SARA: Salience-Aware Reinforced Adaptive Decoding for Large Language Models in Abstractive Summarization

Nayu Liu[1] ✉ , **Junnan Zhu**[2], **Yiming Ma**[3], **Zhicong Lu**[4], **Wenlei Xu**[1], **Yong Yang**[1], **Jiang Zhong**[5], **Kaiwen Wei**[5*] ✉

[1]Tianjin Laboratory Autonomous Intelligence Technology and Systems, School of Computer Science and Technology, Tiangong University

[2]State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, CAS

[3]Department of Data Science & Artificial Intelligence, The Hong Kong Polytechnic University

[4]University of Chinese Academy of Sciences, [5]College of Computer Science, Chongqing University

Abstract

LLMs have improved the fluency and informativeness of abstractive summarization but remain prone to hallucinations, where generated content deviates from the source document. Recent PMI decoding strategies mitigate over-reliance on prior knowledge by comparing output probabilities with and without source documents, effectively enhancing contextual utilization and improving faithfulness. However, existing strategies often neglect the explicit use of salient contextual information and rely on static hyperparameters to fix the balance between contextual and prior knowledge, limiting their flexibility. In this work, we propose **S**alience-**A**ware **R**einforced **A**daptive decoding (SARA), which incorporates salient information and allows the model to adaptively determine reliance on the source document's context, salient context, and the model's prior knowledge based on pointwise mutual information. Moreover, a tokenwise adaptive decoding mechanism via reinforcement learning is proposed in SARA to dynamically adjust the contributions of context and prior knowledge at each decoding timestep. Experiments on CNN/DM, WikiHow, and NYT50 datasets show that SARA consistently improves the quality and faithfulness of summaries across various LLM backbones without modifying their weights.

1 Introduction

Abstractive summarization (Zhang et al., 2020, 2024; Jin et al., 2024; Liu and Lapata, 2019; Ryu et al., 2024a,b) aims to generate concise and informative texts from input documents. The emergence of recent large language models (LLMs) (Zhang et al., 2024; Lv et al., 2024) has improved the fluency and informativeness of abstractive summarization. Despite the impressive performance of LLMs, they still suffer hallucinations (or unfaithfulness) (Li et al., 2024b; Xia et al., 2024), where the generated content misrepresents or distorts the information from the origin documents. This can be attributed to the fact that summaries rely on both the prior knowledge embedded in pre-trained models and the context from input documents, where an over-reliance on the prior knowledge or insufficient utilization of the context impacts the faithfulness of the generated content (Shi et al., 2024).

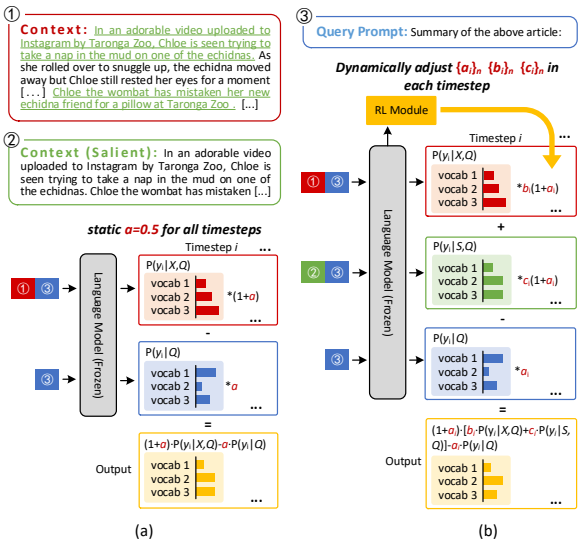

Figure 1: (a) Previous point mutual information (PMI) based decoding; (b) The proposed SARA.

Existing methods employ various model training strategies to alleviate hallucinations, such as enhancing the faithfulness of specific entities/tokens (Shen et al., 2023; Dong et al., 2022) or suppressing the generation of irrelevant content through contrastive learning (Chen et al., 2023b; Choubey et al., 2023). Considering that altering model weights may degrade the general capabilities of LLMs, some works have sought to improve the model during the inference (Lee et al., 2022; Wan et al., 2023); Notably, pointwise mutual information (PMI) decoding strategies (Chae et al., 2024; Xu, 2023; Shi et al., 2024; Van Der Poel et al., 2022) are proposed to reduce the influence of

*Corresponding author (email: weikaiwen@cqu.edu.cn).

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 25450–25463
July 27 - August 1, 2025

prior knowledge that may produce hallucinations by comparing the difference between the output probabilities with and without source documents, which allows generated content to be more faithful.

Despite the progress in PMI decoding, there remain challenges as illustrated in Figure 1: (1) How can the model effectively select salient contextual knowledge for decoding, especially how can it allocate greater attention to supportive content to enhance summary faithfulness (Wang et al., 2022; Aralikatte et al., 2021), while previous decoding strategies often neglect to explicitly leverage salient information. (2) How can the model dynamically determine whether to rely on contextual or prior knowledge during decoding. Existing approaches typically rely on static hyperparameters to assign fixed weights to output probabilities at the sequence-level, which offers limited flexibility.

To tackle these issues, we propose **S**alience-**A**ware **R**einforced **A**daptive decoding (SARA) for LLM-based abstractive summarization. SARA guides the decoding by incorporating salient information and allowing the model to decide, at each generation step, whether to rely on the source document's context, salient information, or its prior knowledge. SARA reduces hallucinations by comparing output probabilities with and without the source document, ensuring a more faithful summary. It uses an extraction model to identify key contextual information and enhances faithfulness by leveraging point mutual information between the salient context and prior knowledge. Unlike previous methods that use fixed sequence-level weights to combine probabilities, SARA adopts a tokenwise adaptive decoding mechanism based on reinforcement learning. This allows the model to dynamically adjust the influence of different knowledge sources at each step, improving the quality and faithfulness of the generated summaries.

Extensive experimental results on CNN/DM (Nallapati et al., 2016), WikiHow (Koupaee and Wang, 2018), and NYT50 (Durrett et al., 2016) datasets, have shown that SARA effectively improves the quality and faithfulness of summaries generated by various LLMs (e.g., GPT-Neo (Black et al., 2021), LLaMA (Touvron et al., 2023), OPT (Zhang et al., 2022b), and Mistral (Jiang et al., 2023)) without requiring any modifications to the LLM weights. Our contributions can be summarized as follows:

1) We propose Salience-Aware Reinforced Adaptive decoding for LLM-based abstractive summarization, which helps the model decide how to rely on the source context, salient context, or prior knowledge based on point mutual information.

2) We introduce a tokenwise adaptive decoding mechanism via reinforcement learning, enabling the model to dynamically balance contextual and prior knowledge at each decoding step.

3) SARA achieves consistent improvements in summary quality and faithfulness across CNNDM, WikiHow, and NYT50 datasets on multiple LLM backbones without modifying model weights[1].

2 Preliminaries

2.1 Problem Definition

Given a language model θ, a source document $X = \{x_1, ..., x_m\}$, and a query prompt $Q = \{q_1, ..., q_n\}$, the model produces a summary $Y = \{y_1, ..., y_k\}$ by autoregressively sampling from the probability distribution generated by the model θ based on source text X and query Q:

$$\begin{aligned} y_t &\sim \log p_\theta(y_t|X, Q, Y_{<t}) \\ &\sim \text{logit}_\theta(y_t|X, Q, Y_{<t}) \end{aligned} \tag{1}$$

2.2 Pointwise Mutual Information Decoding

PMI decoding (Chae et al., 2024; Xu, 2023; Shi et al., 2024; Van Der Poel et al., 2022) calculates the point mutual information score between the input and output, aiming to alleviate the overreliance on the model's prior knowledge (e.g., tokens that are frequent but weakly related to the input context) to improve the faithfulness of summary generation. In our scenario, the output probability distribution of the LLM without the input context (i.e., with only the query prompt) represents the prior knowledge. PMI decoding adjusts the final output probabilities by using pairwise mutual information scores between the output probabilities with and without the input context. We adopt the formulation by Shi et al. (2024):

$$\begin{aligned} y_t &\sim \log \Big(p_\theta(y_t|X, Q, Y_{<t}) \Big(\frac{p_\theta(y_t|X, Q, Y_{<t})}{p_\theta(y_t|Q, Y_{<t})} \Big)^\alpha \Big) \\ &\sim \text{softmax}[(1+\alpha)\text{logit}_\theta(y_t|X, Q, Y_{<t}) - \\ &\quad \alpha \text{logit}_\theta(y_t|Q, Y_{<t})] \end{aligned} \tag{2}$$

where the larger the hyperparameter α, the less the decoding relies on the model's prior knowledge, with $\alpha = 0$ degrading to standard decoding.

[1]The code is released at https://github.com/wkw1259/SARA

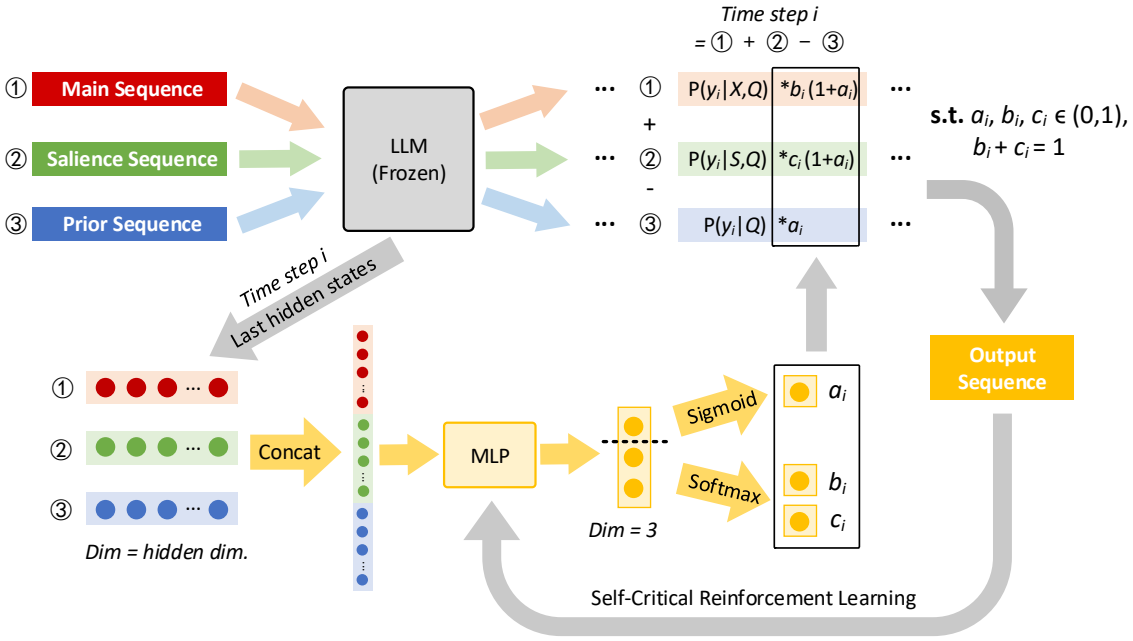

Figure 2: Overall architecture of SARA. In each timestep, LLM receives ① main sequence (i.e., source context with prompt), ② salient sequence (i.e., salient context with prompt) , and ③ prior sequence (i.e., only prompt), and leverages the point mutual information of output logits between the context sequences ①,② and prior sequence ③ to guide decoding process. In addition, tokenwise adaptive decoding via RL is proposed to dynamically adjust the weights of output logits by ①,② and ③ for flexible knowledge combination of different source.

3 Salience-Aware Reinforced Adaptive Decoding Method

3.1 Overview

Figure 2 illustrates the overall architecture of the proposed SARA, which aims to improve summary decoding by incorporating salient information into PMI decoding and dynamically adapting the reliance on source document context, salient context, and prior knowledge at the token level. We introduce SARA through three main part: (1) salient context selection, which identifies key sentences that potentially contain summary-relevant content; (2) salience-aware decoding, which leverages the point mutual information of output probabilities (i.e., logits) between salient context and prior knowledge to guide the decoding process; (3) tokenwise adaptive reinforced decoding, which optimizes the contributions of source context, salient context, and prior knowledge at each decoding timestep through reinforcement learning, enabling the model to flexibly adapt the utilization of different knowledge sources.

3.2 Salient Context Selection

We directly adopt BERTSum (Liu and Lapata, 2019) to generate the salient context. BERTSum is a BERT-based (Kenton and Toutanova, 2019) extractive summarization model that scores each sentence based on its likelihood of being part of the summary, selecting a set of high-scoring sentences. Different from its purpose of extracting a few sentences for forming a concise summary, we relax the extraction threshold to capture more key sentences for forming salient contexts, that aims to comprehensively cover the source content required for generating an abstractive summary. Let $S = \{s_i\}_l$ denote the sequence of key sentences, consisting of l key sentences.

3.3 Salience-Aware Decoding

During decoding, the summary is sampled by combining the word probability distributions generated by an LLM from three sequences: (1) the sequence of source document X accompanied by the query prompt Q, (2) the sequence of key sentences S from the source document also accompanied by the query prompt Q, and (3) the sequence containing only the query prompt Q without the context. For simplicity, we refer to them as (1) the main sequence $X_Q = [X : Q]$, (2) the salient sequence $S_Q = [S : Q]$, (3) the prior sequence Q, where $[:]$ denotes concatenation along the sequence dimension.

Specifically, at each decoding time step, the word probability distribution generated from the original main sequence X_Q are accompanied by incorporating the probability distribution from the salient sequence S_Q to increase faithfulness to the key context. Also, the probability distribution of the prior sequence Q are subtracted to further mitigate potential hallucinations arising from the model's reliance on high-frequency training data or conflicting knowledge. The core formulation of SARA's mutual information computation is as follows:

$$\begin{aligned} y_t \sim & \log\Big(p_\theta(X_Q)^{b_t} p_\theta(S_Q)^{c_t} \Big(\frac{p_\theta(X_Q)^{b_t} p_\theta(S_Q)^{c_t}}{p_\theta(Q)}\Big)^{a_t}\Big) \\ \sim & \operatorname{softmax}\{(1+a_t)\,[b_t \operatorname{logit}_\theta(y_t|X_Q, Y_{<t}) + \\ & c_t \operatorname{logit}_\theta(y_t|S_Q, Y_{<t})] - a_t \operatorname{logit}_\theta(y_t|Q, Y_{<t})\}, \end{aligned} \tag{3}$$

Here, we use $p_\theta(\cdot)$ as a shorthand for $p_\theta(y_t|\cdot, Y_{<t})$, where $\cdot$ can represent X_Q, S_Q, or Q. a_t, b_t, c_t in $A = \{a_i\}_k, B = \{b_i\}_k, C = \{c_i\}_k$ represent dynamic weights at timestep t assigned to the output probability vectors (i.e., logits) from the model θ for sequences Q, X_Q, S_Q. We impose the constraints defined in Equation 4 on a_t, b_t, c_t to prevent their excessive growth.

$$a_t, b_t, c_t \in (0, 1)\,; b_t + c_t = 1 \tag{4}$$

3.4 Tokenwise Adaptive Reinforced Decoding

Tokenwise adaptive reinforced decoding allows the model to compare the output probabilities from main sequence X_Q, salient sequence S_Q, and prior sequence Q, and to dynamically allocate weights $A = \{a_i\}_k, B = \{b_i\}_k, C = \{c_i\}_k$ for the three output probability vectors of X_Q, S_Q, Q at each decoding timestep via reinforcement learning. The weighted combination of the three probability vectors produces a final probability distribution used for sampling the summary output $\hat{Y} = \{y_i\}_{|\hat{Y}|}$.

Concretely, the sequences X_Q, S_Q, Q are passed through the LLM θ, and we take the last hidden layer features, that is, the features used for projecting the logits:

$$\begin{aligned} [Q : \hat{Y}_t^{(Q)}] &\leftarrow \mathrm{LLM}_\theta^{\text{last layer}}([Q : \hat{Y}_t^{(Q)}]) \\ [X_Q : \hat{Y}_t^{(X_Q)}] &\leftarrow \mathrm{LLM}_\theta^{\text{last layer}}([X_Q : \hat{Y}_t^{(X_Q)}]) \\ [S_Q : \hat{Y}_t^{(S_Q)}] &\leftarrow \mathrm{LLM}_\theta^{\text{last layer}}([S_Q : \hat{Y}_t^{(S_Q)}]) \end{aligned} \tag{5}$$

where $\leftarrow$ is used to represent the update of input text feature representations for simplifying the notation, and $\hat{Y}_t^{(Q)}, \hat{Y}_t^{(X_Q)}, \hat{Y}_t^{(S_Q)} \in \mathbb{R}^{t \times d}$ represents the last hidden layer feature of the summary sequence autoregressively generated by sequence Q, X_Q, S_Q at time step t in the LLM, respectively. Outside the LLM, we constructed a small MLP network, trained via reinforcement learning, which generates the weights A, B, and C respectively for the weighted combination of the LLM's output probabilities from the sequences Q, X_Q, and S_Q. The purpose of this MLP network is to model the interactions of independent features produced by the LLM for different input sequences, allowing the LLM to determine how to allocate weights for combining the output probabilities of input sequences. The computation is expressed as follows:

$$\begin{aligned} [A : B : C]^* = & \mathrm{ReLU}([Y_t^{(Q)} : Y_t^{(X_Q)} : \\ & Y_t^{(S_Q)}]^* W_1 + b_1) W_2 + b_2 \end{aligned} \tag{6}$$

where $[:]^*$ denotes concatenation along the feature dimensions; $W_1 \in \mathbb{R}^{d \times h}, b1 \in \mathbb{R}^h, W_2 \in \mathbb{R}^{h \times 3}, b2 \in \mathbb{R}^3$; The concatenated matrix $[A : B : C]^* \in \mathbb{R}^{k*3}$ represents the weights assigned to the output logits vectors of Q, X_Q, and S_Q at each time step. Next, A, B and C are normalized to satisfy the constraints defined in Equation 4, using a left arrow to denote the updates of A, B and C for brevity:

$$\begin{aligned} [B : C]^* &\leftarrow \operatorname{softmax}([B : C]^*) \\ A &\leftarrow \operatorname{sigmoid}(A) \end{aligned} \tag{7}$$

Note that the MLP's role is to compute weights for the logits of the three sequences produced by the LLM in each timestep, allowing for a dynamic combination of three sequences' logits, rather than allocating fixed weights to the entire sequence as hyperparameters in previous methods. It does not participate in the LLM's own inference and does not alter the LLM's weights or the text feature representations. Based on the combined output probabilities, the summary $\hat{Y}$ can be predicted using various sampling strategies. Inspired by the self-critical sequence training (Rennie et al., 2017), we enable the MLP to adjust A, B, and C through reinforcement learning. For a predicted summary sequence $\hat{Y} = \{y_i\}_{|\hat{Y}|}$, we set the ROUGE (Lin, 2004) and FactKB (Feng et al., 2023) metrics that measure quality and faithfulness as reward, which is defined as:

$$\begin{aligned} r(\hat{Y}) = & \mathrm{R}_1(\hat{Y}, Y) + \mathrm{R}_2(\hat{Y}, Y) \\ & + \mathrm{R}_\mathrm{L}(\hat{Y}, Y) + \lambda\, \mathrm{FKB}(\hat{Y}, Y) \end{aligned} \tag{8}$$

where $\mathrm{R}_1, \mathrm{R}_2, \mathrm{R}_\mathrm{L}(\cdot)$ and $\mathrm{FKB}(\cdot)$ represent the scores of ROUGE-1,2,L, and FactKB, respectively, in relation to the predictions $\hat{Y}$ and ground truth

Y. Then the MLP layer is optimized by gradient ascent based on rewards for adjusting the weights A, B, and C, represented as:

$$\begin{aligned}\nabla_\theta J(\theta) &= (r(\hat{Y}) - \frac{1}{N}\sum_\tau r(\hat{Y}^\tau))\nabla_\theta \log \text{softmax}(\{P_\theta\}_{|\hat{Y}|}) \\ P_\theta &= (1+a_i)\,[b_i \log\text{it}_\theta(y_i|X_Q, Y_{<i}) + c_i \log\text{it}_\theta(y_i|S_Q, Y_{<i})] \\ &\quad - a_i \log\text{it}_\theta(y_i|Q, Y_{<i}) \\ \theta &\leftarrow \theta + \eta\nabla_\theta J(\theta)\end{aligned} \tag{9}$$

where $\hat{Y}^\tau$ denotes the additional sampling sequence for calculating the reward deviation from the base prediction $\hat{Y}$, N is the number of sampling sequences; P_θ denotes the combined output logits obtained from the sequences Q, X_Q, and S_Q as expressed in Equation 9.

4 Experiments

4.1 Datasets

The proposed method is evaluated on three summarization datasets: CNN/DM (Nallapati et al., 2016), WikiHow (Koupaee and Wang, 2018), and NYT50 (Durrett et al., 2016). CNN/DM consists of news articles in CNN and Daily Mail paired with summaries, providing a widely-used summarization benchmark. WikiHow contains instructional articles with corresponding summaries, offering a diverse range of topics and writing styles. NYT50 is a filtered version of the New York Times Annotated Corpus dataset (Sandhaus, 2008), where articles with summaries shorter than 50 words are excluded.

4.2 Setup and Metrics

Implementation Details. Please refer to Appendix B for complete implementation details, including the prompts for summarization, and reinforcement learning and inference hyperparameters in LLMs.

Metrics. Following prior works (Shi et al., 2024; Xu, 2023), ROUGE (Lin, 2004), BERTScore-Prediction (Zhang et al.), and FactKB (Feng et al., 2023) are used to measure the quality and factual consistency, respectively. In addition, SacreBLEU (Post, 2018) is also introduced to evaluate the summaries.

Furthermore, we utilize GPT-4[2] (Achiam et al., 2023) to evaluate the quality and faithfulness of the generated summaries, which provides human-like preference assessments compared to traditional evaluation metrics (Gao et al., 2023; Chen et al., 2023a). Specifically, GPT-4 ranks the summaries generated by different methods based on the original documents and reference summaries. Prompts for GPT-4 are provided in Appendix C.

4.3 Models and Baselines

We implement the proposed method on four LLMs: GPT-NEO[3] (3B) (Black et al., 2021), LLaMA-2-Chat (7B)[4] (Touvron et al., 2023), OPT[5] (7B) (Zhang et al., 2022b), and Mistral-Instruct-v0.2 (7B)[6] (Jiang et al., 2023). The proposed method is compared with CAD (Shi et al., 2024; Xu, 2023) and the vanilla usage of LLMs, where CAD applies PMI decoding to LLMs and adjusts the proportion of context and prior knowledge with hyperparameters. All baselines maintaining the same inference settings as detailed in Appendix B.

4.4 Overall Performance

Table 1 presents the performance of different methods on CNNDM, WikiHow, and NYT50 in terms of ROUGE, SacreBLEU, BERTScore-P, and FactKB metrics. Overall, SARA outperforms both CAD and the vanilla decoding across all three datasets. Notably, the performance gains are more pronounced on CNNDM and NYT50 compared to WikiHow. We attribute this to the more extractive nature of CNNDM and NYT50 while WikiHow leans towards abstractive, and the proposed method showing a more notable advantage in scenarios that lean towards extractive summarization, where key contextual information plays a crucial role. Conversely, on the WikiHow dataset, while CAD improves factual consistency, its ROUGE scores drop on LLaMA, likely due to an overemphasis on contextual knowledge at the expense of the model's prior knowledge. Our method, with adaptive token-level weight allocation through salient context and reinforcement learning, effectively maintains both summary quality and faithfulness.

4.5 Effect of Weight Allocation on Triple Sequences

Can reinforcement learning (RL) effectively find the reasonable weight allocation among the main sequence, salient sequence, and prior sequence log-

[2]https://chatgpt.com/

[3]https://huggingface.co/EleutherAI/gpt-neo-2.7B

[4]https://huggingface.co/meta-llama/Llama-2-7b-chat-hf

[5]https://huggingface.co/facebook/opt-6.7b

[6]https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

Datasets	Model	Decoding	ROUGE-1	ROUGE-2	ROUGE-L	SacreBLEU	BERTScore-P	FactKB
CNN/DM	GPT-NEO	Vanilla	29.46	8.89	26.34	6.76	85.27	70.49
		CAD	34.10	12.71	30.66	9.90	86.07	83.60
		SARA	**35.25**	**13.63**	**31.71**	**10.63**	**86.34**	**85.17**
	LLaMA	Vanilla	37.69	14.28	33.63	9.81	88.06	91.34
		CAD	38.18	15.07	34.02	10.38	87.92	93.14
		SARA	**38.74**	**15.41**	**34.57**	**10.69**	**88.06**	**93.95**
	OPT	Vanilla	33.22	12.07	29.80	9.42	86.02	85.59
		CAD	35.49	13.99	31.85	11.02	86.28	89.90
		SARA	**36.36**	**14.66**	**32.74**	**11.62**	**86.55**	**90.74**
	Mistral	Vanilla	38.16	14.26	33.98	9.85	87.94	90.36
		CAD	38.46	14.86	34.18	10.29	87.90	92.26
		SARA	**38.92**	**15.16**	**34.61**	**10.54**	**88.00**	**92.38**
WikiHow	GPT-NEO	Vanilla	20.77	3.67	19.45	2.03	83.24	64.91
		CAD	22.88	4.77	21.31	2.91	**83.59**	76.03
		SARA	**23.10**	**4.88**	**21.47**	**2.99**	83.56	**77.50**
	LLaMA	Vanilla	24.93	5.90	23.18	2.95	84.50	83.59
		CAD	24.86	5.89	22.91	**3.13**	84.34	84.84
		SARA	**25.08**	**5.97**	**23.19**	3.03	**84.47**	**85.52**
	OPT	Vanilla	20.29	3.62	18.95	2.10	83.21	60.71
		CAD	22.64	4.71	21.07	2.99	83.68	73.55
		SARA	**23.05**	**4.99**	**21.36**	**3.19**	**83.85**	**73.60**
	Mistral	Vanilla	24.98	5.72	23.32	2.66	84.67	69.92
		CAD	25.30	5.95	23.48	2.78	84.60	**77.96**
		SARA	**25.92**	**6.18**	**24.09**	**2.92**	**84.82**	77.14
NYT50	GPT-NEO	Vanilla	25.63	7.32	22.30	5.82	84.64	57.92
		CAD	30.19	11.51	26.68	9.22	85.43	70.54
		SARA	**30.44**	**11.57**	**26.98**	**9.24**	**85.61**	**73.28**
	LLaMA	Vanilla	35.76	14.16	31.36	9.38	88.20	85.31
		CAD	37.66	16.79	33.33	11.42	88.26	88.05
		SARA	**37.84**	**16.96**	**33.50**	**11.51**	**88.31**	**89.46**
	OPT	Vanilla	27.74	9.71	24.51	7.77	85.10	69.43
		CAD	31.04	13.07	27.76	10.37	85.35	74.72
		SARA	**31.70**	**13.47**	**28.31**	**10.69**	**85.66**	**76.72**
	Mistral	Vanilla	36.43	14.48	31.83	9.82	88.19	84.03
		CAD	37.73	16.59	33.23	11.38	88.05	87.67
		SARA	**37.88**	**16.65**	**33.45**	**11.46**	**88.14**	**87.78**

Table 1: Performance comparison on the CNN/DM, WikiHow, NYT50 datasets with various LLM backbones and decoding strategies. Overall, SARA consistently outperforms both CAD and the vanilla decoding across different datasets and backbones.

its at different time steps? To answer this, we manually set various weight configurations and observed the model's performance across different metrics. Experiments were conducted using GPT-NEO as the backbone on the CNNDM dataset, with results shown in Table 2. It can be observed that: (1) As the weight for the prior sequence a increases, the model's performance initially improves and then declines, peaking when $a = 0.5$; (2) Compared to using only the main sequence, the combination of weights for the main and salient sequences b, c enhances the model's performance; (3) Compared to multiple manual weight configurations, the proposed method consistently achieves the best results, demonstrating the advantage of adaptive token-level weight allocation over fixed sequence-level weight allocation.

4.6 Ablation Analysis

To evaluate each component in the proposed method, we conducted the following ablation experiments: (a) removing the main sequence X_Q, (b) removing the salient sequence S_Q, (c) removing the prior sequence Q, respectively; and (d) removing the FactKB reward and (e) removing the ROUGE reward in RL, respectively. The experimental results presented in Table 3 show that: (1) all three sequences positively contribute to performance, with the prior sequence playing a crucial role; (2) using only the ROUGE reward or only

Models	ROUGE-1	ROUGE-2	ROUGE-L	SacreBLEU	BERTScore-P	FactKB
SARA	35.25	13.63	31.71	10.63	86.34	85.17
a=0, b=1, c=0	29.46	8.89	26.34	6.76	85.27	70.49
a=0.1, b=1, c=0	31.59	10.47	28.32	8.00	85.72	76.37
a=0.3, b=1, c=0	33.83	12.40	30.44	9.70	86.11	82.28
a=0.5, b=1, c=0	34.10	12.71	30.66	9.90	86.07	83.60
a=0.7, b=1, c=0	33.73	12.44	30.12	9.53	85.78	81.52
a=1, b=1, c=0	32.27	11.42	28.38	8.53	85.01	73.87
a=0, b=0.7, c=0.3	29.59	8.93	26.44	6.70	85.33	70.63
a=0, b=0.5, c=0.5	29.71	9.01	26.55	6.80	85.34	70.92
a=0, b=0.3, c=0.7	29.69	8.98	26.52	6.80	85.33	70.46
a=0, b=0, c=1	29.72	9.04	26.52	6.89	85.29	70.64
a=0.3, b=0.7, c=0.3	34.27	12.64	30.87	9.85	86.21	82.77
a=0.3, b=0.5, c=0.5	34.46	12.77	30.99	9.89	86.25	82.57
a=0.3, b=0.3, c=0.7	34.40	12.74	30.96	9.90	86.24	82.73
a=0.3, b=0, c=1	34.32	12.77	30.86	9.91	86.18	82.66
a=0.5, b=0.7, c=0.3	34.95	13.31	31.42	10.40	86.28	84.40
a=0.5, b=0.5, c=0.5	35.08	13.46	31.54	10.47	86.29	84.60
a=0.5, b=0.3, c=0.7	35.01	13.42	31.50	10.43	86.26	84.44
a=0.5, b=0, c=1	34.89	13.34	31.30	10.39	86.19	84.82

Table 2: Effect of weight allocation. a, b, c represent sequence-level weights for prior sequence Q, main sequence X_Q, and salient sequence X_Q, respectively. It degrades into CAD when $a = 0$, and degrades into vanilla decoding when $a = 0, c = 0$.

	Models	ROUGE-1	ROUGE-2	ROUGE-L	SacreBLEU	BERTScore-P	FactKB
	SARA	35.25	13.63	31.71	10.63	86.34	85.17
a	*w/o* Seq. X_Q	34.71	13.18	31.18	10.29	86.19	84.61
b	*w/o* Seq. S_Q	34.25	12.79	30.76	9.94	86.09	82.73
c	*w/o* Seq. Q	29.83	9.15	26.65	6.93	85.31	70.60
d	*w/o* Reward FKB	35.35	13.68	31.74	10.65	86.34	84.60
e	*w/o* Reward R	34.32	12.66	30.72	9.64	85.33	85.42

Table 3: Ablation analysis of main sequence X_Q, salient sequence X_Q, prior sequence Q, and different RL rewards on CNN/DM test set.

the FactKB reward for weight adjustment improves the corresponding metric but leads to a decline in others.

4.7 Effect of Different Salient Sequence Lengths

How does the length of the salient sequence S_Q affect model performance? To explore this, we varied the number of salient context sentences l extracted from the source document, including 3, 5, 10, 15, 20 sentences, and observed the experimental results with the GPT-NEO backbone on the CNN/DM test set. As shown in Figure 3, excessively long salient sequences do not enhance ROUGE scores, with the best ROUGE performance observed at 5 and 10 sentences. Meanwhile, the FactKB metric steadily improves with longer salient sequences, stabilizing after reaching 10 sentences. The complete experimental results are provided in Table 8 in Appendix D.1.

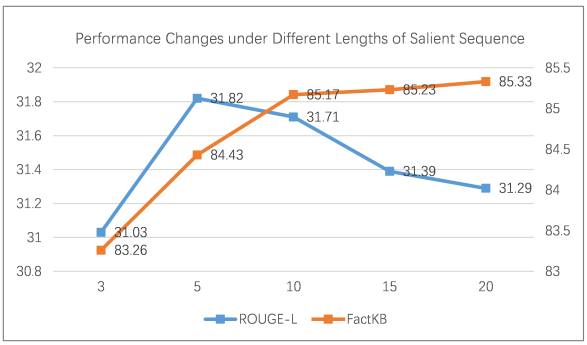

Figure 3: ROUGE-L and FactKB score changes under the salient sequence with different number of sentences.

Dataset	Method	Quality	Faithfulness
		1st-2nd-3rd	1st-2nd-3rd
CNN/DM	Vanilla	0.09-0.31-0.60	0.09-0.36-0.55
	CAD	0.35-0.42-0.23	0.31-0.40-0.29
	SARA	**0.56**-0.27-0.17	**0.60**-0.24-0.16
WikiHow	Vanilla	0.05-0.14-0.81	0.04-0.20-0.76
	CAD	0.45-0.45-0.10	0.36-0.50-0.14
	SARA	**0.50**-0.41-0.09	**0.60**-0.30-0.10
NYT50	Vanilla	0.16-0.13-0.71	0.16-0.16-0.68
	CAD	0.29-0.55-0.16	0.31-0.46-0.23
	SARA	**0.55**-0.32-0.13	**0.53**-0.38-0.09

Table 4: Quality (Q) ranking, Faithfulness (F) ranking via GPT-4 evaluations. SARA outperforms both CAD and vanilla decoding with a winning rate exceeding 50% across different datasets.

4.8 GPT-4 Evaluations

Table 4 presents the ranking results of GPT-4 for the quality and faithfulness of summaries generated by vanilla decoding, CAD, and SARA. We employed GPT-NEO as the backbone model. As can be seen from the experimental results that, SARA obviously outperforms both CAD and vanilla decoding with a winning rate exceeding 50% across CNN/DM, WikiHow, NYT50 datasets, effectively reducing hallucinations. This indicates the advantage of SARA in leveraging ROUGE and FactKB as rewards to dynamically incorporate source documents, salient contexts, and the model's prior knowledge for summary decoding.

5 Related Work

Hallucination in LLM refers to the phenomenon where language models generate factually incorrect or inconsistent information. Recent works focused on the detection and evaluation of hallucinations (Jia et al., 2023; Manakul et al., 2023; Bang et al., 2023; Guerreiro et al., 2023; Mündler et al., 2024). To mitigate hallucinations, a number of studies improved LLMs during the fine-tuning stage, including factual consistency objectives training (Wan and Bansal, 2022), reinforcement learning (Roit et al., 2023), contrastive learning (Sun et al., 2023), post-processing (Gou et al., 2024), and prompt engineering (Wang et al., 2023a; Dhuliawala et al., 2024; Lv et al., 2023). Additionally, some works have attempted to alleviate hallucinations during the inference stage, such as retrieval-augmented generation (Shuster et al., 2021; Peng et al., 2023; Xu et al., 2024; Yang et al., 2025) and various decoding strategies (Lee et al., 2022; Xu, 2023; Wan et al., 2023; Shi et al., 2024).

Faithfulness in summarization is an important topic, which refers to the consistency between the generated and original text. When the generated content lacks faithfulness, hallucinations often arise. Researchers have proposed various methods to improve the faithfulness of summarization, with recent works focusing primarily on entity/token-specific training (Shen et al., 2023; Zhang et al., 2022a; Dong et al., 2022; Nan et al., 2021), post-processing (Balachandran et al., 2022; Fabbri et al., 2022), loss truncation (Kang and Hashimoto, 2020), chain-of-thought (Wang et al., 2023b; Wei et al., 2025), active learning (Xia et al., 2024; Li et al., 2024a), contrastive learning (Feng et al., 2024; Chen et al., 2023b; Choubey et al., 2023; Chen et al., 2021; Cao and Wang, 2021), and factual consistency evaluation (Feng et al., 2023; Cao et al., 2022; Jia et al., 2023; Luo et al., 2024).

Closely related to our work is PMI-based decoding (Chae et al., 2024; Xu, 2023; Shi et al., 2024; Van Der Poel et al., 2022), which mitigates the influence of model prior knowledge that induces hallucinations by contrasting the output probabilities with and without the source document, thereby enhancing the role of contextual knowledge. The goal of SARA is to further introduce the mutual information of salient information and prior knowledge, as well as incorporating reinforcement learning to adaptively adjust the weights of contextual and prior knowledge at each time step.

6 Conclusion

In this work, we propose SARA, salience-aware reinforced adaptive decoding for abstractive summarization, which guides LLMs to determine how to rely on source documents' context, salient context, and models' prior knowledge during decoding based on pointwise mutual information. Moreover, we propose a tokenwise adaptive reinforced decoding mechanism in SARA that dynamically adjusts the contributions of contextual knowledge and prior knowledge at each timestep, enabling the flexible integration of knowledge from different sources in decoding. Experimental results on the CNN/DM, WikiHow, and NYT50 datasets show that the proposed method consistently improves the quality and faithfulness of generated summaries across various LLM backbones without compromising their general reasoning capabilities.

Limitations

We consider SARA as a feature-level voting mechanism for LLMs: it combines word probabilities generated from different important sequences, voting at each timestep to determine the final word probability for decoding, while reinforcement learning assigns weights to the voting process. Although SARA improves summarization guided by RL rewards on LLMs, this work has not further validated its applicability in traditional encoder-decoder summarization architectures. On the other hand, SARA computes point mutual information based on word probabilities from the same language. Given LLMs' multilingual capabilities, whether the proposed method can effectively guide summary decoding in cross-lingual summarization scenarios (where the source and target languages differ) remains an interesting question for future exploration.

Acknowledgements

The work is supported by the National Natural Science Foundation of China (62406223, 62176029), Natural Science Foundation of Tianjin (24JCZDJC00130), research funding from Cangzhou Institute of Tiangong University (TGCYY-Z-0303), China Postdoctoral Science Foundation Funded Project (2024M763867), Chongqing Key Project of Technological Innovation and Application Development (CSTB2023TIAD-KPX0064).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Rahul Aralikatte, Shashi Narayan, Joshua Maynez, Sascha Rothe, and Ryan McDonald. 2021. Focus attention: Promoting faithfulness and diversity in summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6078–6095.

Vidhisha Balachandran, Hannaneh Hajishirzi, William Cohen, and Yulia Tsvetkov. 2022. Correcting diverse factual errors in abstractive summarization via post-editing and language model infilling. In *Empirical Methods in Natural Language Processing*.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–718.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. Gpt-neo: Large scale autoregressive language modeling with mesh-tensorflow.

Meng Cao, Yue Dong, and Jackie Chi Kit Cheung. 2022. Hallucinated but factual! inspecting the factuality of hallucinations in abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3340–3354.

Shuyang Cao and Lu Wang. 2021. Cliff: Contrastive learning for improving faithfulness and factuality in abstractive summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6633–6649.

Kyubyung Chae, Jaepill Choi, Yohan Jo, and Taesup Kim. 2024. Mitigating hallucination in abstractive summarization with domain-conditional mutual information. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1809–1820.

Shiqi Chen, Siyang Gao, and Junxian He. 2023a. Evaluating factual consistency of summaries with large language models. *arXiv preprint arXiv:2305.14069*.

Sihao Chen, Fan Zhang, Kazoo Sone, and Dan Roth. 2021. Improving faithfulness in abstractive summarization with contrast candidate generation and selection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5935–5941.

Wei-Lin Chen, Cheng-Kuang Wu, Hsin-Hsi Chen, and Chung-Chi Chen. 2023b. Fidelity-enriched contrastive search: Reconciling the faithfulness-diversity trade-off in text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 843–851.

Prafulla Kumar Choubey, Alex Fabbri, Jesse Vig, Chien-Sheng Wu, Wenhao Liu, and Nazneen Rajani. 2023. Cape: Contrastive parameter ensembling for reducing hallucination in abstractive summarization. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10755–10773.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason E Weston. 2024. Chain-of-verification reduces hallucination in large language models. In *ICLR 2024 Workshop on Reliable and Responsible Foundation Models*.

Yue Dong, John Wieting, and Pat Verga. 2022. Faithful to the document or to the world? mitigating hallucinations via entity-linked knowledge in abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1067–1082.

Greg Durrett, Taylor Berg-Kirkpatrick, and Dan Klein. 2016. Learning-based single-document summarization with compression and anaphoricity constraints. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1998–2008.

Alex Fabbri, Prafulla Kumar Choubey, Jesse Vig, Chien-Sheng Wu, and Caiming Xiong. 2022. Improving factual consistency in summarization with compression-based post-editing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9149–9156.

Huawen Feng, Yan Fan, Xiong Liu, Ting-En Lin, Zekun Yao, Yuchuan Wu, Fei Huang, Yongbin Li, and Qianli Ma. 2024. Improving factual consistency of news summarization by contrastive preference optimization. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11084–11100.

Shangbin Feng, Vidhisha Balachandran, Yuyang Bai, and Yulia Tsvetkov. 2023. Factkb: Generalizable factuality evaluation using language models enhanced with factual knowledge. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 933–952.

Mingqi Gao, Jie Ruan, Renliang Sun, Xunjian Yin, Shiping Yang, and Xiaojun Wan. 2023. Human-like summarization evaluation with chatgpt. *arXiv preprint arXiv:2304.02554*.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yujiu Yang, Nan Duan, Weizhu Chen, et al. 2024. Critic: Large language models can self-correct with tool-interactive critiquing. In *The Twelfth International Conference on Learning Representations*.

Nuno M Guerreiro, Duarte M Alves, Jonas Waldendorf, Barry Haddow, Alexandra Birch, Pierre Colombo, and André FT Martins. 2023. Hallucinations in large multilingual translation models. *Transactions of the Association for Computational Linguistics*, 11:1500–1517.

Qi Jia, Siyu Ren, Yizhu Liu, and Kenny Zhu. 2023. Zero-shot faithfulness evaluation for text summarization with foundation language model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11017–11031.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Hanlei Jin, Yang Zhang, Dan Meng, Jun Wang, and Jinghua Tan. 2024. A comprehensive survey on process-oriented automatic text summarization with exploration of llm-based methods. *arXiv preprint arXiv:2403.02901*.

Daniel Kang and Tatsunori B Hashimoto. 2020. Improved natural language generation via loss truncation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 718–731.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1. Minneapolis, Minnesota.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*.

Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale N Fung, Mohammad Shoeybi, and Bryan Catanzaro. 2022. Factuality enhanced language models for open-ended text generation. *Advances in Neural Information Processing Systems*, 35:34586–34599.

Dongyuan Li, Ying Zhang, Zhen Wang, Shiyin Tan, Satoshi Kosugi, and Manabu Okumura. 2024a. Active learning for abstractive text summarization via llm-determined curriculum and certainty gain maximization. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8959–8971.

Taiji Li, Zhi Li, and Yin Zhang. 2024b. Improving faithfulness of large language models in summarization via sliding generation and self-consistency. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8804–8817.

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

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740.

Ge Luo, Weisi Fan, Miaoran Li, Guoruizhe Sun, Runlong Zhang, Chenyu Xu, and Forrest Bao. 2024. Summacoz: A dataset for improving the interpretability of factual consistency detection for summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3689–3702.

Bo Lv, Xin Liu, Shaojie Dai, Nayu Liu, Fan Yang, Ping Luo, and Yue Yu. 2023. DSP: Discriminative soft prompts for zero-shot entity and relation extraction. pages 5491–5505.

Bo Lv, Chen Tang, Yanan Zhang, Xin Liu, Ping Luo, and Yue Yu. 2024. URG: A unified ranking and generation method for ensembling language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4421–4434.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017.

Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. 2024. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. In *The Twelfth International Conference on Learning Representations*.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290.

Feng Nan, Ramesh Nallapati, Zhiguo Wang, Cicero dos Santos, Henghui Zhu, Dejiao Zhang, Kathleen Mckeown, and Bing Xiang. 2021. Entity-level factual consistency of abstractive text summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2727–2733.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*.

Matt Post. 2018. A call for clarity in reporting bleu scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7008–7024.

Paul Roit, Johan Ferret, Lior Shani, Roee Aharoni, Geoffrey Cideron, Robert Dadashi, Matthieu Geist, Sertan Girgin, Leonard Hussenot, Orgad Keller, et al. 2023. Factually consistent summarization via reinforcement learning with textual entailment feedback. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6252–6272.

Sangwon Ryu, Heejin Do, Yunsu Kim, Gary Lee, and Jungseul Ok. 2024a. 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.

Sangwon Ryu, Heejin Do, Yunsu Kim, Gary Geunbae Lee, and Jungseul Ok. 2024b. Key-element-informed sllm tuning for document summarization. In *Proceedings of the Annual Conference of the International Speech Communication Association*, pages 1940–1944.

Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752.

Jianbin Shen, Junyu Xuan, and Christy Liang. 2023. Mitigating intrinsic named entity-related hallucinations of abstractive text summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15807–15824.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. 2024. Trusting your evidence: Hallucinate less with context-aware decoding. 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 783–791.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803.

Weiwei Sun, Zhengliang Shi, Shen Gao, Pengjie Ren, Maarten de Rijke, and Zhaochun Ren. 2023. Contrastive learning reduces hallucination in conversations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13618–13626.

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

Liam Van Der Poel, Ryan Cotterell, and Clara Meister. 2022. Mutual information alleviates hallucinations in abstractive summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5956–5965.

David Wan and Mohit Bansal. 2022. Factpegasus: Factuality-aware pre-training and fine-tuning for abstractive summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1010–1028.

David Wan, Mengwen Liu, Kathleen Mckeown, Markus Dreyer, and Mohit Bansal. 2023. Faithfulness-aware decoding strategies for abstractive summarization. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2864–2880.

Fei Wang, Kaiqiang Song, Hongming Zhang, Lifeng Jin, Sangwoo Cho, Wenlin Yao, Xiaoyang Wang, Muhao Chen, and Dong Yu. 2022. Salience allocation as guidance for abstractive summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6094–6106.

Sirui Wang, Kaiwen Wei, Hongzhi Zhang, Yuntao Li, and Wei Wu. 2023a. Let me check the examples: Enhancing demonstration learning via explicit imitation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1080–1088. Association for Computational Linguistics.

Yiming Wang, Zhuosheng Zhang, and Rui Wang. 2023b. Element-aware summarization with large language models: Expert-aligned evaluation and chain-of-thought method. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8640–8665.

Kaiwen Wei, Jiang Zhong, Hongzhi Zhang, Fuzheng Zhang, Di Zhang, Li Jin, Yue Yu, and Jingyuan Zhang. 2025. Chain-of-specificity: Enhancing task-specific constraint adherence in large language models. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 2401–2416. Association for Computational Linguistics.

Yu Xia, Xu Liu, Tong Yu, Sungchul Kim, Ryan Rossi, Anup Rao, Tung Mai, and Shuai Li. 2024. Hallucination diversity-aware active learning for text summarization. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8657–8669.

Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Retrieval meets long context large language models. In *The Twelfth International Conference on Learning Representations*.

Zhichao Xu. 2023. Context-aware decoding reduces hallucination in query-focused summarization. *arXiv preprint arXiv:2312.14335*.

Yuming Yang, Jiang Zhong, Li Jin, Jingwang Huang, Jingpeng Gao, Qing Liu, Yang Bai, Jingyuan Zhang, Rui Jiang, and Kaiwen Wei. 2025. Benchmarking multimodal RAG through a chart-based document question-answering generation framework. *CoRR*, abs/2502.14864.

Haopeng Zhang, Semih Yavuz, Wojciech Kryściński, Kazuma Hashimoto, and Yingbo Zhou. 2022a. Improving the faithfulness of abstractive summarization via entity coverage control. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 528–535.

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.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022b. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

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.

Datasets	Train/Val/Test Partition	Avg. Article/ Summary Len.
CNN	90,266/1,220/1,093	760.50/45.70
DM	196,961/12,148/10,397	653.33/54.65
WikiHow	168,126/6,000/6,000	579.8/62.1
NYT50	96,834/4,000/3,452	800.04/45.54

Table 5: Dataset statistics.

Hyperparameters	**CNN/DM**	**WikiHow**	**NYT50**
Max input length	1,024	1024	1,024
Min new tokens	30	20	30
Max new tokens	70	80	90
Salient context sent. num l	10	10	10
LLM float type	bf16	bf16	bf16
Batch size	4	4	4
Gradient accumulation	1	1	1
Learning rate	5e-5	1e-5	1e-5
Top-k for base pred. $\hat{Y}$	10	10	10
Top-p for base pred. $\hat{Y}$	0.5	0.5	0.5
Top-k for sampling pred. $\hat{Y}$	100	100	100
Top-p for sampling pred. $\hat{Y}$	0.95	0.95	0.95
Sampling sequence num N	5	5	5
Training steps	2000	2000	2000
Warmup ratio	0.05	0.05	0.05
Optimizer	Adam	Adam	Adam
Adam beta1	0.9	0.9	0.9
Adam beta2	0.999	0.999	0.999
GPU	A100×1	A100×1	A100×1
Avg. num of algorithm runs	3	3	3
Temperature	1.0	1.0	1.0
Beam size	1	1	1
Length penalty	1.0	1.0	1.0
Repetition penalty	1.0	1.0	1.0
MLP hidden size h	4,096	4,096	4,096
FactKB reward weight λ	1.0	1.0	1.0

Table 6: Reinforcement learning hyperparameters.

Hyperparameters	CNN/DM	WikiHow	NYT50
LLM float type	bf16	bf16	bf16
Top-k	50	50	50
Top-p	0.9	0.9	0.9
Temperature	1.0	1.0	1.0
Beam size	1	1	1
Length penalty	1.0	1.0	1.0
Repetition penalty	1.0	1.0	1.0
Max input length	1,024	1024	1,024
Min new tokens	30	20	30
Max new tokens	70	80	90
Salient context sent. num l	10	10	10

Table 7: Inference hyperparameters.

Prompt (Separate Quality and Faithfulness)

You will be provided with an article, a reference summary, and three different generated summaries (Method A, Method B, Method C). Your task is to rank the generated summaries in terms of (1) overall quality and (2) faithfulness with the article and reference summary.

1. **Quality**: How well the summary covers the key points of the article, considering coherence and informativeness.
2. **Faithfulness**: How accurately the summary represents the information in the article, avoiding hallucinated or unsupported content.

Please rank the summaries from best to worst based on their overall performance in both criteria.

Article: <Article text>

Reference Summary:
<Reference summary>

Method A Summary:
<Method A summary>

Method B Summary:
<Method B summary>

Method C Summary:
<Method C summary>

Ranking:
1. Overall Quality Ranking: <Rank from A to C>
2. Factual Consistency Ranking: <Rank from A to C>

Provide only the rankings as output.

Figure 4: Prompt to rank generated summaries based on quality and faithfulness respectively.

A Datasets

The proposed method is evaluated on three summarization datasets, CNN/DM (Nallapati et al., 2016), WikiHow (Koupaee and Wang, 2018), and NYT50 (Durrett et al., 2016), and the statistics are shown in Table 5.

Prompt (Both Quality and Faithfulness)

You will be provided with an article, a reference summary, and three different generated summaries (Method A, Method B, Method C). Your task is to rank the generated summaries based on two criteria:

1. **Quality**: How well the summary covers the key points of the article, considering coherence and informativeness.
2. **Faithfulness**: How accurately the summary represents the information in the article, avoiding hallucinated or unsupported content.

Please rank the summaries from best to worst based on their overall performance in both criteria.

Article: <Article text>

Reference Summary: <Reference summary>

Method A Summary:
<Method A summary>

Method B Summary:
<Method B summary>

Method C Summary:
<Method C summary>

Ranking:
1. <Best method>
2. <Second best method>
3. <Third best method>

Provide only the rankings as output.

Figure 5: Prompt to rank generated summaries based on both quality and faithfulness.

B Implementation Details

B.1 Prompt for Summarization

The prompts we used for summarization follow the settings of previous works (Xu, 2023). For all LLM backbones, we use the following prompt for the Wikihow dataset,

Article: <Article text>. Summary of the above article:

and use the following prompt for the CNN/DM and NYT50 datasets,

Article: <Article text>. Summary of the above news article:

For salient sequences, we directly replace <Article text> with salient content. For prior sequences, <Article text> is replaced with an empty input "".

#Sent. of Seq. S	ROUGE-1	ROUGE-2	ROUGE-L	SacreBLEU	BERTScore-P	FactKB
3	34.57	12.87	31.03	9.93	86.31	83.26
5	35.39	13.67	31.82	10.61	86.44	84.43
10	35.25	13.63	31.71	10.63	86.34	85.17
15	34.92	13.39	31.39	10.40	86.25	85.23
20	34.84	13.27	31.29	10.35	86.21	85.33

Table 8: Complete experimental results under the salient sequence with different number of sentences, corresponding to Figure 3 in main text of the paper.

Dataset	Method	Both Q&F
		1st-2nd-3rd
CNN/DM	Vanilla	0.10-0.31-0.59
	CAD	0.32-0.46-0.22
	SARA	**0.58**-0.23-0.19
WikiHow	Vanilla	0.04-0.16-0.80
	CAD	0.37-0.52-0.11
	SARA	**0.59**-0.32-0.09
NYT50	Vanilla	0.12-0.22-0.66
	CAD	0.30-0.52-0.18
	SARA	**0.58**-0.26-0.16

Table 9: Both Quality and Faithfulness (Q&F) ranking via GPT-4 evaluations.

Model	Decoding	R-L	FKB	ms/batch
GPT-NEO	Vanilla	26.34	70.49	5,143
	CAD	30.66	83.60	6,429
	SARA	31.71	85.17	8,571
LLaMA	Vanilla	33.63	91.34	5,857
	CAD	87.92	93.14	7,286
	SARA	88.06	93.95	9,714
OPT	Vanilla	29.80	85.59	5,571
	CAD	31.85	89.90	6,857
	SARA	32.74	90.74	8,857

Table 10: Inference speed of different methods.

B.2 Hyperparameters

Table 6 lists the detailed hyperparameters for reinforcement learning, and Table 7 provides the detailed hyperparameters for inference in this work, which mainly follow the settings of Shi et al. (2024) and Xu (2023).

C Prompt for GPT-4 Evaluations

We utilize GPT-4 to evaluate the quality and faithfulness of the generated summaries. GPT-4 ranks the summaries generated by vanilla decoding, CAD, and SARA based on the original documents and reference summaries. For each method, we test 200 samples. We used two types of prompts: a. Ranking the summaries based on quality and faithfulness separately, as shown in Figure 4; and b. Ranking the summaries based on both quality and faithfulness jointly, as shown in Figure 5. We include only the separately ranked GPT-4 evaluation results in the main text.

D Experimental Results

D.1 Effect of Different Salient Sequence Lengths

How does the length of the salient sequence affect model performance? To address this question, we varied the number of salient context sentences extracted from the source document (3, 5, 10, 15, 20 sentences) to conduct experiments. The complete experimental results are listed in Table 8, corresponding to Figure 3 in the main text of the paper.

D.2 GPT-4 evaluations

Table 9 presents GPT-4's evaluation results for ranking the summaries based on both quality and faithfulness jointly, corresponding to the prompt in Figure 5.

D.3 Inference Speed

Considering that SARA incorporates salient context sequences, which may increase inference time, we compared the runtime of different decoding methods. Specifically, we measured the inference time (in milliseconds per batch) for Vanilla decoding, CAD, and SARA on GPT-NEO, LLaMA, and OPT backbones using a single A100 GPU, with a batch size of 16. As shown in the experimental results in Table 10, SARA achieves the best performance with a slight trade-off in inference speed. Therefore, the proposed method is particularly well-suited for scenarios where performance is prioritized over inference speed.