

Prompt Chaining or Stepwise Prompt? Refinement in Text Summarization

Shichao Sun^{1,5}, Ruifeng Yuan¹, Ziqiang Cao⁴, Wenjie Li^{1*}, Pengfei Liu^{2,3,5*}

¹ The Hong Kong Polytechnic University, ² Shanghai Jiao Tong University

³ Shanghai Artificial Intelligence Laboratory, ⁴ Soochow University

⁵ Generative AI Research Lab (GAIR)

cswjli@comp.polyu.edu.hk, pengfei@sjtu.edu.cn

Abstract

Large language models (LLMs) have demonstrated the capacity to improve summary quality by mirroring a human-like iterative process of critique and refinement starting from the initial draft. Two strategies are designed to perform this iterative process: *Prompt Chaining* and *Stepwise Prompt*. Prompt chaining orchestrates the drafting, critiquing, and refining phases through a series of three discrete prompts, while Stepwise prompt integrates these phases within a single prompt. However, the relative effectiveness of the two methods has not been extensively studied. This paper is dedicated to examining and comparing these two methods in the context of text summarization to ascertain which method stands out as the most effective. Experimental results show that the prompt chaining method can produce a more favorable outcome. This might be because stepwise prompt might produce a simulated refinement process according to our various experiments. Since refinement is adaptable to diverse tasks, our conclusions have the potential to be extrapolated to other applications, thereby offering insights that may contribute to the broader development of LLMs.

1 Introduction

Large language models (LLMs) can enhance the summary via iterative refinement (Zhang et al., 2023). This is motivated by how humans refine their written text. The main idea contains three sequential steps: (1) **Drafting**: LLMs generate an initial summary; (2) **Critiquing**: LLMs provide critical feedback and helpful suggestions for its output; (3) **Refining**: LLMs use the feedback to refine the initial summary.

More generally, this refinement can be applied to various text generation tasks to improve the outcomes (Madaan et al., 2023; Gou et al., 2023; Ye

et al., 2023; Akyurek et al., 2023). Moreover, the improved outcomes can also help train a more helpful and harmless model (Huang et al., 2022; Bai et al., 2022; OpenAI, 2023; Scheurer et al., 2023). Implementing this refinement process can be approached in two distinct methods: *Prompt Chaining*¹ and *Stepwise Prompt*.² Prompt chaining undertakes drafting, critiquing, and refining phases through a sequence of three discrete prompts. It means that LLMs will run three times. Although LLMs can concentrate on solving one particular problem without being overwhelmed by the complexity of the multiple tasks, it is trivial and troublesome for humans to provide three comprehensive prompts. Conversely, stepwise prompt completes these three phases within a single generation. Stepwise prompt only needs a simple prompt to contain three sequential steps, but it is challenging for LLMs to generate a long and complex output. Currently, the effectiveness of these two methods remains underexplored in any text generation task.

In this short paper, we compare prompt chaining and stepwise prompt to find the better method for refinement in text summarization. Specifically, we conduct experiments on the dataset **InstruSum** (Liu et al., 2023) introduced to evaluate the capabilities of LLMs. It involves instruction controllable text summarization, which summarizes the article based on the specific requirement. We evaluate the quality of initial summaries, critiques, refined summaries to show the effect of prompt chaining and stepwise prompt. Experimental results indicate that the prompt chaining is better than stepwise prompt. Moreover, various experiments imply that **stepwise prompt might produce a simulated refinement**

¹Prompt chaining is introduced in https://www.promptingguide.ai/techniques/prompt_chaining

²Stepwise prompt is similar to specifying the steps required to complete a task at <https://platform.openai.com/docs/guides/prompt-engineering/tactic-specify-the-steps-required-to-complete-a-task>

* Corresponding Authors.

Work done while visiting GAIR Lab.

process, where LLMs intentionally produce errors only to subsequently correct them. Intuitively, this conclusion will work on other domains and further facilitate future research.

2 Related Works

Recent work has proved that refinement can significantly improve LLMs performance. Self-Refine (Madaan et al., 2023) uses LLMs for drafting outcomes, providing feedback, and refining initial generation. In a series of 7 varied tasks, ranging from dialogue response to mathematical reasoning, outputs created using the Self-Refine method are favored over those produced through one-step generation with the same LLM, as judged by human evaluators and automated metrics. Critic (Gou et al., 2023) proposes to leverage external tools to critique generated text and refine the initial generation via evaluation feedback. SelFee (Ye et al., 2023) collects generations, feedback, and revised generations to finetune LLaMA models (Touvron et al., 2023). Akyurek et al. (2023) propose to train a better critique model to help repair the model outputs. Zhang et al. (2023) introduce a refinement paradigm to enhance the faithfulness and controllability in text summarization. Moreover, the refined outcomes can also help train a more helpful and harmless model (Huang et al., 2022; Bai et al., 2022; OpenAI, 2023; Scheurer et al., 2023).

3 Prompts

Figure 1 illustrates the prompts of prompt chaining and stepwise prompt within the context of instruction controllable text summarization. Prompt chaining requires a human to segment the refinement process into three steps. Each step leverages the output from the preceding one. In contrast, stepwise prompt specifies the same three steps to be executed within a single operation. Therefore, they can generate the equivalent results, including: (1) **Draft Summary** is the initially generated summary. (2) **Critique** is the critical comment and the helpful suggestion. (3) **Refined Summary** stems from refining the draft summary based on the critique. Correspondingly, these outcomes are obtained from each step in prompt chaining or the sequential items in the prompt chaining outcome.

4 Experiments and Results

4.1 Dataset

We conduct experiments on the dataset **InstruSum** (Liu et al., 2023), which is produced to evaluate the capabilities of LLMs to summarize the article based on the specific requirement. InstruSum contains 100 article-requirement pairs in total. The articles contain around 1000-1200 words, stemming from the BBC news website.³ Requirements for a summary are designed to reflect diverse information needs that readers may have at different stages of their reading journey. These requirements include (a) Informational requirement, which supplies pertinent details about the topic or subject being discussed within the articles; (b) Formatting requirement, which enhances the summary’s structure, such as incorporating bullet lists, to improve its readability and facilitate quicker comprehension; (c) Meta requirement, which reflects a high-level overview of the article.

5 Models and Metrics

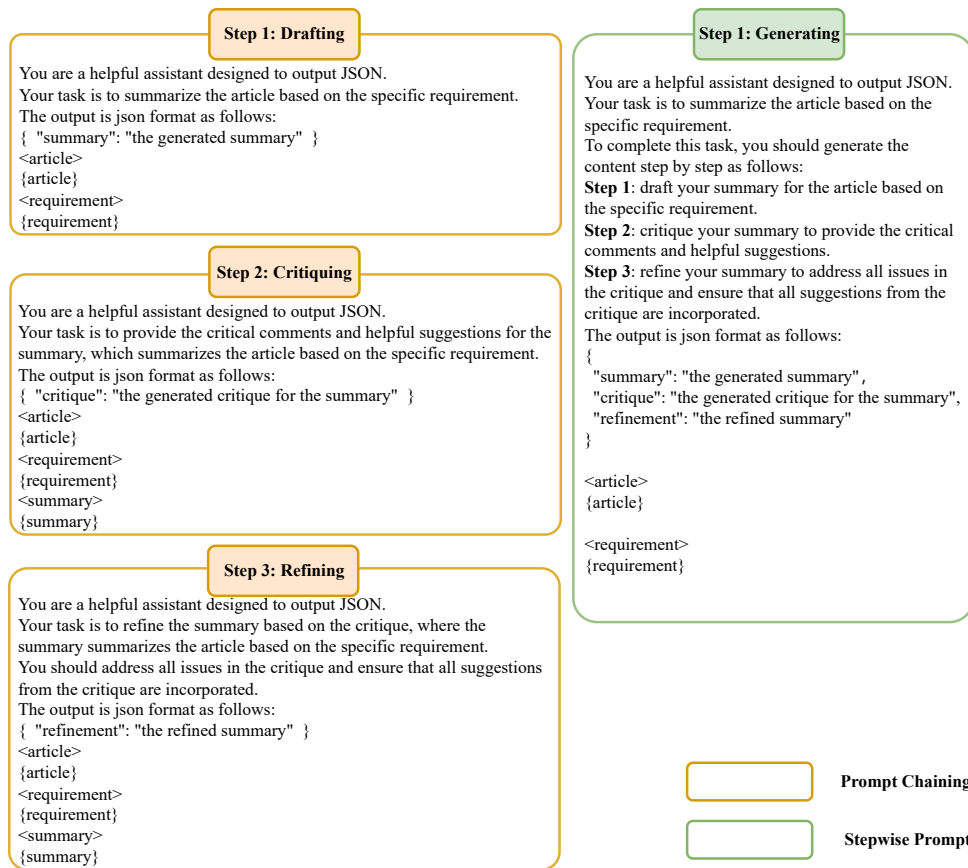
Refinement can be powered by various LLMs. In this paper, we choose the newest versions of GPT-3.5 (gpt-3.5-turbo-0125) and GPT-4 (gpt-4-0125-preview) models from OpenAI⁴ to draft, critique, and refine the outcomes due to their strong instruction-following capabilities. We also explore the performance of a strong open-source LLM (Mixtral 8×7B (Jiang et al., 2024)).

We use the LLMCompare as our evaluation protocol, which compares two candidate outputs and then selects the better one (Zheng et al., 2023; Wang et al., 2023). This is because LLMCompare coupled with GPT-4 is the best evaluation protocol, as mentioned in Liu et al. (2023). The evaluation prompts are shown in Appendix A.

We evaluate the generated summaries from the three quality dimensions as introduced in Liu et al. (2023): (1) **Overall Quality** measures the overall excellence of the summary following the summary requirements. (2) **Missing Information** assesses whether the summary omits any essential article details pertinent to the summary requirements. (3) **Irrelevant Information** examines whether the summary contains extraneous information that falls outside the scope of the summary requirements.

³<https://www.bbc.com/news>

⁴<https://platform.openai.com/docs/models>



Step 1: Generating

You are a helpful assistant designed to output JSON.
Your task is to summarize the article based on the specific requirement.
To complete this task, you should generate the content step by step as follows:
Step 1: draft your summary for the article based on the specific requirement.
Step 2: critique your summary to provide the critical comments and helpful suggestions.
Step 3: refine your summary to address all issues in the critique and ensure that all suggestions from the critique are incorporated.
The output is json format as follows:
{
 "summary": "the generated summary",
 "critique": "the generated critique for the summary",
 "refinement": "the refined summary"
}

<article>
{article}

<requirement>
{requirement}

 Prompt Chaining
 Stepwise Prompt

Figure 1: Prompt Chaining v.s. Stepwise Prompt.

5.1 Exp I: Summarization Benchmark

Setup Consistent with the settings employed in previous research on automatic LLM benchmarking (Dubois et al., 2023; Zheng et al., 2023), we use GPT-4 (gpt-4-0125-preview) one-step outcomes as the baseline. We assess the performance of various methods through direct comparison with the baseline GPT-4 results. To mitigate the potential positional bias, the summary pairs are randomly shuffled before the evaluation. We perform the LLMCompare prompts via gpt-4-0125-preview.

Results Table 1 shows the automatic benchmarking results. More win times or less lose times mean a stronger performance. Generally, the draft summary is enhanced via refinement, regardless of the methods used. Notably, the performance of gpt-3.5-stepwise-refine (and Mixtral-stepwise-refine) is comparable to that of gpt-3.5-chaining-draft (and Mixtral-stepwise-draft). It indicates that stepwise prompt might lead to a simulated refinement process in which LLMs intentionally produce errors only to subsequently correct them.

Q1: Which is the better method of prompt chain-

ing and stepwise prompt?

Prompt chaining achieves the highest win times (77 out of 100), considerably outshining stepwise prompt in producing higher-quality summaries. Moreover, prompt chaining coupled with a better backbone model can lead to better performance by comparing the outcomes of GPT 3.5 and GPT-4.

Q2: How does prompt chaining or stepwise prompt affect the initial outcome?

Notably, summaries initially drafted using stepwise prompt frequently fall short in quality. This may be due to the anticipation that its outputs will subsequently undergo critique and refinement, potentially influencing the initial drafting process.

5.2 Exp II: Robustness

Setup Based on the understanding that different models used for LLMCompare evaluation can yield varied results as indicated by Liu et al. (2023), we employ two iterations of the GPT-4 model, gpt-4-1106-preview and gpt-4-0125-preview, to validate the stability and robustness of prompt chaining’s superiority over stepwise prompt. We do not use the GPT-3.5 models for powering

Models	Overall			Missing			Irrelevant			Length
	Win	Tie	Lose	Win	Tie	Lose	Win	Tie	Lose	
Mixtral-stepwise-draft	12	29	59	13	35	52	8	33	59	111.19
Mixtral-chaining-draft	18	27	55	19	41	40	11	46	43	119.63
Mixtral-stepwise-refine	19	25	56	20	30	50	11	29	60	124.35
Mixtral-chaining-refine	27	21	52	31	29	40	14	48	38	127.3
gpt-3.5-stepwise-draft	10	14	76	8	30	62	5	37	58	86.58
gpt-3.5-chaining-draft	12	22	66	13	28	59	7	37	56	94.76
gpt-3.5-stepwise-refine	12	13	75	14	17	69	2	27	71	85.79
gpt-3.5-chaining-refine	21	17	62	14	24	62	11	38	51	97.24
gpt-4-stepwise-draft	34	40	26	27	53	20	16	60	24	125.73
gpt-4-stepwise-refine	53	29	18	42	49	9	12	57	31	145.85
gpt-4-chaining-refine	77	14	9	57	38	5	19	39	42	174.35

Table 1: Automatic benchmarking results. The summaries of different methods are compared against summaries generated by GPT-4 (gpt-4-0125-preview) one-step generation (i.e., gpt-4-chaining-draft) using the LLM-Compare protocol (Liu et al., 2023). The average length of baseline summaries is 113.03.

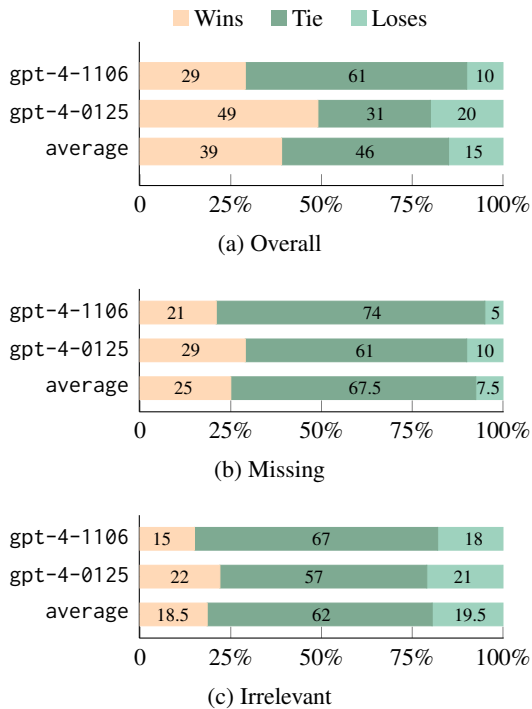


Figure 2: Win rates of refined results from prompt chaining over stepwise prompt. The left-hand models are used to evaluate the refined outcome.

LLMCompare evaluations due to their observed lower consistency with human evaluators. Lastly, average reports the mean value of the two scores.

Results Figure 2 shows the win rates between prompt chaining and stepwise prompt through refined results. The higher win rates of Overall suggest that prompt chaining more effectively adheres to the established summary requirements.

Q3: Does prompt chaining stably outperform

stepwise prompt?

We observe that prompt chaining beats stepwise prompt in both Overall and Missing evaluation across different evaluation models. Meanwhile, prompt chaining exhibits comparable performance to stepwise prompt in Irrelevant. It can confirm the reliability of our conclusion that prompt chaining stably outperforms stepwise prompt.

6 Exp III: Human Evaluation

Setup We engaged two postgraduate students to conduct human evaluation, wherein they compared the refined outcomes of Prompt Chaining against those of the Stepwise Prompt. If Prompt Chaining outperforms Stepwise Prompt, it is notated as a “Win”. For this human evaluation, we randomly selected 30% data from InstruSum dataset. Similar to the automated evaluation, we also use “overall”, “missing”, “irrelevant” as the evaluation metrics.

Results Table 2 presents the quality of critique. A higher score means a better performance. The “win” times significantly exceed the “los” times. It indicates that prompt chaining outperforms stepwise prompt. This conclusion is consistent with GPT-4 automated evaluation. Additionally, we observe that there are fewer “lose” times when we apply the more advanced model, GPT-4. It may imply that Prompt Chaining significantly outperforms Stepwise Prompt when using advanced models.

7 Exp IV: Critique Evaluation

Setup We use METACRITIQUE (Sun et al., 2024) powered by gpt-4-0613 to evaluate the quality of

Models	Overall			Missing			Irrelevant		
	Win	Tie	Lose	Win	Tie	Lose	Win	Tie	Lose
GPT 3.5	16	5	9	15	7	8	6	20	4
GPT 4	14	8	8	13	10	7	9	14	7
Mixtral	11	16	3	7	22	1	6	19	5

Table 2: Human evaluation results.

critiques, which are the intermediate outputs of prompt chaining and stepwise prompt. METACRITIQUE involves three metrics: (1) **Precision** gauges the factuality of the critique; (2) **Recall** measures the comprehensiveness of the critique; (3) **F1 Score** harmonizes the precision score and recall score. We do not assess GPT-4 critiques, as METACRITIQUE uses GPT-4 outcomes as references.

Results Table 3 presents the quality of critique. A higher score means a better performance.

Q4: How does prompt chaining or stepwise prompt affect the critique generation?

Stepwise prompt can generate high-quality critiques that are both more factual and comprehensive. However, in terms of F1 score, prompt chaining achieves only half of that of stepwise prompt, despite the superior performance in refined summaries. These results imply that stepwise prompt produces a simulated refinement process.

Models	MetaCritique		
	Precision	Recall	F1 Score
gpt-3.5-stepwise	78.91	43.29	52.48
gpt-3.5-chaining	40.21	25.62	24.79

Table 3: METACRITIQUE scores.

8 Conclusion

LLMs can enhance summaries by emulating the human-like process of critique and refinement of their initial drafts. This paper explores two distinct strategies for implementing this process: *Prompt Chaining* and *Stepwise Prompt*. We conduct rigorous experiments in the context of text summarization. Our findings indicate that prompt chaining garners a superior performance. Besides, the results imply that stepwise prompt might produce a simulated refinement process. Given that such refinement can be adapted to various tasks, our insights could extend beyond text summarization, potentially advancing the progress of LLMs.

Acknowledgements

We thank the anonymous reviewers for their valuable feedback and helpful suggestions. This project is supported by Research Grants Council of Hong Kong (PolyU/15203617 and PolyU/5210919), National Natural Science Foundation of China (62076212 and 62106165), Qingyuan Research Project and Shanghai Artificial Intelligence Laboratory.

Limitations

Refinement can be applied to various natural language processing (NLP) tasks. However, this paper only compares prompt chaining and stepwise prompt in the scope of text summarization. Future research is warranted to validate the effectiveness of these strategies on an expansive range of NLP tasks, thereby enhancing the generalizability of our findings and their potential utility across the field.

Ethical Considerations

Our experimental data stems from InstruSum, which is well-established and publicly available. Dataset construction and annotation are consistent with the intellectual property and privacy rights of the original authors. This work complies with the ACL Ethics Policy⁵.

References

- Afra Feyza Akyurek, Ekin Akyurek, Ashwin Kalyan, Peter Clark, Derry Tanti Wijaya, and Niket Tandon. 2023. *RL4F: Generating natural language feedback with reinforcement learning for repairing model outputs*. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7716–7733, Toronto, Canada. Association for Computational Linguistics.
- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional

⁵<https://www.aclweb.org/portal/content/acl-code-ethics>

- ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.
- Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. AlpacaFarm: A simulation framework for methods that learn from human feedback. *arXiv preprint arXiv:2305.14387*.
- Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujia Yang, Nan Duan, and Weizhu Chen. 2023. Critic: Large language models can self-correct with tool-interactive critiquing. *arXiv preprint arXiv:2305.11738*.
- Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2022. Large language models can self-improve. *arXiv preprint arXiv:2210.11610*.
- Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.
- Yixin Liu, Alexander R Fabbri, Jiawen Chen, Yilun Zhao, Simeng Han, Shafiq Joty, Pengfei Liu, Dragomir Radev, Chien-Sheng Wu, and Arman Cohen. 2023. Benchmarking generation and evaluation capabilities of large language models for instruction controllable summarization. *arXiv preprint arXiv:2311.09184*.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhunoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*.
- OpenAI. 2023. [Gpt-4 technical report](#). *ArXiv*, abs/2303.08774.
- Jérémy Scheurer, Jon Ander Campos, Tomasz Korbak, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. 2023. Training language models with language feedback at scale. *arXiv preprint arXiv:2303.16755*.
- Shichao Sun, Junlong Li, Weizhe Yuan, Ruifeng Yuan, Wenjie Li, and Pengfei Liu. 2024. The critique of critique. *arXiv preprint arXiv:2401.04518*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Yidong Wang, Zhuohao Yu, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, et al. 2023. Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization. *arXiv preprint arXiv:2306.05087*.
- Seonghyeon Ye, Yongrae Jo, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, and Minjoon Seo. 2023. Selfee: Iterative self-revising llm empowered by self-feedback generation. Blog post.
- Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023. SummIt: Iterative text summarization via ChatGPT. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10644–10657, Singapore. Association for Computational Linguistics.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

A Prompts

We elaborate on the prompts for GPT-4 evaluation in Table 4, 5 and 6 for LLMCompare Overall, LLMCompare Missing, and LLMCompare Irrelevant.

—————SYSTEM MESSAGE—————

You are a helpful assistant designed to output JSON.
 In this task, you will be provided with a news article, a specific summary requirement, and two summaries. The summaries are crafted to meet a specific summary requirement. Note that there may be identical summaries.
 Your task is to compare the overall quality of these two summaries concerning the summary requirement and pick the one that is better (there can be a tie).
 First you will give an explanation of your decision then you will provide your decision in the format of 1 or 2 or tie.
 Please refer to the example below for the format of your response.
 Example Response:

```
{
  "explanation": "Your explanation here",
  "decision": "1 or 2 or tie",
}
```

—————USER MESSAGE—————

```
<article>
{article}
<requirement>
{requirement}
<summary 1>
{summary 1}
<summary 2>
{summary 2}
```

Table 4: Prompt for LLMCompare Overall.

—————SYSTEM MESSAGE—————

You are a helpful assistant designed to output JSON.
 In this task, you will be provided with a news article, a specific summary requirement, and two summaries. The summaries are crafted to meet a specific summary requirement. Note that there may be identical summaries.
 Your task is to compare the quality of these two summaries concerning whether they omit any crucial information from the article with respect to the summary requirement and pick the one that is better (there can be a tie). Crucial information refers to key details or facts that are essential to understanding the article and meeting the summary requirement.
 First you will give an explanation of your decision then you will provide your decision in the format of 1 or 2 or tie.
 Please refer to the example below for the format of your response.
 Example Response:

```
{
  "explanation": "Your explanation here",
  "decision": "1 or 2 or tie",
}
```

—————USER MESSAGE—————

```
<article>
{article}
<requirement>
{requirement}
<summary 1>
{summary 1}
<summary 2>
{summary 2}
```

Table 5: Prompt for LLMCompare Missing.

—————SYSTEM MESSAGE—————

You are a helpful assistant designed to output JSON.
 In this task, you will be provided with a news article, a specific summary requirement, and two summaries. The summaries are crafted to meet a specific summary requirement. Note that there may be identical summaries.
 Your task is to compare the quality of these two summaries concerning whether they include any information that is not relevant to the summary requirement and pick the one that is better (there can be a tie). First you will give an explanation of your decision then you will provide your decision in the format of 1 or 2 or tie.
 Please refer to the example below for the format of your response.
 Example Response:

```
{
  "explanation": "Your explanation here",
  "decision": "1 or 2 or tie",
}
```

—————USER MESSAGE—————

```
<article>
{article}
<requirement>
{requirement}
<summary 1>
{summary 1}
<summary 2>
{summary 2}
```

Table 6: Prompt for LLMCompare Irrelevant.