TRAC: Token-level Reward Assignment for Coherent Abstractive Summarization

Xuanqi Chen[♠,♣], **Ziying Rong**[♣], **Xinfeng Liao**[♢], **Lianxi Wang**[♣], **Ying Gao**[♠*] **& Shengyi Jiang**[♣,†]

[♠]South China University of Technology, [♣]Guangdong University of Foreign Studies
[♢]Jinan University, [†]City University of Macau
{chenxuanqi6, RongZiYing1016, liaoxinfeng218}@163.com
wanglianxi@gdufs.edu.cn, gaoying@scut.edu.cn, Jiangshengyi@163.com

Abstract

Large Language Models (LLMs) have achieved remarkable success in text summarization, particularly through the integration of reinforcement learning. However, maintaining logical coherence and contextual consistency remains a pervasive challenge in long-form generation, often hindering the production of high-quality, unified summaries. To address these persistent issues, we propose TRAC, a framework that introduces a token-level reward function by integrating relative sentence gain, inter-sentence attention, and a Gaussian length penalty. By training a Process Reward Model (PRM) to provide fine-grained, step-wise supervision, TRAC ensures superior structural integrity and fluency during the generation process. Experimental results demonstrate that TRAC outperforms the sequence-level baseline by 11.05% in Fluency and 10.61% in Relevance. Furthermore, it achieves significant gains over competitive baselines such as FIGA and TLCR, underscoring its effectiveness and generalizability in high-quality NLP summarization.

1 Introduction

In the era of rapid information proliferation, the deluge of fragmented textual data across news and social media has exacerbated information overload, complicating the synthesis of core event details (Bawden and Robinson, 2020). While LLMs offer powerful capabilities for extracting key information and generating structured summaries (Aly et al., 2025), their performance is often constrained by existing training paradigms. Specifically, mainstream reinforcement learning approaches, such as Proximal Policy Optimization (PPO) using sequence-level rewards (Ryu et al., 2024; Fikri et al., 2024), typically rely on discrete and delayed signals. Such sparse feedback struggles to capture intricate token-level dependencies, frequently resulting in summaries that lack linguistic coherence and factual consistency (Bosselut et al., 2020).

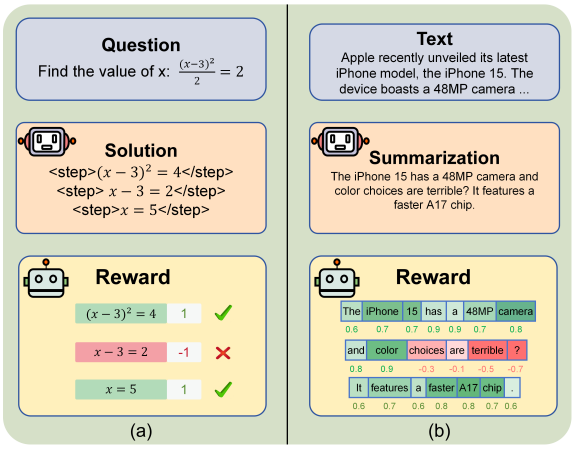

Figure 1: PRM model providing fine-grained rewards for (a) mathematical reasoning with step-level reward and (b) text summarization with token-level reward.

Recent research has shifted toward token-level reward mechanisms to mitigate the challenges of delayed feedback (Yoon et al., 2024; Fu et al., 2025; Huang et al., 2024). However, these approaches often suffer from high computational costs and a heavy reliance on manual reward engineering (Khan et al., 2023). While PRMs offer a promising alternative by providing intermediate signals to optimize generation paths (Lightman et al., 2023), their application has remained largely restricted to structured reasoning tasks, such as mathematics and logic (Yuan et al., 2023; Zhang et al., 2024). As illustrated in Figure 1(a), there remains a significant research gap in extending PRMs to open-ended natural language generation tasks, including summarization and dialogue, where capturing nuanced semantic dependencies is critical.

Building on this foundation, we extend the PRM framework to summarization and dialogue generation Figure 1(b). Our proposed token-level reward function integrates coherence, consistency, and a

*Corresponding authors: gaoying@scut.edu.cn, Jiangshengyi@163.com

Findings of the Association for Computational Linguistics: ACL 2026, pages 37563–37578

Gaussian length penalty, leveraging inter-sentence attention to map global signals to individual tokens. By autoregressively predicting rewards from sequence hidden states, the framework captures subtle dependencies while ensuring global structural integrity. This approach provides an efficient, transferable solution for high-quality open-ended generation with significantly reduced computational overhead. To conclude, our primary contributions are summarized as follows:

- **Fine-grained Reward Mechanism**: We develop a novel token-level reward function that decomposes holistic scores into individual tokens by integrating relative sentence gain and inter-sentence attention. A Gaussian length penalty is further incorporated to ensure a balanced and well-constrained reward distribution.
- **Autoregressive Process Reward Model**: We introduce an autoregressive PRM that utilizes hidden states to generate continuous, context-aware reward signals. Compared to conventional approaches, this model enhances computational efficiency and task transferability while ensuring superior alignment with contextual semantics.
- **Empirical Validation**: Extensive experiments demonstrate that the TRAC framework consistently achieves state-of-the-art performance across multiple generation tasks and benchmarks, significantly improving fluency, coherence, and logical consistency.

2 Related Work

2.1 Token-Level Reward

In reinforcement learning, the efficacy of LLMs is fundamentally constrained by reward function design. While conventional mechanisms often rely on sparse, delayed signals, Fine-Grained Reward mechanisms (Wu et al., 2023) have emerged to provide denser, more informative feedback, significantly enhancing training efficiency and reasoning performance.

Beyond global outcomes, fine-grained signals are increasingly derived from structured human or model-based feedback. While standard RLHF utilizes coarse preference rankings, critique-based feedback (Touvron et al., 2023) enables annotators to highlight specific deficiencies, facilitating more precise error correction. Furthermore, the LLM-as-a-Judge paradigm leverages frontier models (e.g., GPT-4) to automate the generation of granular reward signals (Zheng et al., 2023). This is complemented by self-correction frameworks (Shinn et al., 2023), where models iteratively refine outputs through internal evaluation, establishing a closed-loop optimization process.

2.2 Process Reward Model

Early PRM research focused on structured domains like mathematics and code. Lightman et al. (2023) demonstrated the efficacy of step-level supervision, though the reliance on human-annotated solutions poses significant scalability challenges.

To mitigate annotation costs, contemporary studies explore automated training paradigms. Yuan et al. (2023) derived implicit PRMs from outcome-level signals, while Jiao et al. (2024) utilized self-supervised trajectory synthesis for complex planning. Beyond passive evaluation, PRMs now serve as active search guides; when integrated with Monte Carlo Tree Search, they act as node evaluators to prioritize optimal reasoning paths (Zhang et al., 2024; Park et al., 2025). Recent theoretical advancements further ground these signals in Q-value rankings to provide robust decision guidance (Li and Li, 2025).

Despite these strides, generalizing PRMs to open-ended generation remains a frontier. Zeng et al. (2025) addressed domain-specificity by training on large-scale synthetic reasoning data. Extending this to natural language generation, token-level PRMs can capture complex inter-textual dependencies, facilitating reinforcement learning in long-form summarization and dialogue where maintaining logical consistency and linguistic fluency is traditionally problematic for LLMs.

3 Methodology

We propose the TRAC framework, a three-stage pipeline illustrated in Figure 2. First, the base model is fine-tuned for text summarization. Second, this model generates candidate summaries, from which hidden representations and attention scores are extracted to construct token-level rewards and train the PRM. Finally, the trained PRM provides granular guidance for reinforcement learning, optimizing the summarization model for enhanced performance. The full algorithmic procedure is detailed in Appendix A.

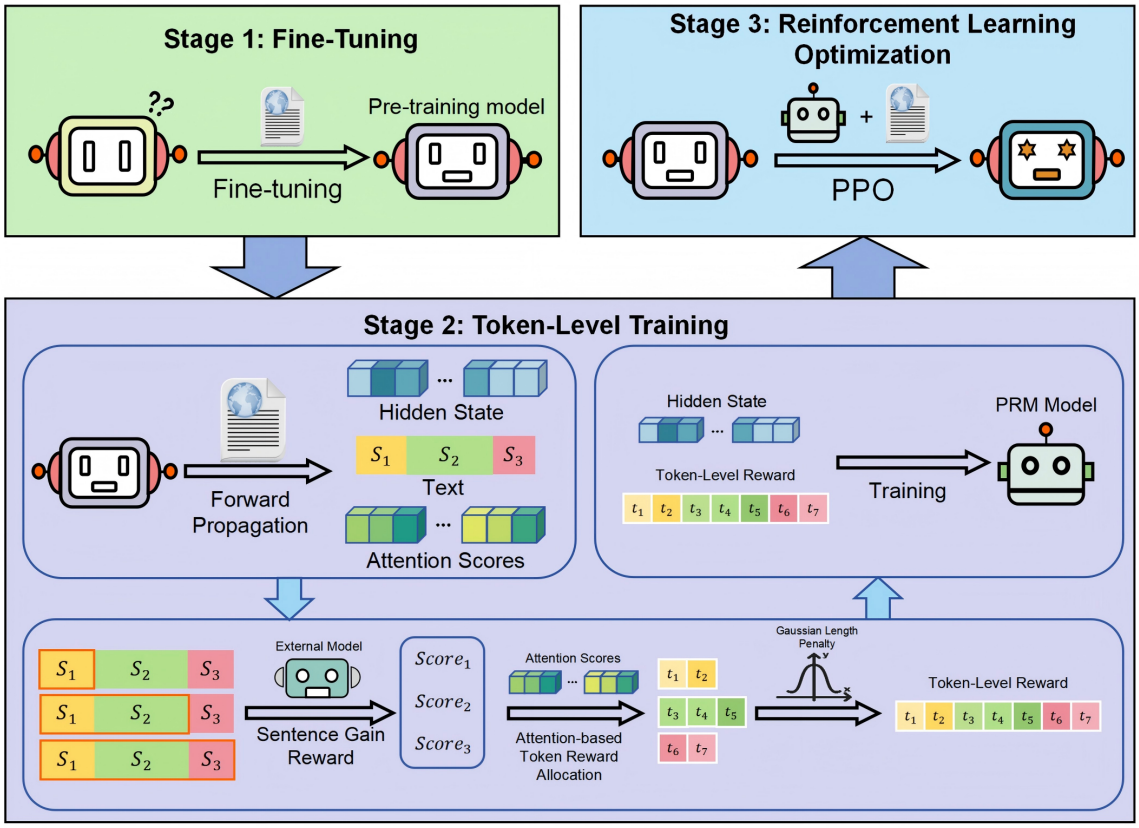

Figure 2: Overall framework of the **TRAC**. **Stage 1** involves fine-tuning the pre-trained model. **Stage 2** generates token-level rewards, which are used to train the sub-regression PRM model. **Stage 3** leverages the trained PRM model to further optimize the PPO process.

3.1 Fine-Tuning

We first perform teacher-forced training on the model using reference summaries $y = \{y_1, y_2, \ldots, y_T\}$ to optimize the token-level prediction probabilities of the language model.

$$\mathcal{L}_{SFT} = -\sum_{t=1}^{T} \log P_\theta(y_t | y_{<t}, x) \tag{1}$$

where x denotes the input text pair, P_θ represents the conditional probability under the current language model parameters θ.

3.2 Token-Level Training

3.2.1 Sentence-Level Gain

We construct the summary sequence incrementally. Suppose the generated summary consists of n sentences, denoted as $S = \{s_1, s_2, \ldots, s_n\}$. After the i-th concatenation, we obtain the partial summary $S_{\leq i} = \{s_1, \ldots, s_i\} (i \leq n)$. We evaluate $S_{\leq i}$ using the *Fluency* and *Coherence* metrics from UniEval, and denote the resulting score as $Q(S_{\leq i})$. The marginal contribution of the i-th sentence, i.e., its relative gain reward, is then defined as:

$$R_{gain}(S_i) = Q(S_{\leq i-1}) + \frac{L(S_i)}{L(S_{\leq i})} \cdot \left[Q(S_{\leq i}) - Q(S_{\leq i-1}) \right] \tag{2}$$

where $L(S_{\leq i})$ denotes the total number of tokens in the first i sentences. By incorporating the token count, we mitigate the sparsity of inter-sentence rewards that may arise from overly long sentences. The resulting score quantifies the contribution of the current sentence to the overall summary quality, thereby guiding the model to generate structurally coherent sentence sequences.

3.2.2 Token-Level Reward Allocation

To achieve a fine-grained distribution from sentence-level rewards to individual tokens, we propose a normalized allocation strategy based on inter-sentence attention. During the decoding process of summary generation, we record the attention scores for each output token. Suppose the i-th sentence consists of $s_i = \{t_{i_1}, t_{i_2}, \ldots, t_{i_m}\}$ tokens, and their corresponding attention scores are given by:

$$A_i = \{a_{i_1}, a_{i_2}, \ldots, a_{i_m}\} \tag{3}$$

where $a_{i_j} \in [0, 1]$ denotes the attention of the j-th token to the contextual information. To prevent reward dilution in long sentences, we first calculate the mean μ_i and standard deviation σ_i of the attention scores for the i-th sentence:

$$\sigma_i = \sqrt{\frac{1}{m}\sum_{j=1}^{m}(a_{i_j} - \mu_i)^2} \tag{4}$$

The attention scores are then normalized using the mean and standard deviation:

$$\bar{a}_{i_j} = \frac{a_{i_j} - \mu_i}{\sigma_i + \epsilon} \tag{5}$$

where ϵ is a small constant to avoid division by zero.

The discriminability of the attention scores is further enhanced using a Softmax operation, yielding the final normalized weights:

$$w_{i_j} = \frac{e^{\bar{a}_{i_j}}}{\sum_{k=1}^{m} e^{\bar{a}_{i_k}}} \tag{6}$$

where w_{i_j} denotes the normalized weight of the j-th token within the i-th sentence, representing its relative importance within that sentence.

Finally, the sentence-level reward $R_{gain}(s_i)$ is distributed across individual tokens, yielding the reward for token t_j as:

$$R_{i_j}^{token} = w_{i_j} \cdot R_{gain}(s_i) \tag{7}$$

This mechanism biases the reward distribution toward key tokens within the sentence while mitigating reward dilution caused by variations in sentence length.

3.2.3 Gaussian Length Penalty Mechanism

To mitigate the inherent trade-off between text length and linguistic quality—where models often favor shorter sequences to artificially inflate fluency and coherence scores—we designed a Gaussian-based length penalty mechanism. By centering on the reference length L_{ref}, this mechanism penalized deviations based on the position L_{t_j} of each generated token, thereby encouraging the model to produce structurally complete and adequately detailed outputs without compromising global structural integrity.

$$\Delta L_{t_j} = |L_{t_j} - L_{ref}| \tag{8}$$

where ΔL_{t_j} denotes the relative distance between token t_j and the reference length L_{ref}. The penalty term is then defined as:

$$P_{t_j}^{len} = -\alpha \cdot \left(1 - \exp\left(-\frac{\Delta L_{t_j}^2}{2\sigma^2}\right)\right) \tag{9}$$

where σ is a hyperparameter that controls the width of the penalty, while α denotes the maximum penalty amplitude.

To prevent excessive gradient fluctuations when the summary length deviates significantly from the reference value, we further introduce a clipping mechanism that restricts the penalty term within a reasonable range $[-c, 0]$. The final length penalty is thus defined as:

$$P_{t_j}^{len} = clip\left(P_{t_j}^{len}, -c, 0\right) \tag{10}$$

This mechanism imposes moderate penalties on unreasonable lengths while preventing instability caused by extreme sequence lengths, thereby enhancing both the length controllability and the overall structural quality of the generated summaries.

3.2.4 Reward-Penalty Integration

For fine-grained and semantically consistent RL, we integrate the inter-sentence reward R_{gain} and length penalty P_{len} into a unified token-level reward. The final reward for each token is defined as their weighted combination:

$$r_t = R_t^{token} + P_t^{len} \tag{11}$$

3.2.5 Token-level PRM Training

To accurately model the quality of each token in the generated sequence, we employ a Process Reward Model to predict token-level scores. Specifically, the PRM takes as input the sequence of hidden vectors produced during the decoding process of the language model, denoted as $H = \{h_1, h_2, \ldots, h_t\}$:

$$\hat{r}_t = \text{PRM}(h_{\leq t}) = f_\phi(h_1, h_2, \ldots, h_t) \tag{12}$$

The scoring network f_ϕ adopts a Transformer decoder architecture (detailed in Appendix C) to assign rewards autoregressively. Unlike conventional value heads that rely on one-shot parallel prediction, f_ϕ captures contextual dependencies through token-level modeling, which better aligns with the causal nature of preference evaluation. The

training objective is to minimize the error between predicted scores and ground-truth rewards:

$$\mathcal{L}_{PRM} = \frac{1}{T}\sum_{t=1}^{T}(\hat{r}_t - r_t)^2 \tag{13}$$

where T denotes the total number of tokens in the summary.

3.3 Reinforcement Learning Optimization

We employ the token-level rewards predicted by the PRM as the reward signal in PPO. For the t-th token, the Value Head estimates the state value V_t, and the temporal-difference (TD) error is computed as:

$$\delta_t = r_t + \gamma V_{t+1} - V_t \tag{14}$$

where r_t denotes the token-level reward predicted by the PRM model, and δ_t represents the TD error at time step t. γ is the discount factor, determining the weight of future rewards. V_t and V_{t+1} are the value estimates of the state at time t and $t+1$, respectively.

Calculate the dominant term through GAE (Generalized Advantage Estimation):

$$A_t = \sum_{l=0}^{T'}(\gamma\lambda)^l\delta_{t+l} \tag{15}$$

where A_t represents the advantage estimate at time step t. T' is the horizon length, and γ is the discount factor. λ is a parameter used to balance the bias and variance in the advantage estimation. δ_{t+l} is the TD error at time $t+l$.

Ultimately, through PPO training, the strategy was optimized as follows:

$$\mathcal{L}_{ppo} = \mathbb{E}\left[\min(\pi_t A_t, clip(r_t, 1-\epsilon, 1+\epsilon)A_t)\right] \tag{16}$$

where $\mathcal{L}_{\text{ppo}}$ is the PPO objective function. π_t denotes the importance sampling ratio at time step t, which is the ratio of the new policy's probability to the old policy's probability: $\pi_t = \frac{\pi_{new}(a_t|s_t)}{\pi_{old}(a_t|s_t)}$. ϵ is a clipping hyperparameter that constrains the update step to ensure training stability. The *clip* function limits the probability ratio to the range $[1-\epsilon, 1+\epsilon]$ to prevent large policy updates.

4 Experimental Settings

4.1 Datasets

We evaluated our framework on four diverse benchmarks: (1) **CNN/DM** (See et al., 2017) for standard news summarization; (2) **XL-Sum** (Hasan et al., 2021) for multilingual adaptability; (3) **NYT Corpus** (Sandhaus, 2008) for long-document summarization; (4) **BookSum** (Kryściński et al., 2022) for hierarchical, narrative summarization of entire books; and (5) **DailyDialog** (Li et al., 2017) for emotional and multi-turn dialogue generation. **Further details are provided in Appendix B.**

4.2 Evaluation Metrics

Automatic Metrics: For summarization, we utilized **ROUGE** (Lin, 2004) for lexical overlap and **UniEval** (Zhong et al., 2022) for semantic consistency. Dialogue tasks primarily used UniEval to measure coherence. Additionally, we employed **G-Eval** (Liu et al., 2023) with Chain-of-Thought prompting to calculate probability-based scores for high-level human alignment (see Figure 8). **Human Evaluation:** We adopted the **FFCI** framework (Koto et al., 2022), assessing *Faithfulness, Fluency, Conciseness,* and *Informativeness*.

4.3 Baselines

To evaluate our fine-grained reward modeling, we utilized **Llama-3.1-8B**, **Qwen-2.5-7B**, and **GPT-2** as backbones. We benchmarked against three representative baselines: **PPO$_{\text{seq}}$**, a sparse approach rewarding only the terminal token; **TLCR** (Yoon et al., 2024), a token-level mechanism based on discriminator confidence; and **FIGA** (Ramamurthy et al., 2023), an edit-based method utilizing Levenshtein distance. This selection ensures a robust comparison across sparse, continuous, and edit-based reward paradigms. By default, we used Llama-3.1 as the backbone model.

5 Result and Analysis

5.1 Summarization Benchmark Evaluation

On the CNN/DM dataset, we evaluated the TRAC framework across three backbone models against competitive baselines. As shown in Table 1, TRAC consistently achieved state-of-the-art performance. Notably, it surpassed the sequence-level baseline PPO$_{seq}$ by **11.0%** in *Fluency* and **10.61%** in *Relevance*. The superiority of TRAC was most pronounced in the G-Eval metric, which reflected high-level human alignment. On Llama-3.1, TRAC

Method	Model	R-1 (↑)	R-2 (↑)	R-L (↑)	Coher (↑)	Consis (↑)	Flu (↑)	Rele (↑)	G-Eval (↑)
SFT	GPT-2	31.20	12.45	21.80	68.40	72.15	70.33	65.80	62.45
	Llama-3.1	41.50	19.85	33.10	74.20	78.90	81.45	76.20	77.64
	Qwen-2.5	39.80	18.90	31.55	73.80	77.45	80.20	74.90	76.58
PPO_{seq}	GPT-2	32.45	14.18	26.36	71.28	79.05	78.53	68.42	65.32
	Llama-3.1	38.12	20.15	32.08	75.42	81.88	80.67	72.35	79.28
	Qwen-2.5	35.27	19.63	31.42	74.87	81.14	79.82	75.83	78.15
FIGA	GPT-2	35.05	14.87	23.01	72.95	81.66	83.33	76.39	69.12
	Llama-3.1	**45.45**	21.03	<u>36.43</u>	77.36	83.21	85.24	<u>81.81</u>	82.45
	Qwen-2.5	41.43	20.08	32.54	76.81	83.14	85.90	80.15	81.04
TLCR	GPT-2	34.54	15.62	24.72	74.22	82.14	82.32	73.56	68.85
	Llama-3.1	43.82	21.34	34.05	79.14	84.62	83.56	80.59	83.92
	Qwen-2.5	40.81	20.76	33.28	79.62	83.28	85.31	78.04	82.16
TRAC	GPT-2	36.59	17.60	25.25	77.88	83.67	87.27	74.40	73.54
	Llama-3.1	<u>44.92</u>	**22.47**	35.92	**81.25**	**85.09**	**90.02**	**83.92**	**87.82**
	Qwen-2.5	43.72	<u>22.04</u>	**36.54**	<u>80.22</u>	<u>84.92</u>	<u>88.11</u>	81.27	<u>86.35</u>

Table 1: Summarization results on CNN/DM and other benchmarks. The best results are highlighted in **bold**, and the second-best are <u>underlined</u>. Metrics include Rouge-1(R-1), Rouge-2(R-2), Rouge-L(R-L). Relevance(Rele), Coherence(Coher), Fluency(Flu), Consistency(Consis) and G-Eval.

achieved a peak score of 87.82, which represented a **10.77%** improvement over PPO_{seq} and maintained an average lead of **5.58%** over FIGA and TLCR. These results underscored that fine-grained reward modeling in TRAC significantly enhanced both linguistic precision and overall semantic quality across diverse architectures.

We further evaluated the proposed TRAC framework on the XL-Sum, NYT Corpus and BookSum datasets to examine its effectiveness in multilingual and long-text summarization scenarios. As shown in Table 2, on the XL-Sum dataset, TRAC achieved state-of-the-art results across coherence, fluency and G-Eval. In particular, it outperformed all baselines with average improvements of **2.26%** and **2.10%** in coherence and fluency, respectively, which demonstrated the robustness of the proposed approach on multilingual data. On the NYT Corpus dataset, TRAC achieved the best results across all evaluation metrics. Notably, it surpassed the second-best model by **4.57%** and **5.43%** on the consistency and fluency metrics, respectively. On the challenging BookSum dataset, TRAC achieved the highest R-1 score, with a substantial gain of **4.57%** in fluency and **4.27%** in coherence compared to the strongest competitor, FIGA. These results provided compelling evidence that TRAC was capable of generating more coherent, fluent, and semantically consistent summaries, particularly for long and complex textual inputs. A detailed analysis of the PRM's intrinsic fitting performance was provided in Section C.5.

5.2 Ablation Experiment

Ablation studies (Table 3) confirmed the necessity of each TRAC module. Replacing the reward mechanism with a Random Reward baseline caused a substantial quality decline—reducing Fluency and Relevance by **21.75%** and **17.12%**, respectively—which highlighted the PRM's critical role in providing effective guidance. Excluding the Length Penalty resulted in severe over-compression; while ROUGE scores were high, average length fell far below the dataset mean (60), leading to an **8.98%** drop in Consistency due to information loss. To investigate the role of Attention Allocation, we replaced it with a uniform reward distribution strategy (w/o AA), which led to a clear performance decline. These results demonstrated that the full TRAC model achieved an optimal balance between linguistic quality, length control, and informational completeness.

5.3 Dialogue generation Benchmark Evaluation

To evaluate the generalization and robustness of TRAC, we further applied it to dialogue generation, which required modeling multi-turn dependencies and context-sensitive responses. This setting enabled a rigorous assessment of TRAC's fine-grained credit assignment in open-ended generation, where sequence-level rewards often strug-

Dataset	Method	R-1 (↑)	R-2 (↑)	R-L (↑)	Coher (↑)	Consis (↑)	Flu (↑)	Rele (↑)	G-Eval (↑)
XL-Sum	SFT	34.50	13.27	23.16	70.21	74.85	73.12	71.45	71.25
	PPO_{seq}	36.05	14.47	25.47	73.78	78.25	76.54	76.07	74.67
	FIGA	**42.26**	**20.39**	**32.89**	75.99	81.19	82.96	80.15	76.39
	TLCR	41.02	18.37	30.35	76.39	**82.14**	82.95	**82.97**	75.14
	TRAC	41.79	18.42	31.73	**78.12**	81.24	**84.70**	81.17	**79.80**
NYT Corpus	SFT	37.84	20.12	28.45	74.55	77.10	79.80	72.37	76.44
	PPO_{seq}	39.63	21.98	30.89	77.95	80.01	82.26	75.49	80.17
	FIGA	43.81	23.79	37.30	80.11	82.94	83.53	81.45	84.29
	TLCR	43.50	23.56	35.39	79.05	83.86	84.62	83.53	84.85
	TRAC	**45.40**	**24.66**	**37.84**	**83.90**	**87.70**	**89.22**	**84.54**	**85.19**
BookSum	SFT	32.14	16.82	19.45	64.82	65.43	72.18	70.64	67.55
	PPO_{seq}	33.56	17.65	21.51	71.34	73.68	75.24	74.12	72.81
	FIGA	35.84	**20.05**	**23.12**	74.28	76.55	78.93	77.21	78.86
	TLCR	34.52	18.41	22.21	73.64	75.82	77.51	76.14	77.92
	TRAC	**36.42**	19.28	22.55	**77.45**	**79.12**	**82.54**	**80.31**	**80.15**

Table 2: Summarization results on multilingual dataset and long-document dataset.

Model	R-1(%↑)	R-2(%↑)	R-L(%↑)	Coher (% ↑)	Consis (% ↑)	Flu (% ↑)	Rele (% ↑)	ΔL (↓)
RR	36.10	16.20	27.50	70.12	71.30	68.45	66.80	21.53
TRAC	44.92	22.47	35.92	**81.25**	**85.09**	**90.02**	**83.92**	**6.33**
w/o LP	**48.62**	**23.63**	**36.47**	79.29	76.11	86.19	82.08	23.19
w/o AA	41.19	19.23	32.57	77.18	83.56	82.66	80.54	18.01

Table 3: Ablation experiment results on the CNN/DM dataset. RR (*Random Reward*), LP (*Length Penalty*), and AA (*Attention Allocation*). ΔL denotes the absolute difference from the average reference summary length (72).

gled to maintain logical continuity.As shown in Table 5, TRAC achieved the best performance across all metrics, with particularly strong improvements in *Fluency* and *Relevance*. Compared with PPO_{seq}, TRAC improved *Fluency* and *Relevance* by **14.21%** and **13.42%**, respectively; gains of **6.58%** and **3.88%** were observed over FIGA, and **4.24%** and **2.71%** over TLCR. TRAC also yielded substantial improvements in *Coherence*, outperforming the three baselines by **18.03%**, **8.91%**, and **6.95%**, respectively. Improvements in *Consistency* were more moderate (3–6%) but remained stable. Overall, these results demonstrated that the proposed token-level reward mechanism provided effective fine-grained optimization signals, leading to consistent improvements in dialogue quality.

5.4 Human Evaluation

In the human evaluation, ten graduate students from computer science, law, and linguistics assessed 500 summaries using a 0–10 scale. As illustrated in Figure 3, TRAC outperformed all baselines on Faithfulness, Fluency, Conciseness, and Informativeness, with particularly strong gains in Fluency and Conciseness (both approaching 9). In contrast, PPO_{seq} achieved the lowest scores, especially on Fluency and Conciseness, which highlighted the limitations of sequence-level reward design. A detailed inter-annotator agreement analysis was provided in Section D.

SFT was the most cost-effective due to its supervised nature, while FIGA and TLCR incurred overhead from data dependencies and external labeling. TRAC optimized the RL process by leveraging an autoregressive PRM to provide dense token-level supervision, which accelerated policy convergence and reduced total training time. In contrast, PPO_{seq} was the most resource-intensive due to frequent external reward queries. A comprehensive analysis of efficiency is elaborated in Section C.6.

5.5 LLM Evaluation

Figure 4 reported the evaluation results for 2,000 summarization and dialogue test cases. The evaluation paired model outputs and tasked GPT-4o with judging preferences, where a "Tie" indi-

σ	α	R-1(%↑)	R-2(%↑)	R-L(%↑)	Coher (% ↑)	Consis (% ↑)	Flu (% ↑)	Rele (% ↑)	ΔL (↓)
*	0.01	42.09	21.50	34.74	78.15	84.28	87.52	81.88	6.91
*	0.05	43.85	21.59	35.63	80.38	83.58	89.99	83.91	3.68
*	0.15	44.73	22.04	35.11	**81.63**	81.22	89.87	76.53	9.92
1	*	42.45	20.32	34.15	77.74	82.51	88.15	82.46	11.63
5	*	43.68	21.03	34.29	77.13	84.57	89.01	83.10	8.71
15	*	44.14	**22.83**	35.73	80.06	83.43	87.06	81.11	1.71
*	*	**44.92**	22.47	**35.92**	81.25	**85.09**	**90.02**	**83.92**	**1.33**

Table 4: Ablation experiment results on the CNN/DM dataset. * indicates the default parameter (σ=0.1, α=10).

Method	Model	Rele	Coher	Flu	Consis
SFT	Llama-3.1	69.82	70.57	71.93	73.88
	Qwen-2.5	72.41	71.24	72.96	75.12
	GPT-2	68.57	67.73	67.31	72.06
PPO_{seq}	Llama-3.1	72.61	71.45	73.64	76.72
	Qwen-2.5	74.33	73.09	74.81	77.00
	GPT-2	70.42	70.56	71.16	75.93
FIGA	Llama-3.1	81.71	80.22	81.25	79.62
	Qwen-2.5	79.56	78.71	78.56	78.36
	GPT-2	76.02	74.23	75.57	77.27
TLCR	Llama-3.1	82.38	81.12	82.87	80.64
	Qwen-2.5	80.55	79.57	80.29	78.59
	GPT-2	77.06	76.70	77.49	77.68
TRAC	Llama-3.1	**85.59**	**87.77**	**85.03**	81.11
	Qwen-2.5	81.70	84.67	83.91	**82.68**
	GPT-2	79.22	81.46	81.83	80.79

Table 5: Dialogue generation results on four metrics. TRAC significantly outperforms baseline models, especially in coherence and fluency.

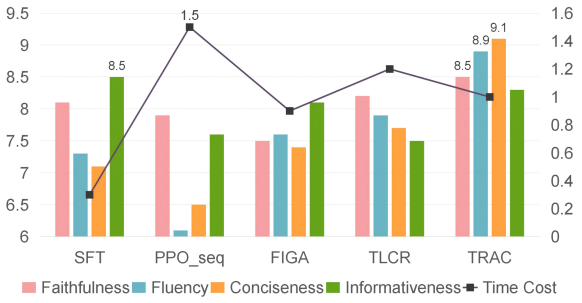

Figure 3: Bar chart representing human evaluations on CNN/DM dataset across four dimensions. The line graph indicates the time cost associated with training.

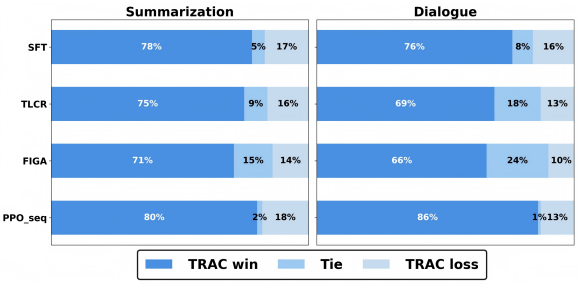

Figure 4: GPT-4o evaluation on CNN/DM and DailyDialog datasets

cated no clear winner. In summarization, TRAC achieved win rates of **78%**, **75%**, **71%**, and **80%** over SFT, TLCR, FIGA, and PPO_{seq}, respectively. In dialogue, TRAC attained win rates of **76%**, **69%**, **66%**, and **86%** against the same baselines. These results shown that TRAC consistently outperformed mainstream baselines on both tasks, with particularly substantial gains observed in dialogue.

5.6 Hyperparameter Experiment

As shown in Table 4, increasing α strengthened the length penalty and yielded shorter summaries, though excessive α led to over-compression and degraded consistency. The parameter σ governed the penalty's sensitivity: while smaller σ imposed stricter constraints that often caused information loss, larger σ allowed for better alignment with reference lengths at the cost of fluency and relevance. Ultimately, these results underscored the necessity of balancing α and σ to optimize overall quality.

6 Conclusion

In this paper, we introduced TRAC, a reinforcement learning framework designed to enhance inter-sentence fluency and coherence in abstractive summarization. By integrating sentence-level gain, inter-sentence attention, and a Gaussian length penalty, TRAC constructs a fine-grained token-level reward mechanism that effectively balances local semantic precision with global structural consistency. Our autoregressive process reward model provides dense, context-aware feedback, enabling the model to navigate complex generation spaces with superior logical alignment. Extensive experiments across multiple benchmarks demonstrate that TRAC significantly improves the output quality and coherence of large language models.

7 Limitations

Despite its performance gains, this study has several limitations. First, although TRAC's attention-based token allocation is inherently **heuristic**, ablation studies have confirmed its superiority over simpler alternatives, such as uniform or randomized allocation. However, the current mechanism still relies on fixed assumptions regarding attention-weight correlations. Future work will focus on investigating more **principled or adaptive token-level reward distribution strategies** to further enhance evaluation precision beyond current heuristics.

Second, computational constraints restricted our experiments to medium-scale models, leaving the scalability of TRAC to large-scale architectures (e.g., >70B parameters) and advanced frameworks like Generalized Reward Policy Optimization (GRPO) for future exploration. Finally, TRAC's robustness in handling highly noisy, real-world data remains to be fully verified. Subsequent research will focus on distributed computing and more efficient RL algorithms to enhance the framework's applicability in complex, large-scale scenarios.

Acknowledgments

Our work is supported by the National Social Science Fund of China (No. 22BTQ045), the Science and Technology Project of Guangzhou Economic and Technological Development Zone (No. 2023GH18), and the Key-Area Research and Development Program of Guangzhou (No. 2024B01W0029).

References

Wael M Aly, Tamer H A Soliman, and A A M AbdelAziz. 2025. An evaluation of large language models on text summarization tasks using prompt engineering techniques. *arXiv preprint arXiv:2507.05123*.

David Bawden and Lyn Robinson. 2020. Information overload: An overview. *Aslib Journal of Information Management*, 72:8–21.

Antoine Bosselut, Asli Celikyilmaz, Xiaodong He, and 1 others. 2020. Discourse-aware neural rewards for coherent text generation. In *Proceedings of NAACL-HLT*, pages 173–184.

Abhimanyu Dubey, Anchal Jauhri, Anshul Pandey, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, page arXiv:2407.21783.

Fikri B Fikri, Kemal Oflazer, and Başak Yanıkoğlu. 2024. Abstractive summarization with deep reinforcement learning using semantic similarity rewards. *Natural Language Engineering*, 30:554–576.

Dongdong Fu, Tianyu Xiao, Ruipeng Wang, and 1 others. 2025. Tldr: Token-level detective reward model for large vision language models. In *The Thirteenth International Conference on Learning Representations*.

Tirthankar Hasan, Abhik Bhattacharjee, Mohammad S Islam, and 1 others. 2021. XL-Sum: large-scale multilingual abstractive summarization for 44 languages. In *Annual Meeting of the Association of Computational Linguistics and International Joint Conference on Natural Language Processing 2021*, pages 4693–4703. Association for Computational Linguistics (ACL).

Chao Huang, Zeqi Wu, Yanfei Hu, and 1 others. 2024. Training language models to generate text with citations via fine-grained rewards. *arXiv preprint arXiv:2407.13529*.

Feng Jiao, Chao Qin, Zhiyuan Liu, and 1 others. 2024. Learning planning-based reasoning by trajectories collection and process reward synthesizing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 334–350.

Shafiq Khan, Md Serajuddin, Zulfiquer Hasan, and 1 others. 2023. Natural language generation (nlg) with reinforcement learning (rl). In *International Conference on Artificial Intelligence and Speech Technology*, pages 303–318. Springer Nature Switzerland.

Fajar Koto, Timothy Baldwin, and Jey Han Lau. 2022. FFCI: A framework for interpretable automatic evaluation of summarization. *Journal of Artificial Intelligence Research*, 73:1553–1607.

Wojciech Kryściński, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. 2022. Booksum: A collection of datasets for long-form narrative summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6536–6558.

Wenjie Li and Yanan Li. 2025. Process reward model with q-value rankings. In *The Thirteenth International Conference on Learning Representations*.

Yizhe Li, Houkun Su, Xiaoya Shen, and 1 others. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995.

Hannah Lightman, Vishaal Kosaraju, Yuri Burda, and 1 others. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

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

Yang Liu, Dan Iter, Shuo Xu, Shuohang Wang, Ruochen Zhu, Kevin Aziz, Haoyang Huang, Furu Wei, Michael Zeng, and Daxin Huang. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 577–591.

Sungjin Park, Xiang Liu, Yuntao Gong, and 1 others. 2025. Ensembling large language models with process reward-guided tree search for better complex reasoning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10256–10277.

Alec Radford, Jeff Wu, Rewon Child, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1:9.

Rohit Ramamurthy, Prithviraj Ammanabrolu, Kristin Brantley, and 1 others. 2023. Is reinforcement learning (not) for natural language processing: Benchmarks, baselines, and building blocks for natural language policy optimization. In *The Eleventh International Conference on Learning Representations*.

Sooyoung Ryu, Hyeonah Do, Yunju Kim, and 1 others. 2024. Multi-dimensional optimization for text summarization via reinforcement learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5858–5871.

Evan Sandhaus. 2008. *The New York Times Annotated Corpus*. Linguistic Data Consortium, Philadelphia. LDC2008T19.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.

Noah Shinn, Federico Cassano, Ashwin Gopinath, and 1 others. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652.

Qwen Team. 2024. Qwen2.5 Technical Report. Alibaba Cloud. Accessed: 2025-09-27.

Hugo Touvron, Louis Martin, Kevin Stone, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *article preprint article:2307.09288*.

Zeqi Wu, Yanfei Hu, Weishui Shi, and 1 others. 2023. Fine-grained human feedback gives better rewards for language model training. *Advances in Neural Information Processing Systems*, 36:59008–59033.

Eunseop Yoon, Hee Suk Yoon, SooHwan Eom, Gunsoo Han, Daniel Nam, Daejin Jo, Kyoung-Woon On, Mark Hasegawa-Johnson, Sungwoong Kim, and Chang Yoo. 2024. Tlcr: Token-level continuous reward for fine-grained reinforcement learning from human feedback. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14969–14981.

Lin Yuan, Wei Li, Hao Chen, and 1 others. 2023. Free process rewards without process labels. In *Forty-second International Conference on Machine Learning*.

Thomas Zeng, Shuibai Zhang, Shutong Wu, Christian Classen, Daewon Chae, Ethan Ewer, Minjae Lee, Heeju Kim, Wonjun Kang, Jackson Kunde, and 1 others. 2025. Versaprm: Multi-domain process reward model via synthetic reasoning data. In *International Conference on Machine Learning*, pages 74197–74239. PMLR.

Deyi Zhang, Shuhao Zhoubian, Zheng Hu, and 1 others. 2024. Rest-mcts*: Llm self-training via process reward guided tree search. *Advances in Neural Information Processing Systems*, 37:64735–64772.

Lianmin Zheng, Wei-Lin Chiang, Yunan Sheng, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Ming Zhong Zhong, Yixuan Liu, Da Yin, and 1 others. 2022. Towards a unified multi-dimensional evaluator for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2023–2038.

Appendix

A Construction Algorithm of the TRAC

The overall optimization process of **TRAC** is illustrated in Algorithm 1. The framework begins with a supervised fine-tuning stage, where the language model is trained to maximize the likelihood of reference summaries. In the second stage, token-level rewards are constructed by decomposing sentence-level quality gains into fine-grained token contributions, further regularized by a Gaussian length penalty to ensure coherent and appropriately sized generations. These rewards are then used to train a Process Reward Model that predicts token-level quality signals. Finally, in the reinforcement learning stage, the PRM-predicted rewards serve as guidance for Proximal Policy Optimization (PPO), enabling stable policy updates and enhancing the fluency, coherence, and structural consistency of the generated summaries.

B Experimental Settings

B.1 Datasets

For the summarization task, we employed the **CNN/DailyMail** (CNN/DM) dataset (See et al.,

Algorithm 1 The TRAC Framework

Require: Pre-trained model $\mathcal{M}_\theta$, dataset $\mathcal{D} = \{(x, y)\}$, reference length L_{ref}, PRM f_ϕ
Ensure: Optimized policy parameters θ^*

// Stage 1: Supervised Fine-Tuning (SFT)

$\theta \leftarrow \arg\min_\theta \mathbb{E}_{(x,y)\in\mathcal{D}}[-\sum_t \log P_\theta(y_t|y_{<t}, x)]$ ▷ Warm-up policy

// Stage 2: Token-Level Reward Derivation & PRM Training

Sample summary $\mathbf{S} = \{s_1, \ldots, s_n\}$ from $P_\theta(\cdot|x)$

for each sentence $s_i \in \mathbf{S}$ **do**

  Compute $R_{\text{gain}}(s_i) = Q(S_{\leq i}) - Q(S_{\leq i-1})$ with length-scaling factor

  Extract attention weights $w_{i_j} = \text{Softmax}(\text{Standardize}(A_i))$

  Assign $r_t = w_{i_j} \cdot R_{\text{gain}}(s_i) + P_t^{\text{len}}(L_{\text{ref}})$ ▷ Construct ground-truth rewards

end for

$\phi \leftarrow \arg\min_\phi \frac{1}{T}\sum_{t=1}^{T}(f_\phi(h_{\leq t}) - r_t)^2$ ▷ Train Process Reward Model

// Stage 3: Proximal Policy Optimization (PPO)

while not converged **do**

  Sample trajectories using current π_θ and predict rewards $\hat{r}_t = f_\phi(h_{\leq t})$

  Compute TD-error $\delta_t = \hat{r}_t + \gamma V_{t+1} - V_t$

  Calculate GAE advantage $A_t = \sum_{l=0}^{\infty}(\gamma\lambda)^l\delta_{t+l}$

  Update θ by maximizing $\mathcal{L}_{\text{PPO}} = \mathbb{E}[\min(\rho_t A_t, \text{clip}(\rho_t, 1-\epsilon, 1+\epsilon)A_t)]$

end while

return $\theta^* \leftarrow \theta$

Table 6: Statistical summary of summarization datasets including total document counts and average token lengths for both source texts and summaries.

Dataset	# Docs	Source	Summary
CNN/DM	311,971	803	59
NYT Corpus	589,000	1,180	78
XL-Sum	301,444	463	29
BookSum	12,630	5,101	505

2017), a widely used benchmark consisting of large-scale news articles paired with human-written summaries. The texts originate from real-world news reports, making the dataset authentic, diverse in topic, and of moderate length, thus well-suited for evaluating a model's ability to compress and abstract long-form news content. In addition, we incorporated **XL-Sum** (Hasan et al., 2021) and the **New York Times** (NYT) Corpus (Sandhaus, 2008)to enhance coverage of multilingual and long-document scenarios. XL-Sum provides multilingual news summaries for testing cross-lingual adaptability, while the NYT Corpus contains lengthy, in-depth articles, making it valuable for assessing a model's performance on complex long-text summarization. Furthermore, we included **BookSum** (Kryściński et al., 2022), a comprehensive dataset specifically designed for long-form narrative summarization. Unlike news-based corpora, BookSum features hierarchical data spanning paragraphs, chapters, and entire books sourced from public domain literature. Detailed statistical characteristics of the datasets are summarized in Table 6.

For dialogue generation, we used the **DailyDialog** dataset (Li et al., 2017), a rich collection of everyday conversations covering emotional expression and information exchange, making it a strong benchmark for assessing fluency and coherence in dialogue generation.

B.2 BaseLine Models

To evaluate the proposed method's efficacy, we benchmarked it against three representative generative models: Llama-3.1-8B (Dubey et al., 2024), Qwen-2.5-7B (Team, 2024), and GPT-2 (Radford et al., 2019). These models provide a diverse spectrum of architectures and capabilities: Llama-3.1-8B serves as a high-efficiency lightweight LLM; Qwen-2.5-7B offers robust multi-task and multilingual performance; and GPT-2 acts as a foundational autoregressive baseline. This selection spans multiple modeling paradigms and resource scales, ensuring a comprehensive and rigorous comparative analysis.

In addition, we compared against multiple token-level reward baselines. **TLCR** (Yoon et al., 2024)is

a token-level continuous reward mechanism designed for RLHF. It trains a discriminator on token preference labels generated by an external LLM and normalizes its confidence scores to continuous rewards ranging from -1 to 1, thereby improving performance in open-ended generation tasks. **FIGA** (Ramamurthy et al., 2023), on the other hand, is an alignment approach between language models and human preferences. By constructing the SPA dataset and leveraging the Levenshtein distance to compare initial and revised responses, it assigns positive reward weights to inserted or replaced tokens, and negative penalties to deleted or substituted tokens, thereby generating token-level rewards. Unlike fine-grained reward models, **PPO**$_{seq}$ serves as a baseline that employs a sparse reward mechanism, where the global sequence score is applied exclusively to the terminal valid token. This signal is further augmented by per-token KL-divergence penalties to maintain distributional stability during alignment.

B.3 Evaluation Metircs

B.3.1 Automatic metrics

In our experiments, we selected evaluation metrics tailored to each task. For summarization, we adopted **ROUGE** (Lin, 2004) and **UniEval** (Zhong et al., 2022)[1]. ROUGE measures n-gram and longest common subsequence overlaps to assess coverage, while UniEval leverages large language models(T5-Large) to provide a more comprehensive evaluation across fluency, consistency, and informativeness. For dialogue generation, we employed UniEval as the primary metric, as it integrates contextual information to evaluate fluency, coherence, and relevance, making it particularly suitable for capturing semantic continuity and logical consistency in multi-turn conversations. Furthermore, we incorporated **G-Eval** (Liu et al., 2023), which utilizes LLMs with Chain-of-Thought prompting and formulates evaluation as a form-filling task. By calculating the expected value of scores based on token probabilities, G-Eval provides a more granular assessment that demonstrates superior correlation with human judgment in nuanced quality dimensions. G-Eval prompt are detailed in Figure 8.

[1] https://github.com/maszhongming/UniEval

Table 7: Hyperparameter table

Hyperparameter	Value
Learning Rate	2e-5
SFT Epoch	50
SFT Batch Size	32
PPO Epoch	5
PPO Batch Size	16
Optimizer	Adam
c	0.05
σ	10
α	0.1

B.3.2 Human Metrics

For human evaluation, we adopt the FFCI framework (Koto et al., 2022), which assesses summaries across four dimensions: **Faithfulness** (fidelity to the source), **Fluency** (grammaticality and readability), **Conciseness** (brevity and non-redundancy), and **Informativeness** (coverage of key content). This complements automatic metrics with subjective judgments of quality.

B.4 Experimental Environment

Our experiments were conducted on Ubuntu 20.04.6 LTS, with hardware equipped with NVIDIA GeForce RTX 3090 and NVIDIA A30 GPUs. To prevent overfitting and optimize training efficiency, we implemented an early stopping strategy based on the convergence of the validation loss. We used Python 3.8 as the programming language and PyTorch 2.3.0 with CUDA 11.8 for model development and accelerated training. The detailed parameter settings are summarized in Table 7.

C Process Reward Model

In this section, we provide the detailed architecture of the **Process Reward Model**, which is designed to assign token-level rewards based on the hidden representations of the summarization model.

C.1 Hidden Representation Extraction

Let the summarization model be parameterized by θ, and a given input sequence be denoted as :

$$X = \{x_1, x_2, \ldots, x_T\}, \tag{17}$$

where x_t represents the t-th token and T is the sequence length. After forward propagation through the encoder-decoder architecture, we obtain the

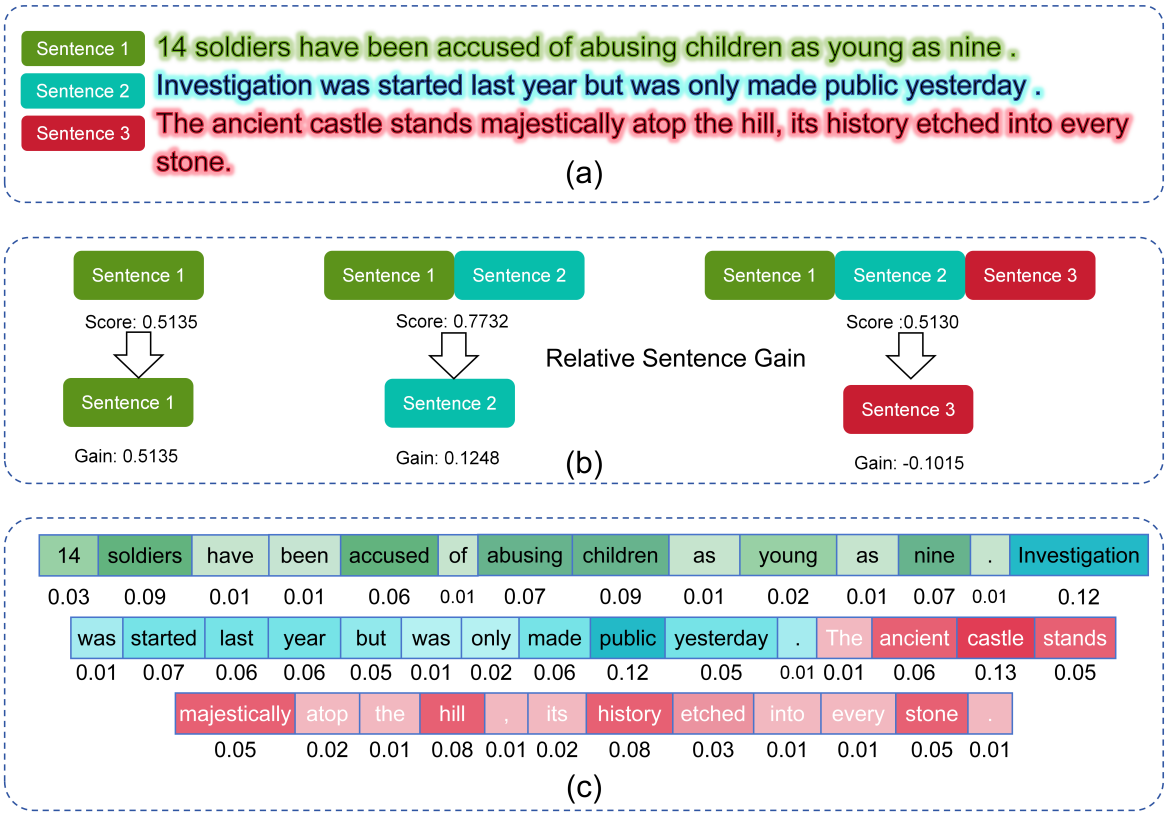

Figure 5: Illustration of sentence gain scoring and token-level reward generation. (a) Example with three sentences from a web news. (b) Relative sentence gain calculation based on incremental scores. (c) Fine-grained token-level reward assignment guided by sentence-level gain and attention scores.

hidden state vectors at each decoding step:

$$H = \{h_1, h_2, \ldots, h_T\}, \quad h_t \in \mathbb{R}^d, \tag{18}$$

where h_t denotes the d-dimensional hidden representation of token x_t at the decoder output layer before projection. These hidden states encapsulate both contextual and syntactic information of the generated summary.

C.2 Token-Level Reward Modeling

The PRM is constructed as a lightweight autoregressive decoder that takes H as input and predicts a token-wise reward score sequence.

$$R = \{r_1, r_2, \ldots, r_T\}, \quad r_t \in \mathbb{R}. \tag{19}$$

At each time step t, the model estimates the conditional reward probability:

$$p(r_t \mid r_{<t}, H) = \text{Decoder}_\phi(h_t, r_{<t}), \tag{20}$$

where ϕ denotes the parameters of the PRM decoder, and $r_{<t}$ represents the previously generated rewards $\{r_1, \ldots, r_{t-1}\}$.

C.3 Autoregressive Decoder Architecture

The decoder is composed of L stacked transformer blocks, each consisting of:

- **Masked Self-Attention Layer:** captures the temporal dependencies among previously predicted rewards. For the l-th layer, the attention output is computed as

$$\text{Attn}^{(l)} = \text{Softmax}\left(\frac{Q^{(l)}(K^{(l)})^\top}{\sqrt{d_k}} + M\right)V^{(l)}, \tag{21}$$

  where $Q^{(l)}, K^{(l)}, V^{(l)}$ are the query, key, and value matrices of dimension d_k, and M is the causal mask ensuring autoregressive prediction.

- **Cross-Attention Layer:** aligns the decoder states with the hidden representations H from the base summarization model, enabling the PRM to leverage semantic and structural information from h_t.

- **Feed-Forward Network (FFN):** applies two linear transformations with a non-linear acti-

vation (e.g., GELU) in between:

$$\mathrm{FFN}(x) = W_2\,\sigma(W_1 x + b_1) + b_2, \quad (22)$$

where W_1, W_2 are learnable weight matrices and $\sigma(\cdot)$ is the activation function.

- **Residual Connections and Layer Normalization:** each sublayer output is wrapped with a residual connection and layer normalization:

$$x^{(l+1)} = \mathrm{LayerNorm}\left(x^{(l)} + \mathrm{Sublayer}(x^{(l)})\right). \quad (23)$$

C.4 Reward Prediction and Training Objective

After the final decoder layer, the model projects the hidden state z_t at each timestep into a scalar reward value:

$$r_t = W_r z_t + b_r, \quad (24)$$

where $W_r \in \mathbb{R}^{1\times d}$ and b_r are trainable parameters.

The training objective is to minimize the mean squared error (MSE) between the predicted rewards and the target token-level reward signals:

$$\mathcal{L}_{\text{PRM}} = \frac{1}{T}\sum_{t=1}^{T}(r_t - \hat{r}_t)^2, \quad (25)$$

where $\hat{r}_t$ denotes the reference reward signal derived from heuristic or external evaluation functions.

By integrating hidden representations H and autoregressive reward estimation, the PRM captures both local and global dependencies of the generated summary. Its design allows fine-grained token-level reward modeling, which serves as a crucial supervision signal in the reinforcement learning phase for optimizing generation quality.

C.5 Intrinsic Evaluation

Table 8: Performance comparison of different token-level process reward models. To account for the fine-grained reward scale ($0.01 \sim 0.1$), the MSE values are scaled by 10^{-4}. The best results are highlighted in **bold**.

Model	MSE (↓)	Spearman ρ (↑)	KL (↓)
FIGA	32.56	0.438	0.482
TLCR	15.89	0.712	**0.356**
TRAC	**12.14**	**0.824**	0.373

As illustrated in Table 8, the proposed TRAC significantly outperforms all competitive token-level baselines across MSE and Spearman evaluation metrics, demonstrating an exceptional capability in fitting fine-grained reward signals within the nuanced range of $0.01 \sim 0.1$. The superiority of TRAC primarily stems from its autoregressive architecture, which, unlike the heuristic-based approach of FIGA or the discriminative-based confidence scoring of TLCR, inherently aligns with the sequential logic of text generation. This generative design enables the model to effectively capture complex, step-wise dependencies and cumulative contextual information, resulting in more precise credit assignment as evidenced by the high Spearman correlation ($\rho = 0.824$). Furthermore, the lower KL divergence indicates that TRAC maintains superior distributional stability, effectively mitigating common pitfalls such as reward sparsity or over-optimization, thereby providing a more reliable and granular guidance signal for fine-grained alignment tuning.

C.6 Efficiency and Computational Overhead Analysis

We evaluate the computational efficiency of the proposed TRAC framework by comparing its latency, maximum batch size (BS), and throughput against existing baselines on a single GPU with 24G VRAM. The results are summarized in Table 9.

Table 9: Comparison of Performance Metrics. BS denotes Max Batch Size (24G VRAM), Throughput is measured in Items/s, and Latency is measured in s/sample.

Model	Latency(↓)	BS(↑)	Throughput(↑)
TLCR	∼0.7s	∼32	∼45
FIGA	∼**0.5s**	∼**40**	∼**82**
TRAC	∼0.6s	∼**40**	∼66

As shown in the experimental results, TRAC demonstrates competitive operational efficiency while maintaining high-granularity evaluation capabilities. Specifically, TRAC achieves a latency of approximately 0.6s per sample, which is a significant improvement over the 0.7s recorded by TLCR. Regarding memory utilization, both FIGA and TRAC exhibit superior scalability, supporting a maximum batch size of ∼40, whereas TLCR is limited to ∼32.

While FIGA maintains a marginal lead in raw throughput (∼82 items/s vs. ∼66 items/s for TRAC), the slight computational overhead of

TRAC is justified by its enhanced qualitative performance in downstream tasks. As TRAC provides a more robust token-level reward signal without a prohibitive increase in latency, it represents a more favorable trade-off between inference speed and generation quality for real-world deployment in summarization and dialogue systems.

D Human Evaluation Consistency

Table 10: Inter-annotator agreement results for human evaluation. We report Krippendorff's α (interval), Fleiss' κ, and ICC(2,1) across all five metrics.

Model	Kripp α (↑)	Fleiss κ (↑)	ICC (↑)
SFT	0.762	0.714	0.785
PPO$_{seq}$	0.618	0.582	0.641
FIGA	0.725	0.689	0.742
TLCR	0.784	0.746	0.801
TRAC	**0.826**	**0.792**	**0.873**

To validate the reliability of the human evaluation results, we conducted an inter-annotator agreement (IAA) analysis using Krippendorff's α, Fleiss' κ, and the Intraclass Correlation Coefficient (ICC). As summarized in Table 10, TRAC consistently achieves the highest consensus among the ten graduate students across all metrics (e.g., $\alpha = 0.826$, $ICC = 0.873$), indicating a "near-perfect" level of agreement that significantly surpasses the baselines. In contrast, PPO$_{seq}$ exhibits the lowest consistency, likely due to the inherent instability of sparse sequence-level rewards which often produce outputs with varying degrees of quality, leading to greater subjective divergence among experts from different disciplines. The superior agreement scores for TRAC suggest that effectively produces more coherent and objectively high-quality summaries.

E Case Study

Figure 5 presented the fine-grained visualization of TRAC, with detailed results provided in the Appendix. As illustrated in Figures 5(a) through 5(c), the framework mapped sentence-level gain scores—such as the positive gain of the second sentence and the negative gain of the third—directly to the token-level reward distribution. Notably, the third sentence exhibited lower token rewards due to its negative gain, while the first sentence, despite its positive gain, was moderated by the Gaussian length penalty to prevent over-generation. This hierarchical mechanism effectively guided the model to maintain an optimal balance between informational coherence and length control.

F Prompt Design

Instruction:
You are an expert summarizer specialized in producing accurate, concise, and coherent summaries for academic or factual content.
Your task:
Given a passage of text, generate a summary that precisely captures the essential information, maintaining factual accuracy, logical flow, and readability. Avoid adding any unverified or speculative details.
Requirements:
1. Accuracy: Ensure all statements in the summary are directly supported by the source text. Preserve the original meaning and do not infer or exaggerate.
2. Conciseness: Eliminate redundant, trivial, or repetitive information. Aim for a compact summary (e.g., 20–30% of the original length).
3. Coherence: Organize information logically to ensure smooth transitions between sentences. Use consistent tense and referential clarity.
Output format:
Write one cohesive paragraph (3–5 sentences). Do not include introductory phrases like "This text discusses…" or "The passage describes…". Directly summarize the key content.

Figure 6: Illustration of the prompt design for text summarization tasks.

Instruction:
You are an intelligent dialogue model designed to generate natural, coherent, and contextually consistent multi-turn conversations.
Your task:
Continue or generate a dialogue that maintains logical coherence, consistency in persona and knowledge, and fluency in expression. Each response should flow naturally from the previous turns.
Requirements:
1. Accuracy: Each utterance must logically follow the previous one. Maintain clear topic continuity without abrupt shifts.
2. Conciseness: Keep the speaker's tone, style, and background consistent across turns. Ensure facts or opinions expressed earlier are not contradicted later.
3. Fluency: Use natural, smooth, and contextually appropriate language. Avoid grammatical errors, repetitions, or overly complex phrasing.
Output format:
Produce a dialogue with alternating speaker turns labeled as "A: " and "B:".

Figure 7: Illustration of the prompt design for dialogue generation tasks.

In this section, we present the prompts employed for guiding large language models in generating text summaries and dialogue responses.

F.1 Prompt for Summarization

In text summarization tasks, prompts are designed to instruct the model to generate concise, informative, and coherent summaries that preserve the

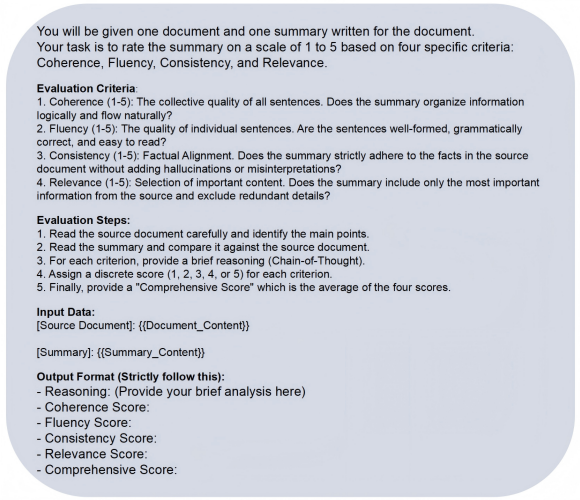

Figure 8: Illustration of the prompt design for G-Eval.

essential content of the source text while minimizing redundancy.

Figure 6 provides a visual illustration of the summarization prompt structure.

F.2 Prompt for Dialogue Generation

For dialogue generation, prompts are crafted to maintain coherence, consistency, and naturalness across multiple conversational turns.

Figure 7 shows a schematic representation of the dialogue prompt.

F.3 Prompt for G-Eval

For G-Eval, the prompt is structured to elicit a multi-dimensional assessment of generation quality through Chain-of-Thought (CoT) reasoning. It defines specific rubrics for Coherence, Fluency, Consistency, and Relevance, guiding the evaluator model to first generate a rationalization and then assign numerical scores. This systematic decomposition ensures that the evaluation remains objective and aligns closely with human judgment.

Figure 8 illustrates the schematic representation of the G-Eval prompt and its integrated scoring workflow.