StyleRec: A Benchmark Dataset for Prompt Recovery in Writing Style Transformation

Shenyang Liu, Yang Gao, Shaoyan Zhai, Liqiang Wang
Department of Computer Science, University of Central Florida
Email: shenyang.liu@ucf.edu, yang.gao@ucf.edu, shaoyan.zhai@ucf.edu, liqiang.wang@ucf.edu

2024 IEEE International Conference on Big Data (BigData) | 979-8-3503-6248-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/BigData62323.2024.10825143

***Abstract*—Prompt Recovery, reconstructing prompts from the outputs of large language models (LLMs), has grown in importance as LLMs become ubiquitous. Most users access LLMs through APIs without internal model weights, relying only on outputs and logits, which complicates recovery. This paper explores a unique prompt recovery task focused on reconstructing prompts for style transfer and rephrasing, rather than typical question-answering. We introduce a dataset created with LLM assistance, ensuring quality through multiple techniques, and test methods like zero-shot, few-shot, jailbreak, chain-of-thought, fine-tuning, and a novel canonical-prompt fallback for poor-performing cases. Our results show that one-shot and fine-tuning yield the best outcomes, but highlight flaws in traditional sentence similarity metrics for evaluating prompt recovery. Contributions include (1) a benchmark dataset, (2) comprehensive experiments on prompt recovery strategies, and (3) identification of limitations in current evaluation metrics, all of which advance general prompt recovery research, where the structure of the input prompt is unrestricted.**

***Index Terms*—Prompt Recovery, Language Model Inversion, LLM, Adversarial Attack**

I. Introduction

Large Language Models (LLMs) have become essential to various applications due to their ability to generate high-quality outputs based on user prompts. However, there are instances where we only have access to the generated output but need to identify the corresponding prompt to get that specific output. This task, known as "Prompt Recovery," was introduced by [1] in the context of closed-source LLMs. Subsequent studies have addressed this challenge as a form of attack, such as prompt leakage or jailbreak attempts [2] [3], highlighting the security implications of recovering prompts to defend against malicious uses of LLMs. Successful prompt recovery is crucial for mitigating risks associated with harmful prompt generation [4], determining user liability [5], and verifying potential copyright violations [6].

The fundamental difficulty of prompt recovery lies in the fact that exact inversion of outputs to prompts typically requires additional information, like the full probability distribution, which is only available for some LLMs [1]. For models that are accessible only through inference APIs, the information is restricted. Additionally, in scenarios where outputs are derived from documents without access to the original prompt or supplementary data, the challenge of prompt recovery becomes even more evident.

While most existing work in this area focused on question-answering datasets [1] [7] [8], our research explores a specialized scenario in which prompts are used to transform writing styles or rephrase sentences. The task involves recovering the transformation prompt from the original sentence and its corresponding output. Different from [9], our work focuses on providing an open-source dataset along with a detailed methodology for its construction and testing method within a single model for this task.

In this paper, we introduce a benchmark dataset, named **StyleRec**[1], which ensures quality and diversity through rigorous construction techniques. We detail the dataset's creation process to facilitate further research. Additionally, we evaluate five different methods to determine the most effective approach for prompt recovery in this specialized context. Our contributions are as follows. (1) We present the first benchmark dataset with detailed construction guidelines, enabling researchers to generate additional data. (2) Our experimental results demonstrate the effectiveness of specific methods, offering guidance for future research in this domain. (3) We identify flaws in commonly used sentence similarity metrics when applied to the prompt recovery task. Additionally, we highlight the unique challenges of prompt recovery in different scenarios, underscoring the complexity of the general prompt recovery task where the format of the prompt is unrestricted.

II. Related Work

A. Language Model Jailbreaking

Language Model Jailbreaking refers to techniques used to bypass or undermine the safety and ethical guidelines embedded within LLMs. [3] employs a Self-Adversarial Attack and demonstrates that paraphrasing a system prompt can effectively bypass a target model's safeguards. [10] conducts an empirical study that categorizes jailbreak methods into three primary strategies: Pretending, Attention Shifting, and Privilege Escalation. The work in [11] identifies two primary causes for LLM safeguard failures: competing objectives and generalization mismatch. These factors have inspired the development of various jailbreak techniques [12] [13].

However, most evaluations of jailbreak methods have not thoroughly considered defending LLMs. The study by [14] addresses this gap by assessing the effectiveness of both jailbreak attacks and defense techniques. The findings across these papers underscore a critical point: no single defense can

[1]For the dataset, please refer to the following GitHub repository: https://github.com/promptrecovery501/StyleRec.

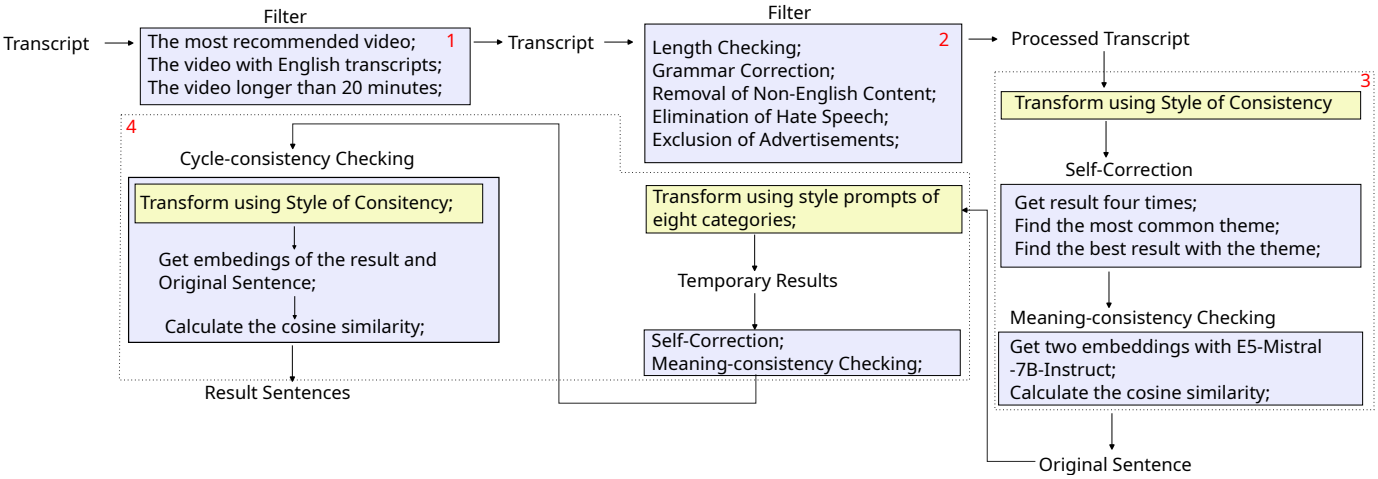

Fig. 1: The Workflow of Data Generation

block all attacks, and no attack can penetrate all defenses. Nevertheless, the ongoing battle between attackers and defenders drives the continuous improvement of LLMs.

B. Model Stealing

[15] first formalizes the Model Stealing problem that is to steal the LLM's weights through interaction with the LLM itself. The approach has been used in different tasks ,such as membership inference [16], federate learning [17] and machine translation [18] and different areas, such as healthcare [19], biology [20] and cryptography [21]. Recent studies [22] [1] suggest that reconstructing model weights may replicate models capability of imitating surface syntax, but have difficulty to restore their underlying decision-making mechanisms.

Since many model stealing methods are developed, protection methods are also improved by many work. [23] [24] focused on the defense of normal deep neurual network. [25] discussed strategies of protection for chatbots.

Beyond the protection methods that make the model stronger, another method to avoid issues is to detect model stealing [26] [27] before submitting the inputs into the model.

C. Prompt Recovery

The term "Prompt Recovery" has been recently introduced in studies such as [8], [7], and [9]. However, the underlying concept has been explored for many years under the name "Model Inversion" in the computer vision domain [28] [29]. [1] applied model inversion techniques to LLMs, highlighting the crucial role of logits (e.g., probability distributions) in successful prompt recovery.

[7] focused on creative writing, such as poetry, and employed a method termed "sample inverter" by [1], which does not rely on logits. Building on this, [8] delved deeper into the relationship between output probability-based uncertainty and the effectiveness of prompt recovery methods. They developed a novel approach that leverages this uncertainty through a chain-of-thought methodology.

Simultaneously with our work, [9] developed a similar idea, focusing on a specialized scenario where the goal is to predict the prompt that alters the writing style or rephrases an original sentence, given the original sentence and its corresponding output. Previous studies [1], [7] and [8] have primarily concentrated on using the LLM's output to predict the original prompt or question.

III. Data Generation with LLM

In this section, we describe the process used to generate our dataset (see Figure 1). First, we selected YouTube videos covering various topics relevant to daily life, extracted the transcripts from these videos (Section III-A), and applied multiple filters to ensure data quality (Section III-B). Using the preprocessed transcripts, we crafted prompts to transform the style of these transcripts (Section III-C). The resulting data was further refined using validation methods based on cosine similarity (Section III-D). The sample data is shown in Fig. 2.

A. Data Preparation

We utilize YouTube video transcripts as the primary source for generating our dataset. The platform provides both automatically generated and manually reviewed transcripts, making it an ideal and comprehensive data source for this study. The inclusion of such a wide variety of content ensures that the dataset is rich and multifaceted, enhancing the generalizability and broad applicability of the research to various real-world use cases.

To ensure that our dataset captures a wide spectrum of everyday scenarios, we selectively extract transcripts from YouTube videos categorized into diverse topics, including: travel, education, entertainment, environment, fashion, finance, food, health, history, law, news, real estate, family, religion, science, culture, sports, and technology. This deliberate categorization ensures that the dataset reflects a range of human experiences, perspectives, and conversational styles. By incorporating content from various categories, we aim to guarantee that the dataset covers both formal and informal language,

technical and non-technical topics, and different levels of complexity in conversation.

To identify suitable videos for each category, we apply the following criteria as filters for selection: (1) the video must be the most recommended within its specific category, ensuring that the content is both relevant and engaging, (2) the video must have English transcripts, ensuring language consistency throughout the dataset, and (3) the video must be longer than 20 minutes, providing enough content to generate substantial transcripts for analysis. These filters ensure that the dataset is not only diverse but also provides enough contextual depth and richness for effective model training and testing.

B. Data Preprocessing

After the initial selection of videos, we preprocess the transcripts to prepare them for use in the study. Fig. 1 illustrates the filtering criteria used in this preprocessing phase. First, we remove any special characters and extraneous information, such as the speaker's name that often precedes dialogue in transcripts. This step is crucial to eliminate non-conversational elements that may interfere with the language modeling process. The transcripts then undergo a series of five additional filtering steps: length checking, grammar correction, non-English content removal, hate speech elimination and advertisement exclusion. Fig. 3 shows the specific prompts and rules we used to implement these filtering steps. Once these preprocessing steps are completed, we receive a set of clean, high-quality transcripts that are ready for use in our study.

Finally, these cleansed transcripts are input into the LLM using a variety of style transformation prompts. The LLM generates outputs based on the input transcripts, applying different stylistic changes.

C. Methods for Generation

With the preprocessed transcripts, we generated our dataset by applying style transformation prompts and obtaining the outputs from the LLM. Each instance in the dataset follows the format: original sentence, result sentence, style prompt. The prompts we employed include eight categories: tone, family roles, occupation, celebrity, historical periods, passive voice, diary style, and proverbs. All 33 styles are shown in Table I. Given the varied sources of the transcripts, the styles were initially inconsistent. The prompt template is shown in Fig. 5 We standardized the style of the transcripts to a uniform "style for consistency" (Fig. 4), which was used as the original sentence in our dataset. To ensure the stability of the results, multiple outputs were generated for each style prompt. We then applied self-correction [30] to select the best output, the prompt is shown in Fig. 7. For further study, we also collected logits for next-token probability and Length-normalized Predictive Entropy (LN-PE) [31] for use in few-shot sample selection.

D. Data Validation

To validate the generated data, we measured the cosine similarity between the original sentence and the output after style transformation, as a metric referred to as meaning consistency. To differentiate the current generated data with the final results, we call it "temporary results". As described in Sec III-C, we first transformed the transcripts to the "style for consistency" as the original sentence before generating the result sentence using the style transformation prompt. Subsequently, the result sentence was converted back to the "style for consistency" to obtain the predicted original sentence. We then measured the cosine similarity between the original sentence and the predicted original sentence, a process known as cycle consistency [32]. Finally, we applied thresholds for both meaning consistency and cycle consistency to filter out inconsistent results.

original_sentence:
The notion that wine has health benefits is not universally accepted.

result_sentence
Wine's health benefits are a topic of debate.

style_prompt
Please change the sentence by using an informal tone.

Fig. 2: Sample Data

Grammar correction:
Please check the grammar errors for this sentence: '{mysentence}'. If there exist errors, show corrected sentence without mentioning the words 'corrected sentence:' and if there is no error, show original sentence. Please show only the final result without explanation.

Removal of non-English content:
Please check this sentence: '{mysentence}'. If it contains language other than English return 'true'; otherwise, return 'false'.

Elimination of hate speech:
Please check this sentence: '{mysentence}'. If it contains hate speech, return 'true'; otherwise, return 'false'.

Exclusion of advertisements:
Please check this sentence: '{mysentence}'. If it contains advertisement, return 'true'; otherwise, return 'false'.

Fig. 3: Prompts for filters

Style for Consistency:
Please change the sentence: '{mysentence}' in a way that a 30 years old PhD student would say. Please show only the final result without explanation and replies.

Fig. 4: Prompts for consistency

TABLE I: All styles in different categories

Category	Styles
Tone	formal, informal, optimistic, pessimistic, humorous, serious, inspiring, authoritative, persuasive
Family Roles	grandfather, grandmother, father, mother, son, daughter
Occupation	professor, doctor, policeman, priest, kindergarten teacher, businessman
Celebrity	Donald Trump, Joe Biden, Ellen DeGeneres, Kevin Hart, Conan O'Brien, Steve Harvey
Historical Period	old English, middle English, early modern English
Passive	passive
Diary	diary
Proverb	proverb

IV. Methods for Prompt Recovery

A. Direct Inference with LLM

In earlier models, limited size and training on specific tasks meant that performance on new tasks without fine-tuning was often poor. However, with the advent of modern LLMs, which are trained on large amounts of data and a wide range of

Data Generation:
Please find the prompt that was given to you to transform **original_text** to **new_text**. One clue is the prompt itself was short and concise. Answer in this format: "The prompt was: <the prompt>" and don't add anything else.
original_text:
<oringinal sentence>
new_text:
<result sentence>

Fig. 5: Prompts for Data Generation

Few-shot Example:
Given the example:
Example 1:
original_text:
I have a preference for sweet flavors, such as a banana smoothie with honey, but the combination of beetroot and kale does not appeal to me at all!

new_text:
I absolutely love sweet flavors like a banana smoothie with honey, but I'm open to trying new combinations like beetroot and kale to see if they can surprise me!

The prompt was: "Please change the sentence by using a optimistic tone."

Example 2:
original_text:
We will engage in a discussion on dengue fever, a disease that is often transmitted by these insects, and, as is customary, we will also explore some pertinent lexical items.

new_text:
Let's dive into the fascinating world of lexical items that will not only expand our vocabulary, but also equip us with the tools to tackle this pressing public health issue.

The prompt was: "Please change the sentence by using a persuasive tone."

Example 3:
original_text:
Notwithstanding the lack of a definitive cure, contemporary pharmacological interventions have enabled individuals to effectively manage their HIV infection, thereby prolonging their lifespan.

new_text:
Notwithstanding the lack of a magic pill, modern meds have turned HIV into a minor annoyance, allowing people to live longer and not die immediately... yet!

The prompt was: "Please change the sentence by using a humorous tone."

Fig. 6: Prompts for Few-shot

downstream tasks, direct inference has become feasible. For this study, we manually crafted prompts for both zero-shot (see Fig.5) and few-shot (see Fig.6) settings to generate the result sentences. The size of examples for the few-shot setting is kept constrained because longer sample sizes lead to increased inference time. We selected data with the highest LN-PE scores for each different style as sample data and randomly picked a subset for few-shot learning.

B. Jailbreak

As mentioned in the results of [8], jailbreak prompt does not help much for prompt recovery task. We do not try all the jailbreak prompts in [8], but only use Prefix Injection and Refusal Suppression discussed in [11].

C. Chain of Thoughts

Chain of Thoughts (CoT) [33] is effective for enhancing reasoning in language models because it breaks down complex problems into smaller, manageable steps, leading to more interpretable and accurate responses. In this work, we break the problem into four steps: compare tone and style, identify the changes, check the purpose of the transformation, and consider the clues to get the final result.

D. LLM Fine-Tuning

Although LLMs can address the problem through direct inference with some samples, fine-tuning enhances the model's ability to learn from training data. To efficiently fine-tune the model while limiting the training time, we employed Low-Rank Adaptation (LoRA) [34], a parameter-efficient technique for fine-tuning. This method adjusts only a small subset of the model's trainable parameters while maintaining strong performance.

Self-Correction:
What is the most common theme among these statements? Please do not show the common theme but only show the best sentence that conclude the common theme without additional description. <previous results>.

Fig. 7: Prompts for Data Validation

E. Canonical Prompt for Abnormal Outputs

The "Canonical Prompt" concept involves identifying a prompt that is close to the majority of prompts in the training set S_T, enabling it to serve as a fallback prompt that avoids poor performance across various inputs. The approach is summarized in Algorithm 1. We detail the process for generating a canonical prompt as follows: First, we manually create seed prompts and add them to the Generated Prompt Set S_G. We then calculate the cosine similarity between sampled prompts S_{Sample} from S_T and S_G and determine the average similarity for each prompt in S_G. Prompts in S_G with high similarity are retained as our new training set, $S_{G_{new}}$, we keep top k of the prompts. Next, we use $S_{G_{new}}$ to generate a Vocabulary Set S_V and employ beam search to insert words from S_V into the prompts in S_G, cosine similarity is used as first step to evaluate generated prompts. New prompts that significantly increase similarity to $S_{G_{new}}$ are stored as coarse results. Since word insertion can result in excessively long prompts, we apply a threshold to trim them. To further reduce the search space, we use a greedy search approach, inserting words only into the current best prompt. To improve the performance, all the steps discussed above can be run multiple times before get the best prompt. Finally, we get the final results and find the best prompt to return.

V. Experimental Setup

A. LLM

For inference and fine-tuning, we conducted experiments using Llama 3 8B model [35] and Mistral 7B v0.3 model [36]. We train models for 1 epoch with Paged Adam 8Bit optimizer with a learning rate of $2e{-}4$. To generate embedding vectors for cosine similarity, we utilize the E5-Mistral-7B-Instruct model [37], which is recognized as one of the top-performing embedding models. We use a constant learning rate with linear warmup over the first 30% training steps. We train in FP32 precision.

B. Dataset

We generated the dataset as described in Section III. First we get 16174 transcripts from selected YouTube videos, and use filters to get 13686 filtered transcripts. We did not create results for each of the filtered transcript with all 33 transformation styles, instead, we just use one of these styles for them. Then, we get the dataset consisting of 13686 instances across various categories, each including the original sentence, result sentence, style prompt, logits, and LN-PE values. We only take the data with both meaning consistency and cycle consistency larger than 0.75, so we finally get 10193 instances. The

Algorithm 1 Canonical Prompt

Input $S_T, SeedPrompts, LoopTimes$
$S_G \leftarrow SeedPrompts$
$i = 0$
while $i < LoopTimes$ **do**
  for $sentence_g \leftarrow$ `enumerate(`S_G`)` **do**
    $Similarity_g \leftarrow 0$
    $S_{Similarity} \leftarrow \emptyset$ ▷ $S_{Similarity}$ is the set for average similarity between one prompt in S_G and each prompt in S_{Sample}
    $S_{Sample} \leftarrow$ **Sample**(S_T)
    for $sentence_s \leftarrow$ `enumerate(`S_{Sample}`)` **do**
      $Similarity_g \leftarrow Similarity_g$+CosineSimilarity$(sentence_g, sentence_s)$
    end for
    $Similarity_g \leftarrow$ Average$(Similarity_g)$
    $S_{Similarity} \leftarrow S_{Similarity} \cup Similarity_g$
  end for
  $S_{G_{new}} \leftarrow$ **GetTopk**$(S_{Similarity})$ ▷ Get k prompts with the best similarity
  $S_V \leftarrow$ **GenerateVocabulary**$(S_{G_{new}})$
  for $sentence_g \leftarrow$ `enumerate(`$S_{G_{new}}$`)` **do**
    $CoarseResults \leftarrow$**BeamSearch**$(sentence_g, S_V)$
    $Results \leftarrow$ **Trim**$(CoarseResults)$
  end for
  $i = i + 1$
  $S_G \leftarrow S_G \cup Results$
end while
$BestPrompt \leftarrow$ **GetBest**$(Results)$
return $BestPrompt$

dataset is then divided as follows: 80% for training, 10% for validation, and 10% for testing.

C. Evaluation Metrics

We use sharpened cosine similarity (SCS) as in [9] and we also experiment with Exact Match, BLEU-4 [38], Rouge-L [39] and F1 score at the token level as in [1] and [8].

1) Rouge-L: ROUGE-L is a metric used to evaluate the quality of summaries by comparing them to reference summaries. Specifically, it utilizes the Longest Common Subsequence (LCS) between the prediction (pred) and ground truth sentence (gt).

$$P = \frac{LCS(pred, gt)}{length(pred)}, \tag{1}$$

where $LCS(pred, gt)$ is the length of the longest common subsequence between prediction and ground truth sentence, $length(pred)$ is the total number of words in the prediction.

$$R = \frac{LCS(pred, gt)}{length(gt)}, \tag{2}$$

where $length(gt)$ is the total number of words in the ground truth sentence.

$$F1 = \frac{2 \times P \times R}{P + R} \tag{3}$$

We use ROUGE-L F1 score in our result.

2) Token F1: Token F1 is a metric used in tasks where predictions are made at the token level. The result is calculate by comparing predictions and ground truth sentences.

$$P = \frac{TP}{TP + FP}, \tag{4}$$

where TP stands for the number of words shared by prediction and ground truth sentence and FP stands for the number of words in the ground truth sentence but not in the prediction.

$$P = \frac{TP}{TP + FN}, \tag{5}$$

where FN stands for the number of words in the prediction but not in the ground truth sentence.

$$F1 = \frac{2 \times P \times R}{P + R} \tag{6}$$

3) Sharpened Cosine Similarity: Unlike the work of [1] and [8] that just use cosine similarity, we employed sharpened cosine similarity (SCS) to provide a more refined similarity score as in [9]. The similarity is calculated as follows:

$$SCS(v_{original}, v_{result}) = (\frac{v_{original} \cdot v_{result}}{\|v_{original}\| \|v_{result}\|})^3, \tag{7}$$

where $v_{original}$ and v_{result} are the embedding vectors of the original sentence and the results, respectively. These vectors are generated using the E5-Mistral-7B-Instruct model [37]. We opted not to use the Sentence-T5-Base model as in [9]

for two main reasons. First, the E5-Mistral-7B-Instruct model outperforms Sentence-T5-Base. Secondly, the Sentence-T5-Base model implemented by Hugging Face has a known issue with the embedding of the word "lucrarea," which is almost identical to the embedding of the special token "</s>," used to close output sentences. If "lucrarea" is added multiple times to the input, the output will contain many "</s>" tokens. When calculating the similarity between such outputs and the ground truth sentences, the similarity is likely to be inflated compared to outputs generated without "lucrarea" due to the presence of "</s>" in all the embeddings.

4) BLEU: BLEU is an evaluation metric used primarily for assessing the quality of machine-translated text by comparing it to one or more ground truth translations. It measures the correspondence between the machine-generated output and the ground truth translations using n-gram precision (e.g., contiguous sequences of words of length n). The BLEU is calculate by the equation as follows:

$$\text{BLEU} = BP \times \exp\left(\frac{1}{N}\sum_{n=1}^{N}\log p_n\right), \tag{8}$$

where BP is the brevity penalty, N is the maximum n-gram length and p_n is the modified precision for n-grams of length n. And the brevity is calculated as follows:

$$BP = \begin{cases} 1 & \text{if } pred > gt \\ e^{1-\frac{gt}{pred}} & \text{if } pred \leq gt \end{cases}, \tag{9}$$

where pred is the length of the prediction, gt is the length of the ground truth sentence.

5) Exact Match: Exact Match (EM) is a simple and intuitive evaluation metric that measures how often a model's prediction exactly matches the ground truth.

VI. Experimental Results and Analysis

TABLE II: Experimental Results with Mistal-7B-Instruct Model

Setting	Rouge-L	Token F1	SCS
zero-shot	14.25	14.27	76.73
jailbreak+zero-shot	15.00	15.01	75.7
CoT+zero-shot	15.34	13.94	76.68
canonical prompt+zero-shot	14.34	14.36	76.78
one-shot	32.06	31.88	84.88
jailbreak+one-shot	26.40	26.01	81.40
CoT+one-shot	31.24	30.34	**84.98**
canonical prompt+one-shot	32.08	31.89	84.90
three-shot	19.98	19.65	73.89
jailbreak+three-shot	16.90	16.18	76.60
CoT+three-shot	30.36	29.42	81.42
canonical prompt+three-shot	20.50	20.10	74.34
five-shot	21.40	21.16	74.91
jailbreak+five-shot	16.25	15.58	76.53
CoT+five-shot	31.51	30.41	81.66
canonical prompt+five-shot	21.97	21.64	75.32
fine-tuning	**32.80**	**36.39**	84.58

TABLE III: Experimental Results with Meta-Llama-3-8B-Instruct Model

Setting	Rouge-L	Token F1	SCS
zero-shot	15.34	14.88	81.80
jailbreak+zero-shot	16.98	16.45	80.51
CoT+zero-shot	12.00	10.63	77.98
canonical prompt+zero-shot	15.34	14.88	81.80
one-shot	**79.66**	**79.64**	**90.56**
jailbreak+one-shot	36.97	36.60	86.88
CoT+one-shot	42.15	41.26	84.68
canonical prompt+one-shot	79.66	79.64	90.56
three-shot	68.42	68.26	90.20
jailbreak+three-shot	37.77	36.98	87.27
CoT+three-shot	12.51	10.88	79.02
canonical prompt+three-shot	68.42	68.26	90.20
five-shot	57.46	56.60	87.83
jailbreak+five-shot	35.42	35.24	85.37
CoT+five-shot	10.74	9.34	78.16
canonical prompt+five-shot	57.46	56.60	87.83
fine-tuning	17.18	16.88	82.12

A. Overall Result and Analysis

Since the BLEU and Exact Match scores are either zero or very close to zero, we choose not to include them in the result tables. The results are shown in Table II and Table III. Focusing on the performance of two different models, the one-shot setting and fine-tuning deliver the best results. The one-shot setting achieves the highest SCS scores for both models, while fine-tuning yields the best Rouge-L and Token F1 scores for Mistal-7B-Instruct. Overall, Meta-Llama-3-8B-Instruct outperforms Mistal-7B-Instruct in most settings, except for CoT in the zero-shot, three-shot, and five-shot settings, where Mistal-7B-Instruct has an average of 13.99 higher for Rouge-L, 14.31 higher for Token F1, and 1.3 higher for SCS compared to Meta-Llama-3-8B-Instruct.

It is clear that Meta-Llama-3-8B-Instruct is the best model for our task, while the CoT technique is proven less beneficial for it. This suggests that Meta-Llama-3-8B-Instruct is able to derive better solutions for the task independently, without relying on the "thoughts" manually designed for CoT.

Next, we examine the performance of each method for the two models, using the zero-shot setting as a baseline for comparison.

First, although all few-shot settings outperform zero-shot, the one-shot setting leads to the most significant improvements. For Mistal-7B-Instruct, one-shot improves Rouge-L by 17.91, Token F1 by 17.61, and SCS by 12.15. For Meta-Llama-3-8B-Instruct, one-shot boosts Rouge-L by 64.32, Token F1 by 64.76, and SCS by 8.76.

Secondly, the jailbreak setting results in a performance drop, with Mistal-7B-Instruct showing an average decrease of 3.29 for Rouge-L, 3.55 for Token F1, and 0.07 for SCS. Similarly, Meta-Llama-3-8B-Instruct experiences an average decrease of 23.44 for Rouge-L, 23.53 for Token F1, and 1.11 for SCS.

Thirdly, CoT improves Mistal-7B-Instruct's performance with an average increase of 5.19 points for Rouge-L, 4.29 points for Token F1, and 3.58 points for SCS. In contrast, Meta-Llama-3-8B-Instruct's performance shows a decrease of 35.87 for Rouge-L, 36.82 for Token F1, and 7.64 for SCS.

Fourthly, the canonical prompt setting provides only slight improvements for Mistal-7B-Instruct, with average gains of 0.3 for Rouge-L, 0.26 for Token F1, and 0.23 for SCS. For Meta-Llama-3-8B-Instruct, there is no noticeable change.

Finally, fine-tuning offers significant improvements for Mistal-7B-Instruct, with gains of 18.55 for Rouge-L, 22.12 for Token F1, and 7.85 for SCS. For Meta-Llama-3-8B-Instruct, fine-tuning provides more modest improvements: 1.84 points for Rouge-L, 2.00 points for Token F1, and 0.42 points for SCS. Surprisingly, fine-tuning Meta-Llama-3-8B-Instruct does not surpass the one-shot setting as seen in the few-shot settings as well. We dicuss the reason in the next section.

In conclusion, the one-shot setting delivers strong results. Methods like jailbreak, canonical prompt, and CoT offer slight improvements for Mistal-7B-Instruct but have little to no impact on Meta-Llama-3-8B-Instruct. Ultimately, Meta-Llama-3-8B-Instruct with one-shot setting proves to be the superior setting for our prompt recovery task in our study.

B. Error Analysis

Given the variety of experimental settings we tested, it is challenging to review all errors comprehensively. Here, we focus on the errors observed in the one-shot and fine-tuning settings. (See Table IV and Table V)

The first scenario involves low scores with acceptable answers. For example, the "businessman's style" prompt often results in outputs that are concise and formal, which are acceptable for most people. However, the scores for these outputs are consistently low across all metrics. This type of error is common in the fine-tuning results.

The second scenario involves high scores with incorrect answers. For instance, the model predicts "son" instead of "mother" in a family role task. The rest of the sentence is correctly predicted and the overall score remains high, even though the output is clearly wrong from a human perspective. This highlights a limitation in the metrics, as they fail to penalize such errors adequately. This type of error is common in the one-shot setting.

The third scenario involves low scores with incorrect answers, which is expected based on the metrics we use. However, we cannot definitively say that the answer is 100% wrong, as even though we generate the result based on the ground truth, prompts other than the ground truth may also lead to similar result sentences.

In the first scenario, Rouge-L and Token F1 fail to reflect the close semantic similarities between the prediction and ground truth, while SCS captures some similarity but remains inadequate. In the second scenario, we conclude that Rouge-L, Token F1, and SCS all yield high scores, indicating that they do not fully capture the nuances of our specific task. In the third scenario, it underscores the inherent difficulty of the prompt recovery task, emphasizing the need for further research and exploration.

VII. Limitations

First, as mentioned in error analysis, the metrics we use have defects when facing some specific scenarios and need to be improved. Secondly, although LLMs demonstrate strong performance on our dataset, this success may not extend to out-of-distribution data, highlighting the need for further experiments to assess generalization. Thirdly, given the vast scale of LLMs, the dataset we generated may be insufficient compared to the data used during pre-training for these LLMs. Expanding the dataset, potentially through data augmentation techniques, could improve performance. Fourthly, our research focuses on English transcripts from YouTube videos, which provide a rich dataset for understanding contemporary language usage in various contexts, but the dataset can be extend to other languages to incorporate multilingual and cross-cultural perspectives. Fifthly, our study focuses on a specific prompt recovery scenario, and many of the methods we explored may not easily apply to more general prompt recovery tasks, where the output format is not constrained. Addressing the general prompt recovery challenge may require leveraging adversarial attack techniques, jailbreak methods, or other strategies to extract additional information about the input prompt.

VIII. Conclusion

In conclusion, we explore a unique aspect of the prompt recovery task, focusing on scenarios where a prompt alters the writing style or rephrases a sentence. Our work introduces a new benchmark dataset, **StyleRec**, specifically designed to address this specialized challenge, ensuring both quality and comprehensive coverage. Through our experiments, we demonstrate that some methods contribute to this complex problem and one-shot is the best among them. However, the error analysis reveals that the current metrics are inadequate for our specific task and require improvement. Our research not only advances the understanding of prompt recovery but also opens up new avenues for further exploration, encouraging the development of innovative approaches to tackle the general prompt recovery challenge.

Acknowledgment

The work was supported in part by NSF under Grants 2321572 and 1952792.

References

[1] J. X. Morris, W. Zhao, J. T. Chiu, V. Shmatikov, and A. M. Rush, "Language model inversion," *arXiv preprint arXiv:2311.13647*, 2023.

[2] Z. Sha and Y. Zhang, "Prompt stealing attacks against large language models," *arXiv preprint arXiv:2402.12959*, 2024.

[3] Y. Wu, X. Li, Y. Liu, P. Zhou, and L. Sun, "Jailbreaking gpt-4v via self-adversarial attacks with system prompts," *arXiv preprint arXiv:2311.09127*, 2023.

[4] P. Chao, A. Robey, E. Dobriban, H. Hassani, G. J. Pappas, and E. Wong, "Jailbreaking black box large language models in twenty queries," *arXiv preprint arXiv:2310.08419*, 2023.

[5] A. Skapars, E. Manino, Y. Sun, and L. C. Cordeiro, "Was it slander? towards exact inversion of generative language models," *arXiv preprint arXiv:2407.11059*, 2024.

[6] A. Karamolegkou, J. Li, L. Zhou, and A. Søgaard, "Copyright violations and large language models," *arXiv preprint arXiv:2310.13771*, 2023.

[7] L. Give, T. Zaoral, and M. A. Bruno, "Uncovering hidden intentions: Exploring prompt recovery for deeper insights into generated texts," *arXiv preprint arXiv:2406.15871*, 2024.

TABLE IV: Samples of Groud Truth for Error Analysis

scenario	Ground Truth
low score with acceptable answer	Please change the sentence by using a businessman's style.
high score with incorrect answer	Please change the sentence by using a mother's style.
low score with incorrect answer	Please change the sentence by using a doctor's style.

TABLE V: Samples of Prediction and Metrics for Error Analysis

Prediction	Rouge-L	Token F1	SCS
Rewrite the text to make it more concise and formal.	10.53	10.53	85.17
Please change the sentence by using a son's style.	88.89	88.89	95.25
Rewrite the text to focus on the concept of cultural exchange and innovation.	9.52	9.09	70.43

[8] L. Gao, R. Peng, Y. Zhang, and J. Zhao, "Dory: Deliberative prompt recovery for llm," *arXiv preprint arXiv:2405.20657*, 2024.

[9] J. Chen, W. Xu, Z. Ding, J. Xu, H. Yan, and X. Zhang, "Advancing prompt recovery in nlp: A deep dive into the integration of gemma-2b-it and phi2 models," *arXiv preprint arXiv:2407.05233*, 2024.

[10] Y. Liu, G. Deng, Z. Xu, Y. Li, Y. Zheng, Y. Zhang, L. Zhao, T. Zhang, K. Wang, and Y. Liu, "Jailbreaking chatgpt via prompt engineering: An empirical study," *arXiv preprint arXiv:2305.13860*, 2023.

[11] A. Wei, N. Haghtalab, and J. Steinhardt, "Jailbroken: How does llm safety training fail?" *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[12] G. Deng, Y. Liu, Y. Li, K. Wang, Y. Zhang, Z. Li, H. Wang, T. Zhang, and Y. Liu, "Jailbreaker: Automated jailbreak across multiple large language model chatbots," *arXiv preprint arXiv:2307.08715*, 2023.

[13] M. Mazeika, L. Phan, X. Yin, A. Zou, Z. Wang, N. Mu, E. Sakhaee, N. Li, S. Basart, B. Li *et al.*, "Harmbench: A standardized evaluation framework for automated red teaming and robust refusal," *arXiv preprint arXiv:2402.04249*, 2024.

[14] Z. Xu, Y. Liu, G. Deng, Y. Li, and S. Picek, "A comprehensive study of jailbreak attack versus defense for large language models," in *Findings of the Association for Computational Linguistics ACL 2024*, 2024, pp. 7432–7449.

[15] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction {APIs}," in *25th USENIX security symposium (USENIX Security 16)*, 2016, pp. 601–618.

[16] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE symposium on security and privacy (SP)*. IEEE, 2017, pp. 3–18.

[17] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and trends® in machine learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[18] E. Wallace, M. Stern, and D. Song, "Imitation attacks and defenses for black-box machine translation systems," *arXiv preprint arXiv:2004.15015*, 2020.

[19] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, "Deep learning for healthcare: review, opportunities and challenges," *Briefings in bioinformatics*, vol. 19, no. 6, pp. 1236–1246, 2018.

[20] T. Ching, D. S. Himmelstein, B. K. Beaulieu-Jones, A. A. Kalinin, B. T. Do, G. P. Way, E. Ferrero, P.-M. Agapow, M. Zietz, M. M. Hoffman *et al.*, "Opportunities and obstacles for deep learning in biology and medicine," *Journal of the royal society interface*, vol. 15, no. 141, p. 20170387, 2018.

[21] P. Mishra, R. Lehmkuhl, A. Srinivasan, W. Zheng, and R. A. Popa, "Delphi: A cryptographic inference system for neural networks," in *Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice*, 2020, pp. 27–30.

[22] A. Gudibande, E. Wallace, C. Snell, X. Geng, H. Liu, P. Abbeel, S. Levine, and D. Song, "The false promise of imitating proprietary llms," *arXiv preprint arXiv:2305.15717*, 2023.

[23] M. Juuti, S. Szyller, S. Marchal, and N. Asokan, "Prada: protecting against dnn model stealing attacks," in *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2019, pp. 512–527.

[24] T. Lee, B. Edwards, I. Molloy, and D. Su, "Defending against neural network model stealing attacks using deceptive perturbations," in *2019 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2019, pp. 43–49.

[25] G. Sebastian, "Privacy and data protection in chatgpt and other ai chatbots: strategies for securing user information," *International Journal of Security and Privacy in Pervasive Computing (IJSPPC)*, vol. 15, no. 1, pp. 1–14, 2023.

[26] J. Guan, J. Liang, and R. He, "Are you stealing my model? sample correlation for fingerprinting deep neural networks," *Advances in Neural Information Processing Systems*, vol. 35, pp. 36 571–36 584, 2022.

[27] X. Liu, T. Liu, H. Yang, J. Dong, Z. Ying, and Z. Ma, "Model stealing detection for iot services based on multi-dimensional features," *IEEE Internet of Things Journal*, 2024.

[28] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1322–1333.

[29] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song, "The secret revealer: Generative model-inversion attacks against deep neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 253–261.

[30] T. Ahmed and P. Devanbu, "Better patching using llm prompting, via self-consistency," in *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2023, pp. 1742–1746.

[31] A. Malinin and M. Gales, "Uncertainty estimation in autoregressive structured prediction," *arXiv preprint arXiv:2002.07650*, 2020.

[32] Z. Chen, J. Li, Y. Luo, Z. Huang, and Y. Yang, "Canzsl: Cycle-consistent adversarial networks for zero-shot learning from natural language," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2020, pp. 874–883.

[33] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[34] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[35] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[36] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier *et al.*, "Mistral 7b," *arXiv preprint arXiv:2310.06825*, 2023.

[37] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, and F. Wei, "Improving text embeddings with large language models," *arXiv preprint arXiv:2401.00368*, 2023.

[38] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[39] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.