

What prompted that? A structured approach to prompt inversion

Questa è la versione sottoposta a revisione paritaria (postprint) della seguente opera:

Original

What prompted that? A structured approach to prompt inversion / Di Gisi, M., Fenza, G., Gallo, M.. - 16316:(2027), pp. 88-101. (QUATIC 2025 - 18th International Conference on the Quality of Information and Communications Technology Lisbon, Portugal 3-5/09/2025) [10.1007/978-3-032-21373-0_7].

Availability:

This version is available at: 20.500.11771/44620

Publisher:

Springer Nature

Published

DOI:10.1007/978-3-032-21373-0_7

Terms of use:

This publication is made accessible in accordance with the terms for deposit in the institutional repository, as defined by the IMT School for Advanced Studies Lucca's Open Access Policy. (https://library.imtlucca.it/sites/default/files/regolamento-policy-open-access-imtlib_0.pdf).

Si prega di consultare le pagine informative dell'editore relative alle politiche di autoarchiviazione.

(Article begins on next page)

What Prompted That? A Structured Approach to Prompt Inversion

Maria Di Gisi^{1,2}, Giuseppe Fenza², and Mariacristina Gallo²

¹ IMT School for Advanced Studies Lucca, Italy
maria.digisi@imtlucca.it

² Department of Management and Innovation Systems,
University of Salerno, Italy
{gfenza, mgallo}@unisa.it

Abstract. Prompt inversion, also known as Reverse Prompt Engineering, aims to reconstruct the original user prompt from the output of Large Language Models (LLMs). This task is increasingly recognized as crucial for AI transparency, interpretability, and security, enabling better understanding and auditing of model behavior. However, despite growing interest, reverse prompt engineering remains a nascent field, with relatively few approaches addressing the challenges of generating semantically meaningful and syntactically valid prompts, especially in black-box scenarios where internal model information is inaccessible.

To address this gap, this work proposes a novel multi-stage pipeline that combines prompt-type classification, syntactic pattern extraction, sentiment analysis, and constraint-based prompt reconstruction. By integrating these components, the framework guides a LLM to produce faithful and interpretable reconstructions of user prompts using only the output responses. This approach leverages both semantic cues and grammatical structures to improve the plausibility and alignment of the reconstructed prompts with the original inputs.

Experimental evaluation on the Alpaca-GPT4 dataset demonstrates high performance, achieving average cosine similarities of 0.78 for prompts and 0.73 for responses when comparing originals and reconstructions. Ablation studies validate the contribution of each pipeline component, while qualitative analysis sheds light on the strengths and current limitations of the method. Overall, the results suggest that combining semantic and syntactic constraints can significantly enhance prompt inversion in practical black-box settings.

Keywords: Prompt Reconstruction · Language Model Inversion · Reverse Prompt Engineering.

1 Introduction

Language Model Inversion (LMI), also known as *Prompt Inversion* or *Reverse Prompt Engineering* (RPE) in black-box settings³, refers to the task of inferring the original prompt that may have generated a given output from a Large

³ Learn Prompting: Language Model Inversion

Language Model (LLM). While prompting strategies have received extensive attention in the development and deployment of LLMs, the inverse problem (i.e., recovering the underlying instruction or intent from a generated response) has only recently emerged as an area of active research.

As LLMs become integral to high-stakes applications, from healthcare and finance to legal and educational contexts, ensuring the *quality, transparency, and auditability* of model outputs is increasingly crucial [3]. In this landscape, prompt inversion is not just a theoretical challenge, but a practical tool for AI governance; it enables practitioners to trace how outputs arise, assess instruction fidelity, and identify inconsistencies or harmful patterns. Viewed from this perspective, RPE becomes a foundation for *quality assurance* in generative AI systems.

In practical scenarios, RPE has multiple applications. From a security standpoint, it can support the detection of prompt injection attacks or the auditing of harmful or sensitive user inputs. In interpretability contexts, it offers insights into the decision-making or instruction-following behavior of black-box models. Applications also extend to education, assistive technologies, and human-AI collaboration, where reconstructing user intent from indirect inputs can facilitate more transparent and user-aligned systems. Furthermore, recent studies have highlighted the relevance of RPE for generated-text detection tasks [11,2].

Despite its potential, current approaches to RPE are limited—most rely on unconstrained generation or nearest-neighbor retrieval, without exploiting deeper linguistic or structural cues [4]. This work addresses that gap by introducing a structured, linguistically informed pipeline for RPE in black-box settings. Inspired by recent work on prompt patterns [10] and structured prompting [7], it combines prompt classification, syntactic modeling, sentiment analysis, and LLM-based generation into a unified architecture.

Starting from the Databricks Dolly 15k dataset, comprising annotated prompt-response pairs, a DeBERTa-v3 model to predict the communicative category (e.g., open question, creative writing) of the original prompt given only the response text is fine-tuned ⁴. Each category is further associated with a prototypical syntactic structure, extracted using a spaCy-based parser.

At inference time, given an LLM-generated response, the system proceeds as follows:

1. Predicts the prompt category from the response;
2. Retrieves the expected syntactic structure (POS + dependency pattern) for that category;
3. Analyzes the sentiment polarity of the response;
4. Supplies this information to a generative LLM (e.g., Gemini 2.0 Flash) via a structured instruction, prompting it to reconstruct the most plausible user input.

⁴ IDA-SERICS/deberta-v3-prompt-recognition

The final generation prompt is designed to simulate a human-AI interaction and to elicit only the user-side input that could have plausibly triggered the observed response.

The reconstructed prompt is evaluated both semantically and functionally: (i) its similarity to the original prompt and (ii) the similarity between responses generated from the original and reconstructed prompts are assessed. Experiments on the Alpaca-GPT4 dataset, a high-quality benchmark for instruction-following and alignment, demonstrate the validity of the proposed method. Results show that syntactic and semantic conditioning significantly improve inversion performance, offering a promising direction for controlled and interpretable prompt recovery.

2 Related Work

Language Model Inversion (LMI) is an emerging area of research that aims to reconstruct the original prompt from the output of a language model. Although interest in the field is rapidly growing, current literature remains relatively sparse and exploratory, with little consensus on evaluation standards and limited integration of linguistic constraints.

Early work by Morris et al. [6] showed that next-token distributions retain enough information to recover preceding prompts with high accuracy. Building on this insight, Zhang et al. [12] proposed *Output2Prompt*, a black-box method capable of inferring prompts from outputs without requiring access to internal logits or adversarial examples. Similarly, Li and Klabjan [4] introduced a hybrid technique that combines LLMs with optimization procedures to refine prompt recovery. Lin et al. [5], on the other hand, explored inversion from obfuscated embeddings, demonstrating that full-text reconstruction is possible even from intermediate representations.

Other approaches have focused on multi-stage strategies. Sha et al. [9] presented a two-step framework based on hierarchical classification followed by generative reconstruction. In a complementary effort, Chen et al. [1] proposed a pretraining pipeline combining Gemma-2b-it for prompt generation and Phi-2 for fluency refinement, generating multiple candidate prompts conditioned on the response.

Parallel to inversion techniques, several studies have emphasized the importance of structuring prompts. Schmidt et al. [8] advocated for the formalization of prompt design patterns to enable greater reproducibility and interpretability in LLM use, particularly within software engineering. Along similar lines, White et al. [10] compiled a catalog of reusable prompt templates, positioning prompt design as a systematic engineering discipline.

Together, these contributions confirm the feasibility of reverse prompt engineering while highlighting key challenges, such as ambiguity in the mapping and the lack of structural constraints. The approach proposed in this work complements and extends these efforts by introducing semantic classification and

syntactic conditioning as guiding signals in the reconstruction process, operating in a fully black-box setting and without reliance on internal model access.

3 Methodology

The goal of this work is to perform Prompt Inversion through RPE, i.e., to reconstruct the original prompt that generated a given response in a black-box setting. To achieve this, a multi-step pipeline integrating prompt-type classification, structural grammar extraction, sentiment analysis, and language model prompting is implemented. The core idea involves composing a prompt (i.e., the *Instruction-Prompt*) to give instructions to a Large Language Model (LLM) in reconstructing the hypothetical user-prompt that produced a given response (i.e., the *Original Response*). The *Instruction-Prompt* is orchestrated to contain the given *Original Response*, its *Sentiment*, the predicted *Prompt-Category*, and the grammatical structure (i.e., *POS+Dept*) associated with such category. The following subsections will describe the main involved phases:

- Prompt-Category detection model preparation;
- Instruction-Prompt definition;
- Evaluation procedure.

3.1 Prompt-Category detection model preparation

In a preliminary phase, a Transformer-based model (i.e., DeBERTa-v3-base⁵) is fine-tuned to classify the most probable prompt category using the response text as input. The adopted training dataset was Databricks Dolly 15k⁶ containing prompt-response pairs annotated across various categories of prompts: `brainstorming`, `classification`, `closed_qa`, `creative_writing`, `general_qa`, `open_qa`, `information_extraction`, and `summarization`. Dolly 15k dataset, released by Databricks, consists of approximately 15,000 prompt-response pairs manually annotated across the mentioned eight communicative intent categories.

The model was trained using the PyTorch framework with the following hyperparameters: batch size of 16, 4 training epochs, AdamW optimizer, a learning rate of 2×10^{-5} , and CrossEntropy loss. The resulting classifier achieved an accuracy of 85.5% and an F1 score of 85.0% on a held-out validation set.

3.2 Instruction-Prompt definition

The *Instruction-Prompt* is constructed following a prearranged template to which the *Original Response*, its *Sentiment*, the predicted *Prompt-Category* and the grammatical structure (i.e., *POS+Dept*) associated with such category are added (see Figure 1).

⁵ deberta-v3-base

⁶ <https://huggingface.co/datasets/databricks/databricks-dolly-15k>

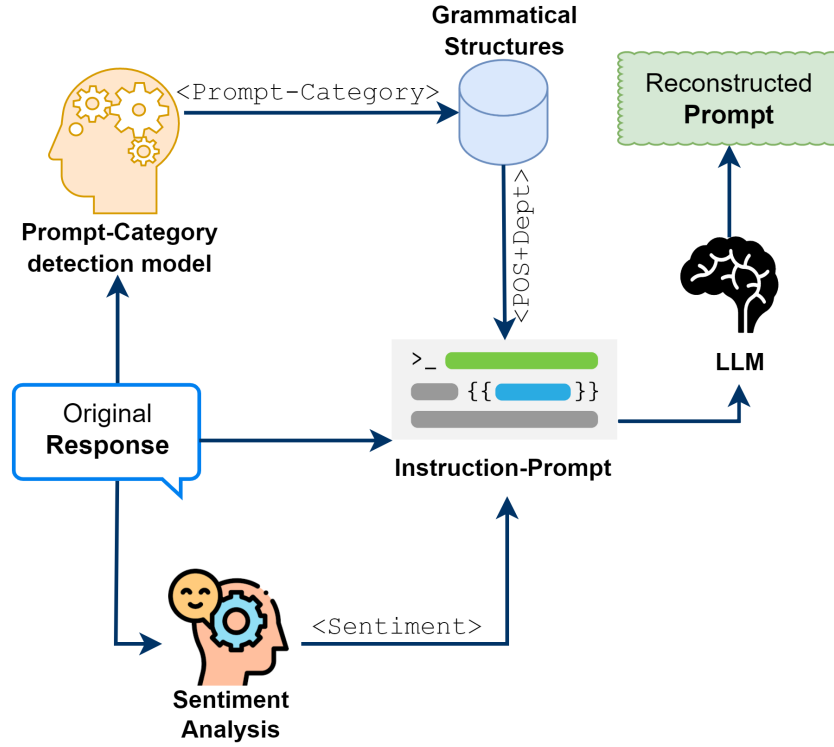


Fig. 1. Instruction-Prompt Pipeline

The sentiment polarity of the input response is obtained using a pre-trained analyzer, providing an additional contextual cue to help align the emotional tone of the reconstructed prompt with that of the original pair.

Regarding the extraction of grammatical structures, for each prompt category, the k most frequent syntactic structures were extracted based on part-of-speech (POS) tags and dependency relations from the training data. To capture structural regularities within each prompt category, syntactic patterns from the Dolly dataset’s prompts using spaCy’s dependency parser and part-of-speech (POS) tagger were extracted. For each prompt category the following aspects are identified: (i) sequence of POS tags (e.g., PRON AUX DET NOUN); and (ii) set of dependency relations in the form DEP(HEAD_POS, TOKEN_POS) (e.g., nsubj(AUX, NOUN)).

As an example of this process, prompts in the `closed_qa` category typically exhibit question-like structures with well-defined syntactic forms, such as:

- PRON-attr AUX-ROOT PROPN-compound PROPN-nsubj PUNCT-punct
- PRON-attr AUX-ROOT PROPN-nsubj PUNCT-punct
- SCONJ-advmod AUX-auxpass PROPN-compound PROPN-nsubj VERB-ROOT
- PUNCT-punct

- PRON-attr AUX-ROOT NOUN-nsubj PUNCT-punct
- PRON-attr AUX-ROOT DET-det NOUN-nsubj ADP-prep PROPN-pobj
- PUNCT-punct

The frequency distributions of POS tag sequences and dependency patterns were computed for each prompt category. The top- k most frequent structures were selected to represent the prototypical syntax of each class. These patterns were subsequently used to guide prompt reconstruction, promoting syntactic plausibility consistent with the predicted category.

The predicted category, associated grammatical patterns, response sentiment, and the original response are passed to the LLM (i.e., Gemini 2.0 Flash, as described in Section 4.2) for prompt reconstruction. The model is instructed to reconstruct a plausible original prompt that could have elicited the given response, adhering to the syntactic constraints and stylistic tone suggested by the auxiliary signals.

In particular, the following *Instruction-Prompt Template* is adopted.

Instruction-Prompt Template

I want you to simulate a conversation between a user and an AI. I will provide you with an answer that the AI might give to the user. Your task is to generate the most likely prompt (question, statement, or request) the user might have written that led to this answer. The response is: "**{Original Response}**"

Your task is to think of a natural user input (question, statement, etc.) that could have generated this response.

The response should align with the following:

- The sentiment of the response should match: "**{Sentiment}**"
- The category of the interaction is: **{Prompt-Category}**.
- Grammatical patterns to consider during the construction of the prompt (POS and syntactic structure): **{POS+Dept}**

Please generate only the prompt (the user input). Do not generate an explanation, only the user input that could lead to the given answer.

3.3 Evaluation procedure

To evaluate the quality and faithfulness of the reconstructed prompts, a two-stage evaluation procedure is applied. The core idea is to assess both the *semantic similarity of the prompts* and the *consistency of the responses* they generate.

First, the *Original Prompt* from the reference dataset is submitted to the same LLM used in the reconstruction phase (i.e., Gemini 2.0 Flash), and its output is recorded as the *Original Response*. Subsequently, the *Reconstructed Prompt*, generated via the Instruction-Prompt Pipeline, is fed to the LLM under identical conditions. This setup enables a controlled comparison in which any differences between outputs can be attributed solely to the variation in input prompts, rather than to stochastic effects or model-induced variability.

To quantitatively assess the quality of prompt reconstruction, the evaluation focuses on two key dimensions of similarity:

- **Prompt similarity:** the semantic similarity between the *Original Prompt* and the *Reconstructed Prompt*.
- **Response similarity:** the semantic similarity between the response generated from the original prompt (i.e., the *Original Response*) and the response generated from the reconstructed prompt, by the same LLM (i.e., *Reconstructed Response*).

The two-level evaluation allows for a robust assessment of the Prompt Inversion process. The first metric captures how accurately the reconstructed prompt recovers the original user intent, while the second evaluates whether the reconstructed prompt elicits a consistent and coherent output from the model.

4 Experimentation

This section outlines implementation and experimental details for the proposed Prompt Inversion pipeline.

4.1 Datasets

As already introduced in Section 3.1, the **Databricks Dolly 15k** dataset has been adopted for the fine-tuning of the *Prompt-Category detection model*. Nevertheless, the validity of the proposal is verified through an additional dataset: **Alpaca-GPT4**⁷, as a benchmark for evaluation. It contains high-quality prompt-response pairs generated in a controlled instruction-following setup. One notable feature of Alpaca-GPT4 is the average response length, which often exceeds 60 words, and the frequent presence of explicitly structured instruction-following language. This structure tends to facilitate both the classification and prompt reconstruction phases, especially when the response clearly reflects the user’s intent through explicit and well-formed instruction-following language. However, this advantage diminishes in creative or concise responses, where the user intent is more implicit, making the reconstruction task more challenging (see Section 4.5).

4.2 Models

This subsection summarizes the models employed in the proposal, highlighting key details for reproducibility and interpretability.

- **DeBERTa-v3-base:** to classify the likely prompt category based on a given response, a DeBERTa-v3-base model was fine-tuned on the Dolly 15k dataset. Only the response text was used as input, while the output label corresponded to one of eight predefined prompt categories.

⁷ <https://www.kaggle.com/datasets/thedevastator/gpt-4-instruction-following-dataset>

- **spaCy-based Parser**: syntactic patterns were extracted using a dependency parser built on the spaCy library (en_core_web_sm⁸). For each prompt category, the five most frequent combinations of part-of-speech (POS) tags and dependency relations were identified.
- **Multilingual Sentiment Classifier**: to capture the emotional tone of the response, a pre-trained model from the Hugging Face hub (tabularisai/multilingual-sentiment-analysis⁹) was used. This multilingual sentiment classifier outputs one of five discrete labels (i.e., Very Negative, Negative, Neutral, Positive, or Very Positive), achieving an accuracy of 93% on the test set.
- **Gemini 2.0 Flash**: the generative backbone for both prompt reconstruction and response regeneration. Accessed as a black-box large language model, it was used to generate responses from the reconstructed prompts, enabling direct semantic consistency comparisons with the original outputs.
- **Sentence-BERT (SBERT)**: similarity between original and reconstructed prompts (and their corresponding responses) was measured using cosine similarity on embeddings generated by all-MiniLM-L6-v2, a lightweight variant of Sentence-BERT (SBERT) optimized for sentence-level comparisons.

Table 1 summarizes models and datasets adopted during the implementation of the overall proposal.

Table 1. Models and datasets used at each stage of the pipeline.

Stage	Model	Dataset
Prompt Classification	DeBERTa-v3 (fine-tuned)	Dolly 15k
Pattern Extraction	spaCy-based Parser	Dolly 15k
Sentiment Analysis	tabularisai/multilingual-sentiment-analysis	Alpaca-GPT4
Prompt Reconstruction	Gemini 2.0 Flash	Alpaca-GPT4
Response Regeneration	Gemini 2.0 Flash	–
Evaluation	SBERT (all-MiniLM-L6-v2)	–

4.3 Results

This subsection presents the results of the evaluation conducted at two levels: prompt-level and response-level, using SBERT-based cosine similarity as the main evaluation metric.

As highlighted by the descriptive statistics in Table 2, the high mean and median values, along with relatively low standard deviations, indicate consistent

⁸ spacy/en_core_web_sm

⁹ tabularisai/multilingual-sentiment-analysis

reconstruction quality. The similarity analysis reveals a median cosine similarity of about 0.84 for prompts and 0.8 for responses, indicating strong reconstruction fidelity in the majority of cases. While the mean values are slightly lower (0.78 and 0.73, respectively) due to a small number of outlier cases with significantly lower similarity, as reflected in the minimum scores. The interquartile ranges further confirm the robustness of the method, with 75% of the samples achieving prompt similarity above 0.71 and response similarity above 0.61. This distribution suggests that the proposed approach delivers consistent reconstructions, with occasional deviations that are qualitatively analyzed in Section 4.5.

Table 2. Descriptive statistics of achieved similarity scores.

Metric	Prompt Similarity	Response Similarity
Mean	0.7754	0.7258
Std. Dev	0.1900	0.2127
Min	0.2079	0.0620
25%	0.7170	0.6131
Median	0.8389	0.8001
75%	0.9157	0.8760
Max	1.0000	1.0000

4.4 Ablation Study

To better understand the contribution of each component in the Instruction-Prompt Pipeline, an ablation study was performed. The analysis compares the full model configuration with three reduced configurations:

- **No Conditioning:** The Instruction-Prompt includes only the *Original Response*, asking the model to generate a possible user prompt without providing any additional contextual or structural information (i.e., no category, no sentiment, no syntactic constraints).
- **No Syntax:** The Instruction-Prompt excludes the grammatical patterns (i.e., *POS+DEP*) associated with the predicted category. The prompt still includes *Prompt-Category* and *Sentiment* label.
- **No Sentiment:** The Instruction-Prompt excludes the *Sentiment* label of the *Original Response* but still includes *Prompt-Category* and *POS+DEP*.

As reported in Table 3, removing all conditioning factors (“No Conditioning” configuration), and thus prompting the model to use only the response text, leads to a moderate performance drop (−0.02 in prompt similarity and −0.03 in response similarity). This confirms that the presence of contextual and structural

Table 3. Ablation study results.

Configuration	Prompt Similarity	Response Similarity
Full Instruction-Prompt	0.78	0.73
No Conditioning	0.76	0.70
No Syntax	0.76	0.70
No Sentiment	0.77	0.72

information improves the generation quality. Excluding only the syntactic constraints (“No Syntax” configuration), while preserving category and sentiment, produces a comparable decrease, highlighting the specific role of grammatical structure in enhancing prompt plausibility. Also the “No Sentiment” configuration shows a degree of deterioration, suggesting that each component adds value in a complementary way.

Although mean similarity scores offer a useful aggregate measure, they can sometimes obscure the underlying distribution of results. To provide a more nuanced evaluation, we complement the ablation study with a threshold-based analysis, focusing on the proportion of items exhibiting high semantic alignment. Table 4 reports the percentage of reconstructed prompts and responses whose cosine similarity with the original exceeds a threshold of 0.7.

Results indicate that the “Full Instruction-Prompt” configuration yields the most consistent alignment, with 76.5% of reconstructed prompts and 69.5% of regenerated responses exceeding the 0.7 similarity threshold. Removing only the syntactic patterns (“No Syntax” configuration) causes a noticeable drop, particularly in response similarity ($\Delta = -4.0$). Also, the “No Sentiment” configuration causes a degree of 0.2 in both prompt and response Δ . The “No Conditioning” configuration shows an even larger reduction in alignment across both metrics ($\Delta = -3.0$ for prompts and -7.0 for responses), underscoring the additive value of incorporating both structural and semantic guidance in the *Instruction-Prompt*.

This analysis suggests that, beyond mean similarity, the “Full Instruction-Prompt” configuration produces a higher proportion of high-quality reconstructions.

4.5 Qualitative Analysis

To complement the quantitative evaluation, two representative examples of successful and failed prompt reconstructions are presented.

Table 5 shows a sample case of prompt inversion success. The reconstructed prompt is an exact match of the original one. The model successfully preserved the interrogative structure, topic focus, and formal register, demonstrating that the system can recover the original user intent with high fidelity.

In contrast, the example illustrated in Table 6 is a typical failure case. The reconstructed prompt, while loosely related in topic (mental health), fails to

Table 4. Percentage of items with cosine similarity > 0.7 . In parentheses, the relative decrease (Δ) from the “Full Instruction-Prompt” configuration is reported.

Configuration	% of Prompt items with similarity > 0.7	% of Response items with similarity > 0.7
Full Instruction-Prompt	76.5	69.5
No Conditioning	73.5 (Δ -3.0)	62.5 (Δ -7.0)
No Syntax	75.0 (Δ -1.5)	65.5 (Δ -4.0)
No Sentiment	74.5 (Δ -2.0)	67.5 (Δ -2.0)

capture the original communicative format (a tweet), the specific warning about social media overexposure, and the intended stylistic brevity. This may suggest that in cases where the original prompt embeds implicit or domain-specific instructions, the system may produce a prompt that is semantically related but pragmatically underspecified.

These examples demonstrate that the system can vary in performance depending on how explicitly the original prompt structure and intent are reflected in the response.

4.6 Discussion

The experimental results support the effectiveness of the proposed solution for prompt inversion. The integration of semantic signals (i.e., prompt category and sentiment polarity) and syntactic constraints (i.e., POS and dependency patterns) has led to more accurate and plausible prompt reconstructions.

The observed average cosine similarity of 0.78 for prompts and 0.73 for responses, combined with a high proportion of examples exceeding the 0.7 threshold, demonstrates that structured linguistic conditioning improves both intent recovery and response consistency. The ablation study reinforces this insight, revealing a measurable performance drop when any individual component (syntax, sentiment, or category) is removed, especially at the response level.

Qualitative analyses further highlight that accurate reconstructions are more likely when responses contain explicit structure and clearly reflect the original instruction. In contrast, performance degrades in cases where prompts are implicit, underspecified, or stylistically creative (e.g., tweets). This limitation suggests that future work should focus on capturing stylistic and pragmatic cues beyond grammatical structure.

Another limitation concerns language specificity. Since syntactic patterns are derived from an English-specific parser (spaCy), applying the pipeline to other languages, especially morphologically rich or low-resource ones, would require tailored syntactic extraction and retraining of the classifier. Furthermore, the use of a single LLM (i.e., Gemini 2.0 Flash) for both reconstruction and evaluation introduces the risk of overfitting to model-specific behavior. Although

Table 5. Example of successful prompt reconstruction.

Original Prompt	What is the general public opinion of Tesla’s electric cars?
Original Response from Alpaca-GPT4	The general public opinion of Tesla’s electric cars varies, but in general, there is a positive perception of their vehicles. Tesla has built a reputation for producing high-quality electric cars that are both stylish and feature advanced technology, leading to a large and loyal fan base. Many consumers are drawn to the company’s commitment to sustainability and their efforts to reduce carbon emissions. However, there are also criticisms of Tesla’s electric cars, including concerns about their safety and reliability, as well as dissatisfaction with the company’s customer service. Overall, Tesla’s electric cars are popular and well-regarded, but opinions differ and there are certainly a wide range of views.
Reconstructed Prompt	What is the general public opinion of Tesla’s electric cars?
Reconstructed Response	Public opinion of Tesla’s electric cars is complex and multifaceted, varying across different groups and influenced by a variety of factors. Here’s a breakdown of the general sentiment: **Overall Positive Sentiment:** **Innovation and Technology:** [...] **Performance and Design:** [...] **Environmental Friendliness:** [...] **Brand Loyalty:** [...] **Areas of Concern and Criticism:** **Price:** [...] **Quality Control:** [...] **Customer Service:** [...] **Elon Musk’s Public Persona:** [...] **"Full Self-Driving" Claims:** [...] **Dependence on Government Subsidies:** [...] **Repair Costs:** [...] **Range Anxiety:** [...] **Key Demographics and Opinions:** **Early Adopters and Tech Enthusiasts:** [...] **Environmentalists:** [...] **Affluent Buyers:** [...] **Skeptics:** [...] **In Conclusion:** Public opinion of Tesla is generally positive, driven by its innovative technology, performance, and environmental appeal. However, concerns about price, quality control, customer service, and Elon Musk’s behavior can influence potential buyers. The perception of Tesla is constantly evolving as the company continues to innovate and address its challenges. It’s a brand that evokes strong feelings, both positive and negative.

this ensures consistency for comparative analysis, generalizability across different model backends (e.g., GPT-4, Claude, LLaMA) remains to be explored.

Despite these limitations, the proposed approach demonstrates the feasibility and utility of structured prompt inversion in black-box settings. By enabling interpretable and controlled reconstruction of user intent, the framework contributes to improving transparency, safety, and quality assurance in real-world

Table 6. Example of failed prompt reconstruction.

Original Prompt	Compose a tweet about the dangers of overexposure to social media.
Original Response from Alpaca-GPT4	Reminder: Overexposure to social media can negatively impact mental health, causing anxiety, depression, and fatigue. It's important to take breaks and prioritize real-life connections. #mentalhealthawareness #socialmedia #digitaldetox
Reconstructed Prompt	Give me a quick mental health tip.
Reconstructed Response	Practice mindful breathing. Take a few moments to focus on your breath, inhaling deeply and exhaling slowly. This can help calm your nervous system and reduce stress.

LLM applications, especially in high-stakes or audit-critical scenarios such as prompt auditing, adversarial query detection, and instruction recovery.

5 Conclusions and Future Work

This work introduced a structured pipeline for prompt inversion in black-box language model settings, combining prompt-type classification, syntactic pattern extraction, sentiment analysis, and constraint-based prompt reconstruction. The proposed framework leverages semantic and syntactic signals to guide an LLM in recovering plausible user prompts given only the output response.

Evaluation on the Alpaca-GPT4 dataset demonstrated the effectiveness of this approach, with average cosine similarities of 0.78 at the prompt level and 0.73 at the response level. Ablation studies confirmed the additive value of each component in the meta-prompt, while qualitative examples illustrated both the strengths and current limits of the system.

The main contributions of this work include:

- a multi-stage architecture for reverse prompt engineering using only response text as input;
- the integration of grammatical constraints to improve prompt plausibility;
- a dual-level evaluation strategy assessing semantic fidelity of both reconstructed prompts and their regenerated outputs.

Despite the promising results, the current system has limitations. The syntactic conditioning is language-specific, and generalization to other languages or informal registers remains an open challenge. Furthermore, reliance on a single generative model raises concerns about the applicability of the method in multi-model or production-scale environments. In this sense, future work will explore the following directions:

- extending the pipeline to support multilingual prompt inversion through cross-lingual syntactic modeling;
- integrating fine-grained stylistic or affective embeddings beyond sentiment polarity;
- incorporating explainability mechanisms to trace how each constraint influences prompt reconstruction;

Overall, this study lays the foundation for interpretable and controllable prompt inversion in LLMs, with potential applications in security, transparency, and alignment of AI systems.

Acknowledgments

This work was partially supported by project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU.

References

1. Chen, J., Xu, W., Ding, Z., Xu, J., Yan, H., Zhang, X.: Advancing prompt recovery in nlp: A deep dive into the integration of gemma-2b-it and phi2 models. In: 2024 IEEE 6th International Conference on Power, Intelligent Computing and Systems (ICPICS). pp. 934–939. IEEE (2024)
2. Chen, Z., Feng, Y., He, C., Deng, Y., Pu, H., Li, B.: Ipad: Inverse prompt for ai detection—a robust and explainable llm-generated text detector. arXiv preprint arXiv:2502.15902 (2025)
3. Gallagher, S.K., Ratchford, J., Brooks, T., Brown, B.P., Heim, E., Nichols, W.R., Mcmillan, S., Rallapalli, S., Smith, C.J., VanHoudnos, N., et al.: Assessing llms for high stakes applications. In: Proceedings of the 46th International Conference on Software Engineering: Software Engineering in Practice. pp. 103–105 (2024)
4. Li, H., Klabjan, D.: Reverse prompt engineering. arXiv preprint arXiv:2411.06729 (2024)
5. Lin, Y., Zhang, Q., Cai, Q., Hong, J., Ye, W., Liu, H., Duan, B.: An inversion attack against obfuscated embedding matrix in language model inference. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing. pp. 2100–2104 (2024)
6. Morris, J.X., Zhao, W., Chiu, J.T., Shmatikov, V., Rush, A.M.: Language model inversion. In: The Twelfth International Conference on Learning Representations (2024)
7. Sahoo, P., Singh, A.K., Saha, S., Jain, V., Mondal, S., Chadha, A.: A systematic survey of prompt engineering in large language models: Techniques and applications. arXiv preprint arXiv:2402.07927 (2024)
8. Schmidt, D.C., Spencer-Smith, J., Fu, Q., White, J.: Towards a catalog of prompt patterns to enhance the discipline of prompt engineering. ACM SIGAda Ada Letters **43**(2), 43–51 (2024)
9. Sha, Z., Zhang, Y.: Prompt stealing attacks against large language models. arXiv preprint arXiv:2402.12959 (2024)

10. White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., Schmidt, D.C.: A prompt pattern catalog to enhance prompt engineering with chatgpt. arXiv preprint arXiv:2302.11382 (2023)
11. Yu, X., Qi, Y., Chen, K., Chen, G., Yang, X., Zhu, P., Shang, X., Zhang, W., Yu, N.: Dpic: Decoupling prompt and intrinsic characteristics for llm generated text detection. *Advances in Neural Information Processing Systems* **37**, 16194–16212 (2024)
12. Zhang, C., Morris, J., Shmatikov, V.: Extracting prompts by inverting llm outputs. In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. pp. 14753–14777 (2024)