RevalSum: Refining LLM Summarization via Fine-grained Feedback

Anonymous ACL submission

Abstract

Large Language Models(LLMs) have demonstrated remarkable capabilities in abstractive summarization tasks. However, traditional one-shot generation approaches and self-iterative LLM methods often suffer from issues such as overconfidence, inconsistent feedback, and overcorrection. To address these limitations, we propose RevalSum—a novel LLM-based iterative summarization framework driven by objective evaluators. RevalSum integrates an external multi-dimensional evaluator that provides fine-grained revision suggestions after each generation step, guiding the LLM to perform targeted refinements. This approach effectively overcomes the key shortcomings of existing self-refinement methods and achieves strong performance across multiple evaluation metrics on the CNN/DM and XSum datasets.

1 Introduction

Text summarization, as a core task in natural language processing, plays a critical role in real-world applications such as news aggregation and information retrieval. In recent years,LLMs have become a driving force behind abstractive summarization due to their remarkable generative capabilities.A wealth of studies has demonstrated that,with carefully designed prompts and fine-tuning, LLMs can produce high-quality summaries that better align with human preferences(Ouyang et al., 2022; Zhang et al., 2024), yielding significant improvements in readability and semantic coverage.

However,traditional one-shot generation methods often suffer from semantic drift and the omission of key information(Goyal et al., 2022; Chhabra et al., 2024), which limits their practical utility.To address these challenges,researchers have proposed iterative strategies based on LLM self-initialization, self-feedback, and self-optimization,such as Self-Refine(Madaan et al., 2023), Reflexion(Shinn et al., 2023),and SummIt(Zhang et al., 2023).These approaches aim to progressively refine generated outputs through multiple rounds of internal feedback.Nonetheless, such self-iterative frameworks still face three major challenges: (1) overconfidence, where the model stubbornly adheres to its own outputs (2) feedback inconsistency,which introduces considerable variability across iterations (3) overcorrection,where the model excessively adjusts based on its internal evaluations,potentially diverging from human preferences.Moreover, existing research indicates that reliable external evaluators can substantially enhance stability and overall performance when guiding LLM self-correction(Kamoi et al., 2024).

To overcome these limitations,we propose RevalSum, a novel LLM-based iterative summarization framework driven by objective evaluator feedback.The key idea of RevalSum is to incorporate an external, multi-dimensional evaluator after each generation step, providing fine-grained corrective suggestions to more effectively guide the LLM's targeted improvements. This paper contributes in three fold:

- We propose RevalSum, a novel iterative summarization framework that incorporates an independent automatic evaluator to provide external feedback for LLMs. This approach overcomes the limitations of existing self-refinement methods and demonstrates superior performance across multiple evaluation metrics.
- To the best of our knowledge, this is the first work that integrates fine-grained feedback from an objective evaluator into the iterative generation process of LLM-based summarization.
- Extensive evaluations across diverse datasets and models demonstrate the effectiveness of the proposed framework.

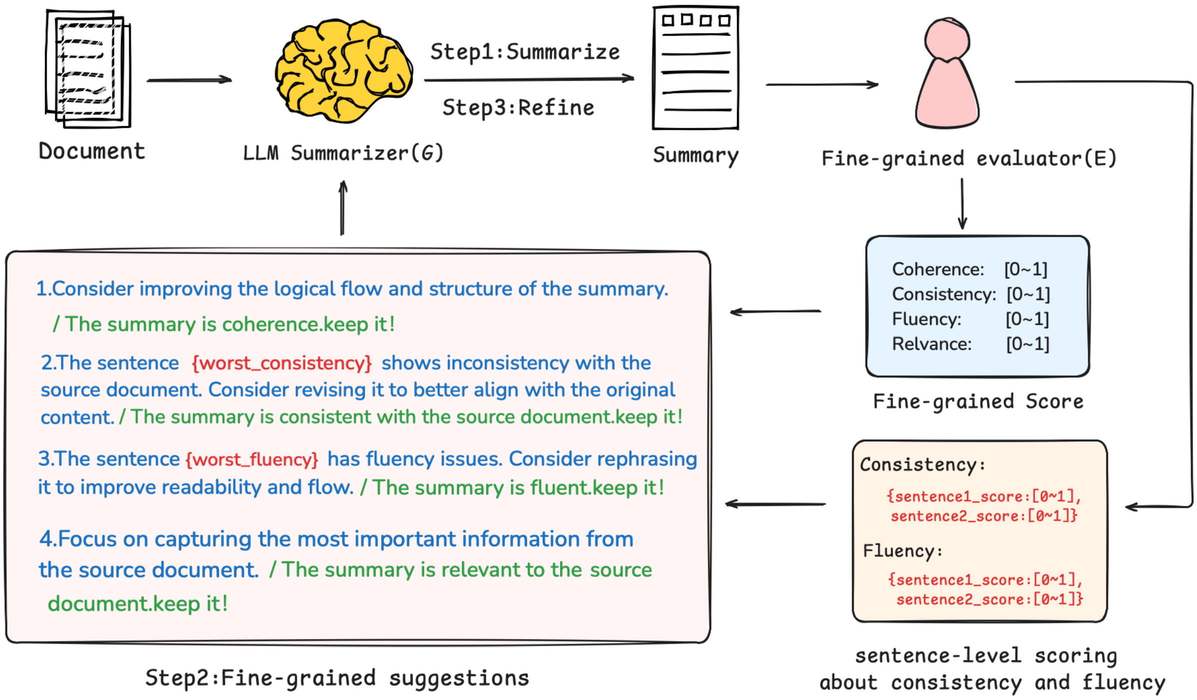

Figure 1: The framework of RevalSum

2 Related Work

LLM-based for Summarization In recent years, the advanced zero-shot paradigm of LLMs has substantially reduced the reliance of text generation tasks on standard datasets (Brown et al., 2020; Chowdhery et al., 2023; Thoppilan et al., 2022). Numerous studies have leveraged LLMs for data augmentation (Wang et al., 2021; Liu et al., 2024). Among these, Madaan et al. (2023) innovatively introduced the self-refine paradigm, wherein a single LLM iteratively refines its outputs, significantly improving performance across various downstream tasks. Building upon this foundation, Zhang et al. (2023) further extended the self-refine approach, systematically validating its effectiveness and applicability specifically in abstractive summarization tasks.

3 Method

3.1 Overview of RevalSum

Given a source document D, RevalSum first generates an initial summary using a generator $\mathcal{G}$. Then, an external evaluator $\mathcal{E}$ scores the summary from multiple dimensions, and we design fine-grained feedback based on the score. Guided by this feedback, $\mathcal{G}$ iteratively optimizes the summary until a pre-defined stopping criterion is met.

RevalSum consists of three core components: a large language model $\mathcal{G}$, a multi-dimensional automatic evaluator $\mathcal{E}$, and two hints for initial generation and iterative optimization. The overall framework is shown in Figure 1 and described in detail in the algorithm 1.

Summarize via LLMs In the RevalSum framework, the generator $\mathcal{G}$ is a large language model (LLM) responsible for generating abstractive summaries. Given a source document D, $\mathcal{G}$ produces the initial summary in response to a specially designed prompt P_{init}, which incorporates the stylistic and content-specific characteristics of the target dataset. In the subsequent iterative process, $\mathcal{G}$ is also used to refine the summary based on feedback.All generations are performed in a zero-shot or few-shot prompting manner.

Fine-grained Feedback via Evaluator In the RevalSum framework,the external evaluator $\mathcal{E}$ is responsible for assessing the quality of the generated summary and providing objective feedback to guide subsequent iterative optimization.We use UniEval(Zhong et al., 2022) as the implementation of $\mathcal{E}$,which quantifies the overall quality of the summary through a multi-dimensional scoring mechanism.Given the source document \$D\$ and the current summary $S^{(t)}$,$\mathcal{E}$ calculates the score $Q^{(t)} = E(D, S^{(t)})$,and evaluates the performance of the summary in terms of consistency,relevance,fluency,and coherence.

To further improve the summary quality,RevalSum constructs fine-grained, targeted revision suggestions based on the scoring results. In the dimensions of factual consistency and fluency, we employ sentence-level scoring to help the LLM precisely identify problematic sentences and provide directions for improvement. Our fine-grained feedback prompts are set as follows:

- **Coherence**:"Consider improving the logical flow and structure of the summary." or "The summary is coherent.Keep it"

- **Consistency**:"<The sentence> shows inconsistency with the source document. Consider revising it to better align with the original content." or "The summary is relevant to the source document.Keep it"

- **Fluency**:"<The sentence> has fluency issues.Consider rephrasing it to improve readability and flow." or "The summary is fluent.Keep it"

- **Relevance**:"Focus on capturing the most important information from the source document." or "The summary is relevant to the source document.Keep it"

Feedback-Guided Iterative Refinement After obtaining the score given by the external evaluator $\mathcal{E}$ and the prompt it constructs, the generator $\mathcal{G}$ will modify the summary item by item based on the suggestions, generating the new summary $S^{(t)} = \mathcal{G}(D, S^{(t-1)}, P^{(t)}_{\text{refine}})$.The complete prompt for RevalSum is shown in Appendix B.

3.2 Loop strategy and Stop criteria

To address overconfidence, error accumulation, and over-correction in LLM self-iteration, RevalSum adopts two termination strategies: (1) stopping after a maximum of T iterations to ensure efficiency, and (2) early stopping when the evaluator score $Q^{(t)} \geq \tau$, indicating satisfactory quality. To prevent excessive revisions, RevalSum maintains the best summary S^* and its highest score Q^* across iterations, updating them whenever $Q^{(t)} > Q^*$.

Algorithm 1 RevalSum: Iterative Summarization with Evaluation Feedback

Require: Document D, Summarizer / Optimizer $\mathcal{G}$, Evaluator $\mathcal{E}$, Max iterations T, Threshold τ, $Q^{(t)}$: evaluation scores from $\mathcal{E}$ at step t, Two types of prompts $\mathcal{P}_{\text{init}}, \mathcal{P}_{\text{refine}}$
Ensure: Best summary S^*

$S^{(0)} \leftarrow \mathcal{G}(\mathcal{P}_{\text{init}} \parallel D)$
$Q^{(0)} \leftarrow \mathcal{E}(D, S^{(0)})$
$S^* \leftarrow S^{(0)}$, $Q^* \leftarrow Q^{(0)}$
for $t = 1$ to T **do**
  if $Q^*.score \geq \tau$ **then**
    break
  end if
  ${\mathcal{P}_{\text{refine}}}^{(t)} \leftarrow \mathcal{Q}^{(t-1)}$
  $S^{(t)} \leftarrow \mathcal{G}({\mathcal{P}_{\text{refine}}}^{(t)} \parallel D \parallel S^{(t-1)})$
  $Q^{(t)} \leftarrow \mathcal{E}(D, S^{(t)})$
  if $Q^{(t)}.score > Q^*.score$ **then**
    $S^* \leftarrow S^{(t)}$, $Q^* \leftarrow Q^{(t)}$
  end if
end for
return S^*

4 Experiment

4.1 Experiment Setting

Datasets We evaluated the proposed framework in this paper on two mainstream abstractive summarization datasets: CNN/DM(Hermann et al., 2015) and XSum(Narayan et al., 2018). Specifically, we randomly sampled 1000 instances from the test set of each dataset as experimental data, and we fixed the random seed to 101 to ensure the reproducibility of the experimental results.

Models To provide a more comprehensive evaluation of the effectiveness of the proposed method, we selected both open-source and closed-source models for experimental validation. For the open-source model, we employed the currently prevalent LLaMA3.1-8B-Instruct [1]. For the closed-source model, we utilized the widely adopted ChatGPT(gpt-4o-0513)[2]. Furthermore, we set the generation temperature to 0 to ensure the determinacy and stability of the generated results.

Baselines We compare our proposed RevalSum method with SummIt(Zhang et al., 2023), a representative self-iterative summarization approach

[1]https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

[2]https://platform.openai.com/docs/models/gpt-4o

Model	CNN/DM								XSum							
	UniEval				ROUGE			G-Eval	UniEval				ROUGE			G-Eval
	Coh	Con	Flu	Rel	R-1	R-2	R-L	score	Coh	Con	Flu	Rel	R-1	R-2	R-L	score
Zero-shot setting																
Pegasus (zero-shot)	0.7771	0.8937	0.8779	0.7594	34.75	13.58	26.03	3.36	0.9399	0.9457	0.7743	0.6684	18.78	2.376	12.14	2.72
BART (zero-shot)	0.8825	0.9249	0.8668	0.8456	35.72	15.51	29.30	3.81	0.9514	0.9485	0.8030	0.7483	19.85	2.73	13.01	2.99
T5 (zero-shot)	0.7772	0.8768	0.8819	0.7480	39.03	17.55	31.82	3.61	0.8308	0.8391	0.8950	0.8604	30.38	11.04	23.76	2.75
LLaMA3.1-8B (SummIt)	0.9417	0.8916	0.9459	0.9258	38.68	15.14	24.80	4.53	0.9364	0.9212	0.9434	0.9050	25.01	6.72	18.08	4.06
LLaMA3.1-8B (RevalSum)	0.9592↑	0.9132↑	0.9523↑	0.9292↑	41.46↑	16.78↑	27.37↑	4.66↑	0.9409↑	0.9203	0.9497↑	0.9157↑	27.18↑	6.67	20.00↑	4.21
ChatGPT (SummIt)	0.8609	0.7797	0.9267	0.9080	36.08	12.12	22.34	4.81	0.8942	0.8828	0.9107	0.8750	26.02	6.63	18.91	4.62
ChatGPT (RevalSum)	0.9293↑	0.8482↑	0.9430↑	0.9278↑	36.48↑	12.21↑	22.74↑	4.88↑	0.9364↑	0.9312↑	0.9503↑	0.9145↑	27.04↑	6.97↑	19.61↑	4.65↑
Few-shot setting																
LLaMA3.1-8B (SummIt)	0.9480	0.8978	0.9470	0.9271	39.21	15.43	25.11	4.51	0.8215	0.8958	0.8824	0.7674	29.78	8.95	22.18	3.69
LLaMA3.1-8B (RevalSum)	0.9519↑	0.9257↑	0.9513↑	0.9311↑	42.86↑	18.57↑	28.88↑	4.67↑	0.9554↑	0.9429↑	0.9547↑	0.9265↑	29.45	9.38	22.25	4.39↑
ChatGPT (SummIt)	0.8921	0.8283	0.9300	0.9120	36.51	12.37	23.01	4.83	0.9114	0.9020	0.9265	0.8898	25.96	6.61	18.72	4.64
ChatGPT (RevalSum)	0.9267↑	0.8548↑	0.9432↑	0.9265↑	36.70↑	12.36	23.13↑	4.88↑	0.9461↑	0.9362↑	0.9534↑	0.9193↑	29.28↑	8.85↑	21.82↑	4.65↑

Table 1: Evaluation results on CNN/DM and XSum datasets using UniEval, ROUGE, and G-Eval under zero-shot and few-shot settings.↑ indicates improvement over the corresponding baseline.Values in blue indicate the best performance for each corresponding metric

based on large language models (LLMs). In addition, to comprehensively evaluate the effectiveness of our method in the zero-shot scenario, we select three widely-used pretrained summarization models as baselines including Pegasus(Zhang et al., 2020), BART(Lewis et al., 2020), and T5(Raffel et al., 2020) for comparison.

Automatic Evaluation We use three automatic metrics to evaluate summary quality: (1) ROUGE(Lin, 2004), a lexical-overlap-based metric for reference similarity; (2) UniEval (Zhong et al., 2022), a BERT-based metric assessing coherence, consistency, fluency, and relevance; and (3) G-Eval(Liu et al., 2023), a high-performing LLM-based evaluation method.

4.2 Results and Analysis

	R-1	R-2	R-L	G-Eval
RevalSum	42.86	18.57	28.88	4.67
-w/o coh	40.68	16.51	27.48	4.66
-w/o con	40.62	16.69	28.44	4.54
-w/o flu	40.12	16.33	27.46	4.58
-w/o rel	40.92	16.65	27.39	4.60

Table 2: Ablation study results for RevalSum

The effectiveness of fine-grained feedback Table 1 reports RevalSum's performance on CNN/DM and XSum. Compared to the self-iterative baseline SummIt(Zhang et al., 2023), RevalSum consistently achieves better results across UniEval, ROUGE, and G-Eval under both zero-shot and few-shot settings, with improvements indicated by ↑.

On CNN/DM, RevalSum surpasses SummIt in all UniEval dimensions, ROUGE scores, and G-Eval, validating the effectiveness of its fine-grained feedback. Similar gains are observed on XSum, especially in consistency and G-Eval. While T5 slightly outperforms RevalSum in ROUGE on XSum, this can be reasonably attributed to the highly abstractive nature of XSum reference summaries, which often include information not explicitly present in the source text—making word-overlap-based metrics less reliable in this context.

Ablation Studies To evaluate each module's impact, we conducted an ablation study on fine-grained feedback using the CNN/DM dataset (Table 2). Results show that each dimension improves RevalSum's performance: the fluency module boosts ROUGE most, while the factual consistency module contributes most to G-Eval. This confirms that sentence-level targeted feedback enhances both lexical and factual evaluation metrics.

5 Conclusion

We propose a novel framework, RevalSum, which leverages an external fine-grained feedback module to guide large language models through iterative refinement. Experimental results demonstrate that fine-grained prompts effectively enhance the model's self-correction capability in summarization tasks, thereby improving summary quality.

Limitations

Our current work primarily relies on UniEval as an external evaluator to validate the effectiveness of the RevalSum framework. While UniEval can provide granular scores and help locate sentences needing revision, as a black-box evaluator, it cannot offer human-understandable evidence for its ratings. This, to some extent, limits the precision of the optimization direction for LLMs. For instance, when evaluating relevance, if the evaluator could not only provide a low score but also point out "the summary is missing key information about <content>," it would offer more instructive revision advice to the model. Therefore, constructing a truly unsupervised external evaluator capable of providing both fine-grained scores and interpretable reasoning for these scores will be a crucial direction for our future research. We believe that exploring more insightful external feedback mechanisms is a key pathway to further enhance the performance and controllability of text generation models.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Anshuman Chhabra, Hadi Askari, and Prasant Mohapatra. 2024. Revisiting zero-shot abstractive summarization in the era of large language models from the perspective of position bias. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 1–11, Mexico City, Mexico. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.

Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. 2024. When can LLMs actually correct their own mistakes? a critical survey of self-correction of LLMs. *Transactions of the Association for Computational Linguistics*, 12:1417–1440.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yang Liu, Dan Iter, Yichong Xu, Shuo Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Conference on Empirical Methods in Natural Language Processing*.

Yixin Liu, Kejian Shi, Katherine He, Longtian Ye, Alexander R Fabbri, Pengfei Liu, Dragomir Radev, and Arman Cohan. 2024. On learning to summarize with large language models as references. In *NAACL-HLT*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Noah Shinn, Beck Labash, and Ashwin Gopinath. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. *ArXiv*, abs/2303.11366.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, and 1 others. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2021. Want to reduce labeling cost? gpt-3 can help. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4195–4205.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023. Summit: Iterative text summarization via chatgpt. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, page 10644–10657. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International conference on machine learning*, pages 11328–11339. PMLR.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. 2024. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 12:39–57.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Peng Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. In *Conference on Empirical Methods in Natural Language Processing*.

A Ethics and Risks

A.1 Ethics

Data Privacy and Source All datasets utilized in this study—CNN/DailyMail[3] and XSum[4] are publicly accessible, ensuring transparency in data sourcing and minimizing ethical concerns regarding data usage.

A.2 Risks

Over-reliance on the Quality of Automatic Evaluator Feedback The core mechanism of RevalSum involves guiding the large language model (LLM) to iteratively generate summaries based on fine-grained feedback from an external evaluator. If the evaluator itself is biased, unstable, or lacks sufficient capability for certain types of texts, it may mislead the model into optimizing in the wrong direction.

B The full prompt template of RevalSum

[3] https://github.com/abisee/cnn-dailymail
[4] https://github.com/EdinburghNLP/XSum

Step	Prompt Template of RevalSum
Init	You are a top expert in the field of summary generation. Now you need to complete the text summary generation task in output format. Before generating a summary, please think about the following points: 1) **Coherence**: The summary should have a logical flow and be easily understood as a cohesive piece. 2) **Consistency**: Ensure factual consistency, with no contradictions in the summary compared to the original content. 3) **Fluency**: The summary should be grammatically correct, well-structured, and natural to read. 4) **Relevance**: The summary should cover the most important points from the original text without adding irrelevant information. 5) **Overall Quality**: Aim for a well-rounded summary that reflects the original content accurately and concisely. Document:{doc} Please provide **only** the summary, formatted EXACTLY as follows: <summary> Your generated summary text here </summary>
Fine-grained suggestions	**Coherence**: "Consider improving the logical flow and structure of the summary." or *"The summary is coherent.keep it"* **Consistency**: "{The sentence} shows inconsistency with the source document. Consider revising it to better align with the original content." or *"The summary is relevant to the source document.keep it"* **Fluency**: "{The sentence} has fluency issues.Consider rephrasing it to improve readability and flow." or *"The summary is fluent.keep it"* **Relevance**: "Focus on capturing the most important information from the source document." or *"The summary is relevant to the source document.keep it"*
Refine	You are an AI model for generating summaries, and your task is to iteratively improve the provided summary based on the given feedback. Current document to summarize:{doc} Current summary (to be improved):{summary} Current suggestions for improvement:{suggestion} Please refer to the detailed suggestions for improvement to refine this summary. Please make sure all suggestions are modified. Now, please provide only the refined summary formatted EXACTLY as follows: <refine summary> Your refined summary text here </refine summary>

Table 3: The full prompt template of RevalSum