Optimising Factual Consistency in Summarisation via Preference Learning from Multiple Imperfect Metrics

Anonymous ACL submission

Abstract

Recent work on language models often applies reinforcement learning with human-annotated preference data to enhance specific capabilities, such as generating informative summaries. However, such data often focuses on overall preferences and overlooks factuality. Since collecting new annotations is costly, we propose to use automatic factuality metrics to obtain factuality preference labels. While individual factuality metrics are limited, their combination can effectively capture diverse factual errors. We introduce an automated training pipeline that improves summarisation factuality via preference optimisation. For each source document, we generate lexically similar summary pairs by varying decoding strategies, ensuring the model learns from minor factual errors. To avoid human annotation, we derive preference labels from weak factuality metrics filtering out conflicting cases to improve reliability. This results in a high-quality preference dataset constructed with only source documents. Experiments show consistent factuality gains across models, ranging from early encoder-decoder architectures to modern large language models, with smaller models reaching comparable factuality to larger ones. Code and data will be released upon acceptance.

1 Introduction

Cutting-edge language models have demonstrated impressive capabilities in generating fluent and coherent responses to a wide range of prompts. However, maintaining faithfulness and factual consistency remains a persistent challenge, particularly in tasks like summarisation. Despite their surface plausibility, model-generated summaries often contain factual inconsistencies or hallucinated details.

Recent research has tried to mitigate this issue by incorporating reinforcement learning (RL) to guide models towards more factually consistent outputs. A critical obstacle lies in designing effective reward signals that can reliably capture and quantify factuality. Many approaches (Gao et al., 2018; Roit et al., 2023; Pasunuru and Bansal, 2018; Ye and Simpson, 2023; Wan and Bansal, 2022) adopt automatic evaluation metrics developed in earlier work (Lin, 2004; Zhang et al., 2020; Laban et al., 2022) as reward signals for RL. However, even state-of-the-art metrics struggle with subtle inconsistencies and may penalise factually accurate outputs (Tang et al., 2023). Using a single metric as an RL signal, as explored in prior work (Roit et al., 2023), is limited by the metric's reliability. Although combining metrics can broaden error detection coverage (Ye et al., 2024), existing RL methods often rely on manual weighting of sub-rewards (Gao et al., 2018; Pasunuru and Bansal, 2018; Ye and Simpson, 2023), reintroducing reward design complexity.

Another alternative is Reinforcement Learning with Human Feedback (Ouyang et al., 2022, RLHF), which uses human annotated preference data. While this approach has seen success in aligning large language models (LLMs) with general human values, its applicability to factuality is limited. Annotator biases, misunderstandings, and the scarcity of factuality-focused datasets reduce its effectiveness in this context (Hosking et al., 2024). Creating high-quality factuality-focused preference datasets is resource-intensive and requires expertise, making scalability a significant concern.

To overcome these barriers, this paper proposes a fully automated training pipeline that improves factual consistency in summarisation without relying on human annotations or reference summaries. Our method is model-driven, using the language model itself to generate two summaries by either selecting alternative candidate outputs from the same decoding strategy or using different decoding strategies, as illustrated in Figure 1. In contrast to previous work (Choi et al., 2024), which paired diverse samples together, our approach ensures that summaries in a pair are lexically similar. This lexical similarity minimises confounding stylistic or structural differ-

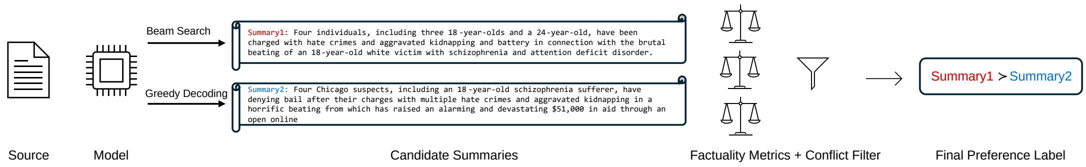


Figure 1: Our method only requires source documents to build a preference dataset.

ences, allowing the model to focus specifically on factual distinctions, which facilitates the factuality improvement on summaries.

With the generated summary pairs, we use an ensemble of factuality metrics to score them and derive preference labels from the scores. To address the unreliability of any single metric, we include only those summary pairs for which all selected metrics agree along with preference learning. This agreement-based filter removes noisy and contradictory signals, enhancing the robustness of the preference signal and making the training process more reliable and scalable.

By leveraging lexically similar summary pairs and agreement-based preference labels derived from multiple factuality metrics, our method enables more targeted factuality training than previous RLHF or model-based approaches (Stiennon et al., 2020; Choi et al., 2024). Importantly, we demonstrate that this pipeline is effective across a diverse set of language models, spanning different architectures and capabilities, including BART (Lewis et al., 2020), GPT-J (Wang and Komatsuzaki, 2021), LLaMA (Grattafiori et al., 2024), and DeepSeek (DeepSeek-AI et al., 2025). Our method consistently improves factuality scores across these various models, showing strong generalisation beyond a single model family or scale. Remarkably, our method empowers older and smaller models, such as BART, to achieve factuality performance comparable to that of significantly larger and more recent models, effectively revitalising their potential to produce accurate summaries at lower computational cost.

Our contributions are three folds:

- We introduce a novel, fully automated training pipeline for improving factuality in summarisation, which does not rely on human annotations or reference summaries.
- We introduce an agreement-based approach to generate preference labels for fine-tuning. By leveraging multiple factuality metrics and using agreement-based filtering, we ensure that only reliable signals are used in training.
- We show that lexically similar summary pairs are more effective for enhancing factuality for summarisers.

2 Related Work

2.1 Factuality Evaluation in Summarisation

Factuality has become one of the most critical properties to evaluate in recent language models. Depending on the methodologies applied, existing factuality evaluation metrics can be broadly categorised into 3 types.

Similarity-based metrics Traditional similarity-based metrics, such as ROUGE (Lin, 2004) and BLEU (Papineni et al., 2002), assess the factuality of a system summary by comparing it to a reference summary, using lexical overlap as a proxy for similarity. Subsequent work like BERTScore (Zhang et al., 2020) replaced exact word matching with embedding-based cosine similarity to enhance robustness for evaluation. More recent methods improve factual consistency evaluation by using sentence embedding similarity between the summary and the source document directly (Ye et al., 2024). These metrics are straightforward and somewhat interpretable, making them suitable for using as reward signals in RL to avoid reward hacking.

Question Answering-based metrics This line of work frames factuality evaluation as a reading comprehension task. Key phrases are extracted from the summary, and questions are generated based on their context. A question-answering model answers these questions using the source document, then checks whether the answers are consistent with the summary (Durmus et al., 2020; Scialom et al., 2021; Fabbri et al., 2022). While this approach has shown empirical effectiveness, it usually involves multiple processing stages and models, making it computationally expensive.

Natural Language Inference-based metrics These methods assess whether the content of a summary can be inferred from the source document using natural language inference (NLI) models. Early approaches that used entire documents and summaries as input to NLI models often underperformed. Recent methods have improved performance by segmenting the source document (Laban et al., 2022; Zha et al., 2023) or extracting relational structures for inference (Goyal and Durrett, 2020; Qiu et al., 2024). The final factuality score is computed by aggregating the inference results across text segments or extracted relation pairs.

2.2 RL for Fine-tuning Language Models

Reinforcement learning is often applied to fine-tune pre-trained language models, especially to improve capabilities that are difficult to formalise mathematically. Early research introduced interactive or preference learning to define reward functions in RL (Gao et al., 2018; Shapira et al., 2022). Other previous studies used evaluation metrics as direct reward signals for training (Pasunuru and Bansal, 2018; Ye and Simpson, 2023), but these approaches often suffered from distribution shift and required careful reward design to prevent catastrophic forgetting and to combine multiple, sometimes contradictory, reward components.

With the advent of LLMs, RL has been widely used with human feedback to enforce desirable properties such as safety, which are difficult to guarantee through supervised fine-tuning alone (Grattafiori et al., 2024). More recently, DeepSeek-R1 have demonstrated that RL can also facilitate emergent capabilities, such as reasoning (DeepSeek-AI et al., 2025). However, this depends on sparse rule-based rewards that may be difficult to learn from. While the human feedback can tune the model for properties that are hard to define, the annotators make an overall judgment that might ignore factual errors (Hosking et al., 2024), leading to underperformance in terms of factuality (Wang et al., 2024; Augenstein et al., 2023).

An alternative proposed by Choi et al. (2024) avoids the limitations and costs of human annotation by using rules to automatically label pairs of summaries. We suggest that this leads to noisy labels, and propose instead to use a combination of evaluation metrics that directly target factual consistency. Our experiments provide a thorough comparison of the two approaches.

3 Methods

3.1 Summary Generation

Given a source document $\mathbf{x}$, different decoding strategies can lead to various outputs $\mathbf{y}$.

Beam Search selects the top-k most likely partial sequences at each timestep t, by extending each of the k token sequences from the previous timestep, $\mathbf{y}_{<t}$, with all possible tokens. Each sequence is scored by its log probability conditioned on the source document $\mathbf{x}$. The hyperparameter k is known as the beam size. The output $\mathbf{y}_{beam}$ with length L can be expressed as:

$$\mathbf{y}_{beam} = \arg\max_{\mathbf{y} \in B} \sum_{t=1}^{L} \log P(y_t | \mathbf{y}_{<t}, \mathbf{x}) \quad (1)$$

where B is the set of top-k candidate sequences identified during decoding.

Greedy Decoding chooses the most likely token at each timestep:

$$y_t = \arg\max_{y_t} \log P(y_t | \mathbf{y}_{<t}, \mathbf{x}) \quad (2)$$

Random Sampling samples each token from the vocabulary's probability distribution at each timestep. The distributions are derived from logits using the softmax function:

$$y_t \sim \text{softmax}\left(\frac{LM(y_t | \mathbf{y}_{<t}, x)}{\tau}\right) \quad (3)$$

where $LM(\cdot)$ denotes the logit output of each timestep, and temperature τ controls the sampling distribution. A higher τ increases diversity by adding more variance to the outputs.

Recent LLMs often employ the sampling-based decoding strategies to enhance output diversity (Grattafiori et al., 2024; DeepSeek-AI et al., 2025). Prior research has shown that beam search tends to yield higher factuality scores compared to other decoding strategies, especially random sampling (Wan et al., 2023; Choi et al., 2024). In contrast, greedy decoding generally produces outputs that are lexically similar but less factually consistent than beam search outputs, as it is biased toward locally optimal token choices.

In this paper, we aim to train a model to avoid generating highly probable but factually inconsistent summaries. To do this, we can generate pairs of summaries with minimal differences from the same decoding strategy. For example, we can take the second most probable sequence produced by beam search as follows, where $\mathbf{y}_{beam}$ is the standard beam search output from Equation 1.

$$\mathbf{y}_{beam'} = \underset{\mathbf{y} \neq \mathbf{y}_{beam}, \mathbf{y} \in B}{\arg\max} \sum_{t=1}^{L} \log P(y_t | \mathbf{y}_{<t}, x) \quad (4)$$

This ensures that $\mathbf{y}_{beam}$ and $\mathbf{y}_{beam'}$ differ only slightly, enabling the evaluation metrics to focus on factuality differences, rather than stylistic or structural variations that could bias the evaluation.

3.2 Data Annotation

In this subsection, we leverage multiple factuality metrics to score summaries generated in the previous step. Prior research (Choi et al., 2024) used a heuristic to identify target summaries, rather than scoring each one, where beam search-generated summaries were always selected as the winning completions in preference learning. This introduces noise into the training data: it assumes that the higher average factuality score of beam search necessarily corresponds to more factual summaries individually, but it struggles when beam search and greedy decoding produce similar outputs, in which cases the greedy decoding could be more accurate.

To address this issue, instead of over-trusting beam search-generated summaries, we use multiple weak factuality metrics to score the summaries and derive preference labels from them. Since scores from different metrics are not directly comparable, we convert these heterogeneous scores to binary preference labels so that they can be aggregated. Then we employ a conflict resolution strategy to filter out inconsistent preference labels. The annotation process works as follows:

1. For each metric m, we obtain score $S_m(\mathbf{y}, \mathbf{x})$ for summary $\mathbf{y}$ given source $\mathbf{x}$.

2. For each pair of summaries $(\mathbf{y}_1, \mathbf{y}_2)$ related to the same source document $\mathbf{x}$, we obtain its binary preference label under the metric m, which can be written as $P_m(\mathbf{y}_1, \mathbf{y}_2, \mathbf{x}) = \text{sign}(S_m(\mathbf{y}_1, \mathbf{x}) - S_m(\mathbf{y}_2, \mathbf{x}))$

3. We apply a conflict resolver on $P_{m_i}(\mathbf{y}_1, \mathbf{y}_2, \mathbf{x})$ and only keep the data with consistent preference labels under all metrics m_i.

3.3 Training with DPO

Using the preference data obtained from the previous step, we apply Direct Preference Optimization (Rafailov et al., 2023, DPO) to train the language models towards improved factuality. Compared to RL, DPO directly optimises models without requiring a separate reward model, reducing complexity and improving training efficiency. Given summary pairs with corresponding preference labels, DPO adjusts the model parameters to increase the likelihood of generating the preferred summary. The loss function of DPO can be written as:

$$L(\theta) = \mathbb{E}_{(\mathbf{x}, \mathbf{y}_{\{w,l\}})}[\log \sigma(\beta(f_\theta(\mathbf{x}, \mathbf{y}_w) - f_\theta(\mathbf{x}, \mathbf{y}_l)))]$$

where σ is the sigmoid function, f is the log probability that the model assigns to a summary, θ represents the model parameters to optimise, β is a temperature parameter, and $\mathbf{y}_{\{w,l\}}$ denote the winning and losing summaries in the pair, respectively.

4 Experiments

4.1 Experimental Setup and Implementations

4.1.1 Dataset and Evaluation Metrics

To ensure consistency with prior work (Choi et al., 2024), we evaluate our approach on the XSUM (Narayan et al., 2018) and TL;DR (Völske et al., 2017) datasets. Both datasets require the summarisation of long articles or Reddit posts into single-sentence summaries, posing challenges for the summarisers to identify key information and assemble it correctly. Table 1 presents the characteristics of the two datasets.

Dataset	Size	Source Length	Summary Length	Compression Rate
XSUM	204045(11334)	430(433)	23(23)	5.35%(5.31%)
TL;DR	116722(6553)	313(314)	31(31)	9.90%(9.87%)

Table 1: Characteristics of XSUM and TL;DR datasets. Numbers in parentheses refer to the test split while other numbers are for the train split. Length refers to the total number of words in the text. Compression Ratio is computed between source length and summary length.

We train the models using the dataset built upon the train split and evaluate the trained language models on the test split. For automatic factuality evaluation, we utilise AlignScore (Zha et al., 2023), a state-of-the-art metric, which also aligns our settings with the evaluation setup in previous works (Choi et al., 2024). To assess the overall quality of summaries, we compute the ROUGE-L score (Lin, 2004) that reflects the overlap with the reference summary. In addition, we employ ChatGPT to compare our approach against the baselines as LLMs have shown promising results in directly evaluating generative tasks (Gekhman et al., 2023; Luo et al., 2023). We further analyse shifts in common types of factual consistency error types to understand the impact of our training pipeline, again using ChatGPT to categorise mistakes.

4.1.2 Language Model Selection

Model	Size	Architecture	Pre-release Fine-tuning	Main Ability	Fine-tuning Scale
BART-large	406M	Encoder-Decoder	SFT	Summarisation	Full
GPT-J	6B	Decoder	SFT	Open-ended Generation	Adapter
LLaMA-3.2	3B	Decoder	SFT+RL	Instruction	Adapter
DeepSeek-R1 (Distill-Qwen)	7B	Decoder	SFT+RL	Reasoning	Adapter

Table 2: Specifications of the selected language models.

To demonstrate the robustness of our method, we select a variety of language models with different scales and capabilities. Model specifications are listed in Table 2. We select BART-large (Lewis et al., 2020) to represent encoder-decoder models that were widely employed before the advent of LLMs. We select GPT-J-6B (Wang and Komatsuzaki, 2021), LLaMA-3.2-3B (Grattafiori et al., 2024), and DeepSeek-R1-Distill-Qwen-7B (DeepSeek-AI et al., 2025) as they are representative LLMs trained for different purposes. Due to their large sizes, we apply LoRA (Hu et al., 2021) and only train an adapter during fine-tuning.

GPT-J is an alternative for GPT-3 (Brown et al., 2020) and was only tuned with SFT. It can perform specific tasks given a prompt but it is suggested to apply task-oriented SFT beforehand.

LLaMA-3.2 utilised RL during its training process, specifically through RLHF, to enhance its alignment with human preferences and improve the quality of its responses.

DeepSeek-R1 is a mixture-of-experts model with 671B parameters, providing impressive reasoning ability on a wide range of tasks including math and coding. In this paper, we use its distilled model based on Qwen2.5 (Team, 2024) to balance the training efficiency and reasoning quality.

For GPT-J, SFT is required before RL, so we only use a simple prompt as it will learn to summarise during SFT. For LLaMA and DeepSeek, we avoid fine-tuning them on specific tasks before applying RL, simulating real-world conditions where they are provided only with task instructions. To maintain consistency across experiments, we use the same generic summarisation prompt for all LLMs. Details of the prompt are available in Appendix B, along with the processing steps for DeepSeek's chain-of-thought output.

4.1.3 Decoding Strategies

As highlighted in prior studies (Holtzman et al., 2019; Choi et al., 2024), decoding strategies can impact factuality. In this section, we explore how decoding strategies influence factual accuracy and select which to use in the consequent experiments.

Dataset	Model	AlignScore(↑)				
		BS#1	BS#2	RS#1	RS#2	Greedy
XSUM	BART	**61.9**	61.5	19.2	18.4	58.9
	GPT-J	**59.7**	58.3	17.4	17.3	50.5
	LLaMA	**86.1**	85.3	67.3	66.5	83.6
	DeepSeek	**82.5**	82.4	60.2	59.6	80.5
TL;DR	BART	**84.9**	84.7	42.5	41.0	80.6
	GPT-J	**89.6**	89.0	60.3	60.2	83.6
	LLaMA	**91.4**	90.6	83.7	83.6	90.7
	DeepSeek	**89.1**	88.9	75.6	75.8	87.9

Table 3: AlignScore of different decoding strategies.

From Table 3, we observe that the first candidate from beam search (BS#1) consistently outperforms other decoding strategies, including greedy decoding and random sampling. The latter strategies introduce excessive randomness or focus too narrowly on local token probabilities, leading to lower factuality. Therefore, in our experiments, we primarily use beam search and greedy decoding, as these strategies provide relatively high factual accuracy while the mix of strategies allows us to generate different summaries for the same source. For final evaluation, we use the first beam search output to ensure the highest factuality.

4.2 Factuality Scoring Metrics

Among the metrics mentioned in 2.1, we utilise SBERTScore (Ye et al., 2024) and SummaC-Conv (Laban et al., 2022), representing similarity-based and NLI-based metrics respectively. These metrics, while slightly less powerful than state-of-the-art alternatives, are more computationally efficient. We exclude QA-based metrics not only due to their high computational cost, but also because they require a question generation model trained on the same dataset, which is not available for Reddit posts in TL;DR.

4.3 Baselines

We compare our proposed approach with three baselines: supervised fine-tuning (SFT), reinforce-

Model	Strategy	AlignScore	Δ	ROUGE-L
BART	SFT	61.9	\	36.4
	MPO$_{(\text{BS\#1,BS\#2})}$	62.0	+0.1	33.5
	MPO$_{(\text{BS\#1,Greedy})}$	36.3	-25.6	21.4
	Ours$_{(\text{BS\#1,BS\#2})}$	**86.6**	**+24.7**	33.9
	Ours$_{(\text{BS\#1,Greedy})}$	86.1	+24.2	30.5
GPT-J	SFT	59.7	\	25.0
	MPO$_{(\text{BS\#1,BS\#2})}$	53.5	-6.2	23.6
	MPO$_{(\text{BS\#1,Greedy})}$	44.2	-15.5	22.9
	Ours$_{(\text{BS\#1,BS\#2})}$	70.9	+11.2	22.8
	Ours$_{(\text{BS\#1,Greedy})}$	**75.8**	**+16.1**	22.3
LLaMA	SFT	86.1	\	19.2
	MPO$_{(\text{BS\#1,BS\#2})}$	78.9	-7.2	18.2
	MPO$_{(\text{BS\#1,Greedy})}$	79.8	-6.3	18.8
	Ours$_{(\text{BS\#1,BS\#2})}$	**88.7**	**+2.6**	18.3
	Ours$_{(\text{BS\#1,Greedy})}$	87.1	+1.0	18.7
DeepSeek	SFT	82.5	\	14.8
	MPO$_{(\text{BS\#1,BS\#2})}$	80.8	-1.7	15.4
	MPO$_{(\text{BS\#1,Greedy})}$	81.3	-1.2	12.5
	Ours$_{(\text{BS\#1,BS\#2})}$	83.0	+0.5	13.7
	Ours$_{(\text{BS\#1,Greedy})}$	**83.2**	**+0.7**	14.0

Table 4: Comparison of our approach against SFT and MPO on XSUM dataset. Δ refers to the performance difference over SFT results. The best results for each model are highlighted in **bold**.

Model	Strategy	AlignScore	Δ	ROUGE-L
BART	SFT	84.9	\	25.8
	RLHF	73.1	-11.8	22.6
	MPO$_{(\text{BS\#1,BS\#2})}$	88.1	+3.2	24.2
	MPO$_{(\text{BS\#1,Greedy})}$	71.1	-2	20.4
	Ours$_{(\text{BS\#1,BS\#2})}$	94.1	+9.2	23.0
	Ours$_{(\text{BS\#1,Greedy})}$	**94.2**	**+9.3**	22.4
GPT-J	SFT	89.6	\	26.8
	RLHF	81.5	-8.1	23.4
	MPO$_{(\text{BS\#1,BS\#2})}$	92.3	+2.7	23.7
	MPO$_{(\text{BS\#1,Greedy})}$	84.7	-4.9	22.0
	Ours$_{(\text{BS\#1,BS\#2})}$	93.7	+4.1	19.7
	Ours$_{(\text{BS\#1,Greedy})}$	**93.8**	**+4.2**	22.3
LLaMA	SFT	91.4	\	15.6
	RLHF	90.2	-1.2	18.3
	MPO$_{(\text{BS\#1,BS\#2})}$	86.4	-5	15.4
	MPO$_{(\text{BS\#1,Greedy})}$	82.2	-9.2	14.7
	Ours$_{(\text{BS\#1,BS\#2})}$	**93.5**	**+2.1**	15.1
	Ours$_{(\text{BS\#1,Greedy})}$	92.9	+1.5	15.3
DeepSeek	SFT	89.1	\	15.8
	MPO$_{(\text{BS\#1,BS\#2})}$	88.4	-0.7	14.9
	MPO$_{(\text{BS\#1,Greedy})}$	89.7	+0.6	15.1
	Ours$_{(\text{BS\#1,BS\#2})}$	**90.9**	**+1.8**	15.1
	Ours$_{(\text{BS\#1,Greedy})}$	89.9	+0.8	16.5

Table 5: Comparison of our approach against SFT, RLHF and MPO on TL;DR dataset. Δ refers to the performance difference over SFT results. The best results for each model are highlighted in **bold**.

ment learning from human feedback (RLHF), and model-based preference optimisation (Choi et al., 2024, MPO). Both SFT and RLHF are common fine-tuning methods that rely on either golden references or human annotations. SFT trains on reference summaries, while RLHF builds on the SFT checkpoint using human preference rankings to optimise via RL rather than direct supervision.

We reuse the official RLHF checkpoint of GPT-J[1]. For the other models, we perform training using the pipelines from the TRL[2] library, applied to the trl-lib/tldr-preference dataset[3], which includes preference labels based on overall human judgments that are not specifically focused on factuality.

MPO (Choi et al., 2024) avoids the need to score summaries by assuming that beam search-generated summaries are more factually consistent than those generated by other decoding strategies. However, while beam-search generates more factual summaries on average, individual summaries are not guaranteed to be the most factually consistent, leading to some mislabelled pairs. This resulted in huge performance degradation for MPO when applied to similar summary pairs in the original study. Our proposed method overcomes this by using multiple computationally efficient metrics to annotate generated summaries, allowing greater resilience to input similarity and better utilization of summaries from various decoding strategies.

4.4 Experimental Results

Tables 4 and 5 present a comparison of our approach with the baselines. We do not report RLHF results for XSUM due to the lack of a human preference dataset, nor do we include DeepSeek RLHF results for TL;DR, as we cannot learn a reward model for it on a preference dataset without chain-of-thought examples.

Our approach consistently outperforms all three baselines, bringing positive effects to all models across both datasets, and the largest improvements across all models. RLHF and MPO sometimes decreased AlignScore, specifically for LLaMA on both datasets. We observe the degradation on MPO when applied to similar summary pairs, as mentioned in the original MPO study (Choi et al., 2024), so we compare our approach against the best MPO setup with dissimilar pairs in Appendix A; our training pipeline still outperforms it.

In terms of the overall quality, we found a slight trade-off between the factuality score and ROUGE-L. ROUGE is computed between the generated summary and the reference summary, which is directly used for SFT. Note that a previous study (Maynez et al., 2020) has indicated that some human written reference summaries are hallucinated.

[1] https://huggingface.co/CarperAI/openai_summarize_tldr_ppo

[2] https://huggingface.co/docs/trl/main/en/ppo_trainer

[3] https://huggingface.co/datasets/trl-lib/tldr-preference

Considering the large factuality improvement obtained from our approach, we think this trade-off is within the acceptable range.

The results show that our approach is more effective at improving summary factuality compared to RLHF on human-labelled datasets or MPO's heuristic preference label generation, while not losing the overall quality comparing to the reference summaries used by SFT. This highlights the benefit of scoring summaries based on factuality metrics rather than relying on heuristic preferences.

Across the four models, BART gained the largest improvement with a score increase of 24.7 on XSUM and 9.3 on TL;DR. It is worth noting that our training pipeline sealed the gap between BART and the LLMs and led to better post-training performance, making it possible to apply BART where computing resources are limited. The DeepSeek reasoning model received the least improvement, coming second last and last on XSUM and TL;DR respectively. We speculate that this is because our preference labels are only decided by the final summary, so errors made in the thinking process generated before it could be overlooked by the scoring metrics, resulting in a noisy training signal.

4.5 Overall Quality Evaluation

Dataset	Model	Baseline		
		SFT	RLHF	MPO
XSUM	BART	51.4	\	52.0
	GPT-J	44.2	\	80.0
	LLaMA	42.0	\	54.0
	DeepSeek	39.0	\	52.4
TL;DR	BART	47.2	40.4	54.8
	GPT-J	46.8	42.8	61.6
	LLaMA	43.4	39.2	74.6
	DeepSeek	40.8	\	58.6

Table 6: The win rates of our approach against SFT, RLHF, and MPO across 4 models and 2 datasets in terms of overall quality of summaries.

To gain a better understanding of the overall quality of the generated summaries, we use ChatGPT-4o-mini to evaluate them based on not just factuality, but also informativeness, coherence, and legibility. We randomly selected 500 source documents from each dataset, applied different models to generate summaries and asked ChatGPT to compare them in pairs. The full evaluation prompt can be found in Appendix B. We compared the summaries from our approach against those from the baselines (SFT, RLHF, and MPO). Some win rates against RLHF are not available due to the availability of the human preference dataset.

Dataset	Model	Pipeline Decoding Strategy	Pair Similarity	Scoring Metric				SFT Results
				SBERT	SummaC	SBERT +SummaC	SBERT +SummaC +Filter	
XSUM	BART	(BS#1,BS#2)	0.940	71.4	79.7	78.5	**86.6**	61.9
		(BS#1,Greedy)	0.826	75.0	81.7	79.9	**86.1**	
	GPT-J	(BS#1,BS#2)	0.973	60.0	54.1	**71.7**	70.9	59.7
		(BS#1,Greedy)	0.773	68.2	73.9	70.0	**75.8**	
	LLaMA	(BS#1,BS#2)	0.938	85.0	86.5	87.5	**88.7**	86.1
		(BS#1,Greedy)	0.889	85.5	84.3	86.3	**87.1**	
	DeepSeek	(BS#1,BS#2)	0.985	81.1	82.6	82.8	**83.0**	82.5
		(BS#1,Greedy)	0.843	80.7	82.2	83.1	**83.2**	
TL;DR	BART	(BS#1,BS#2)	0.954	94.0	91.3	**94.7**	94.1	84.9
		(BS#1,Greedy)	0.802	93.1	91.3	**94.4**	94.2	
	GPT-J	(BS#1,BS#2)	0.943	92.9	95.3	**95.6**	93.7	89.6
		(BS#1,Greedy)	0.751	91.9	91.6	**94.2**	93.8	
	LLaMA	(BS#1,BS#2)	0.909	92.1	90.8	91.8	**93.5**	91.4
		(BS#1,Greedy)	0.868	89.9	91.0	91.5	**92.9**	
	DeepSeek	(BS#1,BS#2)	0.972	88.7	85.6	89.2	**90.9**	89.1
		(BS#1,Greedy)	0.735	89.5	88.8	89.3	**89.9**	

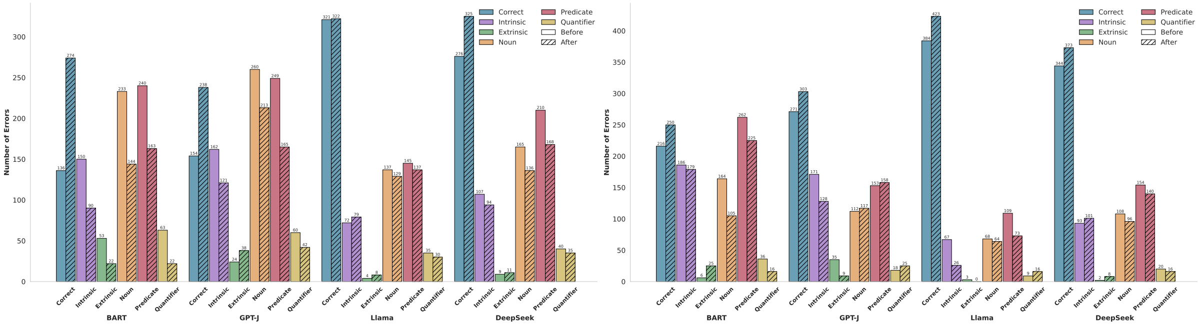
Table 7: AlignScore of language models fine-tuned by different training settings using our approach on the two datasets. The best results are highlighted in **bold**.

Table 6 shows that our summaries were preferred over MPO but less preferred than SFT summaries. This is likely because SFT directly trains on human-written reference summaries, while ours focus on factuality, leading to potentially less fluency or informativeness. RLHF summaries are also more preferred because they are originally trained to align with human values, thus being more likely to be selected by ChatGPT, which has also been trained with the same purpose. However, previous discussion has confirmed the competitive overall quality of our summaries. Therefore, we asked ChatGPT to output the selection reasons and found out that the preferred summaries contained excessive details, while our summaries are more abstract and discarded some of the unnecessary details to reduce the risk of generating inconsistent content (Appendix C). This suggests a trade-off between factual consistency and summary style, which aligns with previous findings (Hosking et al., 2024) that overall judgements may neglect factuality.

5 Analysis

5.1 Ablation Study

We studied the effectiveness of each component in our approach and present their influence in Table 7. Introducing a single factuality metric to score the summary did not always lead to improvements. For example, when only one metric was applied, LLaMA and DeepSeek occasionally showed decreased factuality scores. However, when multiple factuality metrics were applied, all models showed improvement. Additionally, filtering out inconsistent labels further enhanced performance, likely



(a) Error frequency on XSUM.

(b) Error frequency on TL;DR.

Figure 2: Error frequencies before and after training.

because contradicting labels may appear in different batches, thereby adding noise during training.

5.2 Similarity of Summary Pairs

We also examined the impact of similarity between paired summaries, as shown in Table 7. Summary pairs generated by selecting alternative outputs, i.e., (BS#1,BS#2), achieved higher similarities than pairs generated by varying the decoding strategy. Highly similar summary pairs may help the model focus on subtle factual consistency differences. However, the (BS#1,Greedy) strategy is competitive with (BS#1,BS#2) overall, suggesting that an average similarity ~ 0.75 may be sufficient.

Pipeline Decoding Strategy	Pair Similarity	AlignScore
SFT baseline	-	61.9
(BS#1, BS#2)	94.0	86.6
(BS#1, Greedy)	82.6	86.1
(BS#1, Random)	34.9	72.0

Table 8: The effect of using temperature-based random sampling decoding strategy to generate less similar candidate summaries to train BART on XSUM.

Taking BART as an example, we then further investigated the effect of less similar summary pairs generated by beam search and temperature-based random sampling, as shown in Table 8. Less similar summary pairs went through the same preference label generation process. Fine-tuning with these labels still improved factuality but to a lesser degree than the similar pairs (BS#1,BS#2) and (BS#1,Greedy). We show the evaluation accuracy curve during training in Appendix D, which stayed level during training, implying that the model benefitted little from training on these data. Summary pairs generated by beam search and random sampling, which have a greater factuality gap (as shown in Table 3), were too straightforward for BART to learn from, resulting in minimal improvements.

Therefore, we can conclude that both our similar summary pair generation process contributes to the final improvement of our approach.

5.3 Inconsistency Type Analysis

Finally, we employ ChatGPT to assess factual inconsistencies in the summaries and analyse how the frequency of factual errors changes before and after training with our approach.

Similar to previous studies (Tang et al., 2023), we defined five inconsistency types, namely *Intrinsic*, *Extrinsic*, *Noun*, *Predicate*, *Quantifier*. Along with *Correct* summaries, we asked ChatGPT to identify them according to a given definition and count the frequency of each. The definition and prompt can be found in Appendix B.

Figure 2 shows that the error frequencies of *Noun* (Orange bars), *Predicate* (pink bars), and *Quantifier* (yellow bars) mostly decreased. Consequently, our approach achieved many more *Correct* summaries (blue bars) than SFT checkpoints, demonstrating the effectiveness of our approach across different models.

6 Conclusion

We introduce a novel automatic training pipeline for improving the factual consistency of summarisers. Our approach can be generalised over different model architectures and scales. It requires only source documents, utilising multiple factuality evaluation metrics to score the summary and obtain labels for preference optimisation. The experimental results suggest that our approach outperforms supervised and RLHF baselines and boosts the factuality performance of smaller models to a comparable levels to LLMs, revealing the effectiveness of preference learning over similar summary pairs.

Limitations

We only applied SBERTScore and SummaC to score the generated summaries in this paper. There are various other metrics available but we were not able to test them all. While were were able to demonstrate that it is possible to improve factuality using our chosen imperfect metrics, this could raise concerns about the generalisation ability of our approach to other automated scoring methods. On the other hand, we rely on AlignScore to evaluate our output. Although AlignScore is considered state-of-the-art for factuality evaluation for now, it is not perfect, so will still miss some factual errors in the summary.

In overall quality evaluation, we found that our approach generated summaries that were less preferred by ChatGPT when comparing to SFT/RLHF summaries. This reveals the challenge of how to fine-tune the summariser towards better factuality without trading off other qualities. It also highlights the difficulty of judging the overall quality of summaries, where a human or LLM judge my put more weight on certain qualities (e.g., readability, brevity) at the expense of others (e.g., factual consistency). The trade-off between these qualities may need to be judged within the context of a specific application: how important it is that a summary is factually consistent versus stylistically compelling will depend on its use case.

References

Isabelle Augenstein, Timothy Baldwin, Meeyoung Cha, Tanmoy Chakraborty, Giovanni Luca Ciampaglia, David Corney, Renee DiResta, Emilio Ferrara, Scott Hale, Alon Halevy, Eduard Hovy, Heng Ji, Filippo Menczer, Ruben Miguez, Preslav Nakov, Dietram Scheufele, Shivam Sharma, and Giovanni Zagni. 2023. Factuality challenges in the era of large language models. *Preprint*, arXiv:2310.05189.

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

Jaepill Choi, Kyubyung Chae, Jiwoo Song, Yohan Jo, and Taesup Kim. 2024. Model-based preference optimization in abstractive summarization without human feedback. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18837–18851, Miami, Florida, USA. Association for Computational Linguistics.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

Alexander Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. QAFactEval: Improved QA-based factual consistency evaluation for summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2587–2601, Seattle, United States. Association for Computational Linguistics.

Yang Gao, Christian M. Meyer, and Iryna Gurevych. 2018. APRIL: Interactively learning to summarise by combining active preference learning and reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4120–4130, Brussels, Belgium. Association for Computational Linguistics.

Zorik Gekhman, Jonathan Herzig, Roee Aharoni, Chen Elkind, and Idan Szpektor. 2023. TrueTeacher: Learning factual consistency evaluation with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2053–2070, Singapore. Association for Computational Linguistics.

Tanya Goyal and Greg Durrett. 2020. Evaluating factuality in generation with dependency-level entailment. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *CoRR*, abs/1904.09751.

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

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

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

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Zheheng Luo, Qianqian Xie, and Sophia Ananiadou. 2023. Chatgpt as a factual inconsistency evaluator for text summarization. *Preprint*, arXiv:2303.15621.

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.

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, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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.

Haoyi Qiu, Kung-Hsiang Huang, Jingnong Qu, and Nanyun Peng. 2024. AMRFact: Enhancing summarization factuality evaluation with AMR-driven negative samples generation. 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 594–608, Mexico City, Mexico. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741.

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.

Thomas Scialom, Paul-Alexis Dray, Patrick Gallinari, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, and Alex Wang. 2021. Questeval: Summarization asks for fact-based evaluation. *arXiv preprint arXiv:2103.12693*.

Ori Shapira, Ramakanth Pasunuru, Mohit Bansal, Ido Dagan, and Yael Amsterdamer. 2022. Interactive query-assisted summarization via deep reinforcement learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2551–2568, Seattle, United States. Association for Computational Linguistics.

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. *Advances in neural information processing systems*, 33:3008–3021.

Liyan Tang, Tanya Goyal, Alex Fabbri, Philippe Laban, Jiacheng Xu, Semih Yavuz, Wojciech Kryscinski, Justin Rousseau, and Greg Durrett. 2023. Understanding factual errors in summarization: Errors, summarizers, datasets, error detectors. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11626–11644, Toronto, Canada. Association for Computational Linguistics.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. TL;DR: Mining Reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 59–63, Copenhagen, Denmark. Association for Computational Linguistics.

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, 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, Dubrovnik, Croatia. 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`.

Wenxuan Wang, Juluan Shi, Zhaopeng Tu, Youliang Yuan, Jen tse Huang, Wenxiang Jiao, and Michael R. Lyu. 2024. The earth is flat? unveiling factual errors in large language models. *Preprint*, arXiv:2401.00761.

Yuxuan Ye and Edwin Simpson. 2023. Towards abstractive timeline summarisation using preference-based reinforcement learning. In *ECAI 2023*, pages 2882–2889. IOS Press.

Yuxuan Ye, Edwin Simpson, and Raul Santos Rodriguez. 2024. Using similarity to evaluate factual consistency in summaries. *arXiv preprint arXiv:2409.15090*.

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.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Results for MPO on Dissimilar Pairs

Figure 9 demonstrates the results of our approach and MPO under the best setup individually. Our methods significantly outperforms MPO.

Dataset	Model	MPO	Ours
XSUM	BART	68.85	86.6
	GPT-J	65.26	75.8
	LLaMA	67.31	88.7
TL;DR	GPT-J	91.61	93.8
	LLaMA	85.33	93.5

Table 9: AlignScore comparison against the best results for MPO, cited from Choi et al. (2024).

B Prompt for LLMs

B.1 Prompt for Summarisation Generation

We only prepare a simple prompt for GPT-J as it needs SFT before applying RL, as shown in Figure 3. *{doc}* denotes the source document which will be changed according to the data being processed. It will learn to summarise the source document into a single sentence during SFT, therefore it only needs a template to ensure the model receives the source document and generate summaries as completion.

Document: {doc}

Summary:

Figure 3: Prompt for GPT-J.

Figure 4 presents the prompts we use to generate summaries using LLaMA on the two datasets. SFT is not involved before we apply it to generate summaries, therefore we provide a more detailed instruction to specify our requirements.

For DeepSeek, we provide our requirements as for LLaMA. Specifically, it requires a special token *<think>* to trigger the thinking process, as shown in Figure 5. Following the prompt, it generates a chain-of-thought that ends with *<\think>* before generating the final output. Therefore, we truncate its output at *<\think>* and take all the following output as the final summary for the metrics to score.

You are a useful AI assistant that helps people to summarize news documents.
Summarize the given document into a single sentence:

Document: {doc}

Summary:

(a) Prompt for XSUM.

You are a useful AI assistant that helps people to summarize news documents.
Summarize the given document into a single sentence:

Document: {doc}

Summary:

(b) Prompt for TL;DR.

Figure 4: Prompt for LLaMA to generate summaries on the two datasets.

You are a useful AI assistant that helps people to summarize news articles. Think first and then summarize the given article into a single sentence.

Document: {doc}

<think>

(a) Prompt for XSUM.

You are a useful AI assistant that helps people to summarize news articles. Think first and then summarize the given article into a single sentence.

Document: {doc}

<think>

(b) Prompt for TL;DR.

Figure 5: Prompt for DeepSeek to generate summaries on the two datasets.

B.2 Prompt for ChatGPT Evaluation

We use a similar prompt in the previous work (Choi et al., 2024) for ChatGPT to compare two summaries, as described in Figure 6. *{source}*, *{summary1}*, *{summary2}* denote the source document and two candidate summaries. We found that ChatGPT-4o-mini tends to claim that both summaries are not good enough due to informativeness, therefore we relaxed the requirement and ask it to choose the most faithful summary if both are not good as we focus on factuality on this paper.

As for inconsistency type analysis, we give the definition in the prompt first and then ask ChatGPT to judge the summary. The prompt is shown in Figure 7. *{source}* and *{summary}* represent the source document and the summary to analyse.

Which of the following summaries does a better job of summarizing the most important points in the given news article, without including unimportant or irrelevant details? A good summary is both precise and concise but not overly specific. If both summaries are not good, choose the one that are most faithful to the original post.
Article: {source}
Summary A: {summary1}
Summary B: {summary2}
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 6: Prompt for ChatGPT win rate evaluation.

Here is the definition of common factual inconsistency types.
Intrinsic Errors: The summary contains misinformation that is present in the original text.
Extrinsic Errors: The summary contains information that is not present in the original text.
Noun Errors: The summary misrepresents details from the source, such as dates, numbers, names, or events.
Predicate Errors: The summary misrepresents the relationships between entities or events in the source.
Quantifier Errors: The summary misrepresents the quantity entities or events in the source.
Can the given summary be supported by the given article? Only consider the errors above.
Article: {source}
Summary: {summary}
FIRST, identify whether the summary is correct. If the summary is correct, please say \"No errors\". THEN, identify the errors in the summary, reply only with the error types \"Intrinsic\", \"Extrinsic\", \"Noun\", \"Predicate\", \"Quantifier\". Your response should use the format:
Error types: <a list of error types>

Figure 7: Prompt for ChatGPT inconsistency type analysis.

C ChatGPT Win Rate Reason Analysis

We print out the common words appeared in the reasons for choosing SFT and RLHF summaries over ours in Figure 8. The main reason for the SFT and RLHF summaries being preferred is that they carry more details, while ours reduced the hallucination risk by generating less of the details.

Figure 8: Prompt for ChatGPT inconsistency type analysis.

D Evaluation Accuracy Curve during Training

Figure 9 shows how well the model learns to distinguish the chosen summary and the rejected summary in the pair. Ideally, the model learns to simulate the chosen summary while differs its behaviour

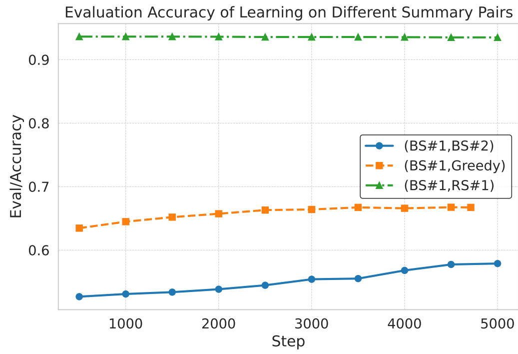


Figure 9: Evaluation accuracies over pairwise labels during DPO training for BART on XSUM.

from the rejected summary so that it gains better accuracies during training.