT's ability to handle unstructured prompts, achieving significant improvements in LLM performance across four BBH tasks and four real-world RAI tasks, with prompts averaging 980 words in length and up to a maximum of 2,644 words. (3) SCULPT is robust to erroneous or perturbed prompts, outperforming existing techniques by systematically refining prompt structures, even in challenging scenarios. (4) We perform a comprehensive evaluation of SCULPT's action distribution, showing that it applies controlled, systematic, and balanced refinements compared to baselines, ensuring stable and generalizable results across tasks.

2 Related Work

Prompt Engineering: Optimizing prompts is crucial to unlocking the full potential of LLMs across various applications, and prompt engineering has emerged as a key area of research for creating more effective prompts (Brown, 2020; Reynolds and McDonell, 2021; Wang et al., 2022; Chang et al., 2024; Sahoo et al., 2024). While prompt engineering through human effort has been effective, it is labor-intensive and demands expertise to craft detailed, task-specific prompts. Automatic prompt optimization methods are increasingly needed to streamline and improve prompts with minimal manual intervention.

Soft Prompting: One prominent line of research involves soft prompting (Lester et al., 2021; Li and Liang, 2021; Liu et al., 2021; Qin and Eisner, 2021), where prompts are optimized as continuous task-specific vectors in the model's embedding space. However, these methods are only applicable in settings where gradient access is available, making them unsuitable for black-box LLMs, where the model's internal structure remains inaccessible.

Automatic Black-box Prompt Optimization: To address limitations in gradient-based methods, research has shifted towards black-box prompt optimization techniques. These approaches can be classified into explicit or implicit reflection-based methods. Explicit reflection-based techniques (Cheng et al., 2023; Pryzant et al., 2023; Ye et al., 2023;

Sun et al., 2023; Dong et al., 2024) generate structured feedback based on model errors, which are iteratively incorporated into the prompt. These methods are advantageous as they provide task-specific refinements to better align the prompt with the desired outcome. SCULPT adopts this explicit feedback approach, systematically refining long, complex prompts in a controlled, iterative process.

In contrast, implicit reflection-based approaches, such as OPRO (Yang et al., 2024), and those utilizing evolutionary algorithms (Xu et al., 2022; Guo et al., 2024; Fernando et al., 2024; Liu et al., 2024), rely on historical data or model outputs to improve prompts without explicitly generating reflections. Additionally, some methods integrate human feedback to align generated prompts with human preferences and reduce exploration costs (Chen et al., 2024b).

Recent studies have identified key limitations in prompt optimization methods, such as the need for repeated refinements and unpredictable behavior that often results in suboptimal outcomes (Ma et al., 2024). SCULPT overcomes these challenges by applying precise, hierarchical updates to the prompt, ensuring controlled, systematic refinements that enhance stability and align more effectively with task requirements.

In parallel, research has focused on automatic prompt generation without an initial prompt. Methods such as (Honovich et al., 2023; Zhou et al., 2022; Chen et al., 2024a) leverage LLMs' generative capabilities to create prompts from input-output pairs. Additionally, work on multi-step AI pipelines explores how textual feedback can optimize the performance of individual components (Khattab et al., 2023; Yuksekgonul et al., 2024; Schnabel and Neville, 2024).

Long Prompt Optimization: While recent methods have extended to optimizing longer prompts by segmenting them and applying predefined operations (Prasad et al., 2022; Hsieh et al., 2024), they remain limited in scope. APEX (Hsieh et al., 2024), for instance, focuses on few-shot chain-of-thought prompts for benchmarks like BBH, rather than comprehensive task instructions. While APEX is a step forward, it struggles with complex prompts that include detailed instructions and intricate structures. Our work highlights these limitations and demonstrates the need for methods specifically designed to optimize long-form prompts with complex structures.

3 The SCULPT Methodology

In this section, we present SCULPT, a powerful framework designed to optimize complex, long prompts used by LLMs. While existing methods primarily focus on short prompts or few-shot examples, SCULPT specifically targets the intricacies of longer prompts with multiple instructions, examples, and layered structures. Our goal is to ensure refined and controlled prompt modifications that lead to robust model performance while maintaining the clarity and task relevance of the prompt.

Let $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{val}}$, and $\mathcal{D}_{\text{test}}$ represent the training, validation, and test datasets, each consisting of input-output pairs (x, y). The LLM $\mathcal{M}$ generates predictions $\hat{y} = \mathcal{M}(\mathcal{P}, x)$ based on the given prompt $\mathcal{P}$, which can contain complex sets of instructions and examples (see Appendix J for examples). The objective of SCULPT is to find an optimized prompt $\mathcal{P}^*$ that maximizes a performance metric $\mathcal{Q}$ (e.g., accuracy) on $\mathcal{D}_{\text{val}}$.

$$\mathcal{P}^* = \arg\max_{\mathcal{P}} \mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{val}}} \left[\mathcal{Q}(y, \mathcal{M}(\mathcal{P}, x))\right]. \quad (1)$$

The optimization process in SCULPT consists of three main components: *Prompt Structuring*, the *Critic Module*, and the *Actor Module*, coupled with a beam search strategy to explore and refine multiple candidate prompts simultaneously. These components, as illustrated in Fig. 1, work together to iteratively refine the prompt, ensuring a systematic and structured approach to improving long, complex prompts.

3.1 Prompt Structuring

Prompt optimization methods designed for short prompts struggle to apply feedback effectively to longer, more complex instructions. When treating a long prompt as a single unit, it becomes difficult for optimizers to attribute error feedback to specific sections, leading to fragmented and less effective refinements. To address this, we transform unstructured prompts into a hierarchical tree structure $\mathcal{T} = (N, E)$, where N represents components such as headings, instructions, and examples, and E defines the containment relationships between these nodes. This structure enables targeted, node-specific modifications while maintaining the coherence of unrelated sections, facilitating more precise and systematic optimizations.

In Fig. 2, we illustrate this process, where different colors represent different types of nodes (e.g.,

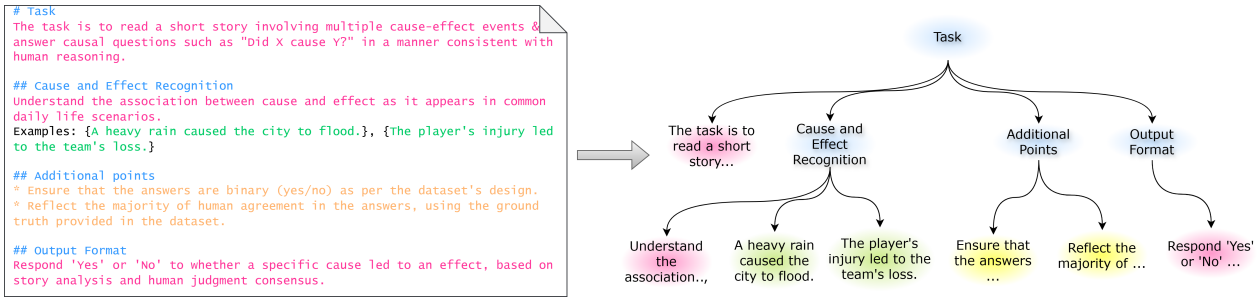


Figure 2: Illustration of SCULPT's Prompt Structuring Process. Unstructured prompt is transformed into a hierarchical tree structure, with different colors represent various node types (e.g., heading, instructions, examples).

headings, instructions, examples), and how the hierarchical tree is formed. To help the LLM better parse and understand the prompt structure, we use markdown formatting for the hierarchical tree, leveraging the model's familiarity with markdown from its training on documentation and code repositories (OpenAI, 2023).

3.2 Critic Module

The Critic Module $\mathcal{C}$ plays a pivotal role in evaluating the prompt and generating reflective feedback for iterative refinement. There are two types of reflections:

1. *Preliminary Assessment* ($\mathcal{R}_{\text{prelim}}$): This reflection is generated independently of task-specific outputs. It focuses on the overall structure, clarity, completeness, and redundancy of the prompt, aiming to enhance general readability and coherence before considering task-specific performance.

2. *Error Assessment* ($\mathcal{R}_{\text{error}}$): This reflection is generated based on model errors when $\hat{y} \neq y$ for a given input-output pair in $\mathcal{D}_{\text{train}}$. It identifies problematic nodes in $\mathcal{T}$ that contribute to model errors and offers targeted recommendations for improvement. These two reflections offer complementary insights: while the *Preliminary Assessment* focuses on improving the overall structure and clarity of the prompt, ensuring it aligns well with the general task requirements, the *Error Assessment* addresses specific weaknesses identified through examples, targeting sections that contributed to model errors.

To enhance generalization and prevent overfitting to isolated training errors, SCULPT employs two distinct aggregation mechanisms for *Error Assessments*, utilizing feedback from multiple examples to capture common error patterns. In the *Similarity-driven Aggregation*, the Critic clusters reflections by identifying recurring error patterns, such as shared node references and similar error types, allowing the system to address higher-level, task-independent issues. Alternatively, in *Node Reference-based Aggregation*, reflections are grouped according to the specific nodes they target, ensuring that common problems linked to particular sections of the prompt are consistently addressed. These aggregation techniques ensure that the refinements are broadly applicable, avoiding excessive tuning to individual instances, thereby improving robustness across diverse tasks.

3.3 Actor Module

The Actor Module $\mathcal{A}$ takes the feedback from the Critic Module and translates it into concrete changes in the prompt. To keep things efficient, the Actor works on a simplified version of the prompt tree, $\mathcal{T}_{\text{induced-subtree}}$, which includes only the parts of the prompt that need to be modified, along with the necessary context from parent nodes.

The Actor starts by addressing the feedback from the *Preliminary Assessment*, making broader structural changes like reordering sections, rephrasing unclear parts, merging similar sections, or adjusting examples for better clarity. After these general improvements, the Actor moves on to the feedback from the *Error Assessment*, which focuses on fixing task-specific errors.

The Actor generates a series of actions, denoted by $a = \{a_1, a_2, \dots\}$, to guide the modification process. Once all the necessary actions have been applied, the modified prompt tree, $\mathcal{T}'$, is transformed back into a standard text prompt $\mathcal{P}'$ by following the structure of the tree. A detailed list of actions that the Actor can perform is provided in Table 1.

3.4 Search Process

SCULPT incorporates a beam search strategy to explore and refine multiple candidate prompts in parallel. At each iteration t, a beam $\mathcal{B}_t$ maintains the top K candidate prompts, allowing for both exploitation of high-performing prompts and exploration of new variations.

Action	Description
Structural Reordering	Changing the order of sibling nodes
Instruction Update	Simplifying or adding new instructions
Example Addition	Adding new examples to a node
Example Deletion	Removing redundant examples from a node
Example Refinement	Improving existing examples in a node
Node Pruning	Removing unnecessary nodes
Node Expansion	Adding new nodes to address gaps
Node Merging	Combining nodes that have similar content

Table 1: Action types in SCULPT for prompt refinement

Algorithm 1 Prompt Optimization in SCULPT [1]

Initialize Beam $\mathcal{B}_0 = \{\mathcal{P}_0\}$, $t \leftarrow 0$, max_steps
while $t <$ max_steps **do**
  Evaluate $\mathcal{B}_t$ on a random subset of $\mathcal{D}_{\text{val}}$, obtain $\hat{\mu}_k$
  Calculate UCB scores $\text{UCB}_k(t)$ for each $\mathcal{P}_t^k$
  Select top K candidate prompts $\{\mathcal{P}_t^k\}_{k=1}^K$
  for each selected candidate prompt $\mathcal{P}_t^k$ **do**
    Critic generates *Preliminary Assessment* $\mathcal{R}_{\text{prelim}}^k$
    Critic generates *Error-Assessment Reflection* $\mathcal{R}_{\text{error}}^k$
    Aggregate $\mathcal{R}_{\text{error}}^k$ into clusters $\{\mathcal{C}_i^k\}_{i=1}^g$
    Actor applies actions $\mathcal{A}_{\text{prelim}}^k$
    for each Reflection cluster $\mathcal{C}_i^k$ **do**
      Actor applies actions $\mathcal{A}_{\text{error}}^{(k,i)}$
      Update to $\mathcal{P}_{t+1}^{(k,i)}$
      Add newly updated prompts to beam $\mathcal{B}_{t+1}$
    end for
  end for
  $t \leftarrow t + 1$
end while
Return top-K prompts from $\mathcal{B}_t$ sorted (non-asc order) by UCB scores

Since evaluating all candidate prompts on $\mathcal{D}_{\text{val}}$ is computationally expensive, we adopt an Upper Confidence Bound (UCB)-based selection strategy (Pryzant et al., 2023). The UCB score for each candidate $\mathcal{P}_t^k$ is calculated as:

$$\text{UCB}_k(t) = \hat{\mu}_k + c\sqrt{\frac{\log t}{n_k}},$$

where $\hat{\mu}_k$ is the estimated performance (using previous evaluations) of the candidate on the validation set $\mathcal{D}_{\text{val}}$, n_k is the number of times the candidate has been evaluated, and c is a hyperparameter controlling the trade-off between exploration and exploitation. This strategy ensures that promising candidates with fewer evaluations are explored further, while continuously refining the most promising candidates. A more detailed explanation of the beam search algorithm and the UCB-based selection strategy is provided in the Appendix A.

The overall optimization process for SCULPT , integrating prompt structuring, reflection-based refinement, and beam search with UCB-based selection, is provided in Algorithm 1. We have included detailed templates for both the Critic and Actor modules in Appendix G, while their step-by-step interactions and refinements are detailed in Appendix C. Fig. 3 visually demonstrates the improvements made to the prompt after optimization, showing the impact of these iterative refinements.

[1] We have removed t from notations for $\mathcal{C}$, $\mathcal{A}$, $\mathcal{R}$ for sake of clarity.

4 Experiments

4.1 Experimental Setup

Datasets: We selected four tasks from the BBH benchmark (Suzgun et al., 2023), a subset of the BIG-BENCH tasks (Srivastava et al., 2023), which focuses on challenging problems for LLMs. The tasks include *Causal Judgement* (**CJ**), which evaluates the model's ability to assess causes and effects, moral reasoning, and actionable conclusions; *Disambiguation QA* (**Dis**), where the model resolves ambiguous pronouns; *Formal Fallacies* (**FF**), testing the model's ability to differentiate between valid arguments and fallacies; and *Salient Translation Error Detection* (**ST**), where the model identifies critical errors in translations. Additionally, we included four RAI tasks: *Inappropriate Detection* (**Inapp**), *Hate-Speech Detection* (**Hate**), *Misinformation Detection* (**Misinfo**), and *Suicidal Ideation and Drug Use Detection* (**Selfharm**), each evaluating four levels of harm: *No Harm*, *Low Harm*, *Moderate Harm*, and *High Harm*. The initial prompts for RAI tasks were created by experts and are provided in Appendix J, while for the BBH tasks, we generated prompts using task descriptions from the README files (cf Appendix I). Additionally, Table 2 lists the word counts of the initial prompts used in each task, highlighting the length and complexity of these prompts.

Task	# Words
Formal Fallacy	382
Causal Judgement	367
Salient Translation	279
Disambiguation	346
Inappropriate	2644
Hate	1554
Misinformation	1335
SelfHarm	933

Table 2: Number of words in the initial prompts

Baseline Methods: We evaluate SCULPT against five baseline methods. (1) *APE* (Zhou et al., 2022), which generates new prompt candidates by leveraging few-shot examples and then rephrases these candidates to create multiple prompt variations, selecting the best based on validation performance. (2) *Long APE*, a variant of APE, which focuses on generating more descriptive prompts using a

Cause-and-Effect Recognition

~~Understand the association between cause and effect as it appears in common daily life scenarios.~~

Understand the association between cause and effect in daily life scenarios. Consider situations where multiple agents or events contribute to an outcome and determine shared responsibility or the primary cause accordingly.

* Recognize potential causes and effects within a given story.

~~* Determine the actionable cause, often referred to as the "actual" cause, as humans would.~~

~~Examples: {A heavy rain caused the city to flood.}, {The player's injury led to the team's loss.}~~

* Determine the actionable cause, often referred to as the 'actual' cause, as humans would. Highlight the importance of communication and adherence to instructions in determining causality.

Examples: {A band is performing at a concert. Did the concert receive a standing ovation because of the lead singer's performance? Yes, but the combined efforts of all band members, including the instrumentalists and backup singers, were significant in receiving the standing ovation.}...

* Recognize scenarios where multiple agents contribute to an event and determine shared responsibility. In such cases, no single agent can be solely responsible for the outcome.

Examples: {A team wins a relay race. Did the team win because of the final runner? No, the combined efforts of all team members contributed to the win.}...

Figure 3: Edits applied to the prompt using SCULPT, where strikethrough represents removed content and blue text indicates additions. These modifications involve example addition, node expansion, and instruction update.

predefined template (cf. Appendix F). (3) *APEX* (Hsieh et al., 2024), which refines prompts by performing sentence-level rephrasing through LLMs while utilizing historical changes for refinement. (4) *OPRO* (Yang et al., 2024), which generates new prompts by relying on historical prompt data and their validation scores. (5) *ProTeGi* (Pryzant et al., 2023), which detects errors in prompts, generates feedback based on these errors, and rephrases the prompts to produce optimized versions.

SCULPT Variants: To evaluate the role of reflection aggregation within SCULPT, we explore three key variants: (1) *SCULPT*, which employs *Node Reference-based Aggregation*, acting as our primary method, (2) *SCULPT+SA* , which utilizes *Similarity-driven Aggregation*, and (3) *SCULPT-NA*, which skips aggregation entirely. Furthermore, since baselines typically expand the search space through rephrasing, we introduce *SCULPT+RP*, where rephrasing is applied alongside *Node Reference-based Aggregation* to assess the impact of rephrasing on SCULPT's performance. Detailed information on rephrasing is provided in Appendix G.5.

Implementation Details: For most tasks, macro F1 scores are used due to the multiclass nature, while accuracy is reported for *ST* and *Dis*, as they are not classification tasks. The results reflect the average performance of the top four generated prompts, evaluated across three trials to ensure consistency. All generations were done using *GPT-4o* (OpenAI, 2024) with a temperature of 0.5, while the evaluation was performed using both *GPT-4o* and *Llama-3.1-8B* (Dubey et al., 2024), with a temperature of 0 to guarantee deterministic outputs.

We ensured fairness by assigning all methods the same search budget of 384 total prompt candidates. APE and Long-APE generated this number directly, while APEX and OPRO, which produce one prompt per step, were run for 384 steps. ProTeGi was run for 6 steps, producing 64 candidates per step. SCULPT, generating up to 16 candidates per step, could have run for 24 steps, but experiments indicated that performance peaked at 8 steps, with additional steps leading to overfitting. Thus, the reported SCULPT performance reflects the results after 8 optimization steps.

5 Results and Analysis

Performance Comparison with Baselines: Table 3 presents the results for SCULPT variants and baseline methods across eight tasks using GPT-4o. SCULPT consistently outperforms all baselines on both BBH and RAI tasks, showing significant improvements over the initial prompt. While ProTeGi and APEX struggle to generate meaningful gains, often performing similarly to the initial prompt on BBH tasks, OPRO shows minor improvements but lacks consistency across different tasks. Long-APE delivers strong performance on tasks such as *Dis* and *FF*, surpassing APE and the initial prompt, yet it underperforms on other tasks. Importantly, SCULPT demonstrates lower variance in performance compared to other methods, signifying greater stability and reliability across multiple runs.

In Table 4, we show results for four tasks using Llama 3.1. Apart from *Misinfo*, SCULPT significantly outperforms all baseline methods across tasks. Interestingly, OPRO delivers better improvements with *Llama-3.1-8B* than GPT-4o, even surpassing ProTeGi on 3 out of 4 tasks, suggesting that model-specific behavior can influence the effectiveness of optimization strategies. SCULPT continues to exhibit robust performance, reinforcing its adaptability across different models and tasks.

Ablation Study of SCULPT: Table 3 highlights the performance of different SCULPT variants.

Method	ST	Dis	CJ	FF	Inapp	Misinfo	Hate	Selfharm	Avg
Initial Prompt	62.3 ± 1.0	74.1 ± 0.7	71.1 ± 0.1	80.2 ± 1.4	46.6 ± 1.3	51.5 ± 0.6	46.8 ± 0.1	66.4 ± 0.5	62.4
APE	51.0 ± 1.6	74.3 ± 1.1	71.8 ± 2.6	75.3 ± 2.5	45.3 ± 0.4	31.9 ± 5.9	29.3 ± 2.7	38.1 ± 0.4	52.1
Long-APE	52.7 ± 4.0	78.0 ± 1.0	72.2 ± 3.3	81.3 ± 0.3	42.4 ± 0.9	43.6 ± 2.1	37.6 ± 0.5	44.2 ± 1.3	56.5
APEX	62.71 ± 0.6	61.5 ± 21.2	70.5 ± 0.7	80.6 ± 1.26	48.0 ± 0.4	50.5 ± 0.8	47.1 ± 0.3	65.2 ± 0.7	60.8
OPRO	64.6 ± 2.0	75.1 ± 1.4	72.7 ± 1.1	80.7 ± 2.1	46.6 ± 2.2	51.0 ± 4.0	40.9 ± 3.6	65.1 ± 2.1	62.1
ProTeGi	65.5 ± 3.4	74.8 ± 1.2	68.9 ± 2.1	70.1 ± 3.9	44.8 ± 5.4	54.8 ± 1.2	51.9 ± 1.6	66.5 ± 2.1	62.1
SCUPLT-NA	65.2 ± 1.1	77.3 ± 1.2	75.4 ± 2.4	83.1 ± 1.1	53.6 ± 0.7	53.6 ± 2.5	51.9 ± 0.3	65.5 ± 0.9	65.6
SCUPLT+SA	66.2 ± 2.1	77.6 ± 1.9	**76.9 ± 1.9**	83.7 ± 1.1	**55.3 ± 0.7**	**56.7 ± 0.8**	53.1 ± 0.3	68.8 ± 0.3	67.3
SCUPLT+RP	67.6 ± 1.9	78.0 ± 0.6	75.1 ± 2.0	**84.7 ± 1.1**	55.0 ± 0.8	55.3 ± 2.4	52.9 ± 1.4	**69.0 ± 1.8**	67.2
SCUPLT	**68.8 ± 1.5**	**80.1 ± 1.9**	75.9 ± 1.5	83.7 ± 2.5	55.0 ± 0.8	54.9 ± 0.6	**53.1 ± 1.2**	68.5 ± 1.0	**67.5**

Table 3: Performance Comparison using GPT-4o across various tasks

Method	Dis	CJ	Misinfo	Selfharm	Avg
Initial Prompt	57.3 ± 1.8	61.8 ± 1.2	36.1 ± 0.8	34.9 ± 0.8	47.5
APE	64.3 ± 0.7	56.0 ± 1.6	33.9 ± 2.3	29.4 ± 1.0	45.9
Long-APE	60.3 ± 1.9	60.0 ± 2.4	39.3 ± 2.9	33.9 ± 3.2	48.4
APEX	61.5 ± 3.6	59.4 ± 3.6	28.8 ± 5.0	39.6 ± 1.0	47.3
OPRO	49.1 ± 18.4	62.9 ± 2.7	**43.1 ± 8.3**	51.8 ± 2.4	51.7
ProTeGi	61.3 ± 3.5	58.4 ± 4.3	34.5 ± 4.7	41.6 ± 1.2	49.0
SCUPLT	**65.3 ± 4.3**	**64.9 ± 1.5**	37.3 ± 2.8	**54.5 ± 5.1**	**55.5**

Table 4: Performance Comparison using Llama 3.1

	CJ		Misinfo	
Method	LP	GP	LP	GP
Initial Prompt	69.5	70.5	42.2	26.5
APEX	69.0	70.2	43.0	28.3
OPRO	70.0	69.6	48.1	43.0
ProTeGi	68.3	69.9	52.4	37.8
SCULPT	**74.1**	**73.2**	**56.4**	**50.2**

Table 5: Performance of methods under Localized (LP) and Global (GP) perturbations in the prompt.

SCULPT, which uses *Node Reference-based Aggregation*, achieves the best overall results. This variant excels because the Actor can apply all reflections related to a specific node in the prompt simultaneously, ensuring that refinements are comprehensive and targeted. In contrast, *SCUPLT+SA* (Similarity-driven Aggregation), which clusters reflections based on similarities in inputs or predictions, may fail to aggregate all reflections for the same node. As a result, some potential improvements for that node may be missed, leading to less precise refinements. While *SCULPT+RP* (rephrasing) delivers results comparable to the standard SCULPT, its impact is inconsistent. Rephrasing does not always lead to further improvements, making it an optional step rather than a core part of the SCULPT.

Robustness to Prompt Perturbations: We conducted a robustness evaluation to assess SCULPT and baseline methods on erroneous or poorly structured initial prompts, simulating real-world conditions with significant noise. Two types of perturbations were applied: *Localized Perturbations* (**LP**), where examples were swapped between categories within the prompt, causing moderate disruptions, and *Global Perturbations* (**GP**), where entire instruction-example pairs were swapped between categories, creating severe misalignment in structure. GP represents the more difficult challenge due to the full mismatch between instructions and examples.

As shown in Table 5, SCULPT consistently outperforms baselines across both settings. In the LP scenario, SCULPT's node-specific approach efficiently repositions misplaced examples, while baselines like APEX and ProTeGi struggle with even moderate inconsistencies, leading to performance drops. In the GP setting, SCULPT demonstrates superior recovery, maintaining robust performance as other methods fail to correct the structural confusion. This evaluation highlights SCULPT's ability to manage both subtle and severe prompt errors, ensuring reliable outcomes in challenging conditions.

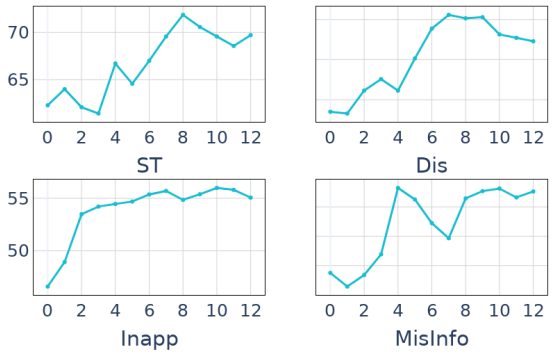
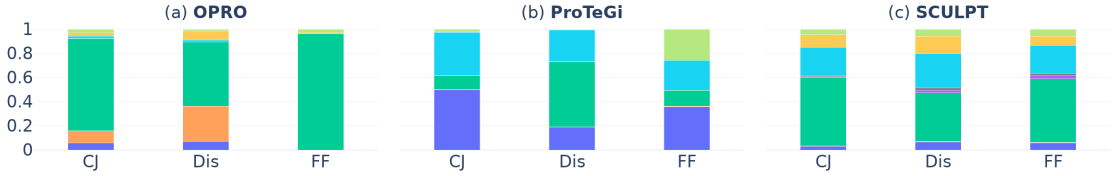
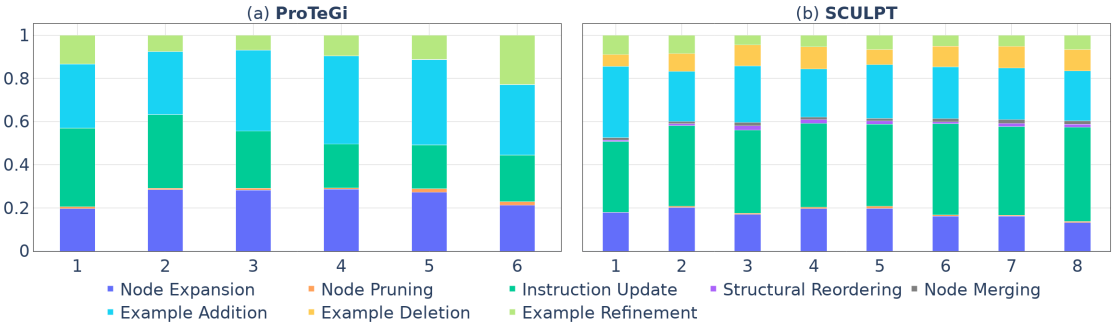


Figure 4: Performance across optimization steps

Performance across Optimization Steps: To assess the impact of the number of optimization steps, we plot the performance progression after each step in Fig. 4. The results indicate that performance tends to plateau around step 8 on average. In some cases, continuing optimization beyond 8 steps may lead to overfitting on the training data. Based on this observation, we report the performance at the end of 8 optimization steps in our evaluation.



(a) Action distribution across tasks, highlighting SCULPT's consistent application of refinements across different scenarios.



(b) Action distribution over optimization steps, illustrating SCULPT's stable approach to prompt refinement across iterations.

Figure 5: Comparison of action distributions for ProTeGi and SCULPT across tasks (a) and optimization steps (b), emphasizing SCULPT's controlled and balanced strategy for optimizing long prompts.

Action Distribution Analysis Across Tasks: In Fig. 5a, we illustrate the distribution of actions applied by OPRO, ProTeGi, and SCULPT across various tasks. Since OPRO and ProTeGi do not explicitly define their action types, we used LLMs to analyze their behavior and classify changes into predefined action categories (cf Appendix H). This classification offers a clearer view of how these methods refine prompts.

SCULPT demonstrates a consistent and balanced distribution of actions across tasks, including *Instruction Updates*, *Example Addition*, *Example Deletion*, and *Node Expansion*. In contrast, OPRO and ProTeGi exhibit more variability. ProTeGi, for instance, relies heavily on *Node Expansion* ($\sim$40%) and *Example Addition* ($\sim$30%), indicating that it tends to address prompt issues by adding content, which may lead to overfitting. OPRO, an implicit reflection method, applies the least controlled refinements, resulting in more varied and less systematic actions. This analysis underscores SCULPT's advantage in applying more controlled, targeted prompt refinements. Its mechanism enables systematic improvements, avoiding the content over-expansion seen in ProTeGi and offering more focused control than OPRO, ultimately leading to more robust, generalizable optimizations.

Action Distribution Analysis Across Steps: Fig. 5b shows how action types evolve across optimization steps, where actions have been averaged across tasks. SCULPT maintains a balanced and consistent distribution of actions throughout steps, ensuring controlled and focused refinements. In contrast, ProTeGi exhibits higher variability, with increasing reliance on *Example Addition* with steps that could lead to overfitting. This further highlights SCULPT's controlled approach, which ensures more generalizable optimizations across steps.

6 Conclusion

In this work, we present SCULPT, a novel framework for systematically optimizing long prompts for LLMs. SCULPT leverages a hierarchical tree structure and a node-specific refinement process, addressing the shortcomings of existing prompt optimization techniques like APEX, which struggle with handling complex, multi-instruction prompts. SCULPT consistently outperforms these baselines by maintaining a balanced distribution of actions across tasks, leading to more controlled, high-quality refinements. Additionally, SCULPT demonstrates robustness against noisy prompts, efficiently correcting errors and misalignments caused by prompt perturbations, outperforming methods like OPRO and ProTeGi in such challenging scenarios. Our experiments across BBH and RAI benchmarks highlight SCULPT's effectiveness in delivering scalable, reliable prompt optimization. Qualitative analysis further shows that SCULPT is remarkably consistent and systematic in its refinements, making it an ideal solution for real-world applications.

7 Limitations and Future Work

While SCULPT demonstrates significant improvements in optimizing long prompts, it has certain limitations. First, SCULPT relies on an initial manually created prompt, which introduces variability and dependence on human expertise. Future work could integrate SCULPT with methods like Long-APE to automate the prompt initialization phase, allowing SCULPT to refine prompts generated from scratch. This would further reduce manual intervention and enhance scalability across diverse applications.

Moreover, our evaluation is limited to two LLMs: GPT-4o and Llama 3.1, due to computational constraints. A more comprehensive study across various LLM sizes and architectures could offer deeper insights into SCULPT's generalizability and effectiveness at different scales. Additionally, SCULPT has only been tested on English-language prompts. Extending its applicability to multilingual settings would make it more versatile in global contexts and broader tasks.

Finally, automating the hierarchical structuring of prompts and improving SCULPT's handling of complex relationships, such as co-references and logical dependencies between examples and instructions, is a promising avenue for future work. Expanding SCULPT's capabilities to manage these intricate structures will enhance its robustness, particularly in handling adversarial inputs and more complex real-world scenarios.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. arXiv preprint arXiv:2404.14219.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. arXiv preprint arXiv:2303.08774.

Tom B Brown. 2020. Language models are few-shot learners. arXiv preprint arXiv:2005.14165.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. arXiv preprint arXiv:2303.12712.

Kaiyan Chang, Songcheng Xu, Chenglong Wang, Yingfeng Luo, Tong Xiao, and Jingbo Zhu. 2024. Efficient prompting methods for large language models: A survey. arXiv preprint arXiv:2404.01077.

Lichang Chen, Jiuhai Chen, Tom Goldstein, Heng Huang, and Tianyi Zhou. 2024a. InstructZero: Efficient instruction optimization for black-box large language models. In Proceedings of the 41st International Conference on Machine Learning, volume 235 of Proceedings of Machine Learning Research, pages 6503–6518. PMLR.

Yongchao Chen, Jacob Arkin, Yilun Hao, Yang Zhang, Nicholas Roy, and Chuchu Fan. 2024b. Prompt optimization in multi-step tasks (promst): Integrating human feedback and preference alignment. arXiv preprint arXiv:2402.08702.

Jiale Cheng, Xiao Liu, Kehan Zheng, Pei Ke, Hongning Wang, Yuxiao Dong, Jie Tang, and Minlie Huang. 2023. Black-box prompt optimization: Aligning large language models without model training. arXiv preprint arXiv:2311.04155.

Yihong Dong, Kangcheng Luo, Xue Jiang, Zhi Jin, and Ge Li. 2024. PACE: Improving prompt with actor-critic editing for large language model. In Findings of the Association for Computational Linguistics ACL 2024, pages 7304–7323, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. arXiv preprint arXiv:2407.21783.

Chrisantha Fernando, Dylan Sunil Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2024. Promptbreeder: Self-referential self-improvement via prompt evolution. In Proceedings of the 41st International Conference on Machine Learning, volume 235 of Proceedings of Machine Learning Research, pages 13481–13544. PMLR.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. 2024. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. In The Twelfth International Conference on Learning Representations.

Or Honovich, Uri Shaham, Samuel R. Bowman, and Omer Levy. 2023. Instruction induction: From few examples to natural language task descriptions. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1935–1952, Toronto, Canada. Association for Computational Linguistics.

Cho-Jui Hsieh, Si Si, Felix Yu, and Inderjit Dhillon. 2024. Automatic engineering of long prompts. In Findings of the Association for

Computational Linguistics ACL 2024, pages 10672–10685, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Ellen Jiang, Kristen Olson, Edwin Toh, Alejandra Molina, Aaron Donsbach, Michael Terry, and Carrie J Cai. 2022. Promptmaker: Prompt-based prototyping with large language models. In CHI Conference on Human Factors in Computing Systems Extended Abstracts, pages 1–8.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? Transactions of the Association for Computational Linguistics, 8:423–438.

Gurusha Juneja, Nagarajan Natarajan, Hua Li, Jian Jiao, and Amit Sharma. 2024. Task facet learning: A structured approach to prompt optimization. arXiv preprint arXiv:2406.10504.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T Joshi, Hanna Moazam, et al. 2023. Dspy: Compiling declarative language model calls into self-improving pipelines. arXiv preprint arXiv:2310.03714.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. arXiv preprint arXiv:2104.08691.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. arXiv preprint arXiv:2101.00190.

Shengcai Liu, Caishun Chen, Xinghua Qu, Ke Tang, and Yew-Soon Ong. 2024. Large language models as evolutionary optimizers. In 2024 IEEE Congress on Evolutionary Computation (CEC), pages 1–8. IEEE.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Z Yang, and J Tang. 2021. Gpt understands, too. arxiv. arXiv preprint arXiv:2103.10385.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Ruotian Ma, Xiaolei Wang, Xin Zhou, Jian Li, Nan Du, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. Are large language models good prompt optimizers? arXiv preprint arXiv:2402.02101.

OpenAI. 2023. Best practices for prompt engineering with the openai api. Accessed: 2024-10-10.

OpenAI. 2024. Introducing gpt-4o. https://openai.com/index/hello-gpt-4o/. Accessed: 2024-09-16.

Archiki Prasad, Peter Hase, Xiang Zhou, and Mohit Bansal. 2022. Grips: Gradient-free, edit-based instruction search for prompting large language models. arXiv preprint arXiv:2203.07281.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with" gradient descent" and beam search. arXiv preprint arXiv:2305.03495.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying lms with mixtures of soft prompts. arXiv preprint arXiv:2104.06599.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In Extended abstracts of the 2021 CHI conference on human factors in computing systems, pages 1–7.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 2024. A systematic survey of prompt engineering in large language models: Techniques and applications. arXiv preprint arXiv:2402.07927.

Tobias Schnabel and Jennifer Neville. 2024. Prompts as programs: A structure-aware approach to efficient compile-time prompt optimization. arXiv preprint arXiv:2404.02319.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Johan Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew M. Dai, Andrew La, Andrew Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakaş, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartłomiej Bojanowski, Batuhan Özyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Bryan Orinion, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, Cesar Ferri, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Christopher Waites, Christian Voigt, Christopher D Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Dan Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, C. Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Moseguí González, Danielle Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito,

Dar Gilboa, David Dohan, David Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Dylan Schrader, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth Donoway, Ellie Pavlick, Emanuele Rodolà, Emma Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A Chi, Ethan Dyer, Ethan Jerzak, Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fanyue Xia, Fatemeh Siar, Fernando Martínez-Plumed, Francesca Happé, Francois Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Xinyue Wang, Gonzalo Jaimovitch-Lopez, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Hannah Kim, Hannah Rashkin, Hannaneh Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Francis Anthony Shevlin, Hinrich Schuetze, Hiromu Yakura, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, Jackson Kernion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, James B Simon, James Koppel, James Zheng, James Zou, Jan Kocon, Jana Thompson, Janelle Wingfield, Jared Kaplan, Jarema Radom, Jascha Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesujoba Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Joan Waweru, John Burden, John Miller, John U. Balis, Jonathan Batchelder, Jonathan Berant, Jörg Frohberg, Jos Rozen, Jose Hernandez-Orallo, Joseph Boudeman, Joseph Guerr, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kaustubh Dhole, Kevin Gimpel, Kevin Omondi, Kory Wallace Mathewson, Kristen Chiafullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras-Ochando, Louis-Philippe Morency, Luca Moschella, Lucas Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros-Colón, Luke Metz, Lütfi Kerem Senel, Maarten Bosma, Maarten Sap, Maartje Ter Hoeve, Maheen Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, Maria Jose Ramirez-Quintana, Marie Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L Leavitt, Matthias Hagen, Mátyás Schubert, Medina Orduna Baitemirova, Melody Arnaud, Melvin McElrath, Michael Andrew Yee, Michael Cohen, Michael Gu, Michael Ivanitskiy, Michael Starritt, Michael Strube, Michał Swędrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimee Xu, Mirac Suzgun, Mitch Walker, Mo Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdeh Gheini, Mukund Varma T, Nanyun Peng, Nathan Andrew Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nicole Martinez, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, Nitish Shirish Keskar, Niveditha S. Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter W Chang, Peter Eckersley, Phu Mon Htut, Pinyu Hwang, Piotr Miłkowski, Piyush Patil, Pouya Pezeshkpour, Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ramon Risco, Raphaël Millière, Rhythm Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbe Raymaekers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, Ronan Le Bras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Russ Salakhutdinov, Ryan Andrew Chi, Seungjae Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib Singh, Saif M. Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Gruetter, Samuel R. Bowman, Samuel Stern Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shane Gu, Shubh Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima Shammie Debnath, Siamak Shakeri, Simon Thormeyer, Simone Melzi, Siva Reddy, Sneha Priscilla Makini, Soo-Hwan Lee, Spencer Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Divic, Stefano Ermon, Stella Biderman, Stephanie Lin, Stephen Prasad, Steven Piantadosi, Stuart Shieber, Summer Misherghi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq Ali, Tatsunori Hashimoto, Te-Lin Wu, Théo Desbordes, Theodore Rothschild, Thomas Phan, Tianle Wang, Tiberius Nkinyili, Timo Schick, Timofei Kornev, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, Vinay Venkatesh Ramasesh, vinay uday prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, Wout Vossen, Xiang Ren, Xiaoyu Tong, Xinran Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yangqiu Song, Yasaman Bahri, Yejin Choi, Yichi Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yufang Hou, Yuntao Bai, Zachary Seid, Zhuoye Zhao, Zijian Wang, Zijie J. Wang, Zirui Wang, and Ziyi Wu. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. Transactions on Machine Learning Research.

Hong Sun, Xue Li, Yinchuan Xu, Youkow Homma, Qi Cao, Min Wu, Jian Jiao, and Denis Charles. 2023. Autohint: Automatic prompt optimization with hint generation. arXiv preprint arXiv:2307.07415.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-

bench tasks and whether chain-of-thought can solve them. In Findings of the Association for Computational Linguistics: ACL 2023, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. In The Eleventh International Conference on Learning Representations.

Hanwei Xu, Yujun Chen, Yulun Du, Nan Shao, Yanggang Wang, Haiyu Li, and Zhilin Yang. 2022. Gps: Genetic prompt search for efficient few-shot learning. arXiv preprint arXiv:2210.17041.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2024. Large language models as optimizers. In The Twelfth International Conference on Learning Representations.

Qinyuan Ye, Maxamed Axmed, Reid Pryzant, and Fereshte Khani. 2023. Prompt engineering a prompt engineer. arXiv preprint arXiv:2311.05661.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Zhi Huang, Carlos Guestrin, and James Zou. 2024. Textgrad: Automatic" differentiation" via text. arXiv preprint arXiv:2406.07496.

JD Zamfirescu-Pereira, Richmond Y Wong, Bjoern Hartmann, and Qian Yang. 2023. Why johnny can't prompt: how non-ai experts try (and fail) to design llm prompts. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, pages 1–21.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In International conference on machine learning, pages 12697–12706. PMLR.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers. arXiv preprint arXiv:2211.01910.

Appendix

Table of Contents

A UCB-based Prompt Selection Strategy **12**

B Task and Initial Prompt Statistics **13**

C Critic and Actor Interactions in SCULPT **13**

D Action Types Distribution across tasks and steps **15**

E APE Template **24**

F Long APE Template **24**

G SCULPT Prompt Templates for Critic and Actor **24**

G.1 Critic Template for Preliminary Assessment 24

G.2 Critic Template for Error Assessment 25

G.3 Critic Template for Error Assessment using Similarity-driven Aggregation 25

G.4 Actor Module Template 26

G.5 Rephrasing Template 27

H Action Identification in OPRO and ProTeGi Optimization **27**

I BBH Prompt Generation **28**

J Initial Prompts **28**

K Perturbed Prompts **36**

A UCB-based Prompt Selection Strategy

Evaluating the generated candidate prompts on the validation set $\mathcal{D}_{\text{val}}$ is a computationally expensive process. To minimize these computations, we have used the Upper Confidence Bound (UCB) Bandit algorithm as mentioned in (Pryzant et al., 2023). This helps to minimize the number of prompts to be evaluated as well as the number of validation set samples to evaluate them on. This is done based on the proposal distribution of prompt performance which is updated after each evaluation round. At the end top b prompts with highest weight in the distribution are selected.

Dataset	Validation Size	Train Size	Test Size
Causal Judgement	19	36	129
Disambiguation QA	24	49	174
Formal Fallacies	24	49	174
Salient Translation	24	49	174
Inappropriate	122	242	851
Misinformation	122	242	851
Hate	122	242	851
SelfHarm	122	242	851

Table 6: Dataset size information

See Algorithm 2 for details, where $Q_t(p_i)$ is the estimated performance of prompt p_i at time step t, $N_t(p_i)$ is the total queries for prompt i so far at time t, and c is the exploration parameter.

Algorithm 2 UCB Bandits Candidate Selection

Require n prompts $p_1, p_2, \ldots, p_n$, dataset $\mathcal{D}_{\text{val}}$, T time steps and metric function m
Initialize: $N_t(p_i) \leftarrow 0$ for all $i = 1,\ldots,n$
Initialize: $Q_t(p_i) \leftarrow 0$ for all $i = 1,\ldots,n$
for $t = 1, \ldots, T$ **do**
  Sample uniformly $\mathcal{D}_{\text{sample}} \subset \mathcal{D}_{\text{val}}$
  $p_i \leftarrow \arg\max_p Q_t(p) + c\sqrt{\frac{\log t}{N_t(p)}}$
  Observe reward $r_{i,t} = m(p_i, \mathcal{D}_{\text{sample}})$
  $N_t(p_i) \leftarrow N_t(p_i) + |\mathcal{D}_{\text{sample}}|$
  $Q_t(p_i) \leftarrow Q_t(p_i) + \frac{r_{i,t}}{N_t(p_i)}$
end for
return $SelectTop_b(Q_T)$

B Task and Initial Prompt Statistics

Table 6 presents the number of examples in the training, validation, and test sets for each task, offering an overview of the dataset size. Additionally, Table 2 lists the word counts of the initial prompts used in each task, highlighting the length and complexity of these prompts. This information emphasizes the challenges posed by long and unstructured prompts, which require systematic optimization to ensure model performance. We have provided the list of initial prompts in Section J.

C Critic and Actor Interactions in SCULPT

This section illustrates the interactions between the Critic and Actor modules within SCULPT by presenting both the preliminary and error-assessment reflections, the Actor's responses to each type of feedback, and the resulting prompt updates. Specifically, we showcase how these actions contribute to prompt refinements during the first round of the Salient Translation task, demonstrating the iterative role of both modules in improving the prompt's clarity and alignment with task requirements.

Critic's Preliminary Assessment: Table 7 shows the preliminary feedback provided in round 1. The feedback identifies multiple areas for improvement, including adding examples for different types of translation errors and rephrasing certain parts of the prompt to enhance clarity and relevance.

Actor Response to Preliminary Assessment: The Actor module processes the Critic's feedback and suggests a set of actions, which are summarized in Table 8. One key action involves adding specific examples for "Named Entities" errors, while another focuses on rephrasing the task description in the 'Task > body' section for greater clarity.

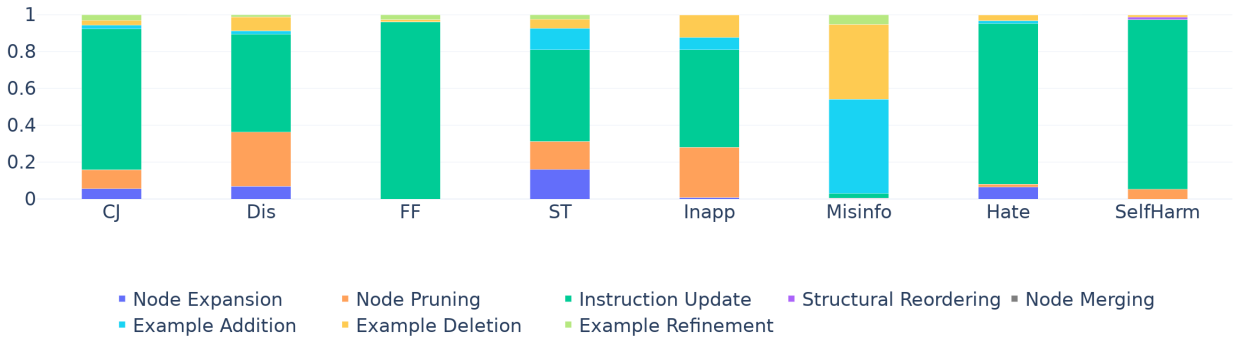
Updated Prompt Based on Preliminary Assessment: After applying the Preliminary Assessment, notable improvements are observed in the prompt. The task description has been rephrased for clarity, and new examples for each translation error type have been added. See Table 11 for the updated prompt, which shows significant refinements compared to the initial prompt in Table J.0.3.

Critic's Error Assessment: Table 9 provides the error assessment based on the Critic's evaluation of the prompt in response to specific model errors. The reflection highlights areas where new examples need to be added and suggests rephrasing sections to clarify definitions of error types, ensuring fewer ambiguities.

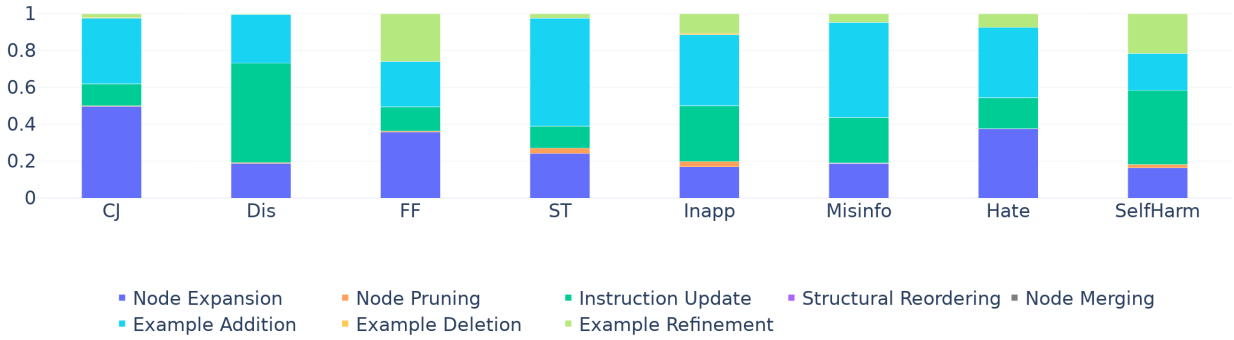
Actor Response to Error Assessment: In response to the error-assessment feedback, the Actor module suggests targeted actions, which are listed in Table 10. These include adding examples for error type 1 and rephrasing sections as needed to avoid confusion and improve clarity.

Updated Prompt Based on Error Assessment: The updated prompt, following both initial assessments and error analysis from the first round, is presented in Table 12. In contrast to the original version (Table J.0.3), the revised prompt integrates additional examples and restructured sections. This demonstrates the capability of SCULPT to systematically refine prompts through controlled reflections and targeted adjustments.

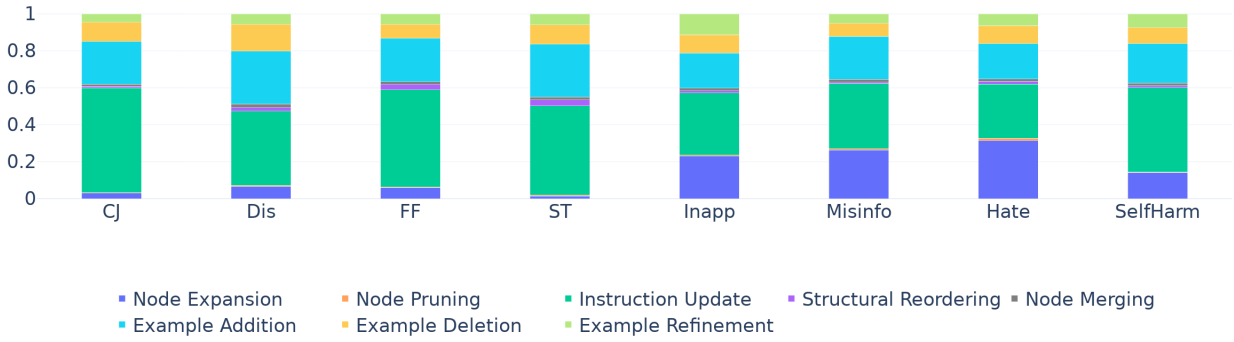
Final Prompt Post-Optimization: The fully optimized prompt, after the entire SCULPT optimization process, is presented in Table 13. This refined version shows significant improvements over the initial prompt (Table J.0.3). Key enhancements include a clearer redefinition of error categories,



(a) OPRO



(b) ProTeGi



(c) SCULPT

Figure 6: Action distribution across tasks of OPRO, ProTeGi and SCULPT.

refined examples, and improved clarity in the final output classes. These adjustments, made after the Preliminary and Error Assessment updates (refer to Table 12), ensure that the user is guided more effectively through the task. This final prompt underscores the systematic nature of SCULPT's iterative refinement process, showcasing its ability to optimize prompt design efficiently and effectively.

D Action Types Distribution across tasks and steps

In Fig. 6, we present the distribution of action types applied by OPRO, ProTeGi, and SCULPT across different tasks, illustrating how each method handles prompt refinements.

Additionally, Figures 7 and 8 show the action distribution employed by SCULPT over successive optimization steps, providing insights into how the system refines prompts over time. Similarly, Figures 9 and 10 illustrate the stepwise action distribution for ProTeGi, highlighting differences in how the two methods evolve their optimizations throughout the process.

Prompt Reference	Preliminary Assessment
- Task> body", - Error Identification> body - Error Identification> 1.> body - Error Identification> 2.> body - Error Identification> 3.> body - Error Identification> 4.> body - Error Identification> 5.> body - Error Identification> 6.> body - Performance Analysis> body - Additional points> body - Options> body - Output format> body	- Add relevant examples to each error class to provide clarity. - Simplify language where possible to make instructions clearer. - Combine sections where appropriate to reduce redundancy and improve flow.

Table 7: Example of Preliminary Critic Reflection

Action Type	Section Reference	Action Description
Example Update- Addition	Error Identification> 1.	Add examples that illustrate 'Named entities' errors, such as changes in names, places, or locations. For instance, 'A person's name changed from John to Jack' or 'A city name changed from New York to Los Angeles'
"Section Rephrase"	"Task> body"	Your task is to identify the type of translation error in a given source-translation pair. You will be given sentences with specific errors manually introduced. Determine which of the six error classes the translation error belongs to.

Table 8: Actor Responses based on Preliminary Assessment

Input Example & Prediction	Prompt Reference	Error Assessment
Input: "Source: Eleonore Lingnau-Kluge war eine deutsche Malerin. Translation: Eleonore Lingnau-Kluge was a German dancer." Expected Output: "(F)" Prediction: "(D)"	- Error Identification> 1.> body - Error Identification> 5.> body - Error Identification> 6.> Examples	- Add examples that clearly differentiate factual errors from named entity changes. - Clarify the definitions of each error class to avoid confusion.
"Input: "Source: Pedro Morenés y Álvarez de Eulate ist ein spanischer Politiker der Partido Popular. Translation: Pedro is a Spanish politician of the Popular Party." Expected Output: "(D)" Prediction: "(E)"	- Error Identification> 1.> body - Error Identification> 6.> Examples	- Add examples that highlight named entity changes, especially when names are shortened or altered.", - Emphasize the importance of preserving named entities in translations.

Table 9: Error Assessment from Critic

Action Type	Section Reference	Action Description
Example Update- Addition	Error Identification> 1.> Examples	Add examples that highlight named entity changes, especially when names are shortened or altered. For instance: 'A politician's name changed from Pedro Morenés y Álvarez de Eulate to Pedro' and 'An actor's name changed from Martin Stephen McCann to McCann'.
Section Rephrase	Error Identification> 1.> body	Named entities: Look for changes in names, places, locations, etc. Ensure that names are preserved accurately, even when shortened or altered.

Table 10: Actor Response based on Error Assessment

```
# Task
Your task is to identify the type of translation error in a given source-translation pair. You will be given sentences with
specific errors manually introduced. Determine which of the six error classes the translation error belongs to.

# Error Identification
Analyze the source-translation pair and identify the error based on the following classes:
* Named entities: Look for changes in names, places, locations, etc.
Examples: A company name changed from Apple to Microsoft, A country name changed from France to Germany, A person's name
     changed from John to Jack, A city name changed from New York to Los Angeles
* Numerical values: Check for alterations in numbers, dates, or units.
Examples: A date changed from 2021 to 2020, A price changed from $50 to $55, A time changed from 3 PM to 4 PM, A measurement
     unit changed from meters to feet, A quantity changed from 100 to 150
* Modifiers or adjectives: Identify changes in descriptors pertaining to a noun.
Examples: The adjective changed from big to small., The descriptor changed from red to blue., The modifier changed from happy
     to sad., The descriptor changed from old to new., The adjective changed from tall to short.
* Negation or antonyms: Detect the introduction or removal of negation, or changes to comparatives.
Examples: The comparative changed from 'less important' to 'more important'., The negation changed from 'is not' to 'is'., The
     phrase changed from 'He is not happy' to 'He is happy'., The comparative changed from 'better' to 'worse'., The sentence
     changed from 'She never goes to the gym' to 'She always goes to the gym'.
* Facts: Spot trivial factual errors not covered by the above classes.
Examples: The fact changed from The capital of France is Paris to The capital of France is Berlin, The fact changed from Humans
     have 206 bones in their body to Humans have 210 bones in their body, The fact changed from The Great Wall of China is
     visible from space to The Great Wall of China is not visible from space
* Dropped content: Notice if a significant clause is missing from the translation.
Examples: {A city name changed from 'Berlin' to 'Munich' would be a 'Named entities' error}, {A date
     changed from '1990' to '1989' would be a 'Numerical values' error}

# Performance Analysis
Understand that language models perform differently across error classes:
* Models like XLM-Roberta may struggle with named entities, dropped content, and modifiers/adjectives.
* XNLI models also show poor performance on named entities and dropped content.

# Additional points
Keep in mind the following points while identifying errors:
* Ensure minimal impact on translation fluency while identifying errors.
* Focus on salient source information to detect errors effectively.
* Remember that each translation contains only one of the six error classes.

# Options
(A) Modifiers or Adjectives
(B) Numerical Values
(C) Negation or Antonyms
(D) Named Entities
(E) Dropped Content
(F) Facts

# Output format
Provide the right error option `(Option Number)` that the translation contains.
```

Table 11: Updated Prompt after Preliminary Assessment. Instruction update are highlighted in blue, Example Addition are marked with mahogany color, and Node Expansion are marked in green.

```
# Task
Your task is to identify the type of translation error in a given source-translation pair. You will be given sentences with
specific errors manually introduced. Determine which of the six error classes the translation error belongs to.

# Error Identification
Analyze the source-translation pair and identify the error based on the following classes:
* Named entities: Look for changes in names, places, locations, etc. Ensure that names are preserved
     accurately, even when shortened or altered.
Examples: A politician's name changed from Pedro Morenés y Álvarez de Eulate to Pedro, A company name changed from Apple to
     Microsoft, An actor's name changed from Martin Stephen McCann to McCann, A country name changed from France to Germany, A
     person's name changed from John to Jack, A city name changed from New York to Los Angeles
* Numerical values: Check for alterations in numbers, dates, or units. Ensure that numerical values are
     preserved accurately.
Examples: A date changed from 2021 to 2020, A price changed from $50 to $55, A time changed from 3 PM to 4 PM, The population
     number changed from 5491 to 5000, A measurement unit changed from meters to feet, A quantity changed from 100 to 150
* Modifiers or adjectives: Identify changes in descriptors pertaining to a noun. Ensure that descriptive
      terms are preserved accurately.
Examples: The adjective changed from big to small., The descriptor changed from red to blue., The modifier changed from happy
     to sad., The description changed from small town to city., The descriptor changed from old to new., The nationality
     changed from German to French., The adjective changed from tall to short.
* Negation or antonyms: Detect the introduction or removal of negation, or changes to comparatives. Ensure that negation and
     antonyms are preserved accurately.
Examples: The comparative changed from 'less important' to 'more important'., The negation changed from 'is not' to 'is'., The
     description changed from rural to urban., The phrase changed from 'He is not happy' to 'He is happy'., The comparative
     changed from 'better' to 'worse'., The sentence changed from 'She never goes to the gym' to 'She always goes to the gym'.
* Facts: Spot trivial factual errors not covered by the above classes. Ensure that factual information
     remains accurate and unchanged.
Examples: The fact changed from The Pacific Ocean is the largest ocean to The Atlantic Ocean is the largest ocean, The fact
     changed from The Earth orbits the Sun to The Sun orbits the Earth, The fact changed from The Great Wall of China is
     visible from space to The Great Wall of China is not visible from space, The fact changed from The capital of France is
     Paris to The capital of France is Berlin, The fact changed from Humans have 206 bones in their body to Humans have 210
     bones in their body
* Dropped content: Notice if a significant clause is missing from the translation.
Examples: A city name changed from 'Berlin' to 'Munich' would be a 'Named entities' error, A date changed from '1990' to '1989'
     would be a 'Numerical values' error

# Performance Analysis
Understand that language models perform differently across error classes:
* Models like XLM-Roberta may struggle with named entities, dropped content, and modifiers/adjectives.
* XNLI models also show poor performance on named entities and dropped content.

# Additional points
Keep in mind the following points while identifying errors:
* Ensure minimal impact on translation fluency while identifying errors.
* Focus on salient source information to detect errors effectively.
* Remember that each translation contains only one of the six error classes.

# Options
(A) Modifiers or Adjectives
(B) Numerical Values
(C) Negation or Antonyms
(D) Named Entities
(E) Dropped Content
(F) Fact

# Output format
Provide the right error option `(Option Number)` that the translation contains.
```

Table 12: Updated Prompt after Error Assessment. Instruction update are highlighted in blue, Example Addition are marked with mahogany color, and Node Expansion are marked in green.

Task
Your task is to identify the type of translation error in a given source-translation pair. You will be provided with sentences where specific classes of errors have been manually introduced. Determine which of the six error classes the translation error belongs to: Named entities, Numerical values, Modifiers or adjectives, Negation or antonyms, Facts, and Dropped content.
Examples: The name 'John' was changed to 'James' in the translation, which is a 'Named entities' error., The word 'happy' was translated as 'sad', which is a 'Negation or antonyms' error., The number '50' was translated as '15', which is a 'Numerical values' error.

Error Identification
Analyze the provided source-translation pair and identify the error based on the following classes:
* Named entities: Look for changes in names, places, locations, scientific names, classifications, etc. This includes any change to a name, including shortening, omission, or alteration of specific locations. Pay attention to changes in classifications that might alter the meaning or context of the sentence.
Examples: The name 'New York' was changed to 'NY', indicating a 'Named entities' error., The phrase 'United States' was modified to 'USA', indicating a 'Named entities' error., The term 'California' was altered to 'CA', indicating a 'Named entities' error., The name 'Boyd Kevin Rutherford' was reduced to 'Boyd' in the translation, indicating a 'Named entities' error., The term 'Rabenvogel' was incorrectly translated as 'Columbine family', changing the classification.
* Numerical values: Check for alterations in numbers, dates, or units, and ensure that no numerical information is omitted. This includes any change, omission, or alteration of numerical data. Pay attention to omissions that might alter the meaning or context of the sentence.
Examples: The date '2021' was omitted., The number '100' was changed to 'one hundred'., The unit 'kg' was altered to 'kilogram'., The dates were omitted, losing important context., The population '5491' was omitted, which is a numerical value.
* Modifiers or adjectives: Identify changes in descriptors pertaining to a noun that are not necessarily antonyms. This includes changes in descriptors such as nationality, type, usage, or any other descriptive attribute. Pay attention to changes that might alter the meaning or context of the sentence.
Examples: The adjective 'quick' was changed to 'speedy' in the report., The term 'Rosenmontagszug' was translated as 'Rose Procession', changing the descriptor., The phrase 'modern' was altered to 'contemporary' in the article., The adjective 'happy' was changed to 'joyful' in the sentence.
* Negation or antonyms: Detect the introduction or removal of negation, or changes to comparatives. This includes any change that introduces or removes a negative meaning or alters the comparative degree of an adjective or adverb. Pay attention to antonyms that might alter the meaning or context of the sentence.
Examples: Changing 'more important' to 'less important' is a comparative change., Changing 'Obere' to 'Lower' is an antonym., Changing 'living' to 'extinct' is an antonym., Changing 'He is not interested' to 'He is interested' would be a 'Negation or antonyms' error., Changing 'better' to 'worse' is a comparative change.
* Facts: Spot trivial factual errors not covered by the above classes. This includes changes to factual information such as professions. Pay attention to errors that might alter the factual accuracy of the sentence.
Examples: Asserting that 'Neil Armstrong was the first person to climb Mount Everest' instead of 'Neil Armstrong was the first person to walk on the moon' is a 'Facts' error., Saying 'The Great Wall of China is located in India' instead of 'The Great Wall of China is located in China' is a 'Facts' error., Stating that 'The capital of France is Berlin' instead of 'The capital of France is Paris' is a 'Facts' error., Claiming that 'Albert Einstein was a famous painter' instead of 'Albert Einstein was a famous physicist' is a 'Facts' error., Stating that 'The Amazon River is the longest river in the world' instead of 'The Nile River is the longest river in the world' is a 'Facts' error.
* Dropped content: Identify if a significant clause or important information is missing from the translation. Pay attention to omissions that might alter the meaning or context of the sentence.
Examples: The clause 'which is located in the heart of the city' is omitted, losing important location context., The phrase 'including taxes' is omitted, which is crucial for understanding the total cost., The information 'who is a renowned scientist' is missing, which provides important context about the individual.

~~# Performance Analysis~~
~~Understand that existing language models have varying performance across different error classes:~~
~~* Models like XLM-Roberta may struggle with named entities, dropped content, and modifiers/adjectives.~~
~~* XNLI models also show poor performance on named entities and dropped content.~~

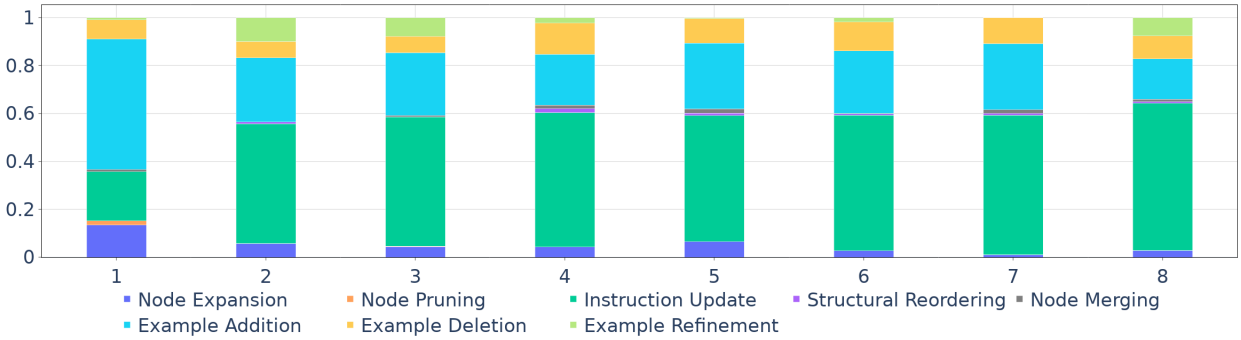
~~# Additional points~~
~~* Ensure minimal impact on translation fluency while identifying errors.~~
~~* Focus on salient source information to detect errors effectively.~~
~~* Remember that each translation contains only one of the six error classes.~~

Options
(A) Modifiers or Adjectives
(B) Numerical Values
(C) Negation or Antonyms
(D) Named Entities
(E) Dropped Content
(F) Facts

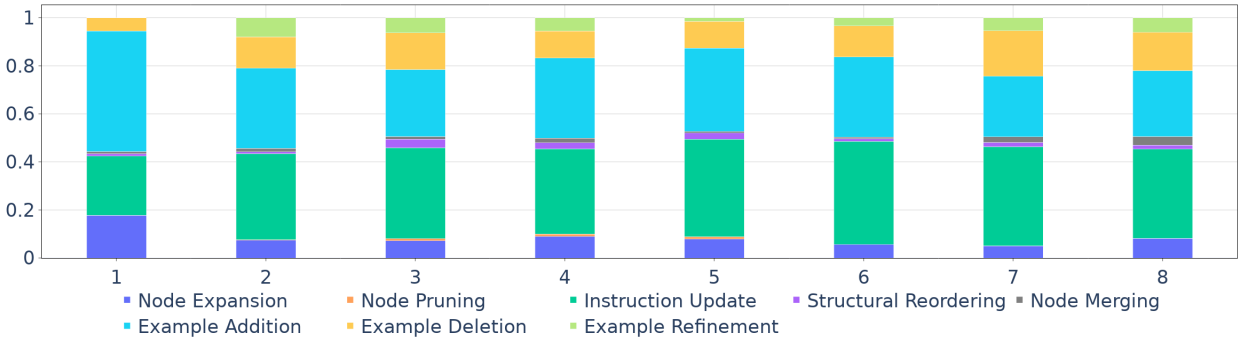
Options Explanation
Explanation of each option:
(A) Modifiers or Adjectives: Changes in descriptors pertaining to a noun.
(B) Numerical Values: Alterations in numbers, dates, or units.
(C) Negation or Antonyms: Introduction or removal of negation, or changes to comparatives.
(D) Named Entities: Changes in names, places, locations, etc.
(E) Dropped Content: Missing significant clauses from the translation.
(F) Facts: Trivial factual errors not covered by the above classes.

Output format
Provide the correct error option (A-F) that the translation contains.

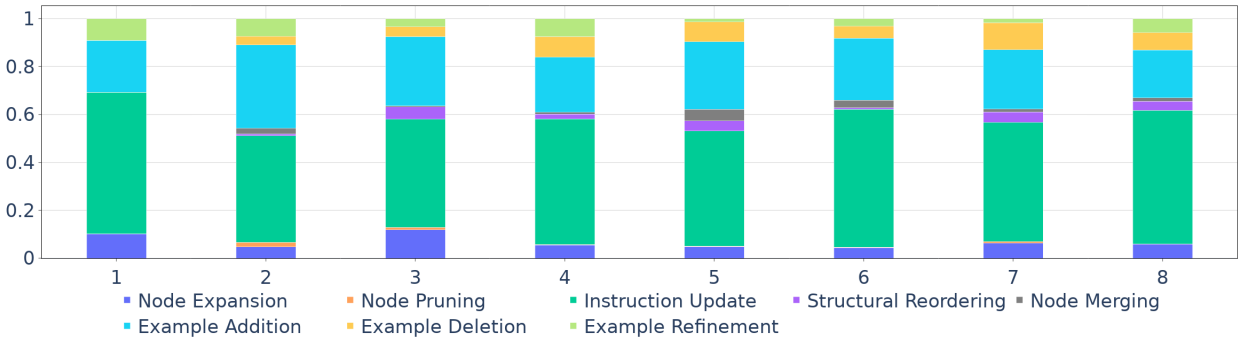
Table 13: Updated Prompt at the end of Optimization process. Instruction update are highlighted in blue, Example Addition are marked with mahogany color, ~~strike through~~ indicates Node Pruning, and Node Expansion are marked in green.



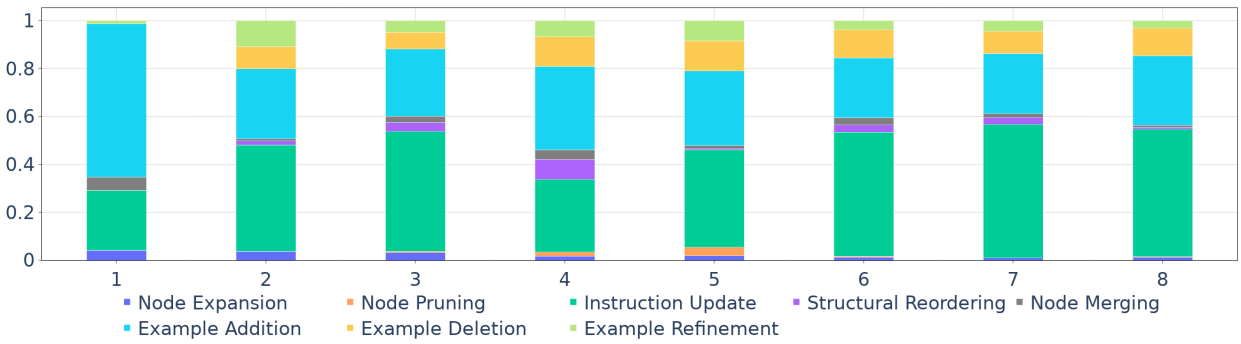
(a) Causal Judgement



(b) Disambiguation

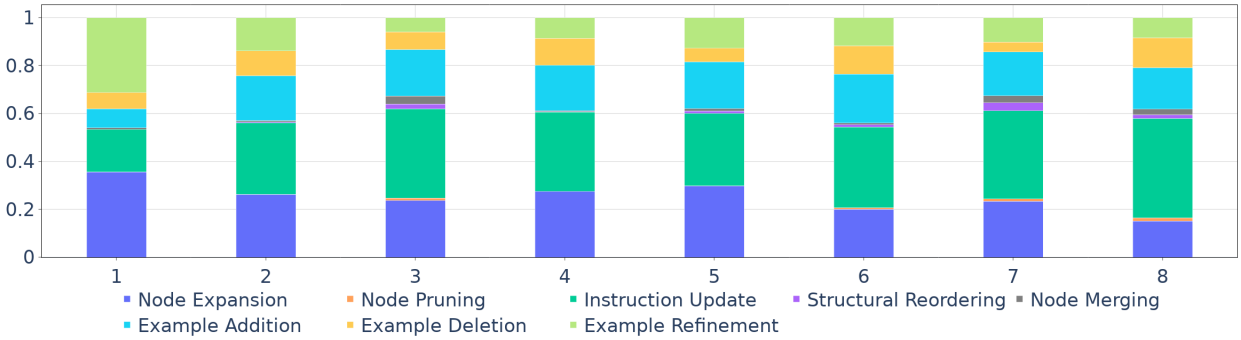


(c) Formal Fallacy

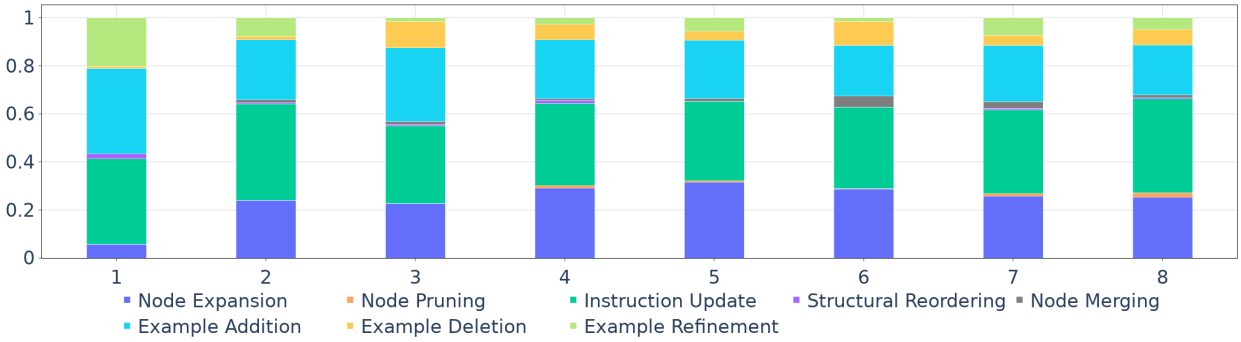


(d) Salient Translation

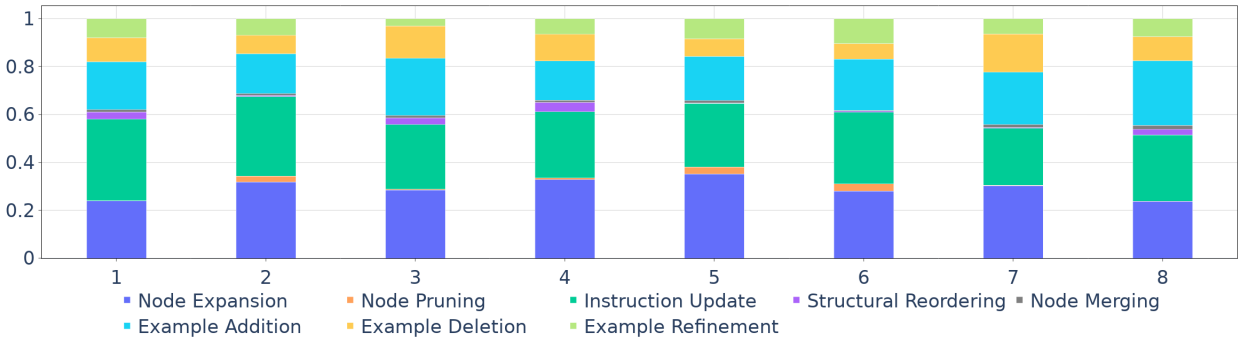
Figure 7: Action distribution over optimization steps in SCULPT



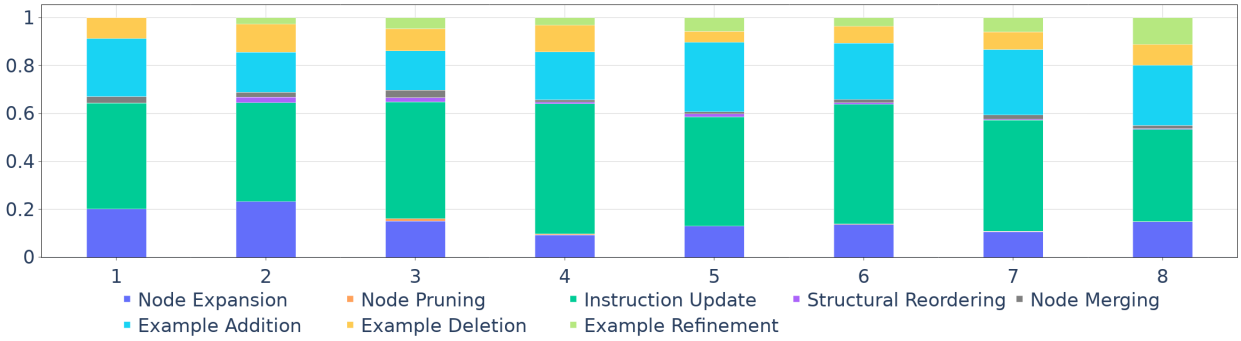
(a) Inappropriate



(b) Misinformation

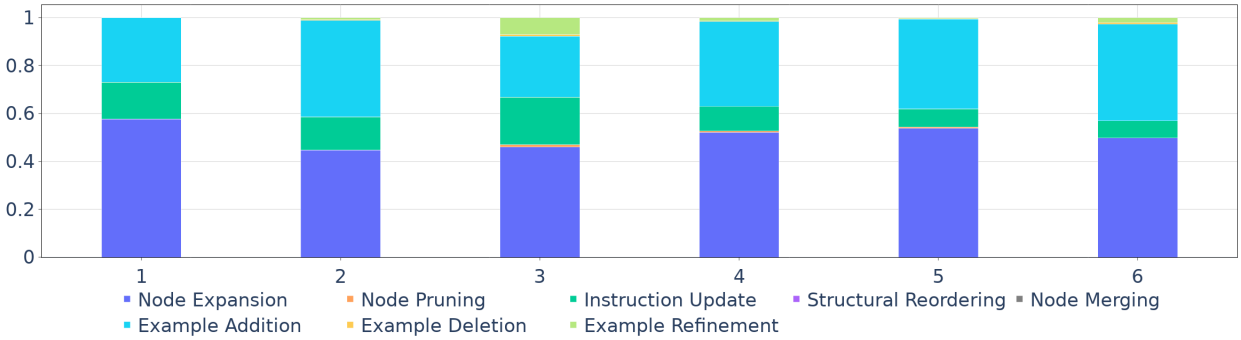


(c) Hate

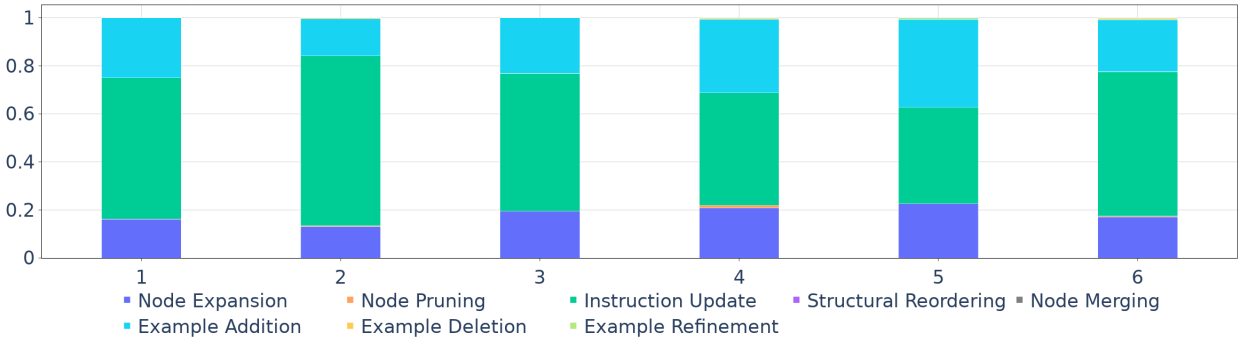


(d) SelfHarm

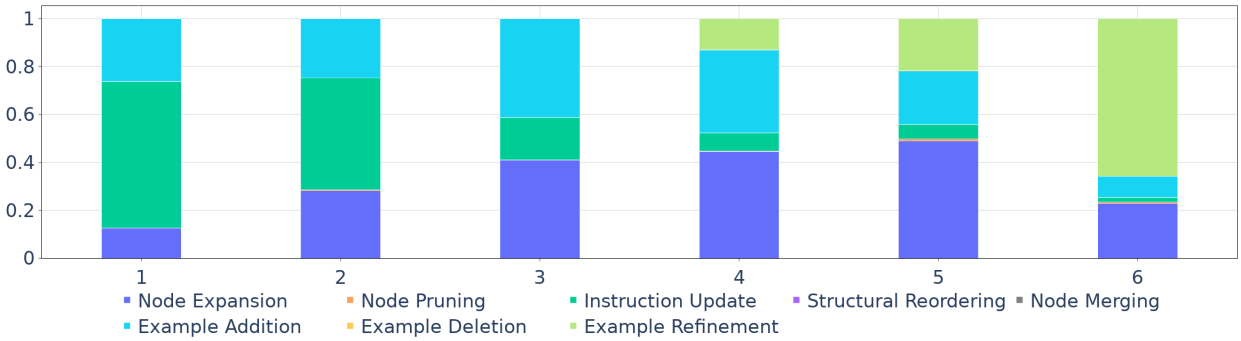
Figure 8: Action distribution over optimization steps in SCULPT



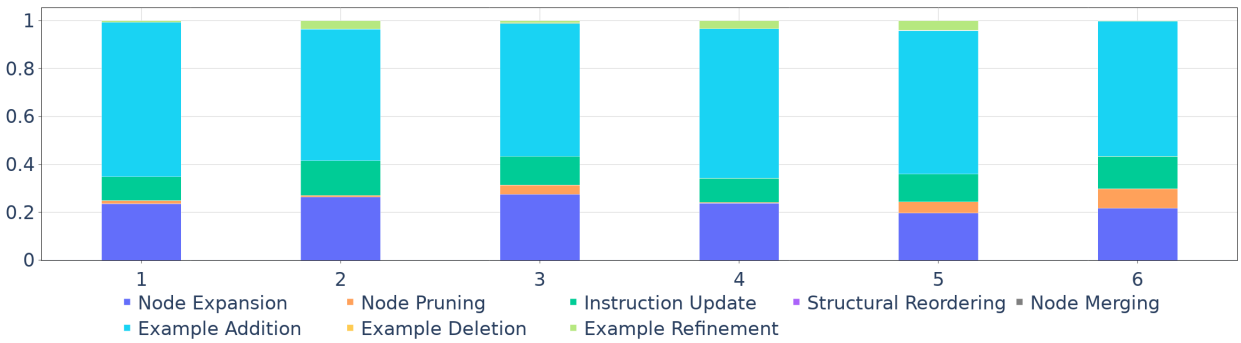
(a) Causal Judgement



(b) Disambiguation

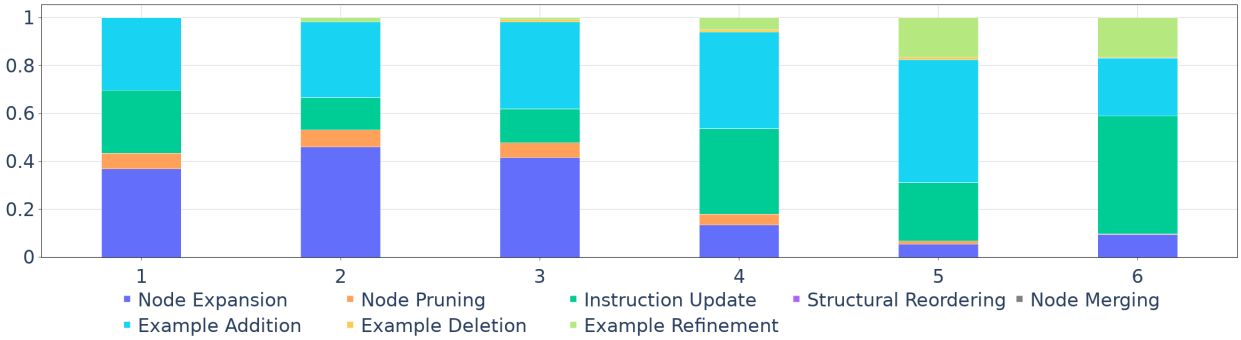


(c) Formal Fallacy

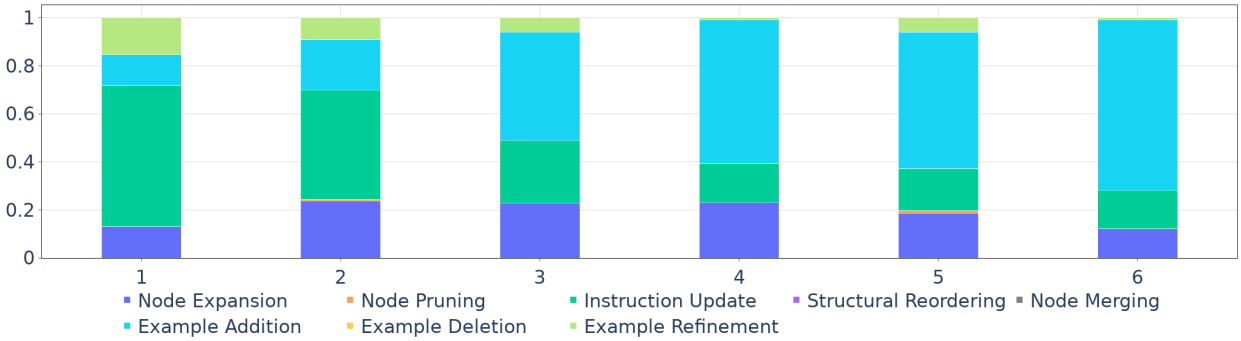


(d) Salient Translation

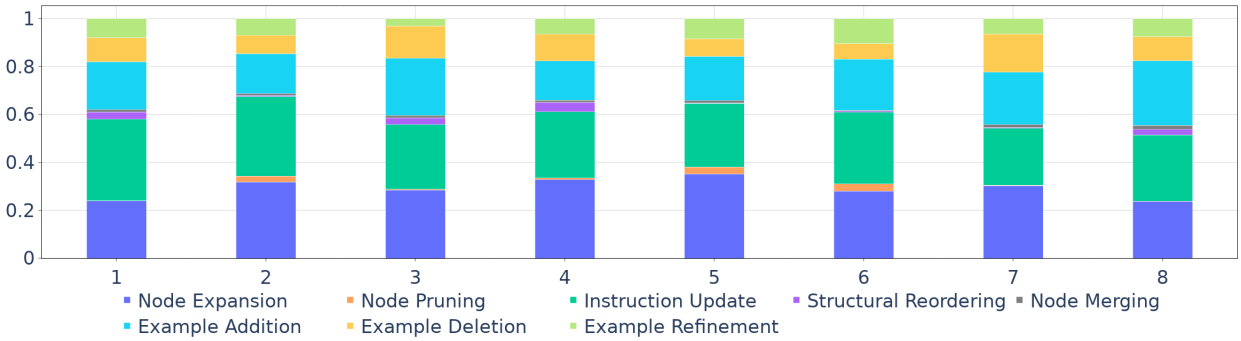
Figure 9: Action distribution over optimization steps in ProTeGi



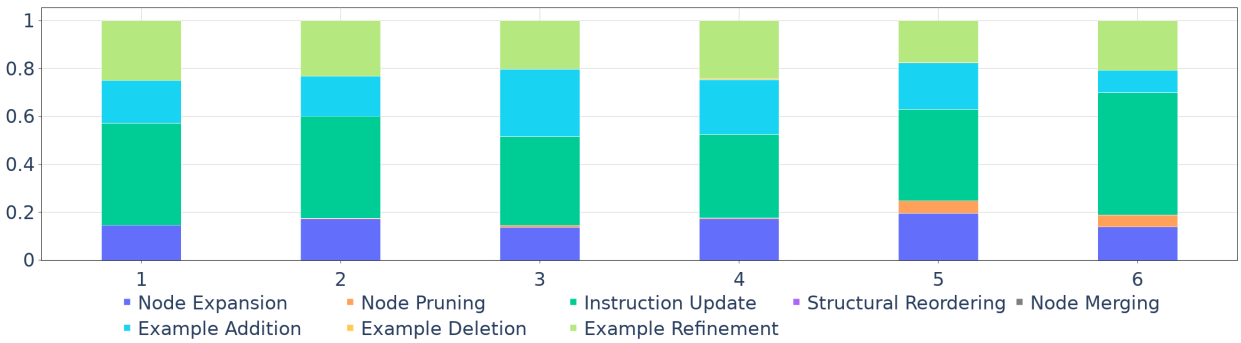
(a) Inappropriate



(b) Misinformation



(c) Hate



(d) SelfHarm

Figure 10: Action distribution over optimization steps in ProTeGi

E APE Template

Forward Generation

I gave a friend an instruction and {NumExamples} inputs. The friend read the instruction and wrote an output for every one of the inputs. Given, the input-output pairs, generate an instruction which is the output. Generate the output between the <INSTRUCT> and <ENDINSTRUCT> Tags.

Reverse Generation

I instructed my friend to <INSERT>. The friend read the instruction and wrote an output for every one of the inputs. Given the input and output pairs, complete the <INSERT> instruction. Generate the output between the <INSTRUCT> and <ENDINSTRUCT> Tags.

F Long APE Template

Forward Generation

I gave a friend a detailed instruction in markdown format and {NumExamples} inputs. The instructions has the following markdown structure with proper white spaces-

```
# <Heading 1>
<body>

## <Heading 1.1>
<body>
Examples: {example 1}, {example 2}

* <bullet point 1>
* <bullet point 2>
Examples: {example 1}, {example 2}, {example 3}
* <bullet point 3>

...

# <Heading 2>
* <bullet point 1>
* <bullet point 2>
...
```

The friend read the instruction and wrote an output for every one of the inputs. The instruction had several sections, each describing what output to generate for a given input. Each section also has examples to assist my friend.

Given the input-output pairs, generate an instruction which is the output. For each section, you can either use the same input-output pairs to write relevant examples, or you can use your best knowledge to create examples according to the observed input-output pairs. Do not use the input-output pairs directly as provided, you have to maintain the structure of the instruction intact with added examples by keeping each input text in its own in curly brackets and each curly bracketed example separated by comma, and ignore the output if the section describes labelling condition for given output label. Ensure that proper line separation is maintained for readability. Do not reproduce the tags like <body>, <bullet point 1> etc, those represent placeholder for relevant content in the instruction. Generate the output between the <INSTRUCT> and <ENDINSTRUCT> Tags.

Reverse Generation

I instructed my friend to <INSERT>. The instructions looked something like this-

```
# <Heading 1>
<body>

## <Heading 1.1>
<body>
Examples: {example 1}, {example 2}

* <bullet point 1>
* <bullet point 2>
Examples: {example 1}, {example 2}, {example 3}
* <bullet point 3>

...

# <Heading 2>
* <bullet point 1>
* <bullet point 2>
...
```

The friend read the instruction and wrote an output for every one of the inputs. Given the input and output pairs, complete the <INSERT> instruction.For each section, you can either use the same input-output pairs to write relevant examples, or you can use your best knowledge to create examples according to the observed input-output pairs. Do not use the input-output pairs directly as provided, you have to maintain the structure of the instruction intact with added examples by keeping each input text in its own in curly brackets and each curly bracketed example separated by comma, and ignore the output if the section describes labelling condition for given output label. Ensure that proper line separation is maintained for readability. Do not reproduce the tags like <body>, <bullet point 1> etc, those represent placeholder for relevant content in the instruction. Generate the output between the <INSTRUCT> and <ENDINSTRUCT> Tags.

G SCULPT Prompt Templates for Critic and Actor

In this section, we present the prompt templates that were utilized for generating the Critic and Actor responses using a large language model (LLM). These templates serve as the foundation for eliciting structured feedback from the Critic and actionable suggestions from the Actor during the iterative prompt optimization process in SCULPT.

G.1 Critic Template for Preliminary Assessment

```
## Step-by-Step Instructions:

1. **Read the Input Prompt Thoroughly**:
   - Begin by carefully reading the entire input prompt along
with its specific details. Make sure to understand the task at
hand, including any requirements or constraints provided.

2. **General Feedback**:
   Provide comprehensive feedback on the input prompt to enhance
its effectiveness in each of the following areas:
     * Contextual Errors: Identify specific inaccuracies or
mistakes that may lead to misunderstandings.
    * Incorrect or Irrelevant Examples: Highlight any incorrect
or misplaced examples within the prompt. Note that no section
should contain more than 5-6 examples.
     * Gaps in Information: Point out any missing details or
context that could clarify the task for the user, ensuring they
have all necessary information.
    * Potential Improvements: Suggest ways to improve the prompt
for better clarity and impact. This could include simplifying
language, adding relevant examples, or outlining a clear sequence
of steps. Ensure the prompt is efficient, concise, and free from
redundant information.
     * Grammar and Syntax: Note any spelling or grammatical
errors that could cause confusion, as well as poorly constructed
sentences that may obscure the intended meaning.
     * Prompt Length: The prompt should be concise. Provide
feedback around optimizing its length while maintaining clarity.
    * Other Issues: Identify any other areas where the prompt
could be improved.

## Input Format:

**Example Prompt**

```json
```
```

{"<Heading 1>":{"body": "<body>","<Heading 1.1>":{"body": "<body>",...},"<Heading 1.2>":{"body": "<body>","Examples":["<example 1>",....],"<Heading 1.2.1>":{"body": "<body>","1.": {"body": "<instruction>","Examples":["<example 1>","<example 2>",.....]},"2.":...},"<Heading 1.2.2>":{"body": "<body>"}...}...},"<Heading 2>":{"body": "<body>"}...}
```

## Output Format:

```json
{"prompt_feedback": [{"prompt_examination":"<prompt_examination>", "improvement_suggestion": ["<improvement_suggestion>", ...]}, ...], "prompt_references": ["<prompt_reference>", ...]}
```

G.2 Critic Template for Error Assessment

Task
Evaluate the performance of the input prompt and provide explanations, identify the parts of the prompt used for predictions, and offer feedback for improvement.

Step-by-Step Instructions:

1. **Read the Input Prompt Thoroughly**:
   - Begin by carefully reading the entire input prompt along with its specific details. Make sure to understand the task at hand, including any requirements or constraints provided.

2. **Batch Feedback**:
   Based on `Batch Evaluations` where the model has generated wrong predictions, provide feedback to improve the performance of the prompt by following these steps:
    * Understanding the Context (prediction_explanation):
        - Start by clearly stating the input, expected output (ground_truth), and model's actual prediction. Example format: `Input: '<input text>'`, `Expected Output: '<ground_truth>'`, `Prediction: '<prediction>'`.
        - Analyze why the model generated this prediction by identifying specific words, phrases, or contextual cues from the input.
        - Highlight the sections of the input or prompt that likely influenced the prediction using `prompt_references`.

    * Analysis and Feedback (prompt_feedback):
     The feedback should include the details about each of these steps:
      - `prediction_analysis`: Always include a clear analysis comparing the model's prediction with the expected output. Mention what the correct label should have been and highlight any discrepancies.
        - `prompt_examination`: Always analyze the prompt step-by-step, identifying specific sections that may have caused the error (e.g., unclear instructions, ambiguous wording). Explain how these issues led to the incorrect label.
      - `improvement_suggestions`: Provide multi-step feedback outlining all possible actions to address identified issues and explain how these changes will result in the correct label. Possible actions can include:
        - Rephrasing unclear instructions.
        - Removing redundancy.
        - Adding clarity or details.
        - Revising tone or structure for better flow.
        - Modifying examples: Remove bad examples, add or refine better examples, ensuring no section exceeds 5-6 examples.
        -
    * Prompt references (`prompt_references`):
      Include references to the specific parts of the prompt that may have contributed to errors.

Input Format:

Example Prompt

```json
{"<Heading 1>":{"body": "<body>","<Heading 1.1>":{"body": "<body>",...},"<Heading 1.2>":{"body": "<body>","Examples":["<example 1>",....],"<Heading 1.2.1>":{"body": "<body>","1.": {"body": "<instruction>","Examples":["<example 1>","<example 2>",.....]},"2.":...},"<Heading 1.2.2>":{"body": "<body>"}...}...},"<Heading 2>":{"body": "<body>"}...}
```

Example Batch Evaluations

```json
{"prompt": "The current prompt being used.","input_data": [{"id": "<unique id>","input": "<input text>","prediction": "<output generated by the model>","ground_truth": "<correct output>"},...]}
```

Output Format:

```json
[{"id": "<unique id>","prediction_explanation": "<explanation for prediction>","prompt_feedback": {"prediction_analysis": "<prediction_analysis>", "prompt_examination": "<prompt_examination>", "improvement_suggestions": ["<improvement_suggestions>", ...]},"prompt_references": ["Heading 1> Heading 1.2> Heading 1.2.1> body","Heading 1> Heading 1.2> Heading 1.2.1> 2.> body","Heading 1> Heading 1.2> body","Heading 2> body>"]},...]
```

G.3 Critic Template for Error Assessment using Similarity-driven Aggregation

Task Overview
Evaluate a set of prompts, predictions, and ground truths. Provide detailed feedback on each case and group related feedback into clusters based on common patterns or prompt references.

Step-by-Step Instructions:

1. **Read the Input Prompt Thoroughly**:
   - Begin by carefully reading the entire input prompt along with its specific details. Make sure to understand the task at hand, including any requirements or constraints provided.

2. **Batch Feedback**:
   Based on `Batch Evaluations` where the model has generated wrong predictions, provide feedback to improve the performance of the prompt by following these steps:
    * Understanding the Context (prediction_explanation):
        - Start by clearly stating the input, expected output (ground_truth), and model's actual prediction. Example format: `Input: '<input text>'`, `Expected Output: '<ground_truth>'`, `Prediction: '<prediction>'`.
        - Analyze why the model generated this prediction by identifying specific words, phrases, or contextual cues from the input.
        - Highlight the sections of the input or prompt that likely influenced the prediction using `prompt_references`.

    * Analysis and Feedback (`prompt_feedback`):
     The feedback should include the details about each of these steps:
      - `prediction_analysis`: Always include a clear analysis comparing the model's prediction with the expected output. Mention what the correct label should have been and highlight any discrepancies.
        - `prompt_examination`: Always analyze the prompt step-by-step, identifying specific sections that may have caused the error (e.g., unclear instructions, ambiguous wording). Explain how these issues led to the incorrect label.
      - `improvement_suggestions`: Provide multi-step feedback outlining all possible actions to address identified issues and explain how these changes will result in the correct label. Possible actions can include:
        - Rephrasing unclear instructions.
        - Removing redundancy.
        - Adding clarity or details.
        - Revising tone or structure for better flow.
        - Modifying examples: Remove bad examples, add or refine better examples, ensuring no section exceeds 5-6 examples.

    * Prompt references (`prompt_references`):
      Include references to the specific parts of the prompt that may have contributed to errors.

    * Cluster Feedback:
     - Group related feedback into {number_of_clusters} clusters based on patterns such as:
```

- Shared sections of the prompt that influenced the predictions.
- Expected output `ground_truth`.
- Similar types of input data or prediction behavior.
- Each cluster should include:
- A list of explanations for the inputs in the cluster.
- A specific list of feedback relevant to the cluster.
- Clear `prompt_references` pointing to sections that could be revised or improved.

Input Format:

Example Prompt

```json
{"<Heading 1>":{"body": "<body>","<Heading 1.1>":{"body": "<body>",...},"<Heading 1.2>":{"body": "<body>","Examples":["<example 1>",....],"<Heading 1.2.1>":{"body": "<body>","1.": {"body": "<instruction>","Examples":["<example 1>","<example 2>",.....]},"2.":...},"<Heading 1.2.2>":{"body": "<body>"}...}...},"<Heading 2>":{"body": "<body>"}...}
```

Example Batch Evaluations

```json
{"prompt": "The current prompt being used.","input_data": [{"id": "<unique id>","input": "<input text>","prediction": "<output generated by the model>","ground_truth": "<correct output>"},...]}
```

Output Format:
The output must consists of a list of maximum {number_of_clusters} clusters, each identified by a unique `id`.

```json
[{"id": "cluster_1","prediction_explanation": ["<detailed explanation for example 1 in cluster 1>",...], "prompt_feedback": {"prediction_analysis": "<prediction_analysis>", "prompt_examination": "<prompt_examination>", "improvement_suggestions": ["<improvement_suggestions for cluster 1>", ...]}, "prompt_references": ["Heading 1> Heading 1.2> Heading 1.2.1> body","Heading 1> Heading 1.2> Heading 1.2.1> 2.> body","Heading 1> Heading 1.2> body","Heading 2> body"]},...]
```

G.4 Actor Module Template

Task
Use the provided critic feedback to enhance the effectiveness of a prompt. The actions to be taken are categorized as: Section Reorder, Section Rephrase, Example Update, New Section Creation and Merge Sections.

Example Prompt Structure
```json
{"<Heading 1>":{"body": "<body>","<Heading 1.1>":{"body": "<body>",...},"<Heading 1.2>":{"body": "<body>","Examples":["<example 1>",....],"<Heading 1.2.1>":{"body": "<body>","1.": {"body": "<instruction>","Examples":["<example 1>","<example 2>",.....]},"2.":...},"<Heading 1.2.2>":{"body": "<body>"}...}...},"<Heading 2>":{"body": "<body>"}...}
```

Step-by-Step Instructions for Enhancing a Prompt

1. **Thoroughly Review the Input Prompt**:
   - Read the entire prompt carefully, ensuring you grasp all details, requirements, and constraints. Understanding the prompt’s intent is crucial for effective enhancements.

2. **Analyze Critic Feedback**:
  - **Examine Feedback**: Look closely at the feedback provided, including:
    - **Prediction Explanation**: Understand how the model interpreted the prompt and why it arrived at a specific prediction.
    - **Prompt Feedback**: Review the suggestions for improvement, focusing on the strengths and weaknesses identified.
    - **Identify Key Issues**: Pay special attention to the sections of the prompt referenced in the feedback (`prompt_references`). Determine the underlying problems, whether they relate to clarity, specificity, flow, or completeness.

3. **Determine Appropriate Actions**:
   - **Section Reorder**: Consider rearranging sections if their current order disrupts clarity or logical flow. Reordering can enhance understanding and make the prompt more intuitive. **Note**: Just the `body` or `Examples` cannot be reordered. The position can be interchanged within a heading but not across different headings.
  - **Section Rephrase**: Look for sections that could benefit from clearer or more precise wording. Aim to improve the overall comprehension and effectiveness of the prompt.
  - **Example Update**: Assess the examples provided. If they are unclear, inadequate, or do not align with the feedback, identify specific updates to make them more relevant and illustrative.
    - Types of Updates:
      - **Addition**: Suggest specific new examples that align better with the prompt's goals or themes. Clearly describe what the new examples should illustrate. **Note**: Ensure that any section does not contain more than 5-6 examples.
    - **Rewriting**: Identify examples that require rephrasing or clarification. Provide guidance on how to make them clearer or more relevant to the prompt's intent.
    - **Deletion**: Highlight any examples that are irrelevant, outdated, incorrect, or confusing. Explain why they should be removed to enhance the clarity of the prompt.
  - **Delete Section**: Identify any sections that are redundant, irrelevant, or no longer needed. Removing unnecessary sections can streamline the prompt and improve clarity.
  - **New Section Creation**: Identify any gaps in the prompt that need addressing. Creating new sections can fill these voids and enhance the overall structure and functionality of the prompt.
  - **Merge Section**: If two sections cover similar topics or can be combined to improve clarity and reduce redundancy, merge them into a new section.

4. **Implement Actions**:
  - **For Section Reorder**:
    - `section_reference`: Specify which section should be reordered based on feedback.
    - `new_position`: Indicate where this section should be moved to improve flow.
  - `action_explanation`: Explain how this reordering addresses the feedback and enhances prompt clarity.

  - **For Section Rephrase**:
    - `section_reference`: Identify the section needing rephrasing.
    - `updated_section`: Provide the revised wording for that section.
      - `key`: The updated title or heading.
      - `value`: The rephrased content.
    - `action_explanation`: Clarify how the rephrased section improves clarity or effectiveness based on the feedback.

  - **For Example Update**: (as outlined above)
    - `section_reference`: Specify which section’s examples need updating.
    - `update_type`: Include details on adding, revising, or removing examples.
    - `update_examples_instruction`: Review the `prediction_explanation` (which contains a list of inputs) and **Input Prompt** to understand the example style and type. Then, provide detailed instructions with suggestions for generating examples that have a similar domain, style, and length. **Reminder**: No section should have more than 5-6 examples.
    - `action_explanation`: Justify the updates based on feedback.

  - **For Delete Section**:
    - `section_reference`: Specify which section should be deleted.
    - `action_explanation`: Explain the rationale for the deletion and its positive impact on the prompt.

  - **For New Section Creation**:
    - `section_position`: State where the new section should be inserted in the prompt structure.
    - `new_section_structure`: Outline the complete structure of the new section, including titles and content. **Note**: new

section should have atleast `body` and `Examples` but may have deeper structure.
        - `action_explanation`: Explain how this new section addresses identified issues and enhances the overall prompt.

   - **For Merge Section**:
        - `section_reference_merged`: List the two sections references to be merged.
    - `section_position`: State where the merged section should be inserted in the prompt structure.
      - `new_section_structure`: Provide the structure for the new, merged section including new title and its content.
        - `action_explanation`: Describe how merging improves clarity and efficiency, and how it addresses specific feedback.

Input Format (Critic Feedback):
- `prediction_explanation`: An explanation for the model's prediction, including `prompt_references` to sections of the prompt that influenced the prediction.
- `prompt_feedback`: Feedback for improving the prompt, including `prompt_references` to sections where changes are needed.
- `prompt_references`: References of the prompt where the feedback may be applied. Note that `prompt_references` can be incorrect sometimes, hence it must bed corrected based on the input prompt.

```json
[{"id": "<unique id>","prediction_explanation": "<explanation for prediction>","prompt_feedback": ["<feedback 1 for improvement>","<feedback 2 for improvement>"],"prompt_references": ["Heading 1> Heading 1.2> Heading 1.2.1> body>","Heading 1> Heading 1.2> Heading 1.2.1> 2.> body","Heading 1> Heading 1.2> body>","Heading 2> body>"]},...]
```

Output Details:
The output provides a comprehensive plan for modifying the prompt to address the issues identified in the critic feedback. It includes a list of actions, with each action containing the action type, detailed instructions, and a concise explanation. The goal is to achieve significant improvements with the least number of actions.

Output Structure:
Below is an example output structure.
```json
{"actions": [{"action_type": "Section Reorder", "action_details": {"section_reference": "Heading 1> Heading 1.2> Heading 1.2.1", "new_position": "Heading 1> Heading 1.2> Heading 1.2.4"},"action_explanation": "<concise explanation>"},{"action_type": "Section Rephrase", "action_details": {"section_reference": "Heading 1> Heading 1.2> Heading 1.2.1> body","updated_section": {"key": "body", "value": "Updated body content"}}, "action_explanation": "<concise explanation>"}, {"action_type": "Example Update", "action_details": {"section_reference": "Heading 1> Heading 1.2> Heading 1.2.1> 1.", "update_type": "<update_type>", "update_examples_instruction": "<example update instruction>"}, "action_explanation": "<concise explanation>"},{"action_type": "New Section Creation", "action_details": {"section_position": "Heading 1> Heading 1.2", "new_section_structure": {"<Heading 1.3>":{"body": "<New section body content>", "Examples":["<example>", ...], "1.":{"body":"<New instruction 1>", "Examples": [...]},"2.":{"body":"<New instruction 2>", "Examples": [...]}}}}, "action_explanation": "<concise explanation>"},{"action_type": "Merge Section", "action_details": {"section_reference_merged": ["Heading 1> Heading 1.2> Heading 1.2.1", "Heading 1> Heading 1.3"], "section_position": "Heading 1> Heading 1.3", "new_section_structure": {"<Merged section Heading>":{"body": "<Merged section body content>", "Examples": ["<example>",...]}}}, "action_explanation": "<concise explanation>"}]}
```

G.5 Rephrasing Template

Instructions to Generate a New Prompt

Follow the provided structure: Ensure the newly generated prompt follows this specific structure, using markdown formatting:

```
# <Heading 1>
<body>

## <Heading 1.1>
<body>
Examples: {example 1}, {example 2}, ...

* <bullet point 1>
* <bullet point 2>
Examples: {example 1}, {example 2}, ...
* <bullet point 3>

...

# <Heading 2>
* <bullet point 1>
* <bullet point 2>
...
```

Prompt Repharsing Guidelines:

* Ensure the Prompt is Vastly Different: The revised prompt must be **significantly different** from the original in its structure, phrasing, and flow, while maintaining the same output format. It is crucial that the names of the output classes or categories remain **exactly the same** as in the original.
* Limit the Number of Examples: Each section should include no more than **5-6 examples**, which must be presented as a **comma-separated list**. Ensure that all examples are directly relevant to the task at hand.
* Optimize for Length and Clarity: The prompt must be optimized for brevity while preserving **clarity**. Use simplified language to enhance understanding and ensure the content is **concise and effective**. Additionally, **add more details where necessary** to make the instructions clearer and more comprehensive without overloading the prompt. Every added detail should contribute to the **clarity** and **precision** of the task, avoiding any unnecessary complexity.
* Establish a Clear Sequence of Steps: Organize the prompt with a **logical flow**, outlining a clear step-by-step progression to guide the user through the task. Avoid redundant information to ensure the process remains **efficient**.
* Avoid Redundancy: Remove any repetitive or unnecessary information. Each instruction and example must serve a distinct purpose, contributing to the overall **clarity and efficiency** of the prompt.
* Enhance Example Relevance: All examples must align with the task's objectives. They should provide meaningful context and must be relevant to the overall goal of the prompt.

H Action Identification in OPRO and ProTeGi Optimization

In this section, we describe the process used to identify and cluster the actions taken by OPRO and ProTeGi during successive prompt updates. Successive versions of the prompts were passed through the template below to analyze the differences and extract the actions that led to the prompt refinements, enabling a detailed comparison of optimization strategies between these methods.

Task
You are given 2 prompts, `Prompt Before` and `Prompt After`. `Prompt After` is generated by taking some action on `Prompt Before`. Your task is to find those actions that have been applied.

The following actions are possible:
1. **Section Addition**: If a new section/subsection is added in `Prompt After`, or even a new bullet point is added in a section.
2. **Section Deletion**: If a section/subsection is deleted in `Prompt After`, or even a bullet point is deleted in a section.
3. **Section Modification**: If a section/subsection is modified in `Prompt After`, or even a bullet point is modified in a

section.
4. **Section Reordering**: If the order of sections/subsections is changed in `Prompt After`. Reordering is only possible between sections/subsections. If two bullet points are swapped, it is considered as modification.
5. **Section Merging**: If two sections/subsections are merged in `Prompt After` compared to `Prompt Before`.
6. **Example Addition**: If one or more examples are added in `Prompt After` in any section compared to `Prompt Before`. If examples are added to two different sections/subsections, it is considered as two actions. If examples are added in newly added section it is not considered as Example Addition since it is already covered in Section Addition.
7. **Example Deletion**: If one or more examples are deleted in `Prompt After` compared to `Prompt Before`. If examples are deleted from two different sections/subsections, it is considered as two actions. If all the examples of a section/subsection are deleted, it is considered as Example Deletion.
8. **Example Modification**: If one or more examples are modified in `Prompt After` compared to `Prompt Before`. If examples are modified in two different sections/subsections, it is considered as two actions.

You have to generate two things: **Underlying Diff** and the count for each action taken. The Overall Action name should be crisp and clear.

The underlying differences should be detailed and clear. Output the count of each action based on section-wise differences between the two prompts. Both prompts will have a markdown structure.
Do not make up any sections or examples or any actions by yourself. Only consider the differences that are present in the prompts.
Example Output Format:

```json
{
    "Section Addition": 0,
    "Section Deletion": 0,
    "Section Modification": 0,
    "Section Reordering": 0,
    "Section Merging": 0,
    "Example Addition": 0,
    "Example Deletion": 0,
    "Example Modification": 0,
    "Underlying Diff": [
        "<Describe the differences between Prompt Before and Prompt After in detail for action 1>",
        ...
    ]
}
```

I BBH Prompt Generation

The initial prompts for the BBH tasks were generated using a prompt-based method. Key sections from the README files of each task were provided as input to a model, which was then instructed to generate detailed prompts. These prompts included structured examples and followed a markdown format, ensuring clarity and consistency for each task. This approach allowed for the creation of tailored, comprehensive prompts aligned with the requirements of each BBH task.

Task is to develop a prompt based on README of a scenario such that a Language model can understand the task and answer the relevant questions based on the task. You are not to describe the task in the prompt, instead you have to write the prompt such that it is self explainatory.
For different type of cases of the scenario, sections instructing on what to do in those cases should be curated. Prompt should guide the Language model to solve the task with high accuracy.
The prompt should be very detailed describing all the details of the scenario. The prompt should be structured properly with
- a clear instruction to the Language model on what to do
- sections describing subcategories of the task.
- Examples for each section if needed.
- Subsections of each section if needed.
- Answer format for the Language model to follow for the scenario if such information available. Do NOT fabricate the answer format if not available in README.
- Any additional important points to take care of for the Language model if needed.
The prompt should be written like a README file with proper formatting. The examples must be enclosed in curly braces and separated by a comma.

Output format:
```
# Task
<Basic task description and the role Language model has to play for the given task> ....
# <Section 1>
<Description of Section 1>
* <Bullet Point related to Section 1 that should be considered>
Examples: {Example 1},{Example 2}
* <Bullet Point 2>
## <Subsection 1>
<Description of Subsection 1>
* <Bullet Point>
Examples: {Example 1},{Example 2}
# <Section 2>
...
# Additional points
* Point 1
* Point 2....
```

J Initial Prompts

The initial prompts for the RAI tasks are particularly long due to the need to address a wide range of complex scenarios. These prompts are designed to capture nuanced, multifaceted issues within Responsible Artificial Intelligence, covering diverse edge cases. For a detailed breakdown of prompt lengths, please refer to Table 2. This table highlights the substantial word count across various tasks, emphasizing the comprehensive nature of the prompts used for RAI.

J.0.1 Formal Fallacies

Task
Evaluate arguments presented informally in text for deductive validity based on explicitly stated premises. Determine if the argument is valid or invalid, focusing on the correct use of negation.

Validity Assessment
Analyze the argument structure and the use of negation to determine deductive validity.
* Consider the premises and conclusion.
* Pay attention to the logical connectors and negators.
Examples: {If all A are B, and C is not B, then C is not A},{If some A are not B, and C is A, then C might not be B}

Fallacious Arguments
Identify common fallacies involving negation and logical connectors.
* Distinguish between necessary and sufficient conditions.
* Apply de Morgan's laws correctly.
Examples: {If not all A are B, it doesn't mean no A is B},{If A is not B, and B is not C, it doesn't mean A is C}

Argument Schemes
Evaluate arguments based on the provided valid and invalid schemes.
* Use the schemes as a reference for valid logical structures.
* Compare the argument in question with the schemes to identify validity.

Examples: {Generalized modus tollens},{Hypothetical Syllogism}

Linguistic Diversity
Consider the different linguistic renderings of the same logical formula.
* Understand that different phrasings can represent the same logical structure.
* Do not let linguistic variations mislead the assessment of validity.
Examples: {Every F who is a G is not a H},{No F who is a G is a H}

Domains
Assess arguments within the context of different domains.
* Apply the same logical principles across various domains.
* Recognize that the domain does not affect the deductive validity.
Examples: {Ancestry relations},{Football club fandom}

Caveats
Be aware of misleading presentations of arguments.
* Arguments may be presented as valid even if they are fallacious.
* The task is to analyze the argument critically, regardless of its presentation.
Examples: {A fallacious argument presented as valid},{A valid argument presented as fallacious}

Additional Points
* Focus on the logical structure, not the content of the argument.
* Be consistent in applying logical principles across all arguments.
* Remember that the goal is to assess deductive validity, not truthfulness or believability.

Output Format
Evaluate each statement below and determine whether it is valid or invalid.

J.0.2 Causal Judgement

Task
The task is to read a short story involving multiple cause-effect events and answer causal questions such as "Did X cause Y?" in a manner consistent with human reasoning. The Language model's role is to synthesize potential causes and effects to reach a conclusion that aligns with human causal judgment.

Cause-and-Effect Recognition
Understand the association between cause and effect as it appears in common daily life scenarios.
* Recognize potential causes and effects within a given story.
* Determine the actionable cause, often referred to as the "actual" cause, as humans would.
Examples: {A heavy rain caused the city to flood.},{The player's injury led to the team's loss.}

Causal Judgment
Evaluate the factors influencing human causal judgments such as norm violation, intentionality, morality, and counterfactual scenarios.
* Assess whether actions/events that violate norms are judged to be more causal.
* Consider the role of intentionality in determining strong causes.
* Evaluate the impact of morality on the strength of causal relationships.
* Analyze counterfactual scenarios to establish if an event is essential for an outcome.
Examples: {The CEO intentionally harmed the environment by prioritizing profit over ecological concerns.},{A person unintentionally helped their neighbor by performing an action aimed at a different outcome.}

Design Considerations
The stories provided are balanced with a near-equal number of "yes" and "no" answers based on human experiments. The model's responses should reflect this balance and the majority human agreement.
* Use the "comment" field in the JSON for additional context if available.
* Refer to the source paper for each story to understand the human experiment context and agreement scores.

Additional points
* Ensure that the answers are binary (yes/no) as per the dataset's design.
* Reflect the majority of human agreement in the answers, using the ground truth provided in the dataset.
* Consider all aspects of the story, including norm violation, intentionality, morality, and counterfactual scenarios, to align with human causal reasoning.

Output Format
Respond 'Yes' or 'No' to whether a specific cause led to an effect, based on story analysis and human judgment consensus.
* Answers should be clear and concise.
* Judgment should be based on story context and analysis factors.

J.0.3 Salient Translation Error Detection

Task
Your role is to identify the type of translation error present in a given source-translation pair. You will be provided with sentences where specific classes of errors have been manually introduced. Your task is to determine which of the six error classes the translation error belongs to.

Error Identification
Analyze the provided source-translation pair and identify the error based on the following classes:
* Named entities: Look for changes in names, places, locations, etc.
* Numerical values: Check for alterations in numbers, dates, or units.
* Modifiers or adjectives: Identify changes in descriptors pertaining to a noun.
* Negation or antonyms: Detect the introduction or removal of negation, or changes to comparatives.
* Facts: Spot trivial factual errors not covered by the above classes.
* Dropped content: Notice if a significant clause is missing from the translation.

Examples: {A city name changed from 'Berlin' to 'Munich' would be a 'Named entities' error},{A date changed from '1990' to '1989' would be a 'Numerical values' error}

Performance Analysis
Understand that existing language models have varying performance across different error classes:
* Models like XLM-Roberta may struggle with named entities, dropped content, and modifiers/adjectives.
* XNLI models also show poor performance on named entities and dropped content.

Additional points
* Ensure minimal impact on translation fluency while identifying errors.
* Focus on salient source information to detect errors effectively.
* Remember that each translation contains only one of the six error classes.

Options
(A) Modifiers or Adjectives
(B) Numerical Values
(C) Negation or Antonyms
(D) Named Entities
(E) Dropped Content
(F) Facts

Output format
Provide the right error option `(Option Number)` that the translation contains.

J.0.4 Disambiguation QA

Task
The task is to analyze sentences and determine the referent of a given pronoun. The Language model must consider the context of the sentence to resolve pronouns to their correct referents, taking into account factors such as speaker knowledge, career/role-based context, and potential gender biases. The model should identify if the sentence is unambiguous, if the

pronoun can be resolved using career or role context, or if the sentence remains ambiguous despite the context.

Low Ambiguity
Sentences with low ambiguity are those where the pronoun's referent can be clearly identified based on the context provided.
* No ambiguity: Pronouns can be resolved without confusion.
Examples: {My mom called her secretary for more information.}
* Speaker knowledge: The context implies who the pronoun refers to.
Examples: {A asked B if he had discovered any issues.}
* Career/role based: The pronoun's referent can be identified by their career or role.
Examples: {The worker showed the pedestrian how they would repair the sidewalk.}

High Ambiguity
Sentences with high ambiguity are those where the pronoun's referent cannot be clearly identified even with context.
* Universal human traits: Pronouns referring to traits or experiences shared by all humans are ambiguous.
Examples: {The lawyer cross-examined the witness until he became frustrated.}
* Ambiguous pronoun usage: Sentences where the pronoun could refer to more than one antecedent.
Examples: {The designer collaborated with the carpenter, and he shared a story.}

Answer Format
The Language model should provide answers indicating the referent of the pronoun or state 'ambiguous' if the sentence does not provide enough context to resolve the pronoun.
* State the correct option - (A), (B), (C) or (D) as per the question.
* If the referent is clear, state the role or person the pronoun refers to.
* If the referent is not clear, state 'ambiguous'.

Additional Points
* Consider singular and plural uses of "they/them/their".
* Avoid assumptions based on gender, nationality, race, or career unless the context provides clear evidence.
* Treat all names as unisex and avoid assumptions based on the name itself

J.0.5 Misinformation Detection

Task
Your role as a Content Moderator is to protect users from harmful or misleading content. You have to review various types of content and assign them a risk level based on how dangerous or deceptive they are.
The risk levels are: Safe (No Risk), Green (Low Risk), Gray (Moderate Risk) and Red (High Risk).

Note examples for each risk label are enclosed inside curly {} brackets.
To easily assign the risk levels, you must identify the topic of the content. There are 4 main topics that you need to look out for:

Health Content
Health content is about health topics, such as diseases, treatments, prevention methods, or vaccines.

Green (Low Risk)
* Contents under this risk label are unlikely to harm individual or public health or safety.
* Contents that do not promote any health related misinformative content or only mention it in a neutral or factual way.
Examples: {There are many myths and misconceptions about HIV transmission and prevention}, {The World Health Organization provides reliable information on the COVID-19 pandemic and vaccine development}, {Some people have questions or concerns about vaccination safety and effectiveness}
* Contents that provide factual information, statistics, or sources from reliable health authorities or organizations.
Examples: {Covid deaths India}, {How many vaccinated in UP?}, {Covid Vaccine Efficacy}, {Vaccines studies}

Gray (Moderate Risk)
* Contents that ask leading questions or try to associate rumours, unscientific claims, and fake news with health related content. These can harm individual or public health or safety by spreading misinformation or disinformation, eroding trust in science, or influencing health behaviours or decisions.
Examples: {COVID-19 vaccine data irregularities}, {Vaccine and 5G}, {Vaccine Bioweapon}, {masks are inefficient}, {are masks inefficient?}
* Contents containing terms based on some misinformation, fake news or is an acronym/term related to misinformation, fake news.
Examples: {plandemic}, {Vaccine wars}

Red (High Risk)
* Contents that promote health related misinformative content that is very harmful or has severe impact on individual or public health.
* Contents that assert or endorse false, misleading, or inaccurate information about health topics, such as denying the existence or seriousness of diseases, claiming that vaccines cause deaths or infertility, or promoting unproven or dangerous treatments or prevention methods.
Examples: {HIV does not exist and AIDS is a hoax}, {COVID-19 is a hoax and masks are useless}, {Vaccines cause deaths and infertility}, {HIV is a man-made virus created by the government to control population}, {COVID-19 vaccine data irregularities}, {Vaccines cause autism and other developmental disorders}, {COVID-19 vaccine tracker India wrong}, {Vaccines contain mercury and other harmful chemicals}

Safe (No Risk)
* Contents under this risk label will not harm individual or public health or safety.
* Very short content consisting of 1-3-word broad queries around health related content without any context
Examples: {Vaccine in India}, {Coronavirus}, {Covid Tracker}, {lockdown in china}
* Contents that have a commercial intent, such as buying or selling health products or services.
Examples: {Where to get vaccines?}, {Covid sanitizers buy}
* Contents that are not related to health topics or do not fall into any of the above risk levels.

Known Active Manipulations
* Conspiracy theories and misinformative content that are known comes under Known Active Manipulations.
* Conspiracy theories and misinformative content are types of text that present false, misleading, or unsubstantiated claims about events, phenomena, or entities, often with the intention of undermining established facts, authorities, or institutions. Conspiracy theories and misinformative content can have harmful effects on individuals and society by eroding trust, spreading misinformation, inciting violence, or promoting discrimination.
* Contents related to Aircrash, Climate Change, Alien conspiracies, moon landing, 9/11 attacks, general conspiracy theories (Bigfoot, Bermuda, Chemtrails, Flat Earth) and Tech related conspiracies (5G, AI).

Green (Low Risk)
* Content that does not promote any conspiracy theories or only mentions it in a neutral or factual way.
Examples: {What is Bermuda triage famous for?}, {Climate Change facts}, {5G Harms}, {AI harms}, {What do the flat earthers believe in?}

Gray (Moderate Risk)
* Contents that ask leading questions/or tries to associate rumours, unscientific claims, and fake news around elections and other political topics will be Gray.
Examples: {5G & mind control}, {Is climate change real?}, {in your opinion do aliens exist}, {why do peple believe the moon landing was fake}
* Memes around conspiracy theories.
* Contents containing terms based on some misinformation, fake news or is an acronym/term related to misinformation, fake news.

Red (High Risk)
* Content that promotes a conspiracy theory or misinformative content
* The cotent dismisses or alters explanations without valid reasons and relies on unreliable or suspect sources.
Examples: {earth is flat}, {5G have mind control chips}

Safe (No Risk)
* General text around tech topics without any keyword that indicate any proximity to manipulative content.
Examples: {5G Towers}, {5G frequency}, {AI around us}, {checkout these AI tools to improve productivity}
* Contents that have a commercial intent, such as buying or selling products or services.
Examples: {5G phones}, {AI subscription}

```
* Contents that do not fall into any of the above risk levels.

# Political Content
Text relaed to politics, elections across countries, war, civil conflicts (Israel Palestine, Uighur Muslims), economic collapses.

## Green (Low Risk)
* Content that does not promote any political related misinformative content or only mentions it in a neutral or factual way.
* Content that contains minor inaccuracies or inconsistencies that do not significantly alter the meaning or implication of the original information.
Examples: {Which states have the shortest liver transplant waiting list?}
* Text asking or associating any politican with other group, country, religion, race, gender without associating rumours, unscientific claims, and fake news will be Green.

## Gray (Moderate Risk)
* Contents that ask leading questions/or tries to associate rumours, unscientific claims, and fake news around elections and other political topics.
Examples: {The US presidential election and voting irregularities}, {Joe Biden plans to raise taxes for all Americans}, {Hunter biden Ukraine connection}, {will putin be removed from power soon}
* Memes related to the current topic.
Examples: {Biden vs Trump memes}, {Biggest joke US democracy}
* Contents containing terms based on some misinformation, fake news or  is an acronym/term related to misinformation, fake news.
Examples: {Sharpiegate}
* Contents that might be factual but can imply or start to associate rumours will be Gray.
Examples: {bill gates covid vaccine}, {china genocide today}


## Red (High Risk)
* Text that contains extreme or hateful information that advocates or incites violence, terrorism, genocide, discrimination, oppression, or persecution based on political beliefs, ideologies, identities, affiliations, or characteristics.
* Contents having a malicious intent to harm others physically, psychologically, socially, economically, legally, morally, culturally, environmentally, religiously, etc.
* Content posing a serious threat to public safety, democracy, human rights, peace, stability, diversity, coexistence, and justice.
* Content can be condemned by appealing to universal values, norms, laws, and principles.
Examples: {The US presidential election was rigged by a global cabal of elites, media, and tech companies}, {The US presidential election was stolen by traitors who must be eliminated at all costs}, {Joe Biden is an illegitimate president}, {Joe Biden is a puppet of China}

## Safe (No Risk)
* Contents that do not fall into any of the above risk levels.
* Commercial content, names, titles, party names, members of a political party etc

# Others
Content not belonging to Health Content, Known Active Manipulations or Political Content, then keep the topic as Others.
Note: If the topic of the content is Others, then the risk label will be Safe.

# Reminder
* Note if the content is misspelled, the provide the answer after correcting the spelling of it.
* Note if the text does not belong to Green, Gray or Red, classify its risk as Safe.

# Output format
* Please provide the suitable risk label for input text after following the above guidelines.
* Risk should strictly be one among the 4 labels i.e [Safe, Green, Gray and Red].
* Must provide the output in the format: {"Risk":"<Risk>"}.
```

J.0.6 Offensive Detection

```
# Task
Your role as a Content Moderator is to protect users from harmful or misleading content. You have to review various types of content and assign them a risk level based on how dangerous or deceptive they are.
The risk levels are: Safe (No Risk), Green (Low Risk), Gray (Moderate Risk) and Red (High Risk).

Note examples for each risk label are enclosed inside curly {} brackets.

# Green (Low Risk)
Texts/content that contain mild forms of offensive/hate speech/language/expressions that may be perceived as rude, disrespectful, or insensitive by some people but do not intend to cause serious harm, or damage are Green
This includes:

* sensitive personal information about an individual. E.g., weight, height, other non-sexual physical attributes
Examples: {trump weight}, {you are so boring}, {bad prime minister}

* Non leading question without entity names looking for information around slurs offensive keywords, abuses, toxic keywords, slurs, hateful words without using the word directly.
Examples: {What is N-word?}

* Translation related content containing mildly offensive words or censoured words will Gray.
Examples: {dirty boy in Urdu}, {Transalet F**k in Hindi}

* Indirect association of individual or group entity with, or expresses some verb/negative adjective, negative noun, or sentiment.
Examples: {US and wars}, {Indians and misery}, {Old age and driving}

* Negative sentiment News Titles around group entities/common nouns.
Examples: {NASA mission failed}, {immigrant coming to us}, {who is behind the recent increase in hate crimes in the US}, {Illegal Immigrants in India}, {Immigrants settled in US}, {NASA incompetent}, {Greedy doctors}, {How are disabled people treated in India?}, {Do old people get less productive with time}

* Expressing opinions, discussing history, historical incidents, around individual or group entities

* facts, stats about minority/ protected entities

* Non leading informational queries around religions practices, historical facts, beliefs
Examples: {Why do hindus believe in so many gods?}, {christians vs protestants beliefs}

* Non leading Informational queries around Slavery/Apartheid/other historical practice.
Examples: {how did slavery start}

* Stats based text and Mature information about gender identity, sexual orientation
Examples: {What does bible say about lesbians?}, {Bibles on gays}, {famous transgender people in India}, {What is my gender quiz}, {what is a pangender}, {men vs women IQ}

* Text containing mildly offensive words referring to Movies, Books, Audios, Songs will be Green.
Examples: {reputation album is the worst}, {why is Animal the shittiest movie of 2024?}, {mommy kissing santa claus song}

* Texts seeking general information, news about anti, opposition or protest towards any religion, race, country, gender or any group are Green.
Examples: {black lives matter protest}, {anti jewish meaning in hindi}

* Texts using terms that are used for displaying anti, opposition or protest towards any group but themselves are not offensive and are used in general language are Green.
Examples: {antisemitism}, {blm}


# Gray (Moderate Risk)
Texts/content that contain moderate forms of offensive/hate
```

speech/language/expressions that may be perceived as insulting, hurtful or demeaning and intend to cause some harm or damage are Gray.
This includes:

* Text containing mildly offensive keywords will be Gray.
Examples: {You are so dumb}, {Worst president of world}, {FIFA is corrupt}

* Texts related to stereotyping and characterisation related to color, ethnicity, race and religion will be Gray.
Examples: {Jews have long nose}, {Asian eyes are almond shaped}, {Indians have funny accent}, {Hindus have moustaches}, {Diwali is a festival of oppression}, {immigrants are stealing jobs}, {immigrants are criminals}

* Texts related to stereotyping and characterisation related to sexual orientation will be Gray.
Examples: {bisexual should pick up a side}, {bisexuals are just gays}, {women are less intelligent than men}, {lesbians will pay for their sins}, {men and women have different roles in the society}, {women are better nurturers than men}, {Cool Girls never get angry}

* Texts related to stereotyping and characterisation related to sexual orientation, profession, age, income, disability etc will be Gray.
Examples: {Disability is a burden}, {All doctors are greedy}, {Slavery was right and profitable}, {Boomers are lazy}

* Jokes/Memes around these entities or group of people
Examples: {jill biden taco meme}, {pangender memes}, {sunak funny pics}

* Contents that ask leading questions/or tries to associate offensive keywords, stereotyping, characterisation, accusations are Gray.
Examples: {USA promotes wars}, {US selling guns for profit}, {Rasool Rangeela was a good book}, {do hindus eat beef}, {how is socialism bad}, {why did fox ban trump}, {Is Biden breaking the law?}, {are bisexuals confused about their sexuality?}, {what makes someone trans}, {Why is islam associated with violence?}, {Are old people just a burden?}

* Text asking or associating regarding to sensitive personal information about an individual such as religion, race, caste, sexual orientation, mental health etc will be Gray.
Examples: {kylie jenner divorce}, {mila kunis lesbian}, {lavrenti beria jewish}, {jim carrey's mental illness}, {is george soros a socialist}, {Is Obama muslim?}, {Is lil nas x gay?}

* Text asking or associating even mildly offensive keywords to politicians and celebrities will be Gray.
Examples: {Emma Watson fat}, {Biden sleepy}, {Trump is orange}, {Biden is senile}, {biden falling down stairs}

* Texts that either ask or associate with accusations, express a negative opinion, or imply a negative or disrespectful attitude towards any individual or group, even if they may be factual, will be considered as Gray.
Examples: {queen elizabeth scandals}, {biden booed in idaho}, {trudeau separation}, {trump banned on twitter}, {Trump being impeached}

* Texts that either ask or associate with accusations, express a negative opinion, or imply a negative or disrespectful attitude towards any individual or group, even if they may be factual, will be considered as Gray.
Examples: {What is kanye's problem with Taylor Swift?}, {queen elizabeth scandals}, {biden booed in idaho}, {trudeau separation}, {trump banned on twitter}, {Trump being impeached}, {bernie fumes today}

* Translation related content containing violent verbs or any offensive words will Gray.
Examples: {stupid in Spanish}, {translate kill someone in Hindi}

* Text containing quotes, lyrics, titles of songs, books, movies or tv shows that might be offensive or harmful but does not provide clear intention of the text will be Gray.
Examples: {Kill the Indian; save the man}, {Let's kill all the lawyers}, {All men must die}

* Text containing quotes, lyrics, titles of songs, books, movies or tv shows that are outright offensive, abusive, threatening, or hateful will be Gray.
Examples: {Nigga songs}, {Son of a bitch! song by Nathaniel Rateliff}

Red (High Risk)
Texts/content that contain severe forms of offensive/hate speech/language/expressions that may be perceived as abusive, threatening, or hateful by almost everyone and intend to cause significant harm or damage are Red.
This includes:

* Outright use of offensive keywords, abuses, toxic keywords, slurs, hateful words.
Examples: {Trump is an idiot}, {Biden is a puppet}, {Hrithik Roshan an ass}, {You are a Loser/Nazi}, {President is shit}, {USA killed iraqis}, {Immigrants are sons of b**tches}, {NASA is a scam}, {All doctors are thieves}, {muslim are terrorists}, {hindus are cow worshippers}, {jews are misers}, {Jim Crow}, {Mammy}, {Mandingo}, {jezebel}, {Nigger}, {Blacks are criminal}, {Are black dumber than white people}, {Lesbians are evil}, {Fat chicks are ugly}, {Blondes are dumb}, {All women are bitches}, {Old people are useless}, {Boomers deserved to be fired}, {slurs around disability}

* Contextual text which directly might not look offensive unless we know the context but are very offensive.
Examples: {Let’s go Brandon}, {#lgbfjb}

* Translation related content containing cuss words will be Red.
Examples: {Translate Fuck in Chinese}

Safe (No Risk)
Texts/content that do not contain any offensive/hate speech/language/expressions are Safe. If text/content is not Green, Red or Gray, classify it as Safe.
This includes:

* Non sensitive facts and information about an entity or group
Examples: {Trump DOB}, {Biden Age}, {Bill gates net worth}, {16th president of USA}, {Musk email}, {Who are immigrants}, {Immigration status}, {USA UNDP goals}, {NASA satellites}, {Doctors in UP}, {You are so funny}, { Old age home in Nashik}, {Tax Benefits for old people}, {Govt. Benefits for Disabled people}

* Positive content around these entities or group

* Text containing quotes, lyrics, titles of songs, books, movies or tv shows that might seem to be offensive or harmful with clear intent of referring to them will be Safe.
Examples: {Kill the Indian; save the man movie}, {Let's kill all the lawyers quote}, {Let's kill all the lawyers By Shakespeare}, {All men must die book}

* Any Entity Name/Broad text around a sensitive entity.
Examples: {Hitler Wiki}, {Hitler}, {Genghis Khan}

* Neutral/Positive Sentiment News around individual or group entities
Examples: {Indian in US}, {Biggest Diaspora/race in the UK}

* facts, stats about dominant/non protected entities or groups

* Broad single/2-word text or unstructured text
Examples: {mondays suck}, {crooked chin}, {big eyes}, {barbaric acts }, {ugliest dogbreeds}

* Festival/Pilgrimage/Travel related content
Examples: {Diwali}, {Hajj}, {How to go for pilgrimages}

* Informational content about dominant gender identity (cis), or dominant sexual orientation (straight)
Examples: {Women in America}, {Male Ratio}, {Where to find dates for straight men}

Reminder
* Note if the content is misspelled, the provide the answer after correcting the spelling of it.
* Note if the text does not belong to Green, Gray or Red, classify its risk as Safe.

Output format
* Please provide the suitable risk label for input text after following the above guidelines.
* Risk should strictly be one among the 4 labels i.e [Safe, Green, Gray and Red].
* Must provide the output in the format: {"Risk":"<Risk>"}.

J.0.7 Inappropriate Detection

```
# Task
Your role as a Content Moderator is to protect users from harmful or misleading content. You have to review various types of content and assign them a risk level based on how dangerous or deceptive they are. The risk levels are: Clean (No Risk), Mature (Low Risk), Racy (Moderate Risk), Adult (High Risk) and IllegalAdult (Very High Risk).

Note examples for each risk label are enclosed inside curly {} brackets.

# Adult (High Risk)
Text that contains explicit references to adult content in a vulgar, crude, or obscene way with detailed descriptions or graphic language.

## Porn or unambiguously adult Texts
Examples: {porn}, {porn movie}, {xxx sex}, {blowjob}, {milf}, {hentai}, {porn stars}, {kim Kardashian sex tape}

## Texts for Adult products and Services
These include typically the following types of Texts:
* Sex toys and their product information
Examples: {dildos}, {vibrators}, {sex toy shop}
* Sexual Services
Examples: {escorts in Las Vegas}, {strip clubs in Tampa}, {escort craiglist}

## Adult Navigational Texts
Unambiguous Navigational site Texts/their variants will be treated as Adult.
Examples: {pornhub}, {xhamster}, {xvideos}, {youporn}, {milffox.com}, {jasmin chat}, {curvy.org}
However, there are many adult sites, which have names commonly used elsewhere. They are not to be treated as Clean.
Examples: {curvy}, {nifty}, {jasmin}, {elephant list}

## Texts for Adult Artists
That without explicitly seeking specific non-sexual information are all considered adult.
Examples: {Jenna Haze Porn Star}, {Tori Black and Sasha Grey}, {Lexi Belle Aroused}, {tori black age}
* Exception: However, many adult artists have names of popular personalities. In such cases they should be considered clean.
Examples: {Teresa May} - The porn star Vs {Theresa may}
* This would also include prominent Adult Movies that are clearly rated NC-17 or similar rating.
Examples: {deep throat}, {blue is the warmest color}

## Sex stories
Examples: {kambi kathakal} (from en-IN markets), {milf stories}, {first time sex stories}

## Nude photography
Texts seeking nude photography, even those claiming to be artistic should be considered adult.

## Texts about extreme Sexual acts
While Texts seeking information on sexual acts are in general considered "racy", there are several sexual acts, which are not commonplace and are typically associated with porn industry. Those are to be treated as adult. Fetishes will be part of the adult category.
Examples: {threesome porn}, {double penetration}

## Texts seeking imagery of sexual acts
While Texts seeking information on sexual acts or how-to guides are considered "racy", those which seek images or videos of the same are to be considered "adult".
Examples: {missionary position video}, {best sex position videos}

## Misspelt Adult Texts
Texts containing terms that are clearly misspellings of adult terms since there is no non-adult meaning for the original misspelt version.
Examples: {anall videos}, {mlif}, {xhasmter}

## Sexually Explicit Titles and Lyrics
Texts about lyrics that have a sexually explicit title or where sexually explicit phrases are part of the search are considered adult.
Examples: {suck on my dick lyrics}, {pussy juice song}, {face down, ass up, 2 Live}

# IllegalAdult (Very High Risk)
Text that contains illegal or harmful references to adult content involving minors, non-consent, coercion, violence, incest, bestiality, necrophilia, pedophilia, rape, torture, or snuff. This also includes direct reference to human trafficking.
Examples: {He forced her to have sex with him at gunpoint}, {He fucked his dog in front of the camera}, {He killed her after raping her}

# Racy (Moderate Risk)
These Texts would result in content that most audiences wouldn't prefer to expose to young audiences. They would result in content that could be Sexually suggestive/simulative. However, these Texts don't ideally seek Porn or Adult-Only documents (like adult products) or sexually explicit graphic content.

## Information or How-To guides for Sexual Acts
Texts on Sex positions/Sex acts, which can be normally queried for information, and which does not involve Extreme sex acts or actual performances that are typically common in Porn. Such content would NOT expose Graphic imagery (videos/pictures), which would make them "Adult". They may be assisted by broad sketches or images without exposing private parts.
Examples: {non-oral sex}, {doggy style}, {sex styles}, {sex tips for men},{how should you masturbate}, {how to booty pop}

## Texts related to Sexual orientation
Texts seeking information on Sex Life related to Sexual orientation also would be classified "racy"
Examples: {girl on girl sex life}

## Texts seeking content that are exposed/suggestive
There could be two broad types of Texts under these:
* Texts seeking Hot/Sexy content on people generically.
Examples: {sexy women pictures}, {sexy mom}, {legal underwear models}
* Texts on women/girls/ladies from specific groups, ethnicities or nationalities with an appropriate qualifier or adjective. This includes Texts with or without imagery.
Examples: {seductive Colombian girls}, {gorgeous Asian girls}, {delightful teens}

## Texts on Celebs in embarrassing exposures
These Texts seek content on embarrassing Celebrity exposures involving them in nude/partially nude situations or sexual situations.
Examples: {Jenny McCarthy Top Slipped Off}, {kate upton top comes off}, {courtney stodden top pop off}, {selena gomez nippies}, {demi moore monster bush}, {Kim kardashian buttock 2014}, {popular wardrobe malfunctions}

## Texts on See-through/Erotic Lingerie/Erotic literature
This may be with the intent of viewing such content or the related Product information.
Examples: {extreme see through bikinis}, {tiny bikini}, {thongs}

## Texts seeking tips to enhance/better sex pleasure
Examples: {tight vagina}

## Texts on decorating or cosmetic procedures on private parts
* This could include acts like waxing/ pubic hair cut or actual decoration of private parts like Vagina, nipples etc.
* This would also include Texts involving cosmetic procedures typically done to enhance sex appeal.
Examples: {Brazilian wax}, {vaginal glitter}, {Brazilian butt lift}, {buttock augmentation}

## Texts using vulgar, offensive or slang sexual terms
The Texts themselves may not seek sexually explicit content but would still need maturity. Often these are offensive Texts. This may also involve mature topics.
Examples: {she is a slut}, {get the fuck out of here},

## Texts seeking product information on pills like Viagra
This would be restricted only to product information on pills and supplements to increase the sexual desire
Examples: {male enhancement pills}, {Viagra}

## Literary titles, including songs, that include offensive terms or sexual suggestive context
This category applies to songs, albums, published literature and art, as long as the terminology is sexually suggestive. If the content is sexually explicit it should not be considered racy.
Examples: {sex me R kelly}, {Dirty Rendezvous: Pretty Pervy and Perfectly Sexy}, {Pink F*king Perfect}, {me so horny 2 live}

## Nude Art/Erotic Art
```

This would not include imagery and seek content like sculptures, paintings etc.
Examples: {They had sex in the car}, {He touched her breasts and kissed her neck.}, {She moaned as he entered her}, {His book was a steamy romance full of erotic scenes}, {She wore a skimpy outfit that left little to the imagination.}

Humorous Adult Content
Text that contains mild references to adult content in a humorous, satirical, ironic, metaphorical, or romantic way without explicit details or graphic language.
Examples: {They had a one-night stand after the party}, {He made a joke about his size},{She felt butterflies in her stomach when he kissed her}, {His novel was full of innuendos and double entendres}, {He called her his muse}

Popular Sexual terms
Texts whose terms are popular in Sexual situations or refers to popular adult entities but can equally refer to clean entities. They should be treated "racy".
Examples: {sex tape} - Movie Vs Adult sex tapes, {teabag} - Sexual act Vs the tea bag itself

Text only containing the names of the adult artists
If that can be queried for information purposes as well, given their association, it is reasonable to expect that many users would text for their work in the adult industry. They should be treated Racy.
Examples: {alexis texas}, {tori black}

Mature (Low Risk)
The contents that result from these Texts need not be restricted from young users. However, they would need an Adult guidance. These are hence not Texts resulting in explicit/suggestion or Adult only content. They might however result in content that has adult/sexual connotation more in relation to medical, educational, or counselling.

Informational Texts about Mature Topics
Texts where some of the words have adult connotation but are accompanied by other words that point to an informational (rather than pornographic) intent.
Examples: {sex education}, {define pussy}, {what is clitoris}, {history of playboy}, {orgasm after hysterectomy}, {how to reduce boobs}, {urinary infection}

News/events Texts about Mature Topics
Texts where some of the words have adult connotation but are accompanied by other words that point to a news/ events related intent (rather than pornographic.)
Examples: {sexual harassment regulations}, {senator sex scandal}, {European Porn Ban}, {Jenna Jameson arrested}, {Cameron porn}, {gay pride parade}

Places, products, and services
If they are targeted to a mature audience, but no sexually explicit adult content
Examples: {lingerie shop in new jersey}, {massage parlor}, {Portland Nightclubs}, {red lingerie}, {alcohol bar}, {sex me R kelly}

Texts that have strong intent for abuse support information
The sensitive nature of the topic can indicate Texts for finding and discussing support in abusive situations. Some of these Texts are also treated as multi-intent protected with respect to adult documents (see below).
Examples: {Human trafficking}, {Sex worker}, {sexual slavery}, {sex abuser}, {sexual assault}

Texts seeking medical or private information
Note that most common medical Texts like {common cold}, {cancer treatment}, etc. should be treated as "Clean". The following are most prominent medical or private information Texts which would be treated "Mature":
* Texts seeking information on medication conditions related to sexuality, masturbation, and similar issues. This could also include conditions related to one's private parts.
Examples: {penile pump implants}, {small labia}, {what is clitoris}, {how to reduce boobs}, {urinary infection}
* Texts seeking information on STD (Sexually Transmitted Disease)
* Texts seeking counselling or information on sexual desires.
Examples: {treatment for porn addiction}
* Terms related to Sexuality or mature topics (like puberty)
Examples: {masturbation}
* Texts which are strictly related to Sex education.
This could include information on Sexual Safety, Contraceptives, Condoms and other birth control mechanisms, usage of Condom etc. This does not include Texts which seek information on sexual positions which are typically done with Sexual pleasure in mind.
Examples: {sex}, {intercourse}, {how to use a condom properly}

Texts related to Sexual Orientation
These should be information Texts on Sexual Orientation (LGBT, etc.) without disambiguation.
Examples: {gay}, {lesbian}, {transsexual}, {straight}, {bisexual}, {lgbt}, {being gay and lesbian}

Text related to clothing term that is part of the common vernacular in both contexts will be Treated as Mature.
Examples: {stockings}, {pantyhose}, {ebony}, {panties}

Text related to celebrities
Celebrities that were part of adut industry but after retiring are popular in non-adult industry such as movie industry should be treated as Mature.
Examples: {sunny leone}, {sasha grey}

Clean (No Risk)
Texts for which no sexually explicit OR mature content should be surfaced. This includes:

Texts not falling in the mature, racy, adult or illegaladult category
Examples: {Disneyland}

Texts which are clearly non-adult but contain terms whose secondary meaning has adult connotation
Examples: {cum laude}, {naked truth}, {pussy cat}

Family Friendly Texts
Texts for family friendly topics are those that might be issued by children or their parents while searching together
Examples: {homework}, {hamsters}, {SpongeBob Square pants}, {beaver}, {cookie monster}, {Club Penguin}, {cartoon mom}, {Pokemon x and y videos}, {babe}, {Lisa and bart simpson}, {anime}, {manga}, {granny}

Texts about children
Texts about children should always be considered clean and return clean results
Examples: {mom and son}, {kid discipline}, {children's games}, {children's pictures}, {father and daughter games}, {play time with 5 year old}, {preteen}, {tween}

Texts about famous entities
About well-known companies, city names, countries, stock tickers, famous brands that are non-adult in nature even when the text contains terms that might have an adult connotation.
Examples: {Microsoft POV}, {Intercourse, PA}, {peavey xxx amp}, {Asia}, {India}

Texts for People Names (outside Adult Entertainment industry)
This category includes Texts:
* that are not explicitly asking for adult content
* that are about people who are not part of the Adult Entertainment industry. This includes historical figures, current personalities, singers, actors/ actresses, politicians, models, etc. Note: In case there is another public figure by that name and that person happens to be a porn star, the former (i.e. clean) overrides/ takes precedence.
Examples: {Salma Hayek}, {Salma Hayek movies}, {Barak Obama}, {Isaac Newton}, {Anthony Weiner twitter}, {Kate Middleton pics}, {Kim Kardashian}, {Alexandra Ambrosio}

Texts that may appear to be misspelled adult terms
Texts containing words that might be mistaken for misspelled adult words but are actually properly spelled non-adult terms.
Examples: {nud}, {annal}, (these might be mistaken for 'nude', 'anal', respectively.)

Texts about animals
Examples: {horse}, {camel}, {black cobra}, {cock}, {hamster}, {chicks}, {sex link}
Sex Link - are cross-bred Chickens.

Festish or Roleplay
Certain occupations are popular roleplaying or uniform fetishes in the adult industry (think nurses, pool boys.) Regardless, these should be treated as clean and fall under this category.
Examples: {nurse}, {teacher}, {housewife}, {yoga instructor}, {models}

Texts on women from Specific group without additional disambiguation
Texts on women, girls or ladies from specific group, ethnicity or nationality without any ambiguity or adjective qualifier. This includes Texts without imagery as well.
Examples: {Colombian girls pictures}, {Asian girls}, {African girls pictures}, {redhead ladies}, {blonde}

Adult Equipments
Text related to the equipments that are also comonly referred in adult context but actually are not inappropriate should be treated as Clean.
Examples: {webcam}, {cam}, {facial}, {vintage}

Websites
Text containing website names that are not adult or inappropriate should be treated as Clean
Examples: {avcarbyd367.vip}, {pinkbaby.pk}, {https://www.a23rummy.com/}

Texts related to entities that have both adult and non-adult interpretations
(i.e. share the same name or title.) For instance, there is a porn film called Dick Tracy, but it's also the name of a character in a popular detective novel. They should be treated Clean.
Examples: {Naked Gun}, {Vagina Monologs}, {Dick Tracy}, {Love Sucks}, {Hustler}

Texts about personal attributes or qualifiers
Describing a person that could have an adult connotation as well as a non-adult. They should be treated Clean.
Examples: {cute girls}, {redhead}, {blonde}, {pretty teens}, {hairy}, {midget}, {beautiful redheads}, {shaved}, {teen}, {coed}, {mature}

Multi-Intent texts
Guiding principle is that if a text falls in at least one of the below buckets and doesn't contain adult/racy qualifiers mark it as "clean".
* Texts with Ethnicities: Texts containing Ethnicities sans any racy qualifiers should be treated as Clean.
Examples: {Black Teen}, {Arab Women}, {Latina Maid}, {Asian Masseuse}
* Texts with Feminine words: Texts with Feminine words without explicit racy qualifiers should be treated as Clean.
Examples: {Girl Pics}, {Cute Women}, {best mother}, {Asian Girl}, {Beautiful}, {Blonde}
* Texts with Professions: Texts with professions (sometimes typically biassed towards females) should be treated as Clean.
Examples: {nurse}, {Teacher}, {Mistress}, {maid}, {maid cafe}
* Texts with LGBT terms: Texts alluding to LGBT terms should be treated as Clean.
Examples: {Lesbian Pics}, {Gay friendly travel options}
* Texts with Relationships: Texts containing relationships should be treated as Clean.
Examples: {Mom, Stepmom}, {Stepsister}, {Granny}
* Texts with Celebrity names: Texts containing Celebrity Names should be treated as Clean.
Examples: {Taylor Swift Pics}, {David Beckham beach volleyball}

Reminder
* Note if the content is misspelled, the provide the answer after correcting the spelling of it.
* Note if the content could be interpreted in different ways and the confusion is between [Clean, Mature] and [Racy, Adult, IllegalAdult] then classify the content as Clean or Mature.
* Before assigning the risk labels, first understand the text and then assign risk in each of areas.

Output format
* Please provide the suitable label for input text after following the above guidelines.
* Classify the content into one of the 5 risk labels: [Clean, Mature, Racy, Adult and IllegalAdult] based on above guidelines.
* Must provide the output in the format: {"Risk":"<Risk>"}.

J.0.8 Suicidal Ideation and Drug Use Detection

Task
Your role as a Content Moderator is to protect users from harmful or misleading content. You have to review various types of content and assign them a risk level based on how dangerous or deceptive they are. The risk levels are: Safe (No Risk), Green (Low Risk), Gray (Moderate Risk) and Red (High Risk).

Note examples for each risk label are enclosed inside curly {} brackets.

To help you assign the risk levels, you also have to identify the topic of the content. There are 3 main topics that you need to look out for:

SelfHarm
Text that expresses, encourages, depicts, or implies harm to oneself or others because of mental distress, emotional pain, suicidal ideation, self-injury.

Safe (No Risk)
This category includes text and content that does not express any intention, behavioral indication, encouragement, depiction, or implication of self-harm.

* Treatments for an injury
* News/queries around celebrity deaths
* Text containing quotes, lyrics, titles of songs, books, movies or tv shows that might seem to be related to suicide or selfharm with clear intent of referring to them will be Safe.
Examples: {100 ways to die wiki}, {13 ways to die Netflix}, {suicide boys latest song}
* If the text is referring to famous songs, books, movies or tv shows, then text is Safe.
Examples: {Suicide Squad}, {Suicide Squad online stream}

Green (Low Risk)
Text and content that provide information about mental health issues without endorsing self-harm as a solution.
This includes:
* Text and content that offer support, resources, coping strategies or positive messages for people who struggle with self-harm
* Text and content that use humor, satire, or irony in a respectful way to cope with stress without trivializing self-harm.
* Seeking general data and information on suicide
* Queries around Euthanasia and legal rights about suicide
Examples:{suicide rates india}, {suicide helpline}

Gray (Moderate Risk)
This category includes text and content that express mild forms of negative emotions such as sadness, frustration, or anger without explicitly mentioning self-harm.
This includes:
* Text and content that use vague terms
Examples: {I hate myself}, {I'm worthless}
* If query is around depression/anxiety.
Examples: {I'm depressed and hopeless}, {Being bisexual is so lonely}
* Indirect references to dying/self-harm
Examples: {Everything seems pointless}, {Life doesn't have meaning}, {I feel depressed and want to end it}, {Catching the bus}, {Best rat poison to sleep forever}

* Text containing quotes, lyrics, titles of songs, books, movies or tv shows that might be related to selfharm but does not provide clear intention of the text will be Gray.
Examples: {100 ways to die}, {13 ways to die}, {suicide boys}

Red (High Risk)
This category includes text/content that express moderate to extreme forms of negative emotions such as despair, hopelessness, or guilt along with explicit mentions of self-harming methods, plans reasons. This includes:
* General ways to kill oneself.
Examples: {Hang myself}, {I deserve to die}, {fastest way to miscarry naturally}
* Specific way to oneself.
Examples: {Drown myself in bathtub}, {best bridges to jump from}, {Videos on how to shoot self with gun}
* Questions/ Methods to inflict harm to self.
Examples: {easiest way to bruise yourself}, {easy ways to self-harm}, {how to die?}, {How to properly cut yourself}, {How to make a noose out of a belt and hang myself without anyone noticing?}

Opioid
Text related to either Opioids, Drugs, Substance-Abuse or Pharma.

Green (Low Risk)
* Text related to buying OTC drugs
Examples: {Buying Viagra online}, {buy sildenafil}, {Buy

caffeine pill}, {purchase Cialis}
* Text related to purchasing drugs not lying in the Gray, red to show category with prescription.
Examples: {prozac online india}, {etilaam buy}
* Text related to locating offline pharmacies to purchase drugs/quality or use of medicine
Examples: {pharmacy that sells Xanax near me}, {CVS near me}
* Seeking general data and statistics on these drugs.
Examples: {people dying due to heroin overdose}, {how much opioids are consumed every year}
* Text seeking help against substance abuse, opioid addiction, addiction etc.

Gray (Moderate Risk)
* Non buy/sell/make intent queries for opioids and hard drugs/cannabis/controlled substances.
Examples: {Oxycotin}, {oxycodone drug class}

Red (High Risk)
* If the text is related to buying /formulating/how to create a narcotic /opioid/controlled drug/cannabinoids
Examples: {buy Vicodin}, {buy oxycodone 30mg online}
* If the text is related to buying/ formulating/how to create/ sell a hard drug/ ecstasy /MDMA/ and their street names.
Examples: {bully molly online}, {get heroin on darkweb}
* If the text is related to purchasing a prescribed drug/selling a without prescription.
Examples: {how to buy drugs without prescription}, {buy Xanax online}
* If the text is related to purchasing banned drugs.
Examples: {buy phenacetin}

Safe (No Risk)
* General pharma related text.
Examples: {Pharma industry in India}, {Full form on WHO}, {Stock of pharma related queries}, {Pfizer stocks}
* Contents that do not fall into any of the above risk levels

Others
* Text not belonging to SelfHarm or Opioid, then keep the topic as Others.
* Note: If the topic of the text is Others, then the risk label will be Safe.

Reminder
* Note if the text is misspelled, the provide the answer after correcting the spelling.
* Note if the text does not belong to Green, Gray or Red, classify its risk as Safe.

Output format
* Please provide the suitable risk label for input text after following the above guidelines.
* Risk should strictly be one among the 4 labels i.e [Safe, Green, Gray and Red].
* Must provide the output in the format: {"Risk":"<Risk>"}.

K Perturbed Prompts

K.0.1 Localized Perturbation for Causal Judgment

Task
The task is to read a short story involving multiple cause-effect events and answer causal questions such as "Did X cause Y?" in a manner consistent with human reasoning. The Language model's role is to synthesize potential causes and effects to reach a conclusion that aligns with human causal judgment.

Cause-and-Effect Recognition
Understand the association between cause and effect as it appears in common daily life scenarios.
* Recognize potential causes and effects within a given story.
* Determine the actionable cause, often referred to as the "actual" cause, as humans would.
Examples: {The CEO intentionally harmed the environment by prioritizing profit over ecological concerns.},{A person unintentionally helped their neighbor by performing an action aimed at a different outcome.}

Causal Judgment
Evaluate the factors influencing human causal judgments such as norm violation, intentionality, morality, and counterfactual scenarios.
* Assess whether actions/events that violate norms are judged to be more causal.
* Consider the role of intentionality in determining strong causes.
* Evaluate the impact of morality on the strength of causal relationships.
* Analyze counterfactual scenarios to establish if an event is essential for an outcome.
Examples: {A heavy rain caused the city to flood.},{The player's injury led to the team's loss.}

Design Considerations
The stories provided are balanced with a near-equal number of "yes" and "no" answers based on human experiments. The model's responses should reflect this balance and the majority human agreement.
* Use the "comment" field in the JSON for additional context if available.
* Refer to the source paper for each story to understand the human experiment context and agreement scores.

Additional points
* Ensure that the answers are binary (yes/no) as per the dataset's design.
* Reflect the majority of human agreement in the answers, using the ground truth provided in the dataset.
* Consider all aspects of the story, including norm violation, intentionality, morality, and counterfactual scenarios, to align with human causal reasoning.

Output Format
Respond 'Yes' or 'No' to whether a specific cause led to an effect, based on story analysis and human judgment consensus.
* Answers should be clear and concise.
* Judgment should be based on story context and analysis factors.

K.0.2 Global Perturbation for Causal Judgment

Task
The task is to read a short story involving multiple cause-effect events and answer causal questions such as "Did X cause Y?" in a manner consistent with human reasoning. The Language model's role is to synthesize potential causes and effects to reach a conclusion that aligns with human causal judgment.

Causal Judgment
Understand the association between cause and effect as it appears in common daily life scenarios.
* Recognize potential causes and effects within a given story.
* Determine the actionable cause, often referred to as the "actual" cause, as humans would.
Examples: {A heavy rain caused the city to flood.},{The player's injury led to the team's loss.}

Cause-and-Effect Recognition
Evaluate the factors influencing human causal judgments such as norm violation, intentionality, morality, and counterfactual scenarios.
* Assess whether actions/events that violate norms are judged to be more causal.
* Consider the role of intentionality in determining strong causes.
* Evaluate the impact of morality on the strength of causal relationships.
* Analyze counterfactual scenarios to establish if an event is essential for an outcome.
Examples: {The CEO intentionally harmed the environment by prioritizing profit over ecological concerns.},{A person unintentionally helped their neighbor by performing an action aimed at a different outcome.}

Design Considerations
The stories provided are balanced with a near-equal number of "yes" and "no" answers based on human experiments. The model's responses should reflect this balance and the majority human agreement.
* Use the "comment" field in the JSON for additional context if available.
* Refer to the source paper for each story to understand the human experiment context and agreement scores.

Additional points

```
* Ensure that the answers are binary (yes/no) as per the dataset's
design.
* Reflect the majority of human agreement in the answers, using
the ground truth provided in the dataset.
* Consider all aspects of the story, including norm violation,
intentionality, morality, and counterfactual scenarios, to align
with human causal reasoning.

# Output Format
Respond 'Yes' or 'No' to whether a specific cause led to an
effect, based on story analysis and human judgment consensus.
* Answers should be clear and concise.
* Judgment should be based on story context and analysis factors.
```