

ENHANCING PROMPT QUALITY IN LANGUAGE MODELS THROUGH SUPERVISED FINE-TUNING: AN EXPLORATION WITH MISTRAL-7B

Mahrugh Rafique^{*1}, Mukhtar Ali Soomro², Hasna Arshad³, Onib Ur Rehman⁴, Isma ul Hassan⁵

^{*1}IT intern Cppa

²PhD Student HITEC University Taxila

³Assistant Prof HITEC University Taxila

⁴Student HITEC University Taxila

⁵Assistant Professor National University of Computer and Emerging Sciences Islamabad

^{*1}mahrughrafique786@gmail.com, ²23sp-phd-cs-003@student.hitecuni.edu.pk,

³hasna.arshad@hitecuni.edu.pk, ⁴onib.rehman@gmail.com, ⁵isma.hassan@nu.edu.pk

DOI: <https://doi.org/10.5281/zenodo.22829220>

Keywords

Prompt Engineering, Large Language Models, Supervised Fine-Tuning, Mistral-7B, G-Eval, Natural Language Processing, Direct Preference Optimization, Automated Prompt Generation.

Article History

Received: 01 June 2026

Accepted: 23 August 2026

Published: 18 September 2026

Copyright @Author

Corresponding Author: *

Mahrugh Rafique

Abstract

When it comes to the field of Natural Language Processing (NLP), the input prompts are a critical factor that influences how the human user interacts with the Large Language Model (LLM). The formulation of prompts has changed from basic, rule-based to sophisticated, context-sensitive prompts. But existing techniques have one drawback: the need for a lot of domain knowledge and manual prompting in creating the prompts. This gap between the theoretical capabilities of LLMs and their actual use by everyday users without AI expertise is a significant hurdle that needs to be addressed.

This paper proposes an automated prompt generation and refinement system that uses Supervised Fine-Tuning (SFT) of the Mistral-7B large language model to close this gap. The proposed system acts as a smart mediator between the user and the target LLM, taking input from the user, converting it into a well-constructed, high-quality prompt following best practices of prompt engineering, and providing the refined prompt to the target LLM. The system can be used as a standalone application or as a middleware element that can be placed into any human-AI interaction pipeline, thereby making the job of prompt optimization easier for the end user and restoring accessibility and usefulness to the system.

To fine-tune, a custom 2500 paired raw and refined prompts were created. The proposed system's effectiveness is quantitatively assessed with the specialized evaluation framework G-Eval, which consists of GPT-4 and the chain of thought evaluation protocol, with scores ranging from 1 to 100 for three dimensions: completeness, clarity, and overall quality. Experimental results show that the fine-tuned Mistral-7B SFT model outperforms the unmodified Mistral-7B base model and a Direct Preference Optimization (DPO) fine-tuned model for all elements of the evaluation in terms of completeness, clarity, and overall.

This work proposes an architecture to refine prompts for deployment, a curated prompt pair training set, and empirical data showing that supervised fine-tuning of prompt pair data can be an effective approach to automating prompt optimization. The knowledge acquired suggests valuable future research opportunities in NLP related to the creation of models that would better comprehend and react to the complete meaning in natural human speech.

I. INTRODUCTION

In the last few decades, NLP (Natural Language Processing) has made a tremendous leap. From basic language understanding tasks, like keyword extraction and rule-based classification, to the current stage, where state-of-the-art models known as Large Language Models (LLMs) can compose almost identical text to human writing. This trend is one of the most prominent technological changes in the history of Artificial Intelligence, and has changed the way people and organizations interact with information systems [1].

Generative LLMs became a household name when OpenAI's ChatGPT came out in November 2022, and the world has seen their number of use cases grow exponentially in research, creative writing, software development, education, healthcare, and more general assistive applications. ChatGPT, BERT, LLaMA, Mistral, among others, have shown great versatility and can guide conversations toward very specific end points and can prompt responses that are very finely tuned to specific needs. This flexibility, however, is a major practical problem: The quality and specificity of the output of an LLM is highly dependent on the quality and specificity of the prompt to which it responds [2].

Prompts are the instructions and questions that provide the LLM with direction regarding the

nature, size, scope, context and output format of its response [1]. With a growing recognition of the significance of speed and quality, prompt engineering has emerged as a distinct field in the study and practice of AI. Prompt engineering refers to a series of methods used to create a prompt that generates consistent, correct, and contextually suitable LLM responses. These techniques include chain-of-thought prompting, few-shot prompting, role-based prompting, injecting contextual memory, and structuring tasks, among others. These techniques are explainable and trainable, but the ability to recognize the patterns and apply the technique in the correct way requires practice and domain knowledge which not all members of the general population will have.

This gap in expertise is the main issue that spurred the current research. If making the best use of these LLMs requires a technical skill that most users don't have, then its potential is limited. Compare to early computers, where the impact on the social world of PCs wasn't realized until graphical interfaces made command line syntax not necessary. Similarly, if the building of good prompts can be automated, then LLMs can truly be accessible to the widest possible audience, even for those who don't have a technical background.

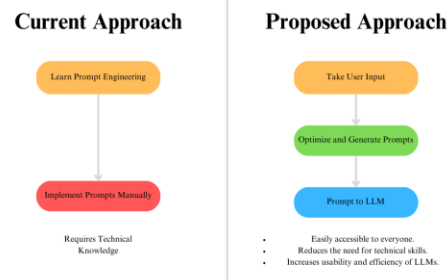


Fig. 1. Current Manual Approach vs. Proposed Automated Prompt Refinement System.

To address this need, the proposed system proposes an automated prompt refinement layer based on a fine-tuned version of the Mistral-7B model, which refines a prompt generated by the user to create a new prompt that is more likely to lead to a satisfactory output from the underlying LLM. The difference between current approach, which requires the user to learn prompt engineering themselves and manually input prompts, and the proposed approach, which automatically works out and implements prompt engineering, but still maintains the intent of the user, is shown in Figure 1.

A. Motivation and Research Problem

While a great deal has been done to enhance the prompt generation techniques, there are still major challenges that have not been overcome. Existing methods are generally not flexible and adaptable to different and complex NLP applications. Prompts that work well in one setting often fail to elicit the desired prompt in another setting, and indicate the need for more responsive/situation-aware approaches [1]. Current systems have a distinct lack of understanding and ability to consider user intent in its entirety: while it may be technically correct, responses often lack the contextual detail and

specificity which humans need in communication. To fill this gap, the need is for methods beyond static prompt templates to dynamic prompt learned optimization.

B. Research Objectives and Questions

- 1. Automated Prompt Generation:** Create a system that automatically generates effective prompts for a variety of NLP tasks, without needing to manually create them or have prompt engineering skills.
- 2. SFT Effect Evaluation:** Assess the effect of using Supervised Fine-Tuning on Mistral-7B on the improved prompt set, with metrics like "completeness" and "clarity" to measure the quality of the responses.
- 3. Prompt Engineering Pattern Identification:** Identify patterns and best practices in prompt engineering that have been identified during the analysis of the improved prompts that lead to more productive human-AI interactions.

The related research questions are as follows:

(RQ1) How to automate prompt generation without compromising effectiveness in different types of NLP tasks?

(RQ2): What is the measurable performance difference between Mistral-7B before and after SFT on the custom refined prompt dataset?

(RQ3) Do you see any patterns or strategy in the refined prompts that are associated with more productive interactions between humans and AI?

C. Major Contributions

There are three key contributions from this research. First, a framework for prompt refinement that can be widely used in various NLP tasks, such as text generation, question answering, code generation, summarization, and translation is proposed. Second, an improvement of LLM interaction quality is evidenced by the careful quantitative assessment using G-Eval, which confirms the significant and stable gains on each of three assessment dimensions. Third, by showcasing that a fine-tuned Mistral-7B model can be used as an effective, scalable and deployable prompt optimization intermediary, it paves the way for more sophisticated and user-friendly NLP applications.

D. Scope and Significance of the Study

This research covers the entire life cycle of an automated prompt refinement system, from building the dataset, training the model, to testing the system quantitatively. The study is intentionally situated at the nexus of three ongoing research areas: prompt engineering, parameter-efficient fine tuning and automated natural language generation evaluation. The research unites the insights and techniques of these areas, and provides a new, integrated, and empirically supported solution that would - until now - have been only superficially handled by research in any one of these areas.

This study has more than just technical implications. As LLM-based tools are increasingly integrated into consumer applications—such as search, productivity, and educational tools—the disparity between what the LLM models can do and what regular consumers can do becomes a growing concern. Such a system, capable of automatically improving the prompt, has the

ability to impact the design paradigm of AI-powered products throughout the industry, moving away from exposing raw prompting interfaces to intelligent, mediated interactions which adapt to the user instead of the user adapting to the system.

Moreover, the comparative analysis of Supervised Fine-Tuning vs. Direct Preference Optimization presented in this paper is methodologically meaningful to the entire fine-tuning community, and can guide which adaptation method would work best on a task characterized by explicit, well-specified input-output transformations, compared to a task that is characterized mainly by subjective quality preferences.

E. Organization of the Paper

The rest of this paper is organized as follows. The second section summarizes the literature in four thematic areas: foundational methods for LLM and NLP, evaluation metrics for NLP, prompting strategies, and fine-tuning methods, which culminates in the identification of the research gap that the study seeks to address. In Section III, the proposed system architecture is described, explaining the choice of models and the fine tuning methods used. The experimental setup is presented in Section IV which includes the dataset construction, training configuration and the evaluation framework. The experimental results are presented and discussed in Section V for both main comparisons. The paper ends with a summary of findings, a discussion of limitations and directions for future work in Section VI.

II. Related Work

This section summarizes some of the previous work in four related fields that form the basis of the proposed approach, including: fundamental LLM and NLP methods, natural language generation evaluation metrics, prompting methods and strategies, and fine-tuning approaches tailored to LLM usage. The review is based on literature published mainly from 2018 to 2024, which were chosen for their methodological relevance and the most recent ones.

A. Large Language Models and NLP

Foundations

Over the last 10 years there have been significant developments in the underlying architectures and training methods used in modern systems that are operating in the field of NLP. In the past decade, the fundamental architectures and training paradigms used for modern NLP systems have progressed significantly. Research on watermarking techniques for content generated by LLM [1] introduced an invisible and algorithmically detectable watermarking scheme that retains high quality of the text while maintaining the ability to protect the IP, however security concerns in more complex and adversarial cases are still under investigation. The former study on Continuous Integration Training (CiT) for vision-language data [2] showed significant data filtering reduction in offline mode, which facilitated the training pipelines for multimodal models faster and with larger scale.

The model adaptation strategy used in this paper is parameter-efficient, which was motivated by the previous work of knowledge distillation for reasoning [3] in which smaller student models were trained to gain competitive reasoning abilities by supervised distillation process from

larger teacher models. The ability to achieve a 50% reduction in the number of parameters while preserving the core linguistic capabilities of large models has been shown by sparseGPT pruning [4] that selectively removes model weights based on an approximation of the loss landscape at second order. Advanced attention mechanisms [5] were also found to enhance the expressiveness and efficiency of NLP models by modifying their core self-attention operation, such as introducing sparse and grouped-query variants, which boosts the scalability of models without loss of expressive power.

Research on NLP embeddings [6] laid the groundwork for comprehending the distributed representations of semantic and syntactic relationships, and research on electronic health record mining [7], time series analysis [8] and clinical machine learning [9] showed that deep learning and NLP techniques could be applied to a wide range of tasks beyond traditional language processing. Detailed analysis of the BERT architecture [10] and the self-attention mechanisms used in LLMs [11] laid the groundwork for all the subsequent fine-tuning and prompting studies, including the current study.

Table I: Summary of Related Work in LLMs and NLP [1-15]

Ref.	Year	Methodology	Strengths	Weaknesses
[1]	2023	Watermark for LLMs	Invisible, detectable watermark; minimal text quality impact	Security concerns in diverse real-world scenarios
[2]	2023	CiT for Vision-Language Data	Speeds training; no offline filtering required	Limited cross-data-type effectiveness explored
[3]	2023	Knowledge Distillation for Reasoning	Improved reasoning in smaller models	Scalability and generalizability concerns remain
[4]	2023	SparseGPT Pruning	50% weight reduction with minimal accuracy loss	Impact on diverse language tasks not fully studied
[5]	2023	Attention Mechanisms in NLP	Enhanced efficiency and scalability of NLP models	Limited assessment in specific NLP applications

[6]	2020	NLP Embeddings Study	Deep theoretical insights into embedding representations	Lacks focus on practical real-world implementation
[7]	2018	Mining EHRs	Useful insights into healthcare data analysis	Privacy concerns and evolving data challenges
[8]	2019	Deep Learning in Time Series	Broad application view across contexts	Limited NLP-specific time series analysis
[9]	2021	ML in Healthcare	Outperforms traditional clinical models	Scope limited to healthcare domain applications
[10]	2021	BERT Exploration	Foundational BERT understanding established	Rapid post-BERT evolution may date findings
[11]	2023	Self-Attention in LLMs	Enhanced understanding of attention mechanisms	Application and scalability questions remain open
[12]	2023	Zero-Shot Prompt Design	Insights into prompt design and optimization	Long-term efficiency and adaptability concerns
[13]	2023	Activation Functions in LLMs	Highlights activation function importance	Exploration of newer activation functions limited
[14]	2019	Normalization in Transformers	Improved Transformer training stability	Effectiveness in varying architectures unexplored
[15]	2023	Sparse Attention in LLMs	Improved processing efficiency demonstrated	Trade-offs in accuracy and comprehensiveness noted

B. Evaluation Metrics for Natural Language Generation

The task of measuring the quality of natural language generation (NLG) output is far more complex than evaluating classification or extraction tasks because high-quality generated text can have many valid forms and cannot be assessed just by a lexical overlap. While n-gram overlap is used as the basis for traditional metrics like BLEU and ROUGE, these metrics have been demonstrated to poorly correlate with human quality evaluations in open-ended generation tasks, where the output needs to be coherent, creative and context sensitive [16].

G-Eval [17] is a major breakthrough in NLG evaluation that introduces a new evaluation agent based on GPT-4 to evaluate generated text using a structured chain-of-thought protocol to evaluate multiple dimensions of quality. This is significantly more correlated with human quality ratings than any reference based measure, and is also reference free, so it can be used where there is no reference text available. The properties guarantee that G-Eval is an ideal tool for assessing the prompt refinement task studied in this paper, where the goal of the prompt is simply to get the LLM to produce better results than the initial raw prompt, not to replicate any specific reference prompt and output.

Table II: Summary of Related Work on NLG Evaluation Metrics

Ref.	Year	Methodology	Strength	Weakness
------	------	-------------	----------	----------

[16]	2024	Evaluation metrics for NLG quality measurement	Surveys current metrics and known limitations comprehensively	Costly and computationally challenging to apply at scale
[17]	2023	G-Eval NLG using GPT-4 with chain-of-thought	Superior human alignment; reference-free; chain-of-thought process	Dependent on LLM quality and potential LLM evaluation biases

C. Prompting Techniques and Prompt Engineering Methods

Prompt engineering has quickly evolved from an informal art to a well-documented and formalized field that has its own patterns and techniques. A systematisation of this knowledge was proposed by White et al. [18] who constructed a comprehensive catalog of reusable prompt patterns for various domains and task types that provides a principled basis on which to build prompts, which is followed for the refinement strategy used in this paper. Based on this, AutoHint [19] further enhanced this work by introducing an automatic prompt optimization framework that uses model error to generate task-specific prompts without the need for any labeled data to achieve a significant improvement in zero-shot and few-shot performance.

Dynamic Prompting [20] proposed a framework to dynamically adapt the content of prompts to

contextual feedback for better response relevance in interactive and streaming situations. Few-shot grounded dialogue generation [21] proved that with a carefully crafted set of minimal prompts, it is possible to obtain high-quality, context-relevant dialogue outcomes from a small amount of training data. GPT4All [22] created an open-source language model ecosystem of compressed, efficiently deployable language models that can be used for various prompting and fine-tuning tasks. Together, these works represent a space of documented prompt generation methods and directly inspire the automated method of prompt refinement presented in this work, instance-dependent prompt generation methods [23] and generative model-based automatic prompt creation methods [24] and comprehensive prompt engineering surveys [25].

Table III: Summary of Related Work on Prompting Techniques

Ref.	Year	Methodology	Strength	Weakness
[18]	2023	Catalogue of reusable prompt patterns	Improves LLM interactions across multiple domains	May need domain-specific adaptation for specialized use
[19]	2023	AutoHint: automatic prompt optimization with hints	Improves zero-shot and few-shot accuracy significantly	Complex initial setup and integration into pipelines
[20]	2023	Dynamic prompting with real-time contextual feedback	Enhances relevance and model responsiveness	Requires continuous tuning and monitoring in production
[21]	2022	Few-shot grounded dialogue generation via prompts	High-quality dialogues produced with limited training data	Performance limited by quality of few-shot examples used

[22]	2023	Compressed open-source LLM ecosystem (GPT4All)	Ensures broad accessibility and deployment efficiency	Resource-intensive initial training and data curation
[23]	2022	Instance-dependent prompt generation	Improves relevance through prompt customization per instance	Requires significant context-specific data to function well
[24]	2022	Automatic prompt creation via generative models	Automates prompt creation and improves LLM deployment scale	Requires robust generative models and extensive training data
[25]	2023	Comprehensive survey of prompt engineering techniques	Broad overview of techniques and their cross-domain application	May lack specific actionable insights for niche domains

D. Fine-Tuning Techniques in LLMs

Meanwhile, techniques that enable the fine-tuning of pre-trained models for specific downstream tasks have also seen rapid advancements, with the ability to adapt a pre-trained LLM to a specific task with lower computational costs than a from-scratch pre-trained LLM. The data-efficient fine-tuning methods [11] adapt a task via few-shot supervised learning, which significantly decreases the amount of labelled data needed and brings in a risk of overfitting to small datasets. Adapter-based Parameter-Efficient Fine-Tuning (PEFT) methods [12] freeze an existing pre-trained model and add small trainable adapter modules to the pre-trained layers, allowing for many different tasks to be adapted using a relatively small number of additional parameters.

Instruction fine-tuning at scale showed that training on a large set of task instructions—from a

diverse set of tasks—can lead to robust generalization across task types, setting a precedent for instruction guided fine-tuning in this paper. Generalization in fine-tuned LLMs [14] gave an empirical study of the impact of task-specific fine-tuning on downstream performance in different evaluation scenarios, demonstrating significant interdependencies between diversity of datasets and quality of training data. LoRA [15] is a technique used in this work to fine-tune Mistral-7B that decomposes weight update matrices into low-rank factors and trains only these factors while keeping the pre-trained weights fixed, thereby significantly cutting down on the number of trainable parameters by several orders of magnitude and allowing near full fine-tuning performance on consumer-grade hardware.

Table IV: Summary of Related Work on Fine-Tuning Techniques

Ref.	Year	Methodology	Strength	Weakness
[11]	2024	Data-efficient few-shot fine-tuning for LLMs	Reduces data and compute requirements substantially	Risk of overfitting on small or low-diversity datasets
[12]	2023	Adapter-based PEFT for smaller-scale LLMs	Diverse adapter configurations tested across datasets	Computational complexity and scalability challenges remain

[13]	2022	Instruction fine-tuning at scale	Strong generalization across diverse task setups	Extensive fine-tuning requires significant computational resources
[14]	2024	Generalization ability via task-specific fine-tuning	Quantifies how fine-tuning affects LLM performance	Generalization ability varies significantly across different tasks
[15]	2021	LoRA: Low-Rank Adaptation for LLMs	Reduces trainable parameters and CPU memory needs	Small parameter sets may bias model toward overfitting

E. Applications of Automated Prompt Optimization

In addition to the core techniques discussed above, there are a number of application-oriented studies that demonstrate the usefulness of prompt optimization for real-world applications. Automated prompt refinement has been investigated as a strategy to improve student query generation in AI tutoring systems, to lessen the friction of learning a new subject and prompt engineering at the same time. In software engineering settings, refined prompts for code generation tasks have been found to decrease the number of syntactically correct but semantically incorrect code suggestions, as these functional requirements, edge cases and style conventions are no longer left to the implicit assumption of the model.

Prompt refinement systems have been suggested for use in customer service and conversational AI applications as a way to convert informal, colloquial customer questions into structured ones that downstream retrieval-augmented generation systems can handle more accurately, leading to better accurate customer service responses. For scientific and technical writing support, it has been demonstrated that using prompts that make clear which reader they are meant for, the level of technical detail to include, and the type of citations to reference, can help minimize the number of revisions needed when using an LLM to create technical content. These application-driven results are typically published as case studies or industry white papers, and are not

a part of a systematic, empirically validated research program, but they all serve to motivate the need for the rigorous fine-tuning-based automated prompt refinement system developed and evaluated in this thesis.

F. Synthesis of the Reviewed Literature

When taken together, the four thematic areas discussed above suggest a research landscape that has reached some degree of maturity in developing the basic architecture of LLMs, in designing methods to evaluate their performance, in devising methods to prompt them, and in fine-tuning them, but has so far not given rise to a tightly-knitted framework for automated prompt optimization. While there have been significant advancements in foundational LLM research, including the development of more complex and efficient model architectures like Mistral-7B, accessibility is not necessarily achieved by changing model capabilities; an additional layer must be added to handle prompt quality for non-expert users. In contrast to previous studies using lexical overlap measures or limited human evaluation panels, evaluation research has led to more credible and human-centric metrics like G-Eval, which enable robust assessment of the effectiveness of an automated refinement system. Research on prompting has yielded a variety of patterns and strategies, including chain-of-thought prompting, role-based prompting, and contextual scaffolding, detailing the ingredients of a good prompt, but has not yet gone so far as to automatically generate prompts at scale. Research

has been fine-tuned, resulting in efficient adaptation techniques like LoRA which enable the training of large models on customized tasks without requiring huge computational resources, but has not been customized specifically to learning a prompt to prompt transformation function. The present study is uniquely situated at the intersection of these four domains and proposes a streamlined and highly efficient fine-tuning approach to train a competent base model that is able to automatically implement the best practices in prompting, as documented, using a state-of-the-art evaluation framework that thoroughly tests the system's quality. What makes the present contribution unique of all the strands of the literature reviewed, in isolation, is the integrative positioning.

G. Research Gap

The above review revealed a number of gaps in the literature. First, although there have been several studies on documentation and evaluation of prompting procedures for a variety of prompting techniques, no research exists on generating and implementing individualized prompting sequences which transform raw prompts from the user to more appropriately contextually and task driven prompts. A fully automated end-to-end system, which learns this transformation with the help of a dataset of prompt pairs has not been demonstrated before that is fine-tuning based. Second, the generation of prompts by hand is an important aspect of current best practices and the primary domain expertise needed for this is not easily learned by most users, and not available in existing tools. Third, although prompt engineering has been investigated to make the model response better, there is no systematic method in the literature to design a fine-tuning curriculum for the whole model based on prompt engineering principles. This thesis directly tackles all three gaps.

Another issue is the comparative assessment of the strategies for fine-tuning prompts for tasks that are prompt-oriented specifically. The author is not aware of any empirical research on the task of prompt refinement with either of these methods nor of any work comparing the two. Making this comparison is important because two completely opposite assumptions are used in the two paradigms: one assumes an explicit and labeled transformation between the two examples and the other assumes a transformation that is subjectively preferred. Furthermore, the suitability of a paradigm within a task that has a very well defined transformation to produce a paired set of examples is not a priori obvious.

Lastly, current techniques for the assessment of prompt engineering research have mainly been qualitative in nature, either through case studies or small-scale human evaluation panels, which are hard to scale and are prone to high subjectivity and inter-rater discrepancies. A methodological contribution is the use of a large language model evaluation framework that has been demonstrated to be highly correlated with human evaluation and can evaluate a larger number of prompts, at a more consistent and more reproducible manner than did previous prompt engineering studies.

III. Proposed System Architecture

The system is designed as an intermediary that is able to refine prompts for any target LLM. The system doesn't require the user to write a well-formed prompt; it reads the natural, informal, and potentially underspecified user input, and generates a high-quality prompt that complies with well-understood prompt engineering best practices. The user receives the resulting text from the target language model. The user experience is the same as interacting directly with the LLM, and the prompt refinement layer works in the background.

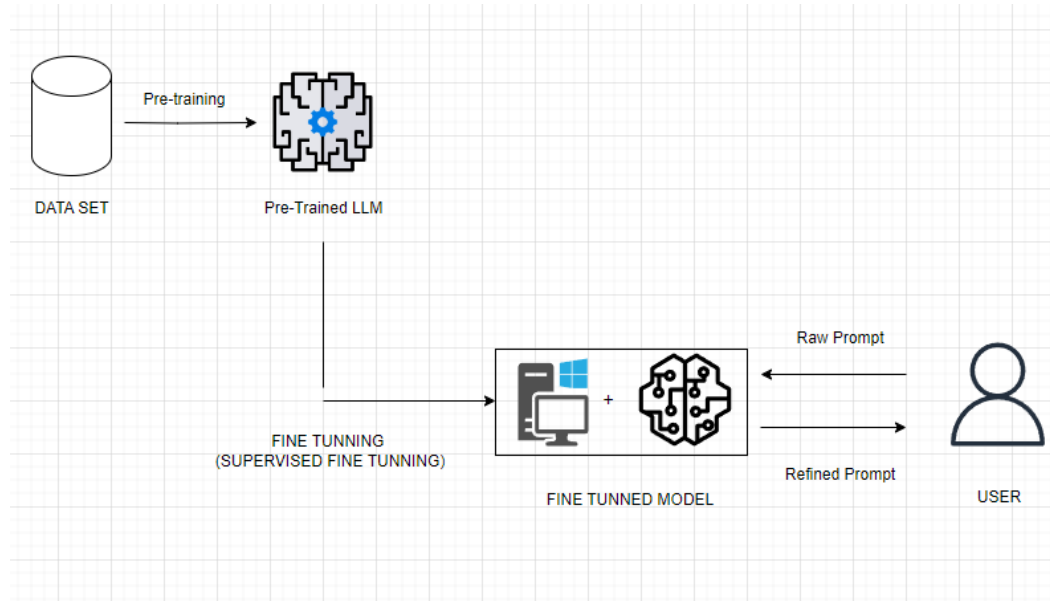


Fig. 2. System Architecture: Pre-Training, Supervised Fine-Tuning Pipeline, and User Interaction Flow.

A. Model Selection: Mistral-7B

The chosen model, Mistral-7B, was chosen due to a combination of technical and practical considerations. Mistral-7B's ability to be efficiently finetuned with iterative updates is further bolstered by its 7 billion number of parameters, which allows for a good balance between representational power and computational feasibility in academic research environments [10]. The model has two attention mechanisms: grouped-query attention and sliding window attention, which enhance the efficiency of the inference compared with the previous transformer architecture of a similar scale.

While Mistral-7B may not have the same level of sophistication or scale as larger models like GPT-3 or LLaMA-65B, it provides significantly faster iteration cycles and reduced deployment expenses without compromising the linguistic depth and complexity needed to accurately interpret and transform user inputs. The model's ability to be trained on big, varied text corpora and to generalize across vastly different styles and content domains of its input is crucial as the raw text prompts in a production system are highly heterogeneous. Moreover, Mistral-7B relies on a vibrant open-source research community and

widely documented fine-tuning procedures, ensuring a current pool of best practices and continual enhancements to the model.

This model was tested against other models like GPT-3.5 and LLaMA-2-7B. GPT-3.5 was not included because it is not directly available for fine-tuning. In addition, LLaMA-2-7B was benchmarked and was found to be underperforming compared to Mistral-7B on initial prompt refinement tasks, where the pretraining data distribution and architectural efficiency may differ. For the above reasons, Mistral-7B was chosen as the best option for this research based on the research objectives and resources available.

B. Supervised Fine-Tuning (SFT)

Supervised Fine-Tuning is one of the crucial steps in the pre-trained language models' development, where they can be significantly fine-tuned to specific tasks or domains. The approach is to receive a task-specific labeled dataset of desired input-output behavior, and then fine-tune the pre-trained model with such data. Each training instance for the prompt refinement task is a pair of a raw user prompt (input) and a high-quality prompt refinement (desired output). This process

enhances the accuracy of the transformation performed by the model and ensures that the outputs are suitable for the context.

The main reason for selecting SFT over other approaches is labeled paired data is available and the transformation desired is specific. Compared to unsupervised methods which are trained on many datasets without explicitly providing an outcome signal, SFT leverages the signal of the target behavior that is present implicitly in the paired prompt set to directly optimize for the target behavior. This direct optimization allows the model to learn the characteristic linguistic changes needed to produce high-quality prompts, including adding contextual framing, specifying output format, and including task-specific constraints.

SFT is also parameter-efficient and uses LoRA (Low-Rank Adaptation), which fine-tunes only a subset of the model's parameters leaving the pre-trained model weights unchanged. This decreases the memory used and the amount of time needed for training significantly without noticeably affecting the capability of the learned prompt refinement, as shown by the experimental results presented in Section V.

C. Direct Preference Optimization (DPO)

A meaningful baseline to compare Direct Preference Optimization (DPO) as an alternative fine-tuning strategy. DPO uses human preference feedback to train the model in such a way that it maximizes the output quality with respect to subjective preference signals without any explicit output labels. It works best for problems situations in which the desired outputs cannot be expressed exhaustively in terms of a set of examples, and where the output is judged as a product of some quality (e.g., coherence, relevance, creativity).

In the context of prompt refinement, DPO fine-tunes the Mistral-7B model by directly optimizing the probability of higher-quality prompt-response pairs being preferred over lower-quality alternatives. In principle, preference data is used, so that DPO does not require labeled pairs of output, but this flexibility comes with the trade-off

of a growing dependence on the quality and representativeness of the preference signal. Based on the comparative experimental results reported in Section V, it is possible to compare directly the suitability of SFT and DPO for the given prompt refinement task automatically.

D. G-Eval Evaluation Framework

For NLG tasks, it is crucial to have sound and dependable evaluation metrics to assess the performance of language models. The study adopts a framework to evaluate NLG systems based on the GPT-4 model with enhanced human alignment, G-Eval [17]. There were a number of reasons why G-Eval was chosen. First, it has been demonstrated that it can do much better than traditional metrics like BLEU and ROUGE in terms of correlation with human evaluation, especially for the task of originality, coherence, and contextual sensitivity. Second, it is reference-free, which means it allows the assessment of the prompt refinement results without having to rely on a ground-truth reference text. Third, it possesses more nuanced and explanatory quality assessments through its chain-of-thought evaluation protocol, as opposed to just point-based metrics.

G-Eval scores each model response on three dimensions: (1) Completeness—whether the response covers all aspects of the prompt; (2) Clarity—readability and comprehensibility of the response; and (3) Overall Score—a quality judgment that combines the three criteria of completeness, clarity, informativeness, and response usefulness. Each dimension is rated on a 100-point scale and can be compared numerically across experimental conditions.

IV. Experimental Setup

A. Dataset Construction

A custom dataset consisting of 2,500 raw and refined prompt pairs was created for fine-tuning. Raw prompts include a wide variety of types and categories of tasks, such as question answering, general information retrieval, summarisation, translation, text generation, and code generation,

and are typed by users in their natural form, often using a single word or a few, and without any strict specifications. A refined prompt is one that contains the same user intent, but in a format that explicitly specifies the task, provides explicit output guidance, frames the context appropriately, and scales down the scope while also giving instructions if needed to get a complete and accurate response from the target LLM.

We used a structured meta-prompting procedure to produce refined prompts, which involves providing a wide range of prompt engineering best practices for each raw input, followed by the Mistral-7B base model. Meta-prompting is a technique that involves structuring the prompting

of the model with a taxonomy of prompting techniques that are organized by task type, such as Chain-of-Thought prompting (encouraging logical step-by-step reasoning), contextual memory injection (providing relevant background before the main request), role-based prompting (providing the model with a persona of an expert), question clarification (identifying ambiguities in the original request), and instruction enhancement (including explicit output format and length specifications). The model is then asked to first detect the category of the task and then use the most relevant techniques before creating the refined prompt.

Table V: Sample Raw and Refined Prompt Pairs from the Training Dataset

Raw Prompt	Refined Prompt (Fine-Tuned Mistral-7B)
give me an advanced python hello world	Please generate an advanced Python "Hello World" script that includes basic error handling, user input to accept a name, and personalized greeting output. Ensure the script adheres to PEP8 guidelines for readability and maintainability.
Teach me the basics of Python	Could you provide a concise introduction to the basics of Python programming, including its syntax, key commands, and core principles? Include practical examples demonstrating variable assignment, conditional statements, loops, and basic function creation.
Why doesn't it thunder during snow storms?	Can you explain why thunder is rarely heard during snowstorms, given that thunder typically accompanies other storm types? Include a scientific explanation of the meteorological conditions involved.
Explain quantum computing in simple terms	Provide a simplified explanation of quantum computing focusing on its basic principles and how it differs from classical computing. Present the information in a way accessible to someone with no prior background in computer science or physics, and include illustrative examples.
how to start hacking	Please provide an introduction to ethical hacking, outlining the basic skills, tools, and legal frameworks required for beginners to start practicing safely and lawfully in authorized environments.

B. Dataset Split and Quality Control

The 2,500-pair data set was split into 3 non-overlapping subsets according to a 90-5-5 split strategy. The training set consisted of 2,250 pairs (90%) which were sufficiently large and diverse to ensure good learning of the refinement function by the model. There were 124 pairs (5%) in the validation set, which was used to track the learning process throughout training, to tune the hyperparameters, and to apply early stopping to avoid overfitting the model. There were 125 pairs (5%) in the test set that was kept aside entirely for the last step of the evaluation process to test the model on unseen prompts.

To ensure the quality of the dataset, this was achieved during the construction of the dataset by omitting refined prompts that were in any way more specific, more contextually rich or longer than the raw prompt. Those pairs that required very little (or no) refinement were eliminated and substituted. The final dataset is balanced across task categories, with approximately 20% of the data for each task type, and 40% of the data for summarization, translation and general information tasks.

C. Model Loading and Quantization

To load the Mistral-7B model, 4-bit quantization was used with the configuration QLoRA, which quantizes the pre-trained model parameters using low-bit weight quantization while keeping the weights of the LoRA adapters trained full-precision. The quantization method can be used to store an 7-billion-parameter model in around 75% less GPU memory, allowing the entire training pipeline to be performed on a single consumer level GPU. Examination of the validation split during training showed that the quality of the responses was not significantly degraded when using 4-bit quantization compared to full-precision loading in the task conditions used in this study.

D. Training Configuration

The SFT training regime was carefully designed to allow for convergence while avoiding catastrophic

forgetting of the pre-trained language model's language skills. The batch size was 2, which was a trivial size, both to generate gradient noise that enhanced the generalization capability in larger models and to fit in the GPU memory. The learning rate was chosen to be $2.5e-5$, intentionally small to ensure that parameters were adjusted slowly to prevent any major changes in the pre-trained representations. A learning rate schedule called cosine schedule was used to decrease the learning rate gradually during the training. Weights of the base model were fixed during training, and the LoRA configuration involved rank 8 and alpha 16, applied to all of the attention projection matrices. The training lasted for 3 epochs and early stopping was performed based on the validation loss. The entire training process has been carried out on Google Colab using a T4 GPU.

E. Prompting Template for Prompt Refinement

A template of a structured meta-prompt is applied in both the training and inference phases in order to represent the model refinement task. The template asks the model to: (1) recognize the type of task embedded in the raw prompt; (2) use an appropriate contextually generated introduction, instruction improvement, and/or clarifying question (if applicable); (3) use advanced prompting from a pre-defined taxonomy based on task type; and (4) return only the refined prompt text, starting with the token "Refined Prompt:". The structured output format guarantees a uniform refinement output that is easy to process in diverse ways for different input types and can be directly used as input for the model without further manipulation.

F. Experimental Design and Evaluation Protocol

The overall experimental design involves two complementary comparisons, which are directed towards different research questions. The first comparison compares the effect of prompt refinement alone by comparing G-Eval scores when the raw prompt is submitted directly to the

fine-tuned Mistral-7B SFT model with scores when the same prompt is first passed to the model, which is then passed to the model for generating a response. The first comparison is designed to assess the effect of prompt refinement itself by comparing the G-Eval scores achieved by submitting the raw prompt directly to the fine-tuned Mistral-7B SFT model with scores achieved by first feeding the prompt into the model before feeding it to the model for the response. This comparison allows for isolating the specific effect of the refinement step, while keeping the underlying model of response generation the same throughout the conditions.

For the second comparison (RQ2, RQ3), the impact of the fine-tuning strategy is assessed by comparing the G-Eval scores of three variants of the model, using the same inputs and prompts from the held-out test set but applying different fine-tuning strategies. This comparison measures the effect of the fine-tuning strategy while keeping the prompt quality the same in all conditions. Enclosed within these two comparisons lie the true explanation of the sources from which quality improvements are made: refinement, fine tuning of the response-generation model; or the merger of the two.

The same G-Eval configuration and prompt template were used for all evaluations to allow for a fair, consistent comparison of the experimental conditions. Each of the 125 test set prompts was scored separately for all experimental conditions that applied and the mean score of the test set for each metric was calculated. The distributions of scores were not visually inspected for statistical outliers and no outliers were removed from the reported results.

V. Results and Discussion

It offers empirical results in two experimental comparisons: raw versus refined prompt performance on the fine-tuned Mistral-7B model; and cross-model comparison between three model variants. G-Eval scores are mean scores across test set of 125 samples, followed by qualitative analysis of refined prompt characteristics.

A. Raw vs. Refined Prompts on Fine-Tuned Mistral-7B (SFT)

The main experiment compares the response quality of the target LLM with raw user prompts against the quality of responses generated by the target LLM after first filtering the prompts with the fine-tuned Mistral-7B SFT model. Table VI shows the mean G-Eval scores on each of the three evaluation dimensions for each condition.

Table VI: G-Eval Scores for Raw Prompts vs. Refined Prompts (Mistral-7B SFT)

Evaluation Metric	Raw Prompt Score	Refined Prompt Score	Absolute Improvement
Completeness	36.50	72.03	+35.53
Clarity	75.07	84.28	+9.21
Overall Score	45.56	76.71	+31.15

The most significant increase is seen with completeness (36.50 to 72.03, an increase of 35.53 points) as refined prompts show significant improvements in the extent that model responses cover all relevant aspects of the user's underlying information need. This is a fundamental problem with raw prompt inputs: No matter how well-

trained an LLM is, it cannot handle underspecified prompts or questions. This ambiguity is addressed by the scope, format, and context of the prompt, which are made explicit in their refinement process, allowing the model to generate truly comprehensive answers.

Measures of clarity also showed meaningful improvements (75.07 to 84.28), suggesting that more refined prompts not only broaden the range of responses, but enhance the organization and accessibility of responses as well. The relatively higher score at baseline clarity (75.07 vs. 36.50 for completeness) indicates that for the raw prompts, Mistral-7B is able to generate a clear output

naturally, while it seems to have a particular problem with completeness when prompted with underspecific instructions. The overall score improvement (from 45.56 to 76.71 points) shows that the refinement process yields overall multidimensional improvements in scores and not just optimizes one dimension at the cost of others.

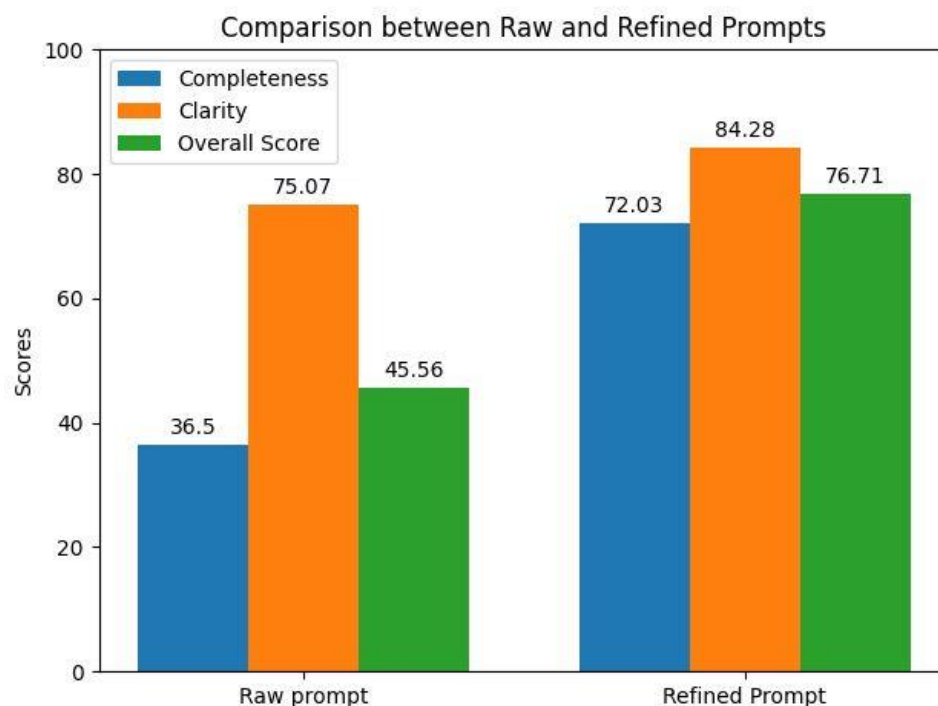


Fig. 3. G-Eval Score Comparison: Raw Prompts vs. Refined Prompts on Fine-Tuned Mistral-7B.

B. Cross-Model Comparative Analysis

To provide a context to the SFT results, two baseline models are considered: the unmodified variant of Mistral-7B and the Mistral-7B fine-tuned using Direct Preference Optimization. During this

comparison, all three models are given the same refined prompts to compare their performance, so any differences in performance are due to the models' performance in response generation rather than due to differences in prompt quality.

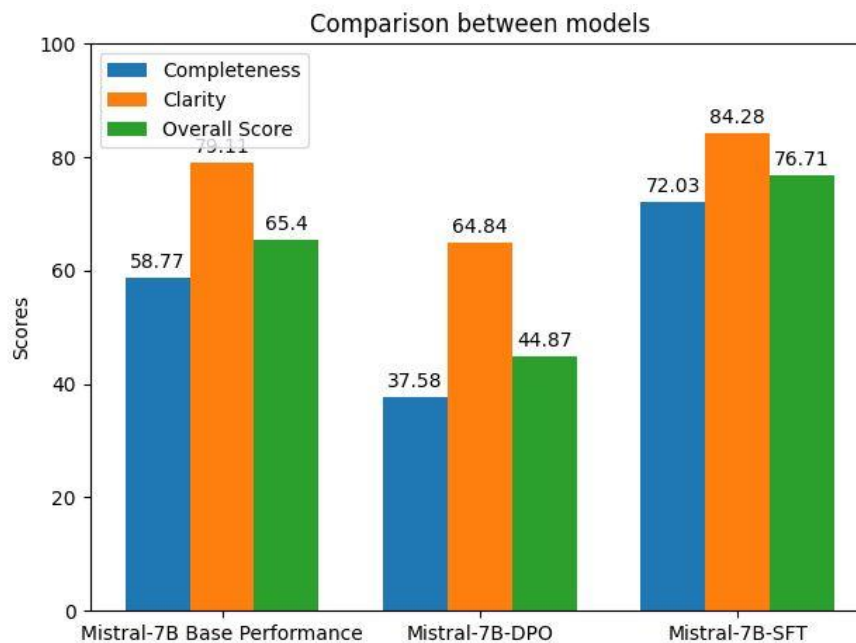


Fig. 4. G-Eval Score Comparison Across Mistral-7B Base, DPO, and SFT Model Variants.

The base Mistral-7B model picked up competitive clarity (79.11) but significantly lower completeness (58.77) and overall score (65.40) than the fine-tuned version of the model with the SFT. This pattern is that pre-trained model generates well-expressed language, but is not always able to generate the correct outcome from a given prompt without further task-specific fine-tuning. Even without fine-tuning, the completeness improvement (58.77 vs. 36.50) of the base model in comparison with the condition of raw prompt suggests that prompt quality is a useful upper bound on the improvement that can be achieved by prompt quality alone, independent of model adaptation.

The reduction in performance on all dimensions (completeness: 37.58, clarity: 64.84, overall: 44.87) is very noticeable for the Mistral-7B DPO variant, and should be interpreted with great caution. The completeness score of the DPO model (37.58) is close to the raw prompt baseline (36.50), indicating that DPO fine-tuning does not enable the model to respond comprehensively to the content of prompts. This could be a disconnect between the preference optimization

task for DPOs and the nature of the prompt refinement task, as the goal is to enhance the response's completeness and specificity instead of its preference-correlated output style. The SFT variant one achieves the best scores on all three dimensions, thus the supervised fine-tuning on paired prompt data is the best adaptation strategy for this task.

C. Qualitative Analysis of Refined Prompts

In addition to the quantitative G-Eval scores, the refined prompts produced by the fine-tuned Mistral-7B SFT model show consistent and interpretable patterns in the model's refinement transformation. The model successfully uses three types of enhancements in most of the items from the test set. The first one is called contextual scaffolding, where the model provides background information to frame what it means to ask for, or the purpose or the domain it fits into, e.g., changing the original prompt to explain quantum computing in simple terms into one that explicitly specifies that the target audience is someone with no physics or computer science background.

The second common pattern is the statement of output format: the model explicitly specifies the desired output in terms of its form, length or presentation style, e.g., asking for comments in the code, a step-by-step explanation, or examples illustrating the solution. This is especially true for the code generation examples in the training set where the specific requirements to handle errors, follow style conventions, and comment with explanations are always given in the prompt, but were nonexistent in the raw prompt. The third pattern is scope and constraint clarification, where the model specifies the scope of the problem to be more narrowly defined, or includes specific constraints that were already not clearly specified in the raw prompt, such as safety in the bedtime routine example, or legal/ethical in the ethical hacking example.

The three patterns described above are very similar to the three patterns which are common in the literature for prompt engineering, as described in Section II, such as the prompt pattern catalog developed by White et al. [18]. Inducing these patterns consistently and appropriately, without being provided a taxonomy at inference time, indicates that the supervised fine-tuning process was able to internalize generalizable prompt engineering principles from the prompt data, and not just memorize surface-level patterns applicable to the training examples.

D. Discussion

The results of the experiments are summarized and three conclusions are drawn. First, the quality of the prompts influences the quality of the responses, with the 35-point gap in completeness of LLM responses between the raw and refined prompt conditions showing that even a very good base model would give very incomplete responses to the informal prompts. The result is consistent with the fundamental idea behind the proposed system, and highlights the significance of a prompt's quality as a factor in LLM utility in practice. Second, the ability to automatically improve prompts with a refined intermediary model is an effective and scalable solution: the

fine-tuned Mistral-7B SFT model successfully learns this transformation between raw and refined prompts, and generalizes it to unseen prompts in the test set, with a performance improvement of more than 35 points in terms of completeness, with no effort required from the user.

Third, the selection of the fine-tuning strategy is crucial: fine-tuning using the paired prompt dataset (SFT) is found to achieve better performances than that of DPO in all evaluation metrics, which shows that the explicit and labeled supervision signal contained in the pair prompt dataset is more informative for the task of prompt refinement than the preference-based signal in DPO. The finding has implications for curriculum design for training prompt optimization systems, and suggests investing in building high quality paired prompt datasets as a key resource for this stream of research.

The proposed system is model-agnostic, meaning that it could be adapted to work with other LLM backends, such as other versions of Mistral, by training the prompt refinement model on data from the target model. It can be seamlessly integrated into human-AI interaction pipelines without changing the LLM, and will introduce only a slight latency overhead for a much higher quality of responses.

E. Threats to Validity

Here, several issues may compromise the internal and external validity of the reported results, and these are explicitly discussed. Internal validity is also supported by the fact that all evaluations are conducted using G-Eval, which may be a proxy of the underlying GPT-4 evaluation model, although validation studies with humans have shown that the correlation between the two is high. However, with regards to internal validity, the assessments are only performed using G-Eval which may be dependent on the underlying GPT-4 assessment model, though previous studies indicated that there is high correlation between human judgment and G-Eval. To partially address this concern, all three model variants and both prompt

conditions were tested with the same G-Eval configurations so that any systematic evaluation bias would be the same across conditions and not be artificially weighted to one condition compared to the other.

External validity: The 2,500 pairs of training data, although widespread in task category coverage, were created with a single approach to prompt design (meta-prompting) and may not adequately reflect the stylistic and linguistic variability of prompts from actual users in varying population groups, native language and technical proficiency. Although the 125-sample test set was adequate for the relatively large effect sizes reported here, it is relatively small compared to production deployment scenarios and it would be helpful to have a larger number of samples in subsequent replications. These potential threats to validity are common limitations in the fine tuning of research under the academic's constraint of limited resources, and don't compromise the overall findings but it does call for caution in overgeneralizing the specific numerical improvements reported here to all possible deployment situations.

F. Ethical Considerations

The creation and use of automated systems that improve prompts bring up ethical issues that should be talked about. First since the process changes what the user writes before it goes to the language model there is a chance that the new version could change what the user really meant, especially for questions that are not clear or involve strong feelings. It is important to design this process making sure the main idea stays the same and not changing the meaning too much. Second the data used to train this system was made up on purpose using a way of creating prompts instead of using real user messages. This keeps privacy issues from happening. Might mean the data does not show how people really use the system.

Third as with any system that helps people talk to AI questions come up about whether people should know that their words are being changed

before they get sent to the AI. The authors say that when they use this work in life they should tell users clearly that there is a step that makes the questions better. This is like ways of being open and honest when AI is involved. No information that could identify a person was used when making the training data. All the testing was done with made-up examples or ones that're already out in the open. This is how people usually do research, in NLP in a way.

VI. Conclusion and Future Work

A. Conclusions

This paper introduced a prompt generation and refining system automatically implemented with the Mistral-7B language model, which is Supervised Fine-Tuned. Since the end users do not have to be prompt engineers, the proposed system automatically converts the raw, underspecified input into well-structured prompts that follow best practices. A custom dataset consisting of 2500 pairs of raw and refined prompts was created, and LoRA was used to fine-tune the model for parameter efficiency. The G-Eval framework was employed to evaluate and found significant and stable enhancements: completeness (+35.53 points), clarity (+9.21 points), and overall score (+31.15 points), with the SFT variant achieving better results in every category than the base model and a DPO-tuned variant.

The results support the key hypothesis: Supervised fine-tuning of prompt pairs data is an effective method for automated prompt optimization. The proposed system not only provides a practical architecture but also offers robust empirical evidence for the research community, addressing the challenge of having high-quality prompts for LLM inputs and offering users intuitive access. This paper provides valuable insights for future research in NLP, particularly regarding the development of more effective and efficient model training and prompt refinement methods, which promise significant advances in the field of natural language understanding and generation.

B. Limitations

There are a few limitations to be noted in the present study. The training set is carefully constructed, but is limited to 2,500 pairs and may not be representative of all prompts and users types used in real-world deployments. The results are architecture (Mistral-7B) specific and might not apply to other model families without retraining. The evaluation framework is based on G-Eval, which has high correlation with human evaluations, but is LLM dependent and can have evaluation biases arising from the underlying GPT-4 model. Other training setups, such as longer training periods or higher rank LoRA models, could have produced additional gains in performance, but were not explored due to computational constraints.

C. Future Work

Future work includes extending the dataset beyond domains, languages, and data from users of different demographic backgrounds. Studying how well does our system generalize to other language models besides Bard to test generalizability of our SFT prompt rewriting system. Modifying or creating new optimization algorithms that require less compute resources so that a wider audience can run our system. Testing the SFT prompt rewriting system on user-facing systems in the wild. For example, we could fine-tune models to serve as personal tutors, giving students additional support. We can also integrate it into customer support chatbots to allow faster support or use it to aid researchers. Once deployed, we can collect user studies to understand how people feel about the assistance. We can measure if users were able to complete their tasks successfully and if they find it easy to use (considering our targeted users may not have machine learning expertise). Last, we can consider applying our approach to multi-turn prompt refinements, using model feedback to iteratively improve the input prompt.

Beyond these extensions a few long-term research ideas are important to mention. One idea is to expand the evaluation system beyond G-Eval to

include feedback from people. This would help compare results from AI models and human opinions. It would also make the findings more reliable and easier to trust. Another idea is to look into mixing ways of training models. This could involve using SFT, which gives instructions and DPO which helps match what humans like. This might be done by training with SFT to get the basics right and then using DPO to make the output more natural and pleasing, to people. A third direction is to look at how the refinement skills learned in this study can be used with completely different large language model architectures. This includes testing on models like GPT-4 or smaller more efficient models that can run on devices, with limited resources. The goal is to find out if the prompt refinement ability we discovered works broadly across models or if it only works because it was trained specifically for Mistral-7B's structure and training data.

I see a direction that looks at how to put user feedback loops right into the refinement pipeline. This lets the system learn from both hidden user corrections over time so the refinement pipeline keeps getting better in a real production setting. A feedback-driven refinement system would go beyond the offline fine-tuning we looked at in this study. It would become a dynamic always improving design that's ready for long-term use. Together these future ideas draw a map toward optimization systems that are smarter stronger and more focused, on the user. They build on the base and real results from this research.

REFERENCES

- [1] P. Liu et al., "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1-35, 2023.
- [2] B. Min et al., "Recent advances in natural language processing via large pre-trained language models: A survey," *ACM Computing Surveys*, vol. 56, no. 2, pp. 1-40, 2023.

-
- [3] R. Lou, K. Zhang, and W. Yin, "Is prompt all you need? No. A comprehensive and broader view of instruction learning," arXiv preprint arXiv:2303.10475, 2023.
- [4] H. Zhang et al., "A survey of controllable text generation using transformer-based pre-trained language models," *ACM Computing Surveys*, vol. 56, no. 3, pp. 1-37, 2023.
- [5] G. M. Hocky and A. D. White, "Natural language processing models that automate programming will transform chemistry research and teaching," *Digital Discovery*, vol. 1, no. 2, pp. 79-83, 2022.
- [6] Z. Ke and B. Liu, "Continual learning of natural language processing tasks: A survey," arXiv preprint arXiv:2211.12701, 2022.
- [7] N. Schopow, G. Osterhoff, and D. Baur, "Applications of the Natural Language Processing Tool ChatGPT in Clinical Practice: Comparative Study and Augmented Systematic Review," *JMIR Medical Informatics*, vol. 11, 2023, e48933.
- [8] Y. Kang et al., "Natural language processing (NLP) in management research: A literature review," *Journal of Management Analytics*, vol. 7, no. 2, pp. 139-172, 2020.
- [9] M. Lai et al., "Evalita 2023: Overview of the 8th evaluation campaign of natural language processing and speech tools for Italian," in *Proc. EVALITA 2023*, CEUR.org, Parma, Italy, 2023.
- [10] A. Neviol et al., "Clinical natural language processing in languages other than English: Opportunities and challenges," *Journal of Biomedical Semantics*, vol. 9, no. 1, pp. 1-13, 2018.
- [11] X. Lin et al., "Data-efficient fine-tuning for LLM-based recommendation," arXiv preprint arXiv:2401.17197, 2024.
- [12] Z. Hu et al., "LLM-Adapters: An adapter family for parameter-efficient fine-tuning of large language models," arXiv preprint arXiv:2304.01933, 2023.
- [13] H. W. Chung et al., "Scaling instruction-finetuned language models," arXiv preprint arXiv:2210.11416, 2022.
- [14] H. Yang et al., "Unveiling the generalization power of fine-tuned large language models," arXiv preprint arXiv:2403.09162, 2024.
- [15] E. J. Hu et al., "LoRA: Low-rank adaptation of large language models," arXiv preprint arXiv:2106.09685, 2021.
- [16] M. Khapra and A. Sai, "A tutorial on evaluation metrics used in natural language generation," in *Proc. 59th Annual Meeting of the ACL*, pp. 15-19, 2021.
- [17] Y. Liu et al., "G-Eval: NLG evaluation using GPT-4 with better human alignment," arXiv preprint arXiv:2303.16634, 2023.
- [18] J. White et al., "A prompt pattern catalog to enhance prompt engineering with ChatGPT," arXiv preprint arXiv:2302.11382, Feb. 2023.
- [19] H. Sun et al., "AutoHint: Automatic prompt optimization with hint generation," arXiv preprint arXiv:2307.07415, Aug. 2023.
- [20] X. Yang et al., "Dynamic prompting: A unified framework for prompt tuning," arXiv preprint arXiv:2303.02909, May 2023.
- [21] C. Zheng and M. Huang, "Exploring prompt-based few-shot learning for grounded dialog generation," arXiv preprint arXiv:2109.06513, 2022.
- [22] Y. Anand et al., "GPT4All: An ecosystem of open source compressed language models," arXiv preprint arXiv:2311.04931, Nov. 2023.
- [23] Z. Wu et al., "IDPG: An instance-dependent prompt generation method," arXiv preprint arXiv:2204.04497, Apr. 2022.
- [24] M. Deng et al., "RLPrompt: Optimizing discrete text prompts with reinforcement learning," in *Findings of ACL: NAACL 2022*, Seattle, USA, Jul. 2022, pp. 30-37.
-

- [25] S. Ekin, "Prompt engineering for ChatGPT: A quick guide to techniques, tips, and best practices," TechRxiv, May 2023. doi: 10.36227/techrxiv.22683919.v2.

