The Impact of Role Design in In-Context Learning for Large Language Models

Anonymous ACL submission

Abstract

In-context learning (ICL) enables Large Language Models (LLMs) to generate predictions based on prompts without additional fine-tuning. While prompt engineering has been widely studied, the impact of role design within prompts remains underexplored. This study examines the influence of role configurations in zero-shot and few-shot learning scenarios using GPT-3.5 and GPT-4o from OpenAI and Llama2-7b and Llama2-13b from Meta. We evaluate the models' performance across datasets, focusing on tasks like sentiment analysis, text classification, question answering, and math reasoning. Our findings suggest the potential of role-based prompt structuring to enhance LLM performance. The code has been released at anonymous GitHub.

1 Introduction

In recent years, the field of Large Language Models (LLMs) has seen remarkable advancements. Models such as GPT-3 (Brown et al., 2020) and Llama (Touvron et al., 2023) have showcased impressive capabilities in various natural language tasks, including question answering (Rouzegar and Makrehchi, 2024b), sentiment analysis (Zhang et al., 2023b), and text classification (Abburi et al., 2023).

A new paradigm known as In-context learning (ICL) has emerged in natural language processing (NLP) (Dong et al., 2022). In ICL, LLMs generate predictions based on provided prompts, which usually include a few training examples. This approach has become a trend for evaluating and extending the abilities of LLMs, allowing them to generalize to new, unseen cases without the need for additional fine-tuning (Dong et al., 2022).

Although several studies have discussed various ICL and prompt engineering methods, the impact of role design in ICL has not been thoroughly explored. Role design involves structuring prompts with distinct roles, such as system instructions, user inputs, and assistant responses. Understanding how these roles influence model performance can provide valuable information for optimizing prompt engineering strategies.

In this paper, we examine the impact of role design in zero-shot and few-shot experiments using two prominent instruction-tuned LLMs (Zhang et al., 2023a) such as GPT-3.5 and GPT-4o from OpenAI and Llama2-7b and Llama2-13b from Meta. We systematically evaluate the models' performance across a range of natural language tasks, including sentiment analysis, text classification, question answering, and mathematical reasoning. Our evaluation employs F1 scores as the primary metric to assess output accuracy and Structural Accuracy as a secondary metric to measure adherence to prompt instructions. This approach allows us to quantify the effectiveness of various pre-defined role configurations in enhancing both the models' predictive accuracy and structural reliability.

This study examines the impact of role-based prompt design across various datasets and LLMs, including tasks that involve complex reasoning. Our findings offer insights into optimal prompt design strategies, potentially improving LLM performance across various natural language processing and reasoning tasks. We observed that incorporating pre-defined roles without altering the main prompts often led to better overall performance. For more complex reasoning tasks, we implemented additional steps in our prompt design to address the increased complexity. The source code, implementation details, and all datasets used in our experiments are available in an anonymous GitHub repository to ensure the reproducibility of our results. [1]

[1] GitHub Code

2 Related Works

Significant research has been conducted in the field of prompt engineering and in-context learning. ICL refers to the ability of LLMs to generate predictions based on their input without additional fine-tuning. (Dong et al., 2022) While some research has focused on training and fine-tuning to improve ICL (Gu et al., 2023; Li et al., 2024; Shi et al., 2023), it generally operates at the inference level of LLMs, relying on prompt engineering rather than model modification. (Bhattamishra et al., 2023)

ICL typically involves providing few-shot examples, and several aspects can be considered in this process: selection, formatting, and instruction. Selection methods focus on choosing optimal examples from the training data for few-shot learning. (Dong et al., 2022) These can include techniques based on k-nearest neighbors (KNN) (Liu et al., 2021), using mutual information (Sorensen et al., 2022), perplexity estimation (Gonen et al., 2022), or reinforcement learning (RL) (Zhang et al., 2022a), for example selection.

Instruction engineering is another crucial aspect of ICL. Instruction engineering in ICL has seen various approaches: Instruction Induction generates instructions from input-output pairs (Honovich et al., 2022), APE optimizes instructions by treating them as programs (Zhou et al., 2022b), and SELF-INSTRUCT improves instruction-following by using self-generated examples (Wang et al., 2022b). These methods aim to enhance LLMs' ability to understand and follow complex directions.

Formatting plays a significant role in both ICL and broader prompt engineering techniques. Methods like Chain-of-Thought (CoT) prompting (Wei et al., 2022) provide reasoning demonstrations, allowing models to develop reasoning abilities naturally when given sufficiently complex tasks. Variants such as complex CoT (Fu et al., 2022) and Auto-CoT (Zhang et al., 2022b) further refine these demonstrations under different conditions.

The Self-Ask method (Press et al., 2022) improves performance by decomposing complex questions into simpler sub-questions, which the model answers sequentially. The Memory-of-Thought (MoT) (Li and Qiu, 2023) approach involves pre-thinking on an unlabeled dataset and storing high-confidence thoughts as external memory, which the model can recall during testing to aid reasoning.

Super In-Context Learning (SuperICL) (Xu et al., 2023) combines large language models with locally fine-tuned smaller models to improve performance on supervised tasks. These smaller models act as plug-ins, providing specific task knowledge while the larger models focus on general language understanding.

Iterative Context-Aware Prompter (iCAP) (Wang et al., 2022a) explores an iterative prompting framework that progressively elicits relevant knowledge from LLMs for multi-step inference. Similarly, the LEAST-TO-MOST Prompting method breaks down complex problems into a series of simpler subproblems and solves them in sequence (Zhou et al., 2022a). The Task-Agnostic Prefix Prompt (TAPP) (Ye et al., 2024) (Ye et al., 2024) is a fixed prompt anchored to the beginning of every input, regardless of the target task, which facilitates zero-shot generalization.

Lastly, recent work has explored role-play prompting to enhance LLMs' reasoning abilities (Kong et al., 2023). They introduced a role-play prompting methodology, evaluating its performance in zero-shot settings across various reasoning benchmarks. Their study experimented with new roles, such as doctor and writer, demonstrating improvements over standard zero-shot approaches in tasks.

3 Methodology and Experimental Setup

3.1 Datasets

Following (Min et al., 2022) and (Rouzegar and Makrehchi, 2024a), our experiments were conducted using samples from a diverse range of datasets to evaluate the performance of different prompt designs across various natural language tasks. To explore more challenging reasoning tasks, we also added a dataset of math word problems. Our experiments included the following datasets:

- commonsense_qa: (Talmor et al., 2019) A dataset designed for common-sense question answering.
- ai2_arc: (Sheng and Uthus, 2020) A benchmark dataset used for evaluating question-answering capabilities.
- wiki_movie_plots: (Abbrivia.com, 2021) A dataset containing movie plots used for genre classification.
- IMDB_reviews: (Npathi, 2021) A sentiment analysis dataset comprising movie reviews

with corresponding sentiment labels.

- MATH Dataset: (Hendrycks et al., 2021) A dataset to test reasoning skills with mathematical word problem questions.

3.2 Prompt Designs

We utilized four LLMs for our experiments: GPT-3.5-turbo-0125 and GPT-4o from OpenAI, and Llama2-7b-chat and Llama2-13b-chat from Meta. These models were selected because of their robust performance and widespread applicability in various NLP tasks. (Brown et al., 2020) (Touvron et al., 2023)

To investigate the impact of role design, we experimented with five basic prompt configurations across all tasks. In role design, the "system" provides high-level instructions that guide the overall task and set the context for the interaction. This includes specifying the format, the rules for the responses, or any additional context necessary for the task. The "user" presents specific queries or prompts that the model needs to respond to, effectively driving the interaction and simulating real-world usage scenarios. The "assistant" is the model response to user queries, which should adhere to the system's instructions and accurately respond to user queries (Figure1). Our prompt configurations include:

- Zero-shot User Prompt (ZeroU): The entire prompt is provided as a user instruction without any role distinctions.
- Zero-shot System and User Prompt (ZeroSU): The initial part of the prompt is provided as a system instruction, followed by the user input.
- Few-shot User Prompt (FewU): Multiple examples of questions and answers are provided within the user prompt.
- Few-shot System and User Prompt (FewSU): Examples are provided with clear distinctions between system and user roles.
- Few-shot System, User, and Assistant Prompt (FewSUA): Includes examples with system instructions, user inputs, and assistant responses.

For the more complex math reasoning tasks, we extended our approach by implementing four sets of these five designs, each set incorporating additional prompt engineering steps:

1. Basic Math: Applied our standard five configurations, tailored for math questions.
2. Specialized Math: Refined the system instructions in each configuration to explicitly request solving the math question and providing the answer as a letter.
3. Explanation Request: Added a request for the model to explain its rationale after providing the answer in each of the five configurations.
4. Reasoning-First: Modified the order of operations in each configuration, asking the model to explain its reasoning process before providing the final answer.

This approach resulted in a total of 20 different prompt designs for the math reasoning tasks, five basic configurations multiplied by four refinement steps. By maintaining the five role-based structures across each refinement step, we were able to examine both the effects of role-based prompting and the impact of task-specific instruction refinements on model performance.

4 Experimental Setup

4.1 Data Preparation and Model Configuration

For each dataset, we selected a balanced subset to ensure a comprehensive evaluation. For classification tasks (such as genre classification and sentiment analysis), we ensured a balanced distribution of categories and labels. For the question-answering tasks and math word problems, we maintained an equal distribution of answers across the multiple-choice options. For the few-shot scenario, we randomly selected three samples from each dataset, ensuring that these examples were not included in the test set.

All GPT and Llama models were configured with a temperature of 0.0. This setting minimizes the randomness in the model's output, ensuring more deterministic responses. Other parameters, such as maximum sequence length and batch size, were optimized for each task to ensure optimal performance.

4.2 Prompt Construction

As demonstrated in Figure 1, we maintained a consistent format for prompt design in all five data sets, with minor configurations for each specific

task. The core structure of each prompt consisted of two main parts:

1. High-level instruction: This part provided the general task description and output format. For example, for the movie genre classification task, it read: "Determine the genre of the movie based on the provided plot. For the plot provided, classify its genre as a single word (without other marks or words like 'genre:'), either "comedy", "action", "drama", or "horror"." This instruction was modified for each dataset with simple word changes, such as "Determine the answer of the example based on the provided question" for question-answering tasks, with the output format adjusted accordingly (e.g., "a single small letter (without other marks or words like 'answer:'), either "a", "b", "c", "d", or "e".").

2. Task-specific input: This part contained the actual content to be analyzed, such as the movie plot, review text, or question, always provided in the user prompt or at the end of the user prompt in few-shot experiments.

Prompt Designs
1. Zero-shot User (ZeroU): • All content in a single user prompt.
2. Zero-shot System and User (ZeroSU): • High-level instruction as system prompt. • Task-specific input as user prompt.
3. Few-shot User (FewU): • Examples and task-specific input in a single user prompt.
4. Few-shot System and User (FewSU): • High-level instruction as system prompt. • Examples and task-specific input as user prompt.
5. Few-shot System, User, and Assistant (FewSUA): • High-level instruction as system prompt. • Examples split between user and assistant prompts. • Followed by task-specific input.

Table 1: Summary of Prompt Designs

As outlined in Table 1, we implemented five distinct prompt designs for each dataset. These designs range from zero-shot to few-shot approaches, with varying degrees of role distinction. The zero-shot designs include ZeroU, which contains all content in a single user prompt, and ZeroSU, which separates high-level instructions into a system prompt. The few-shot designs incorporate examples, with FewU presenting these alongside task-specific input in a single user prompt. FewSU builds on this by distinguishing between system and user roles, while FewSUA further delineates roles by splitting examples between user and assistant prompts.

4.3 Adaptive Prompt Design

In addition to our standard prompt designs, we implemented three more steps of prompt engineering for the MATH dataset, which represents a more complex reasoning task compared to the other question-answering datasets. The progression of these steps was as follows:

1. Basic Math Prompt: This step used a prompt similar to our other datasets, with minor adjustments for math questions. The system instruction was: "Determine the answer of the example based on the provided question. For the question provided, classify its answer as a single small letter (without other marks or words like 'answer:'), either "a", "b", "c", "d", or "e"."

2. Specialized Math Prompt: We refined the system instructions to more explicitly match the math problem-solving process. The new instruction was: "Solve the following math question and determine the correct answer. Provide the answer as a single small letter (without any additional marks or words like 'answer:'), choosing from: "a", "b", "c", "d", or "e"." This change aimed to better align the prompt with the mathematical nature of the task.

3. Explanation Request: In this step, we added a request for the model to explain its rationale after providing the answer. The instruction was extended to: "After providing the letter answer, explain your rationale and how you arrived at the solution." In the few-shot examples for this configuration, we used the format of an answer followed by an explanation.

4. Reasoning-First Approach: In the final configuration, we modified the order of operations, asking the model to explain its reasoning process before providing the final answer. The instruction was changed to: "Solve the following math question and determine the correct

Dataset	LLM	ZeroU		ZeroSU		FewU		FewSU		FewSUA	
		Str. Acc.	F1 Score	Str. Acc.	F1 Score	Str. Acc.	F1 Score	Str. Acc.	F1 Score	Str. Acc.	F1 Score
commonsense_qa	GPT-3.5	24	68	60	68	20	69	55	68	100	**73**
	GPT-4o	100	77	100	80	99	79	100	82	100	**83**
	Llama2-7b	0	**19**	0	**19**	0	18	0	**19**	67	9
	Llama2-13b	0	33	0	**36**	0	**36**	0	**36**	99	28
ai2_arc	GPT-3.5	73	76	39	80	60	80	87	78	99	**85**
	GPT-4o	99	96	100	96	100	95	100	96	100	**97**
	Llama2-7b	0	36	0	39	0	26	0	25	9	**40**
	Llama2-13b	0	**50**	0	52	0	39	0	37	77	48
wiki_movie_plots	GPT-3.5	99	76	99	**79**	99	77	99	76	99	77
	GPT-4o	100	80	100	81	100	81	100	82	100	**84**
	Llama2-7b	0	**75**	0	**75**	0	68	0	70	25	74
	Llama2-13b	0	73	0	75	0	72	0	75	64	**85**
IMDB_reviews	GPT-3.5	100	**94**	100	93	100	93	100	67	100	**94**
	GPT-4o	100	95	100	95	100	96	100	92	100	**97**
	Llama2-7b	1	**87**	1	87	0	62	0	60	18	85
	Llama2-13b	3	91	3	90	0	64	0	67	82	**93**

Table 2: Performance results of various language models using different prompt designs on various datasets. The table includes Structural Accuracy (Str. Acc.) and F1 scores for each method: ZeroU (Zero-shot User-only), ZeroSU (Zero-shot System and User), FewU (Few-shot User-only), FewSU (Few-shot System and User), and FewSUA (Few-shot System, User, and Assistant).

answer. Explain your rationale and how you came to the solution. After explaining how you reached the answer, provide the answer as a single small letter (without any additional marks), choosing from: "a", "b", "c", "d", or "e"."

By implementing these progressive refinements in prompt design for math reasoning tasks, we created a spectrum of instruction complexities. This ranged from basic answer selection to detailed reasoning processes. This approach enabled us to assess how different cognitive demands embedded within the prompts influence the model performance in complex problem-solving scenarios. In addition, it provided information on how models adapt to increasingly sophisticated instruction sets while maintaining consistent role-based structures.

4.4 Evaluation Metrics

The primary evaluation metric was the F1 score, which checks if the output of the prompt matches the class for sentiment analysis and movie genres or the correct answer on QA tasks. The second metric is structural accuracy, which measures how closely the output matches the requested structure in the prompt (Figure 2). For example, the movie genre should be a single word among the specified genres, the sentiment should be a single word, either "positive or "negative," and the QA answers should be a single capital letter among the choices (A, B, C, or D). (Figure 3 and Figure4)

We examined structural accuracy regardless of the correctness of the answer. For instance, if the output was a single letter (A to D) for ai2_arc, it was considered structured. However, outputs like "the answer to this question is D: state park" or simply "D: state park" were considered unstructured because they did not match the desired format. Additionally, for measuring the F1 score, we used post-processing to extract the label from the output and determine if it matched the true label of the dataset.

5 Results

The results of our experiments are summarized in Tables 2 and 3, where we present the F1 scores and structural accuracy (Str. Acc.) for various prompt designs across five datasets, including the more

Prompt Design	LLM	ZeroU		ZeroSU		FewU		FewSU		FewSUA	
		Str. Acc.	F1 Score	Str. Acc.	F1 Score	Str. Acc.	F1 Score	Str. Acc.	F1 Score	Str. Acc.	F1 Score
Basic Math	GPT-3.5	0.03	0.24	0.03	**0.38**	0.62	0.24	0.80	0.24	1.00	0.26
	GPT-4o	1.00	0.33	1.00	0.35	1.00	0.36	1.00	0.37	1.00	**0.40**
Specialized Math	GPT-3.5	0.27	0.31	0.00	**0.53**	0.96	0.23	0.25	0.29	1.00	0.24
	GPT-4o	0.74	0.39	0.63	**0.42**	0.90	0.35	0.99	0.35	1.00	0.41
Explanation Request	GPT-3.5	0.00	0.44	0.00	**0.47**	0.63	0.25	0.02	**0.47**	0.99	0.25
	GPT-4o	0.29	**0.49**	0.30	**0.49**	1.00	0.31	1.00	0.32	0.99	0.40
Reasoning-First	GPT-3.5	N/A	0.47	N/A	0.51	N/A	0.55	N/A	0.50	N/A	**0.58**
	GPT-4o	N/A	0.45	N/A	0.46	N/A	0.49	N/A	0.48	N/A	**0.53**

Table 3: Performance results of GPT-3.5 and GPT-4o models using different prompt designs on the MATH dataset. The table includes Structural Accuracy (Str. Acc.) and F1 scores for each method: ZeroU (Zero-shot User-only), ZeroSU (Zero-shot System and User), FewU (Few-shot User-only), FewSU (Few-shot System and User), and FewSUA (Few-shot System, User, and Assistant). The prompt designs progress from Basic Math to Reasoning-First, reflecting increasing complexity in instruction structure.

complex MATH dataset. The experiments reveal several key insights regarding the impact of role design and prompt engineering in LLMs.

5.1 General NLP Task Performance

For the general NLP tasks (Table 2), in most cases, the few-shot system, user, and assistant prompt (FewSUA) configuration exhibited a superior F1 score across the experiments, indicating that incorporating clear role distinctions and examples significantly enhances model performance.

For Llama models, the use of few-shot user (FewU) and few-shot system and user (FewSU) commands led to a noticeable decrease in F1 scores. This suggests that embedding few-shot examples within user prompts may cause these models to produce more incorrect answers and hallucinations.

The structural accuracy of GPT models was generally high, with both models adhering to the desired output structure in most cases. However, the Llama models frequently generated more than one word, failing to follow the instructions, which is understandable given their smaller size. A standout finding is that the FewSUA prompt configuration helped the Llama models generate outputs in the desired structure, making it potentially useful for applications like chatbots where maintaining a specific structure is crucial. In addition, Llama2-13b outperformed Llama2-7b in structural adherence.

Furthermore, in tasks such as movie genre classification and sentiment analysis, the Llama2-13b model showed results comparable to GPT models, demonstrating its capability in certain NLP tasks despite its smaller size. However, in question-answering tasks, the performance of the GPT models was significantly better and not comparable to that of the Llama models.

The results from these general NLP tasks highlighted the effectiveness of role-based prompting, particularly the FewSUA configuration. However, they also revealed some limitations, especially for smaller models like Llama. These findings prompted us to explore more refined prompt engineering techniques for more complex reasoning tasks, specifically in the domain of mathematical problem-solving. The following subsection details our approach and findings in this more challenging domain.

5.2 Mathematical Reasoning Task Performance

The results of our experiments on the MATH dataset, as presented in Table 3, reveal intriguing and sometimes unexpected patterns in the performance of GPT-3.5 and GPT-4o across different prompt designs.

In the Basic Math prompt, we observed an unusual trend with the superiority of the ZeroSU configuration, particularly for GPT-3.5 (F1 score of 0.38). The overall performance of other approaches was not as strong as anticipated. This led us to refine our prompt by making minor adjustments to better suit mathematical questions, resulting in the Specialized Math prompt.

The Specialized Math prompt showed a similar trend, with ZeroSU maintaining its edge (F1 score of 0.53 for GPT-3.5). Interestingly, we noticed that configurations with lower structural accuracy often yielded better performance in providing correct answers. Upon analyzing the model responses, we found that better-performing answers often included explanations and reasoning rather than just a single-letter response.

This observation prompted us to test a new hypothesis by incorporating reasoning and rationale into our prompts, leading to the Explanation Request design. In this configuration, we altered the few-shot examples to include explanations for each answer. The results showed that FewSU achieved better F1 scores, again with low structural accuracy. We observed that in some cases, the model made mistakes in attempting to maintain the structure, such as providing an initial answer, followed by an explanation, and then a different final answer (e.g., "a, [explanation], the answer is b").

To address these issues, we further evolved the prompt in the Reasoning-First approach, asking the model to explain its reasoning before providing the final answer. This design aimed to avoid the errors observed in the previous configuration. As a result, we did not evaluate structural accuracy for this prompt (marked as N/A in the table).

For GPT-4o, we observed similar trends, although the results across different configurations were more consistent, which aligns with its status as a more advanced LLM.

Regarding the Llama models (results shown in Table 4 in the appendix), we generally observed poor performance across the board, which is understandable given the complex nature of the mathematical reasoning tasks. However, an interesting exception was observed for Llama2-13b in the FewSUA configuration, which achieved a 31% F1 score, significantly outperforming other configurations for this model.

These findings highlight the complex interplay between prompt design, model capabilities, and task complexity in mathematical reasoning. They also underscore the importance of allowing models to provide explanations and reasoning, which often leads to more accurate responses, even if at the cost of strict adherence to a prescribed output structure.

In general, our experiments reveal that the effectiveness of prompt design varies significantly between task types and model architectures. Although FewSUA configuration excelled in standard NLP tasks, mathematical reasoning required more nuanced approaches, often benefiting from explanatory prompts. Larger models consistently outperformed smaller ones, although Llama2-13b showed promise in certain NLP tasks. In particular, strict structural adherence did not always correlate with better performance, especially in complex reasoning tasks. These findings highlight the need for task-specific and model-specific prompt engineering strategies.

6 Limitations and Future Works

Our study has limitations that present opportunities for future research. We focused on a limited range of tasks and datasets. Future work should expand to include a broader array of NLP tasks, such as summarization and translation, to provide a more comprehensive understanding of the effectiveness of prompt design across different language processing challenges.

While we utilized pre-defined roles for instruction-tuned LLMs, future studies could explore creating new, task-specific roles to further optimize prompt designs. Additionally, investigating how these role-based strategies translate to non-instruction-tuned language models such as GPT-3 and Llama Base models could yield valuable insights into the broader applicability of this approach.

Due to resource limitations, we were unable to experiment with larger Llama models, which may show comparable results to the experimented GPT models in certain tasks. Future research with access to more computational resources could explore the performance of these larger models in role-based prompt designs, potentially revealing new insights into the scalability of this approach across different model sizes and architectures.

Lastly, future research could delve deeper into the relationship between structural accuracy and task performance, potentially leading to more optimized prompt engineering strategies. Exploring adaptive prompt designs that dynamically adjust based on the specific task and model could also prove fruitful.

Conclusion

This study examines the impact of role-based prompt design and task-specific prompt engineering on Large Language Models (LLMs) across various NLP tasks and mathematical reasoning

problems. Our experiments with GPT-3.5, GPT-4o, Llama2-7b, and Llama2-13b revealed that the few-shot system, user, and assistant (FewSUA) prompt configuration generally performed best in NLP tasks. However, in mathematical reasoning, we found that configurations allowing for explanations often outperformed those strictly adhering to structural accuracy.

Significant performance differences were observed between model sizes and architectures, with larger models generally performing better. Interestingly, in some simpler NLP tasks, Llama2-13b achieved results comparable to GPT models.

Our findings suggest that optimal prompt design varies not only between task types but also between models of different sizes and architectures. The success of the Reasoning-First approach in mathematical tasks indicates that encouraging models to articulate their thought processes can lead to more accurate outputs. These insights contribute to the growing understanding of effective prompt engineering for LLMs, offering valuable guidance for researchers and practitioners in the field of natural language processing.

References

Abbrivia.com. 2021. Topicmap: Wiki movie plots deduped. Accessed: 2024-09-25.

Harika Abburi, Michael Suesserman, Nirmala Pudota, Balaji Veeramani, Edward Bowen, and Sanmitra Bhattacharya. 2023. Generative ai text classification using ensemble llm approaches. *arXiv preprint arXiv:2309.07755*.

Satwik Bhattamishra, Arkil Patel, Phil Blunsom, and Varun Kanade. 2023. Understanding in-context learning in transformers and llms by learning to learn discrete functions. *arXiv preprint arXiv:2310.03016*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2022. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Hila Gonen, Srini Iyer, Terra Blevins, Noah A Smith, and Luke Zettlemoyer. 2022. Demystifying prompts in language models via perplexity estimation. *arXiv preprint arXiv:2212.04037*.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Pre-training to learn in context. *arXiv preprint arXiv:2305.09137*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Or Honovich, Uri Shaham, Samuel R Bowman, and Omer Levy. 2022. Instruction induction: From few examples to natural language task descriptions. *arXiv preprint arXiv:2205.10782*.

Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, Xin Zhou, Enzhi Wang, and Xiaohang Dong. 2023. Better zero-shot reasoning with role-play prompting. *arXiv preprint arXiv:2308.07702*.

Xiaonan Li and Xipeng Qiu. 2023. Mot: Memory-of-thought enables chatgpt to self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6354–6374.

Yichuan Li, Xiyao Ma, Sixing Lu, Kyumin Lee, Xiaohu Liu, and Chenlei Guo. 2024. Mend: Meta demonstration distillation for efficient and effective in-context learning. *arXiv preprint arXiv:2403.06914*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.

Lakshmi Narasaiah Npathi. 2021. Imdb dataset of 50k movie reviews. Accessed: 2024-09-25.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2022. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*.

Hamidreza Rouzegar and Masoud Makrehchi. 2024a. Enhancing text classification through llm-driven active learning and human annotation. In *Proceedings of The 18th Linguistic Annotation Workshop (LAW-XVIII)*, pages 98–111.

Hamidreza Rouzegar and Masoud Makrehchi. 2024b. Generative AI for Enhancing Active Learning in Education:a Comparative Study of GPT-3.5 and GPT-4 in CraftingCustomized Test Questions. *Proceedings of the Canadian Conference on Artificial Intelligence*.

Emily Sheng and David Uthus. 2020. Investigating societal biases in a poetry composition system. *Preprint*, arXiv:2011.02686.

Weijia Shi, Sewon Min, Maria Lomeli, Chunting Zhou, Margaret Li, Victoria Lin, Noah A Smith, Luke Zettlemoyer, Scott Yih, and Mike Lewis. 2023. In-context pretraining: Language modeling beyond document boundaries. *arXiv preprint arXiv:2310.10638*.

Taylor Sorensen, Joshua Robinson, Christopher Michael Rytting, Alexander Glenn Shaw, Kyle Jeffrey Rogers, Alexia Pauline Delorey, Mahmoud Khalil, Nancy Fulda, and David Wingate. 2022. An information-theoretic approach to prompt engineering without ground truth labels. *arXiv preprint arXiv:2203.11364*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Boshi Wang, Xiang Deng, and Huan Sun. 2022a. Iteratively prompt pre-trained language models for chain of thought. *arXiv preprint arXiv:2203.08383*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022b. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Canwen Xu, Yichong Xu, Shuohang Wang, Yang Liu, Chenguang Zhu, and Julian McAuley. 2023. Small models are valuable plug-ins for large language models. *arXiv preprint arXiv:2305.08848*.

Seonghyeon Ye, Hyeonbin Hwang, Sohee Yang, Hyeongu Yun, Yireun Kim, and Minjoon Seo. 2024. Investigating the effectiveness of task-agnostic prefix prompt for instruction following. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19386–19394.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023a. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.

Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Jialin Pan, and Lidong Bing. 2023b. Sentiment analysis in the era of large language models: A reality check. *arXiv preprint arXiv:2305.15005*.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022a. Active example selection for in-context learning. *arXiv preprint arXiv:2211.04486*.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022b. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2022a. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

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

A Appendix

The appendix provides detailed descriptions and discussions of the examples of prompts and the outputs from different models, specifically for the wiki_movie_plots dataset. The prompt designs include ZeroU (Zero-shot User-only), ZeroSU (Zero-shot System and User), FewU (Few-shot User-only), FewSU (Few-shot System and User), and FewSUA (Few-shot System, User, and Assistant). Each design isolates the effect of role-based structuring on model performance.

While the specific example in Figure 1 is for the wiki_movie_plots dataset, similar prompts are used for other datasets with minor changes in the desired output, such as sentiment for IMDB reviews or answers for commonsense_qa and ai2_arc. By splitting the prompts into distinct roles, the study observes the impact on the models' ability to generate accurate and structured responses.

Figures 2, 3, and 4 present examples of structured and unstructured responses across different datasets, including sentiment analysis, genre classification, and question-answering tasks. Structured responses are critical for consistency in evaluation, requiring specific formats such as single words or letters.

The post-processing technique mentioned in the text is essential for accurately extracting these labels from the model outputs, highlighted by the blue-colored text in the examples. However, there are instances where the extraction process makes mistakes, as shown in Figure 3. For example, in the GPT-4o (commonsense_qa) example, the post-processing incorrectly extracted 'A' as the answer, although the output indicated 'C'. This highlights the challenges in achieving precise output structures and the importance of improving post-processing techniques to avoid such errors.

The appendix emphasizes the importance of clear role distinctions in prompt design, particularly the FewSUA configuration, which consistently improves both F1 scores and structural adherence. The detailed examples and discussions provide valuable insights for developing effective prompt designs, highlighting the crucial role of post-processing in maintaining output accuracy.

A.1 Resource Utilization

For the experiments involving GPT models, we used the OpenAI API, with the total cost amounting to approximately 150 USD. For the Llama models, we utilized a single GPU with 64GB memory for the 7b model and two parallel GPUs, each with 64GB memory, for the 13b model. This setup ensured that we had sufficient computational resources to conduct the experiments efficiently and obtain reliable results.

Prompt Design	LLM	ZeroU		ZeroSU		FewU		FewSU		FewSUA	
		Str. Acc.	F1 Score	Str. Acc.	F1 Score	Str. Acc.	F1 Score	Str. Acc.	F1 Score	Str. Acc.	F1 Score
Basic Math	Llama2-7	0.00	0.23	0.00	0.24	0.00	0.19	0.00	0.20	0.05	0.21
	Llama2-13	0.00	0.23	0.00	0.23	0.00	0.20	0.00	0.20	0.71	0.21
Specialized Math	Llama2-7	0.00	0.23	0.00	0.25	0.00	0.20	0.00	0.20	0.18	0.23
	Llama2-13	0.00	0.22	0.00	0.23	0.00	0.20	0.00	0.20	0.68	0.21
Explanation Request	Llama2-7	0.00	0.24	0.00	0.26	0.00	0.20	0.00	0.20	0.70	0.22
	Llama2-13	0.00	0.19	0.00	0.20	0.00	0.20	0.00	0.20	0.93	0.21
Reasoning-First	Llama2-7	N/A	0.16	N/A	0.22	N/A	0.20	N/A	0.20	N/A	0.23
	Llama2-13	N/A	0.23	N/A	0.19	N/A	0.20	N/A	0.20	N/A	**0.31**

Table 4: Performance results of Llama2-7 and Llama2-13 models using different prompt designs on the MATH dataset. The table includes Structural Accuracy (Str. Acc.) and F1 scores for each method: ZeroU (Zero-shot User-only), ZeroSU (Zero-shot System and User), FewU (Few-shot User-only), FewSU (Few-shot System and User), and FewSUA (Few-shot System, User, and Assistant). The prompt designs progress from Basic Math to Reasoning-First, reflecting increasing complexity in instruction structure.

ZeroU prompt:

role: **"user"**, content: Determine the genre of the movie based on the provided plot:
For the plot provided, classify its genre as a single word (without other marks or words like 'genre:'), either "comedy", "action", "drama", or "horror".
Plot[i]

ZeroSU prompt:

role: **"system"**, content: Determine the genre of the movie based on the provided plot.
For the plot provided, classify its genre as a single word (without other marks or words like 'genre:'), either "comedy", "action", "drama", or "horror".
role: **"user"**, content: Plot[i]

FewU prompt:

role: **"user"**, content: Determine the genre of the movie based on the provided plot:
For the plot provided, classify its genre as a single word (without other marks or words like 'genre:'), either "comedy", "action", "drama", or "horror".
Examples:
Plot[1] - Genre[1]
Plot[2] - Genre[2]
Plot[3] - Genre[3]
...
Plot[i]

FewSU prompt:

role: **"system"**, content: Determine the genre of the movie based on the provided plot.
For the plot provided, classify its genre as a single word (without other marks or words like 'genre:'), either "comedy", "action", "drama", or "horror".
role: **"user"**, content: Examples:
Plot[1] - Genre[1]
Plot[2] - Genre[2]
Plot[3] - Genre[3]
...
Plot[i]

FewSUA prompt:

role: **"system"**, content: Determine the genre of the movie based on the provided plot.
For the plot provided, classify its genre as a single word (without other marks or words like 'genre:'), either "comedy", "action", "drama", or "horror".
role: **"user"**, content: Plot[1]
role: **"assistant"**, content: Genre[1]
role: **"user"**, content: Plot[2]
role: **"assistant"**, content: Genre[2]
role: **"user"**, content: Plot[3]
role: **"assistant"**, content: Genre[3]
role: **"user"**, content: Plot[i]

Figure 1: This figure illustrates the various prompt designs utilized to feed all models. Each prompt design is defined as follows: ZeroU (Zero-shot User-only), ZeroSU (Zero-shot System and User), FewU (Few-shot User-only), FewSU (Few-shot System and User), and FewSUA (Few-shot System, User, and Assistant).

Sentiment (IMDB reviews):
positive / negative (single word)

Genre (wiki_movie_plots):
comedy, drama, action, horror (single word)

QA (ai2_arc):
A to E (single letter)

QA (commonsense_qa):
A to D (single letter)

Figure 2: This figure illustrates the various structured responses for different datasets used to evaluate the models.

GPT-4o (IMDB reviews):
neutral

GPT-4o (commonsense_qa):
Greg's performance was mediocre. What might his habits be like? A: distinguished B: unique C: peculiar D: exceptional E: outstanding
The answer is: C

GPT-3.5 (IMDB reviews):
science fiction

GPT-3.5 (commonsense_qa):
D: to remove toxic material from the stomach before it is absorbed

Figure 3: This figure shows examples of unstructured responses from GPT models. The blue-colored text highlights the extracted labels in the examples.

Llama2-7b (commonsense_qa):

Based on the question provided, the answer is:
A: halving the distance between them
Therefore, the answer is "A".

Llama2-7b (IMDB reviews):

The sentiment of the review is negative. The reviewer expresses disappointment with the sequel compared to the original, citing a lack of fun and too much time spent on plot development. They also mention that Tim Thomerson's wisecracking is toned down, and the performances are adequate but the script lets them down. The action is described as routine and the plot is only mildly interesting, which suggests that the reviewer was not impressed with the film's action sequences or story. The reviewer also states that they need lots of silly laughs to stay entertained during a Trancers movie, but the film fails to deliver on this front. Overall, the review is negative and suggests that the reviewer did not enjoy the sequel as much as the original.

Llama2-7b (wiki_movie_plots):

Based on the provided plot, I would classify the movie as a comedy. The plot involves a group of irresponsible and wealthy individuals who die in an accident and become ghosts, with the goal of freeing their friend from a regimented lifestyle in order to enter heaven. The movie appears to have a lighthearted and comedic tone, with elements of slapstick humor, witty dialogue, and farcical situations. The characters' antics and misadventures are likely to provide most of the comedic elements in the movie.

Llama2-13b (wiki_movie_plots):

Sure! Based on the plot provided, the genre of the movie would be:
Comedy.
The plot involves a man trying to relive his youth and pursue a beautiful woman, with his friends, in a series of comedic and awkward situations. The tone of the plot is lighthearted and humorous, indicating that the movie would be a comedy.

Llama2-13b (IMDB reviews):

Based on the review, I would rate the movie 3 out of 10. Here's a breakdown of the review:
* The reviewer can't knock the movie too hard because it's clear that the filmmakers were trying to make it bad or campy.
* Some of the parts that were intended to be funny actually are, but only for their cheesiness.
* The reviewer notes that the movie goes over the top near the end with a time warp and sexual innuendoes.
* The reviewer is surprised that the movie got a PG rating, implying that it's not suitable for all audiences.
* The reviewer concludes that it's a fun way to waste a couple of hours, but it's not on the same level as Star Wars or Trek.

Llama2-13b (ai2_arc):

Sure! Based on the question provided, the answer is:
D : wildfire

Figure 4: This figure shows examples of unstructured responses from Llama2 models. The blue-colored text highlights the extracted labels in the examples.