Model-based Preference Optimization in Abstractive Summarization without Human Feedback

Anonymous ACL submission

Abstract

In abstractive summarization, the challenge of producing concise and accurate summaries arises from the vast amount of information contained in the source document. Consequently, although Large Language Models (LLMs) can generate fluent text, they often introduce inaccuracies by hallucinating content not found in the original source. While supervised fine-tuning methods that maximize likelihood contribute to this issue, they do not consistently enhance the faithfulness of the summaries. Preference-based optimization methods, such as Direct Preference Optimization (DPO), can further refine the model to align with human preferences. However, these methods still heavily depend on costly human feedback. In this work, we introduce a novel and straightforward approach called Model-based Preference Optimization (MPO) to fine-tune LLMs for improved summarization abilities without any human feedback. By leveraging the model's inherent summarization capabilities, we create a preference dataset that is fully generated by the model using different decoding strategies. Our experiments on standard summarization datasets and various metrics demonstrate that our proposed MPO significantly enhances the quality of generated summaries without relying on human feedback.

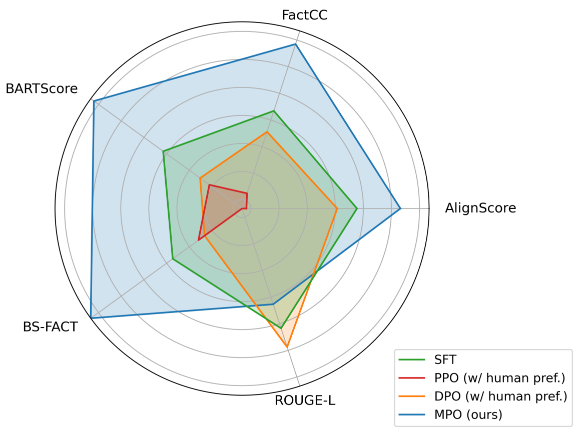

Figure 1: **Summarized results via automated metrics.** Our method MPO, which uses the model-generated summaries for preference optimization, proves to be more effective than PPO and DPO, both of which use human preference datasets for optimization. The results are from using the GPT-J on the TL;DR dataset.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities in generating fluent and plausible text (Wang and Komatsuzaki, 2021; Touvron et al., 2023a; Jiang et al., 2023). However, despite these advancements, LLMs often produce summaries that, while plausible, contain incorrect or contradictory information—a phenomenon known as *hallucination* (Maynez et al., 2020). The fundamental reason for this issue is that LLMs are primarily trained to predict the most likely next token based on maximum likelihood, which is the most common objective for pre-training language models (King et al., 2022). In principle, reinforcement learning based objectives can circumvent these failures by choosing an appropriate reward function (Paulus et al., 2017; Tian et al., 2024). Recently, reinforcement learning from human feedback (RLHF) has focused on aligning language models with human preferences, thereby effectively enhancing the models' summarization abilities (Böhm et al., 2019; Pasunuru and Bansal, 2018; Stiennon et al., 2020; Paulus et al., 2018; Ramamurthy et al., 2023).

While RLHF and other preference-based optimization methods (Rafailov et al., 2023) effectively fine-tune models to align with human preferences, human feedback is not always reliable. For example, even though the quality of text summaries depends on various factors, Hosking et al. (2024) demonstrated that human preferences often overlook factuality and consistency, which are crucial in avoiding hallucination. This implies that a summary judged as good by humans is not necessarily free from hallucination. In other words, preference optimization with human feedback does not guarantee improved faithfulness. Moreover, the use of human preference faces challenges related to the collection of human-annotated data. Although RLHF does not require massive amounts of data to enhance performance, sourcing high-quality human preference data remains an expensive process (Min et al., 2023).

To address these challenges, prior works have aimed to conduct preference optimization without relying on human preferences (Paulus et al., 2018; Tian et al., 2024; Wei et al., 2024; Roit et al., 2023). Such methods often require external metrics or complex filtering processes to establish preference pairs. For instance, Paulus et al. (2018) utilized lexical overlap (ROUGE) to assess salience and an entailment score to evaluate factual consistency. Similarly, Tian et al. (2024) employed FactScore (Min et al., 2023) to gauge reward signals between generated summaries. However, as stated by Goodhart's Law—'*When a measure becomes a target, it ceases to be a good measure*'—relying excessively on these imperfect metrics carries the risk of overfitting to the metrics alone (Strathern, 1997; Ramamurthy et al., 2023).

In response, we propose *Model-based Preference Optimization* (MPO), a novel and straightforward approach that leverages the model's inherent summarization capabilities without relying on any human feedback or external metrics. This method generates faithful summaries by aligning preferences between responses generated using different decoding strategies. In particular, we utilize (1) a deterministic decoding strategy (*e.g.*, beam search decoding) to generate chosen samples and (2) a stochastic decoding strategy (*e.g.*, temperature sampling) to generate rejected samples. Therefore, our approach does not require any external knowledge or metrics to construct preference pairs.

In previous studies, deterministic decoding strategies have been shown to produce results that are less surprising and more aligned with the source, whereas stochastic decoding introduces randomness and is more prone to hallucinations (Yang et al., 2018; Welleck et al., 2020a; Holtzman et al., 2020). Specifically, Wan et al. (2023) presented empirical evidence indicating that beam search yields the most faithful summaries, while the randomness introduced by sampling reduces faithfulness. Based on these findings, we align our model's preference toward summaries generated via beam search rather than those naively sampled. As illustrated in Figure 1, our approach outperforms models trained with standard supervised fine-tuning (SFT) or those optimized with human preferences (*e.g.*, PPO, DPO) in terms of faithfulness and relevance to the source text.

Our main contribution is Model-based Preference Optimization (MPO), a simple and straightforward approach for fine-tuning language models to improve abstractive summarization without relying on any human feedback or external metrics. Our experimental results demonstrate that MPO achieves superior overall performance compared to models optimized with human preferences, and it exhibits generalizability across various language models and datasets.

2 Preliminaries

2.1 Problem Setup

Let $\mathcal{V}$ denote the vocabulary for both input and output. We represent the input document as $\mathbf{x} \in \mathcal{X}$ and the output summary as $\mathbf{y} = \langle y_0, \ldots, y_T \rangle \in \mathcal{Y}$. The sequence $\mathbf{y}$ consists of $T+1$ elements, starting with the beginning-of-sequence token y_0 and ends with the end-of-sequence token y_T.

A language model (LM) is an auto-regressive model of a sequence distribution $P(\mathbf{y} \mid \mathbf{x})$, where each conditional probability is parameterized by a neural network p_θ. We assume that the model computes the probability of the entire generated text $\mathbf{y}$ using a common left-to-right decomposition. Thus, the distribution can be expressed as a product of conditional probabilities:

$$P(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{T} p_\theta(y_t|\mathbf{y}_{<t}, \mathbf{x}). \quad (148)$$

2.2 LM for Summarization

Given an input document $\mathbf{x}$, the optimal summary $\mathbf{y}$ from the set of valid strings $\mathcal{Y}$ is obtained using a scoring function:

$$\mathbf{y}^* = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}} p_\theta(\mathbf{y}|\mathbf{x}).$$

However, finding the optimal summary is not tractable. Therefore, the scoring function for the optimal string $\mathbf{y}$ varies according to decoding strategies to approximate the best possible output. There are two types of decoding strategies: stochastic and deterministic.

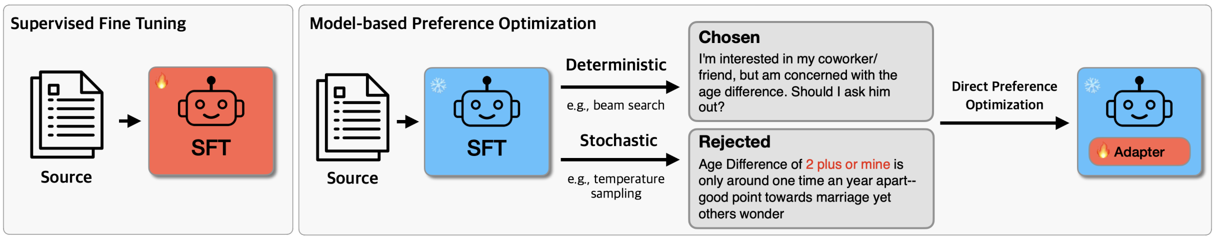

Figure 2: **Model-based Preference Optimization.** Our method follows a two-step process: 1) *Supervised Fine-Tuning* (SFT): we fine-tune a pre-trained model (*i.e.*, LLM) on a given dataset. 2) *Model-based Preference Optimization* (MPO): we build a preference dataset using different decoding strategies. In this step, the chosen samples are derived from deterministic decoding results, while the rejected samples utilize results generated by stochastic decoding.

Stochastic Decoding The simplest approach in decoding strategies is to sample directly from the probabilities predicted by the model. This method involves sampling from the conditional probability distribution at each step, represented as:

$$y_{\text{temp}} \sim P(y_t|\mathbf{x}, \mathbf{y}_{<t}).$$

However, this method exhibits high variance. To adjust for this variance, the temperature of the softmax function can be modified:

$$P(y_t|\mathbf{x}, \mathbf{y}_{<t}) = \text{softmax}\left(\frac{p_\theta(y_t|\mathbf{x}, \mathbf{y}_{<t})}{\tau}\right),$$

where τ is the temperature parameter. Increasing τ causesthe model's conditional probability distribution to approach a uniform distribution, which can lead to the generation of random tokens that are irrelevant to the source documents. Consequently, this increases the risk of the model producing hallucinations. For this reason, we classify samples generated through stochastic decoding as rejected samples in our preference dataset.

Deterministic Decoding The other strategies are deterministic decoding algorithms. The most straightforward algorithm, called greedy decoding, simply selects the most probable token at each step (Welleck et al., 2020a). This can be expressed as:

$$y_{\text{greedy}} = \underset{y \in \mathcal{V}}{\text{argmax}} \log p_\theta(y_t|\mathbf{y}_{<t}, \mathbf{x}).$$

In contrast to greedy decoding, beam search decoding considers the top-k samples for token generation. At each time step t, it tracks the k most likely sequence hypotheses, where k is the beam size. This can be represented as:

$$\mathbf{y}_{\text{beam}} = \underset{y \in \mathcal{V}}{\text{argmax}} \sum_{t=1}^{L} \log p_\theta(y_t|\mathbf{y}_{<t}, \mathbf{x}),$$

where L is the length of the final candidate sequence. These deterministic decoding strategies tend to produce tokens that are more closely related to the source document, resulting in more faithful summaries than those generated by stochastic decoding strategies. Therefore, we align our model's preference toward summaries generated via the deterministic decoding strategies and define them as chosen samples in our preference dataset.

3 Proposed Method

In this section, we detail our process for encouraging faithfulness in abstractive summarization. We follow the typical pipelines of preference optimization (Rafailov et al., 2023; Ziegler et al., 2020; Stiennon et al., 2020; Ouyang et al., 2022). However, by leveraging the differences between deterministic and stochastic decoding strategies, our pipeline does not require any external knowledge (*e.g.*, evaluation metrics) or human feedback. This pipeline is depicted in Figure 2.

3.1 Superveised Fine-Tuning (SFT)

For the summarization task, we first fine-tune a pre-trained language model using supervised learning on training data (*i.e.*, ground truth data), denoted as $\mathcal{D}^{\text{train}} = \{(\mathbf{x}, \mathbf{y}_{\text{ref}})\}$. Based on this supervised fine-tuning (SFT) approach, the model is trained to generate a single-sentence summary from a source document. In this work, we utilize existing SFT models with minimal modifications or apply SFT to pre-trained language models using QLoRA (Dettmers et al., 2023).

3.2 Preference Optimization

For preference optimization, we employ Direct Preference Optimization (DPO, Rafailov et al., 2023). DPO simplifies the process by eliminating the need for an explicit reward function, making it preferable to RL-based algorithms, which incur significant computational costs by training multiple language models and sampling from the policy.

Given a dataset of preference pairs $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i^w, \mathbf{y}_i^l)\}_{i=1}^N$, where $\mathbf{x}_i$ represents source documents, $\mathbf{y}_i^w$ are chosen responses, and $\mathbf{y}_i^l$ are rejected responses, the probability of observing a preference pair is modeled using the Bradley-Terry model (Bradley and Terry, 1952):

$$p(\mathbf{y}^w \succ \mathbf{y}^l) = \sigma(r(\mathbf{x}, \mathbf{y}^w) - r(\mathbf{x}, \mathbf{y}^l)),$$

where σ is the sigmoid function, and $r(\cdot, \cdot)$ is a reward function.

Rafailov et al. (2023) demonstrated that models directly learn this policy from collected data without modeling the reward function. In other words, the 2-stage policy can be simplified into 1-stage policy. DPO loss can be expressed as:

$$\begin{aligned}
&\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = \\
&- \mathbb{E}_{(\mathbf{x}, \mathbf{y}^w, \mathbf{y}^l) \sim \mathcal{D}} \Bigg[\log \sigma \Bigg(\beta \log \frac{\pi_\theta(\mathbf{y}^w \mid \mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}^w \mid \mathbf{x})} \\
&\qquad - \beta \log \frac{\pi_\theta(\mathbf{y}^l \mid \mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}^l \mid \mathbf{x})} \Bigg) \Bigg],
\end{aligned}$$

where π_{ref} is the SFT model and β is a coefficient that controls the trade-off between reward and divergence. By optimizing this objective, the model aligns with the reward function while remaining close to the pre-trained reference model, thus minimizing over-optimization (Tian et al., 2024).

3.3 Constructing Preferences Pairs without Human Feedback

By exploiting the differences between deterministic and stochastic strategies, we construct a dataset of preference pairs, denoted as $\mathcal{D}^{\text{valid}} = \{(\mathbf{x}, \mathbf{y}_{\text{beam}}^w, \mathbf{y}_{\text{temp}}^l)\}$. This strategy is based on the observation that deterministic decoding typically produces more factual summaries (Wan et al., 2023). This significant difference in output quality suggests that summaries generated through beam search decoding can be used as chosen samples, while those from temperature sampling can be designated as rejected samples. We then conduct preference optimization with this generated data to refine the language model, ensuring it avoids generating hallucinated or irrelevant text.

4 Experiments

4.1 Experimental Setup

Dataset We used the TL;DR dataset and the eXtreme Summarization (XSUM) dataset (Cachola et al., 2020; Narayan et al., 2018). The TL;DR dataset is constructed by Reddit posts and their corresponding TL;DR summaries, while the XSUM dataset consists of BBC articles and their single-sentence summaries. Both datasets are widely used for abstractive summarization tasks.

Models To verify the generalizability of our method, we utilized GPT-J (6B) (Wang and Komatsuzaki, 2021), Mistral-7B (Jiang et al., 2023) and LLaMA2-7B (Touvron et al., 2023b) for TL;DR dataset and Mistral-7B and LLaMA2-7B for XSUM dataset. For GPT-J model, we used a checkpoint from Huggingface[1], that was already fully fine-tuned on the train dataset. For LLaMA2-7B and Mistral-7B models, we performed Supervised Fine-Tuning (SFT) on each training dataset using QLoRA, and then merged the adapter into the models for further preference optimization experiments. We limited our experiments to 7B models due to the constraints of our experimental environment.

Evaluation Metrics We adopt the evaluation protocol proposed by Chae et al. (2024). They categorized the evaluation into three key divisions: *Faithfulness*, *Relevance* (with the source), and *Similarity* (with the target). For *Faithfulness*, we used AlignScore (Zha et al., 2023) and FactCC (Kryscinski et al., 2020). To measure *Relevance*, we employed BARTScore (Yuan et al., 2021) and BS-FACT. Lastly, to evaluate *Similarity*, we used ROUGE-L. It is important to note that ROUGE-L compares the generated summary with the target summary rather than the source text, which is not our primary concern.

Implementation Details For the SFT training, we utilized QLoRA with a batch size of 2 and a learning rate of 1e-4, training for one epoch in training split. After training, the SFT-trained QLoRA was merged with the pre-trained model. For preference optimization, we set the DPO hyperparameter β to 0.5. The learning rate was set to 1e-4 with a batch size of 4, and training was also conducted for one epoch in the validation split. During summary

[1] CarperAI/openai_summarize_tldr_sft

Dataset (Model)	Method	Response Ratio	*Faithfulness*		*Relevance*		*Similarity*
			AlignScore (↑)	FactCC (↑)	BARTScore (↑)	BS-FACT (↑)	ROUGE-L (↑)
TL;DR (GPT-J)	*with ground-truth data*						
	SFT	81.2% (99.4%)	89.21 (83.54)	64.18 (53.48)	-1.25 (-1.63)	91.53 (90.30)	26.74 (26.01)
	SFT++	93.8% (99.7%)	87.29 (82.30)	61.50 (57.05)	-1.37 (-1.63)	91.06 (90.11)	**27.47 (26.53)**
	with human feedback (preference dataset)						
	PPO	100.0% (100.0%)	83.10 (75.88)	54.40 (47.52)	-1.35 (-1.80)	91.32 (89.78)	23.55 (23.28)
	DPO	98.3 (99.8%)	88.12 (82.55)	61.70 (54.09)	-1.33 (-1.65)	91.27 (90.22)	27.24 (26.28)
	without human feedback						
	Preferred-FT	66.8% (99.6%)	89.90 (82.04)	**76.58 (64.48)**	-1.39 (-1.73)	91.24 (90.09)	24.38 (24.39)
	MPO (Ours)	99.9% (99.9%)	**91.61 (86.82)**	72.10 (59.39)	**-1.10 (-1.41)**	**92.20 (91.20)**	26.10 (26.49)

Table 1: **Results of the GPT-J model on the TL;DR dataset.** We compared our Model-based Preference Optimization (MPO) with two main baselines: *supervised fine-tuning* and *human preference*. All main results are based on a beam search decoding strategy, while the results in parentheses are based on a greedy decoding strategy. MPO showed overall better performance in terms of *faithfulness* and *source relevance* compared to other baselines. The SFT model is a fine-tuned model on the training split and the SFT++ model is the SFT model further fine-tuned on the validation split. PPO and DPO are SFT models optimized on human-preference datasets. Preferred-FT is a model fine-tuned only on the chosen samples of MPO.

generation, the maximum number of generated tokens was limited to 50. For beam search decoding, we used beam size of 6. For temperature sampling, we employed temperatures of 5.0 for GPT-J, and 1.0 for Mistral-7B and LLaMA2-7B.

Baselines We compared our method with two main baselines: *supervised fine-tuning* and *human preference*. First, we compared our approach against models fine-tuned using either human-annotated summaries or summaries generated through deterministic decoding. Second, we compared our method with PPO and DPO models trained on human preference pairs to demonstrate that the contrast between beam search decoding and random sampling is more effective than human-annotated preferences in terms of faithfulness.

SFT is a fine-tuned model on the train split of each dataset. **SFT++** is a model further trained on a validation split from the SFT model. **Preferred-FT** is fine-tuned to maximize likelihood only on the chosen samples (*i.e.*, y_{beam}). **PPO** and **DPO** are optimized from SFT models on human preference dataset provided by Stiennon et al. (2020). For PPO, we used a Huggingface checkpoint[2], already optimized with the provided human preference dataset. For DPO, we optimized in the same way as MPO but with the human preference dataset.

4.2 Comparison with Fine-Tuned Models

In Table 1, MPO consistently outperforms fine-tuned baselines (*i.e.*, SFT, SFT++, Preferred-FT). SFT++ and Preferred-FT did not significantly improve over SFT. However, MPO shows a substantial increase of up to 3.28 in AlignScore, 7.92 in FactCC, 0.22 in BARTScore, and 0.9 in BS-FACT over SFT. These results suggest that our approach is more effective at mitigating hallucinations than simply fine-tuning with either gold summaries or summaries generated through deterministic decoding. In Table 2, MPO demonstrates robust and generally applicable results across various language models (Mistral-7B, LLaMA2-7B) on both the TL;DR and XSUM datasets.

Dataset	Model	Method	AlignScore (↑)	BARTScore (↑)	ROUGE-L (↑)
TL;DR	Mistral	SFT	87.85 (82.74)	-1.48 (-1.81)	25.32 (25.02)
		MPO	92.12 (89.39)	-1.25 (-1.37)	24.85 (25.01)
	LLaMA2	SFT	84.92 (77.68)	-1.65 (-2.05)	24.31 (23.33)
		MPO	85.33 (78.03)	-1.64 (-2.03)	24.16 (23.29)
XSUM	Mistral	SFT	66.31 (60.00)	-1.96 (-1.97)	30.65 (31.16)
		MPO	68.58 (64.57)	-1.85 (-1.90)	31.11 (31.35)
	LLaMA2	SFT	65.80 (57.57)	-1.80 (-2.06)	30.36 (27.76)
		MPO	67.31 (60.48)	-1.81 (-2.02)	30.32 (28.36)

Table 2: **Comparison of MPO with SFT**. MPO demonstrates generally robust results across various language models (Mistral and LLaMA2) on both the TL;DR and XSUM datasets. The results are based on a beam search decoding strategy, while the results in parentheses are based on a greedy decoding strategy.

4.3 Comparison with Human Preference Optimized Models

In Table 1 and 3, we compared MPO with human preference optimized models (*e.g.*, PPO, DPO). From the perspective of automatic metrics in Table 1, MPO shows overall better results compared to the human preference optimized models. As noted in Hosking et al. (2024), utilizing a human preference dataset can underestimate the faithfulness

[2] CarperAI/openai_summarize_tldr_ppo

GPT-3.5	SFT (vs. MPO)		DPO (vs. MPO)	
	Greedy	Beam	Greedy	Beam
# of compared samples	6061	5376	5962	5332
MPO win rate (%)	**51.30**	**59.36**	**50.27**	47.30

Table 3: **Comparing GPT-3.5 win rates on TL;DR summarization samples.** Samples from different methods are compared only if they are not exactly the same.

aspect.

On the other hand, as shown in Table 3, the MPO did not exhibit a dominant performance compared to others in the win rate evaluation based on GPT-3.5. For details on the win rate prompts, refer to Appendix A.1. This discrepancy arises because summary evaluation involves various factors (Hosking et al., 2024; Yuan et al., 2021). While MPO excels in faithfulness and source relevance, it may fall short in aspects like fluency (refer to Table 4). Additionally, human preference optimized models were trained on significantly more data pairs than MPO, utilizing multiple pairs per source text, whereas MPO is optimized on only one pair per source.

4.4 Comparison with Decoding Strategies

Table 5 shows the results of applying MPO models to various decoding strategies using the LLaMA2-7B model. Despite not being specifically optimized for various decoding strategies (*i.e.*, Nucleus (Holtzman et al., 2020), ITI (Li et al., 2023), DoLa (Chuang et al., 2023)), MPO models are generally applicable to all decoding strategies and consistently produces enhanced summarization results compared to the standard SFT model in terms of faithfulness and relevance.

5 Analysis

5.1 Other Combinations for Preference Pairs

Decoding strategies primarily include two methods: deterministic decoding and stochastic decoding. Our method uses summaries from deterministic decoding as chosen responses and summaries from stochastic decoding as rejected responses. To justify this choice, we explored different combinations of chosen and rejected responses, and the accuracy is summarized in Table 6.

Deterministic decoding preference pairs To test whether improving the quality of rejected responses would enhance the model's summarization performance, we used beam search decoding for the chosen responses and greedy decoding for the rejected responses. However, this approach significantly reduced accuracy (see row 3 in Table 6). Generated sample can be found in Appendix A.2. One reason we identified is that the summaries generated by beam search decoding and greedy decoding are too similar, causing confusion for the model. Specifically, the similarity between the summaries produced by the two methods, shown in row 1 of Table 7, is indicated by very high ROUGE scores.

Method	Text
Source	TITLE: [19/f] What does this guy [20/m] actually want from me? POST: . . . became really good friends, . . . We then somehow from kissing gently . . . basically said he likes me but nothing can happen because I'm not his type... I JUST DON'T KNOW WHAT THE BOY WANTS FROM ME.
SFT	ive been friends with a guy for a while, then we kissed, **then we didn't, then we did again, then we didn't, then we did again.**
DPO	I don't know what the boy wants from me, **and I don't know what I want from the boy.**
MPO (Ours)	Became really good friends with a guy, then we kissed, then he said he likes me but I'm not his type. What does he want from me?

Table 4: **Example summaries of MPO model and human preference optimized model.** Inconsistent words are highlighted in **red**. The summary generated by the MPO model is clearly superior to those by SFT and DPO (w/ human pref.) models in terms of faithfulness and source relevance.

Decoding Strategy	Method	AlignScore (↑)	BARTScore (↑)	ROUGE-L (↑)
Greedy	SFT	77.68	-2.05	23.33
	MPO	78.03	-2.03	23.29
Nucleus	SFT	76.25	-2.11	22.82
	MPO	76.99	-2.09	22.79
ITI	SFT	76.95	-1.88	23.15
	MPO	77.15	-1.87	23.23
DoLa	SFT	82.47	-1.76	24.61
	MPO	82.57	-1.75	24.55
Beam	SFT	84.92	-1.65	24.31
	MPO	85.33	-1.64	24.16

Table 5: **Results of applying various decoding strategies**. MPO aligns well with different decoding strategies. When combined with faithfulness-aware decoding strategies (*i.e.*, ITI, DoLA), it can lead to further improvements. The results are from using the LLaMA2-7B on the TL;DR dataset.

Combination	AlignScore (↑)	BARTScore (↑)	ROUGE-L (↑)
SFT	89.21	-1.25	26.74
$(y^w_{\text{beam}}, y^l_{\text{greedy}})$	51.96	-4.63	0.87
$(y^w_{\text{temp5}}, y^l_{\text{beam}})$	87.59	-1.36	27.24
$(y^w_{\text{greedy}}, y^l_{\text{temp5}})$	90.57	-1.20	26.87
$(y^w_{\text{beam}}, y^l_{\text{temp5}})$	91.61	-1.10	26.10

Table 6: **MPO with different combinations of preference pairs.** The result show that using a deterministic decoding strategy pair significantly inhibit summarization ability. For pairs combining deterministic and stochastic decoding, setting beam search as the chosen and temperature-based sampling as the rejected maximizes the language model's summarization performance. The results are from using the GPT-J on the TL;DR dataset.

Pairs	ROUGE-1 (↑)	ROUGE-2 (↑)	ROUGE-L (↑)
y^w_{beam} vs. y^l_{greedy}	47.38	35.06	43.24
y^w_{greedy} vs. y^l_{temp5}	12.93	0.49	9.00
y^w_{beam} vs. y^l_{temp5}	10.56	0.41	7.40

Table 7: **ROUGE score comparison.** Deterministic decoding generated summaries exhibit high similarity, whereas there is low similarity between summaries generated by deterministic decoding and those generated by stochastic decoding.

This suggests that using overly similar summaries as chosen and rejected responses in preference optimization can have adverse effects (Pal et al., 2024).

Stochastic decoding as chosen responses To test whether the model's summarization performance improves whenever there is a clear distinction between chosen and rejected samples, we used sampling-based stochastic decoding for the chosen samples and beam search decoding for the rejected samples. As a result, while this approach did not cause the degeneration seen in cases where the similarity between samples was very high (refer to Table 8 in Appendix A.2), it led to lower faithfulness compared to the original SFT model (see Table 6). This indicates that if the chosen samples have lower source-alignment compared to the rejected samples, preference optimization can degrade the model's existing summarization capabilities.

5.2 Faithfulness-Abstractiveness Tradeoff from Iterative Training

Recent studies by Pang et al. (2024) and Chen et al. (2024) have demonstrated that iteratively constructing the preference dataset using the trained model from the previous iteration improves dataset quality. Building on these works, our approach extends Preference Optimization to Iterative Preference Optimization.

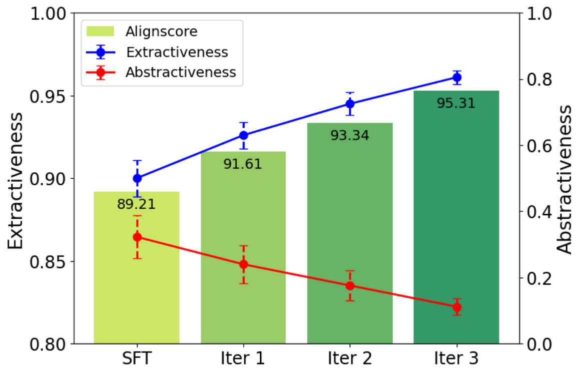

Figure 3: **Analysis for each training iteration.** The average abstractiveness of summaries generated for the TL;DR test set across training iterations, measured by the MINT score, with dotted lines indicating variance. The average extractiveness is measured by extractive fragment coverage.

For this experiment, We employed beam search decoding outputs from the previous iteration as chosen data for subsequent training phases, while summaries generated by random sampling outputs from the SFT model were used as rejected data. We dynamically adjusted the difficulty of the tasks by decreasing the temperature settings—5.0, 3.0, 1.0—for each iteration to adapt to the continuous enhancements in model performance.

We observed a notable trend where the model increasingly produced more extractive summaries, often directly incorporating sentences from the source documents. This trend can be attributed to the slightly extractive nature of the summaries generated by the SFT model using beam search decoding, which were used as the chosen samples (Ladhak et al., 2022). Conversely, the rejected samples, generated through temperature-scaled sampling, suppressed the creativity of summaries. Consequently, as shown in Figure 3, the model's faithfulness improved with increased extractiveness over successive iterations[3].

Qualitative study In Appendix A.2, Table 9 provides an example of summaries generated by the SFT model and by the MPO model at different iterations in response to a given prompt. As the iterations progress, the summaries tend to become more extractive for the document. Notably, the summary generated in the third iteration is quite similar to the title.

[3]To quantitatively assess the abstractiveness and extractiveness, we utilized the MINT (Metric for lexical independence of generated text) (Dreyer et al., 2023) and *extractive fragment coverage* (Grusky et al., 2018), respectively.

5.3 Encoder-Decoder Model

To verify the generalizability of our method across different model architectures, we evaluated our approach using an encoder-decoder model, such as BART (Lewis et al., 2019). As shown in Appendix A.3, MPO outperforms SFT in terms of AlignScore, improving from 61.86 to 66.42. Furthermore, we compared MPO with another decoding strategy baseline, *Faithfulness-aware Lookahead* (Wan et al., 2023), which has shown effectiveness in encoder-decoder models. Interestingly, by using the summary from Faithfulness-aware Lookahead as the chosen samples instead of the beam search summaries (*i.e.*, MPO*), MPO* increased the AlignScore by 2.43 over MPO. This indicates that utilizing better decoding strategies in MPO can further enhance the summarization performance.

6 Related Work

In the realm of auto-regressive language models, there are two primary approaches aimed to enhance the model's summarization capabilities: adjusting the learning algorithm or refining the decoding strategy (Welleck et al., 2020b). The former involves updating the model's parameters through a learning objective, while the latter entails improving the decoding algorithm during generation while maintaining the existing pre-trained parameters frozen. In this paper, we will review two approaches in abstractive summarization aimed at alleviating hallucination.

Faithfulness-aware Decoding Strategies Several methods have been proposed to rectify hallucinations during generation. Inference-time intervention (ITI) shifts activations along truth-correlated directions (Li et al., 2023), repeating the same intervention auto-regressively until the entire answer is generated. Decoding by contrasting layers (DoLa) uses an early-exit strategy by contrasting the differences in logits obtained from projecting the later layers versus earlier layers (Chuang et al., 2023). Lastly, Wan et al. (2023) extend the idea of lookahead (Lu et al., 2022) to improve faithfulness in abstractive summarization, showing that the deterministic decoding strategy outperforms nucleus sampling (Holtzman et al., 2020) in terms of faithfulness. However, it is important to note that decoding strategies do not change the underlying model.

Faithfulness-aware Learning Algorithms To mitigate hallucinations, naively fine-tuning with faithfulness-aware objectives might seem straightforward. FactPegasus (Wan and Bansal, 2022) employs a tailored pre-training setup with contrastive learning to generate more faithful summaries. It modifies sentence selection by combining ROUGE and FactCC (Kryscinski et al., 2020). However, this method risks overfitting to the metrics used, potentially degrading overall summarization performance (Chae et al., 2024).

As an alternative, RL-based objectives can be utilized to enhance faithfulness (Böhm et al., 2019; Roit et al., 2023; Paulus et al., 2018). RL provides a natural path for optimizing non-differentiable objectives in LM-based generation. Ramamurthy et al. (2023) show that RL techniques generally align language models to human preferences better than supervised methods. On the other hand, Direct Preference Optimization (DPO)(Rafailov et al., 2023) simplifies the process by eliminating the need for an explicit reward function of RL-based algorithms. Leveraging DPO, Tian et al. (2024) have suggested optimizing language models for factuality in long-form text generation using FactScore (Min et al., 2023).

In this paper, we train the underlying model to provide summaries faithful to source documents, based on findings from research on decoding strategies. Our approach does not require external metrics or human feedback during the optimization process. Furthermore, the model trained on our framework is versatile enough to integrate enhanced decoding techniques, thereby more effectively reducing hallucinations.

7 Conclusion

This study introduces Model-based Preference Optimization (MPO), a novel approach to improve the faithfulness and quality of abstractive summaries generated by Large Language Models (LLMs). Unlike traditional methods that rely heavily on costly human feedback, MPO leverages the model's inherent summarization capabilities to create a preference dataset using different decoding strategies. Our extensive experiments demonstrate that MPO significantly enhances the summarization performance, providing an efficient and scalable solution to address the challenges of hallucination in LLM-generated summaries.

Limitation

In our experiments, we employed QLoRA to maintain the performance of the SFT model, but this method may have imposed limitations on potential performance improvements. The lack of comparative experiments to substantiate the effectiveness of QLoRA leaves some uncertainty regarding its impact. Due to computational cost constraints, it is also unclear whether similar results can be achieved with larger language models, raising questions about the scalability of our approach.

During iterative training, we observed a trend where the model increasingly adopted an extractive approach, often replicating sentences from the input documents directly in the summaries. This trend poses a challenge to our goal of producing more faithful abstractive summaries.

Ethical Concerns

We propose MPO, which leverages the outputs of a language model as a dataset for preference optimization, relying extensively on the outputs from the SFT model. Previous researches (Sheng et al. (2019), Nangia et al. (2020)) has shown that self-supervised language models, which are trained on unlabeled web-scale datasets, can unintentionally learn and perpetuate social and ethical biases, including racism and sexism. If such biases are inherent within the data, our proposed self-feedback framework may unintentionally reinforce them. We used the TL;DR dataset for training, derived from Reddit posts, which may contain unmoderated and biased expressions. The presence of offensive content in this dataset risks influencing the model's outputs, potentially perpetuating these biases in further training within MPO. Moreover, as MPO progresses and the model increasingly favors extractive summarization, it may struggle to effectively paraphrase and filter out offensive expressions.

References

Florian Böhm, Yang Gao, Christian M. Meyer, Ori Shapira, Ido Dagan, and Iryna Gurevych. 2019. Better rewards yield better summaries: Learning to summarise without references. 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 3110–3120, Hong Kong, China. Association for Computational Linguistics.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel Weld. 2020. TLDR: Extreme summarization of scientific documents. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4766–4777, Online. Association for Computational Linguistics.

Kyubyung Chae, Jaepill choi, Yohan Jo, and Taesup Kim. 2024. Mitigating hallucination in abstractive summarization with domain-conditional mutual information. In *2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2023. Dola: Decoding by contrasting layers improves factuality in large language models.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Markus Dreyer, Mengwen Liu, Feng Nan, Sandeep Atluri, and Sujith Ravi. 2023. Evaluating the tradeoff between abstractiveness and factuality in abstractive summarization. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2089–2105.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration.

Tom Hosking, Phil Blunsom, and Max Bartolo. 2024. Human feedback is not gold standard. In *The Twelfth International Conference on Learning Representations*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Daniel King, Zejiang Shen, Nishant Subramani, Daniel S. Weld, Iz Beltagy, and Doug Downey. 2022.

Don't say what you don't know: Improving the consistency of abstractive summarization by constraining beam search. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 555–571, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Faisal Ladhak, Esin Durmus, He He, Claire Cardie, and Kathleen McKeown. 2022. Faithful or extractive? on mitigating the faithfulness-abstractiveness trade-off in abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1410–1421, Dublin, Ireland. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model.

Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, Noah A. Smith, and Yejin Choi. 2022. NeuroLogic a*esque decoding: Constrained text generation with lookahead heuristics. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 780–799, Seattle, United States. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation.

Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel Bowman. 2020. Crows-pairs: A challenge dataset for measuring social biases in masked language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1953–1967.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. 2024. Smaug: Fixing failure modes of preference optimisation with dpo-positive.

Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. 2024. Iterative reasoning preference optimization.

Ramakanth Pasunuru and Mohit Bansal. 2018. Multi-reward reinforced summarization with saliency and entailment. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 646–653, New Orleans, Louisiana. Association for Computational Linguistics.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model.

Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 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*.

Paul Roit, Johan Ferret, Lior Shani, Roee Aharoni, Geoffrey Cideron, Robert Dadashi, Matthieu Geist, Sertan Girgin, Leonard Hussenot, Orgad Keller, Nikola Momchev, Sabela Ramos Garea, Piotr Stanczyk, Nino Vieillard, Olivier Bachem, Gal Elidan, Avinatan Hassidim, Olivier Pietquin, and Idan Szpektor. 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, Toronto, Canada. Association for Computational Linguistics.

Emily Sheng, Kai-Wei Chang, Prem Natarajan, and Nanyun Peng. 2019. The woman worked as a babysitter: On biases in language generation. 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 3407–3412.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021. Curran Associates, Inc.

Marilyn Strathern. 1997. 'improving ratings': audit in the british university system. *European Review*, 5(3):305–321.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. 2024. Fine-tuning language models for factuality. In *The Twelfth International Conference on Learning Representations*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, and Shengyi Huang. 2020. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`.

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, Seattle, United States. Association for Computational Linguistics.

David Wan, Mengwen Liu, Kathleen McKeown, Dreyer Markus, 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*.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Jiaheng Wei, Yuanshun Yao, Jean-Francois Ton, Hongyi Guo, Andrew Estornell, and Yang Liu. 2024. Measuring and reducing llm hallucination without gold-standard answers.

Sean Welleck, Ilia Kulikov, Jaedeok Kim, Richard Yuanzhe Pang, and Kyunghyun Cho. 2020a. Consistency of a recurrent language model with respect to incomplete decoding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5553–5568, Online. Association for Computational Linguistics.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020b. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*.

Yilin Yang, Liang Huang, and Mingbo Ma. 2018. Breaking the beam search curse: A study of (re-)scoring methods and stopping criteria for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3054–3059, Brussels, Belgium. Association for Computational Linguistics.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348, Toronto, Canada. Association for Computational Linguistics.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. Fine-tuning language models from human preferences.

A Appendix

A.1 GPT-3.5 Judgment Prompts

We use *GPT-3.5-turbo* to evaluate win rates using prompts proposed in Rafailov et al. (2023). The order of summaries or responses is randomly chosen for each evaluation. The prompt examples we used can be seen in Figure 4.

```
Which of the following summaries does a better job of summarizing the most \
important points in the given forum post, without including unimportant or \
irrelevant details? A good summary is both precise and concise.

Post:
<post>

Summary A:
<Summary A>

Summary B:
<Summary B>

FIRST provide a one-sentence comparison of the two summaries, explaining which \
you prefer and why. SECOND, on a new line, state only "A" or "B" to indicate your \
choice. Your response should use the format:
Comparison: <one-sentence comparison and explanation>
Preferred: <"A" or "B">
```

Figure 4: **Summarization win rate prompt.**

A.2 Example Cases

Table 8 shows examples of summaries with different combinations of preference pairs. Table 9 shows examples summaries from iterative preference optimization.

A.3 Encoder-Decoder Model

We conducted experiments with the BART-Large model fine-tuned on the XSUM dataset (SFT). In Table 10, we demonstrate that our approach can also be applied to encoder-decoder models. Moreover, the results of MPO* demonstrate that using faithfulness-aware decoding instead of beam search as the chosen response can yield further improvements compared to MPO.

A.4 License Information of The Assets Used in This Work

Datasets We report known license information of the assets used in this work. The following datasets used in this paper are under the MIT License: XSUM (Narayan et al., 2018). The following datasets used in this paper are under the CC BY 4.0 License: TL;DR (Cachola et al., 2020).

Models We report known license information of the assets used in this work. The following datasets used in this paper are under the Apache 2.0 License: GPT-J (Wang and Komatsuzaki, 2021), Mistral-7B (Jiang et al., 2023), BART (Lewis et al., 2019). The following datasets used in this paper are under the Llama2 License: LLaMA2-7B (Touvron et al., 2023b)

Source code We use the implementation of existing baseline methods for reporting their results in this paper. The source code utilized in this paper is subject to the MIT License: MINT (Dreyer et al., 2023), ITI (Li et al., 2023), AlignScore (Zha et al., 2023), DoLa (Chuang et al., 2023), DCPMI (Chae et al., 2024) The following source code utilized in this paper is subject to the BSD 3-Clause License: FactCC (Kryscinski et al., 2020) The following source code utilized in this paper is subject to the CC-BY-NC-4.0 License: Lookahead (Wan et al., 2023) The following source code utilized in this paper is subject to the Apache 2.0 License: BARTScore (Yuan et al., 2021), trl/examples/research_projects/stack_llama_2 (von Werra et al., 2020)

A.5 Statistics for Data

We utilized two abstractive summarization datasets, TL;DR and XSUM. The TL;DR dataset is constructed by Reddit posts and their corresponding summaries, with 117k samples in the train split, 6.45k in the validation split, and 6.55k in the test split. The XSUM dataset consists of BBC articles and their corresponding summaries, totaling 204k samples in the train split, 11.3k in the validation split, and 11.3k in the test split. Both datasets are in English.

The train splits from each dataset were used during the SFT phase, the validation splits during the preference optimization phase, and the test splits during the evaluation phase.

A.6 Analysis on Error Bars

All experiments were evaluated in single run, fixing the seed at 42. Additionally, all summary generations were conducted in the order of the provided test dataset.

A.7 Reproducibility

We conducted our experiments using computing clusters equipped with NVIDIA RTX 6000 (GPU memory: 48GB) and NVIDIA RTX 3090 GPUs (GPU memory: 24 GB), allocating a single GPU for each experiment.

Based on NVIDIA RTX 6000, model preference optimization typically required an average of 1 hour and 30 minutes. When generating summaries, using GPT-J (6B) with beam search decoding took approximately 20 hours, and with greedy decoding, about 5 hours and 30 minutes. Using Mistral-7B and LLaMA-7B models with beam search decoding took around 5 hours, while with greedy decoding, it took about 1 hour and 30 minutes.

A.8 Parameters for Package

For evaluating summaries, we loaded ROUGE and BERTScore from the evaluate package (version: 0.4.1).

Method	Text
Source	SUBREDDIT: r/relationships TITLE: Is she [21] playing hard to get or uninterested in me?[22/M]POST: Hey guys first post here. So I'll try to make this quick, I've been out of the dating scene for a few years now and need advice with a girl i currently like. Her and I met in class and have been talking for not too long a month or so. We have tons in common I have gotten her number and we text every now and then (more on that later.) But I have really I've made the most progress in the past week. So everything explained further takes place in a span of a week. I've hung out with her a few times. The times we have hung out have been good we seem to hit it off. She's kinda touchy and takes lots of pictures of us (bff on sc if that means anything.) She said things like I'm special for getting to see her act in her goofy way. She even made and brought me a sandwich for when we were studying together. But ever since then she seems less interested in me and we do not text as often. Which is weird cuz that was to me at least the time we hit it off the most. Before We had been texting all day but now barely at all and remember this is all in a span of less than a week. Most recently we were supposed to hangout but she said she forgot (which is a big red flag I know) but we did reschedule for another specific day. So is she uninterested? Playing hard to get? Or other? TL;DR:
MPO $(\mathbf{y}^{w}_{\text{beam}}, \mathbf{y}^{l}_{\text{greedy}})$	\ufffd \ufffd
MPO $(\mathbf{y}^{w}_{\text{beam}}, \mathbf{y}^{l}_{\text{temp5}})$	Girl I like seems less interested in me after we hit it off for a week. Is she uninterested? Playing hard to get? Or other?
MPO $(\mathbf{y}^{w}_{\text{temp5}}, \mathbf{y}^{l}_{\text{beam}})$	ive been talking to a girl for less than a month and we seem to hit it off but now she seems less interested.

Table 8: **Example of summaries with different combinations of preference pairs.** In the case of $(\mathbf{y}^{w}_{\text{beam}}, \mathbf{y}^{l}_{\text{greedy}})$, the quality of the generated summaries significantly deteriorated. When there is a clear distinction between preferred data and rejected data, as observed in the two models below, the generated summaries remain similar even if the preferred and rejected data are swapped.

Method	Text
Source	SUBREDDIT: r/AskReddit TITLE: **Any doctors or med students lurking? What is the process to become a doctor like?** POST: I would like to become an anesthesiologist one day. I have thought about this a lot and would like to hear what it's like from some insiders. I am really not the type who can sit behind a desk all day. I'm prepared to sit at desks for studying, but I can't do it for 30+ years. I don't want to wear a suit and tie. I like to be up running around. Nothing else really interests me aside from extreme sports. I want to put my work in now and come out making a high salary. I don't want to climb the rungs of a corporate ladder. I've had friends with dads in construction, working 80+ hours a week. My dad as an engineer and programmer worked 80+ hours a week for years and his salary still isn't as high as an anesthesiologists (it's great for an engineer). I really just want to become an anesthesiologist.TL;DR:
SFT	I want to become an anesthesiologist. I'm not sure if I'm going about it the right way.
MPO (iteration 1)	I want to become an anesthesiologist. I want to know what it's like from an insider. What is the process like?
MPO (iteration 2)	I want to become an anesthesiologist. I want to know what it's like from an insider. **What is the process to become a doctor like?**
MPO (iteration 3)	**Any doctors or med students lurking? What is the process to become a doctor like?** I want to become an anesthesiologist one day.

Table 9: **Example summaries for iterative preference optimization.** As the iterations progress, an increase in the extractiveness of the summaries is observed, with summaries increasingly incorporating sentences directly from the source. Sentences in **bold** indicate exact matches to the source text.

Dataset	Method	AlignScore (↑)	BARTScore(↑)	ROUGE-L(↑)
XSUM	SFT	61.86	-1.80	**36.42**
	MPO $(\mathbf{y}^{w}_{\text{beam}}, \mathbf{y}^{l}_{\text{temp}})$	66.42	-1.80	35.78
	Lookahead (Wan et al., 2023)	67.78	-1.76	34.3
	MPO* $(\mathbf{y}^{w}_{\text{Lookahead}}, \mathbf{y}^{l}_{\text{temp}})$	**68.85**	**-1.73**	34.93

Table 10: **Results of Experiments for the Encoder-Decoder Model.**