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

Yuxuan Ye and **Raul Santos-Rodriguez** and **Edwin Simpson**
Intelligent System Laboratory
University of Bristol
United Kingdom
{yuxuan.ye, enrsr, edwin.simpson}@bristol.ac.uk

Abstract

Reinforcement learning with evaluation metrics as rewards is widely used to enhance specific capabilities of language models. However, for tasks such as factually consistent summarisation, existing metrics remain underdeveloped, limiting their effectiveness as signals for shaping model behaviour. While individual factuality metrics are unreliable, their combination can more effectively capture diverse factual errors. We leverage this insight to introduce an automated training pipeline that improves factual consistency in summaries by aggregating scores from different weak metrics. Our approach avoids the need for complex reward shaping by mapping scores to preferences and filtering out cases with high disagreement between metrics. For each source document, we generate lexically similar summary pairs by varying decoding strategies, enabling the model to learn from factual differences caused by subtle lexical differences. This approach constructs a high-quality preference dataset using only source documents. Experiments demonstrate 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[1].

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 (Huang et al., 2025).

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 existing automatic evaluation metrics (Lin, 2004; Kryściński et al., 2019; Zhang et al., 2020; Laban et al., 2022) have been adopted 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). Furthermore, using a single metric directly as a scalar reward for RL can lead to unstable training, as explored by Roit et al. (2023). The training process is influenced by the metric's reliability and the distribution of reward scores, but the distributions of existing evaluation metrics are not well-studied. Although combining metrics can broaden error detection coverage (Ye et al., 2024), applying metric combination in RL often requires manual weighting of sub-rewards (Gao et al., 2018; Pasunuru and Bansal, 2018; Wan and Bansal, 2022; Ye and Simpson, 2023), thus the use of RL is impeded by the complexity of reward shaping.

An alternative is Reinforcement Learning from Human Feedback (Ouyang et al., 2022, RLHF), which uses human-annotated preferences and has proven effective at aligning large language models (LLMs) with broad human values. However, recent work (Hosking et al., 2024; Xue et al., 2024) shows that existing RLHF datasets often overlook factuality, despite instructions to annotators to account for it. Annotator judgments integrate various considerations, such as trade-offs among properties, individual biases, and occasional misunderstandings, so the resulting overall preferences can fail to reliably capture factuality. Therefore, while constructing a factuality-focused preference dataset is crucial, doing so requires substantial resources and expertise, making scalability a major concern.

To overcome these barriers, this paper proposes a fully automated training pipeline that improves factual consistency in summarisation without rely-

[1]Code is available at https://github.com/Haruhi07/MultiMetric

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 17342–17355
November 4-9, 2025

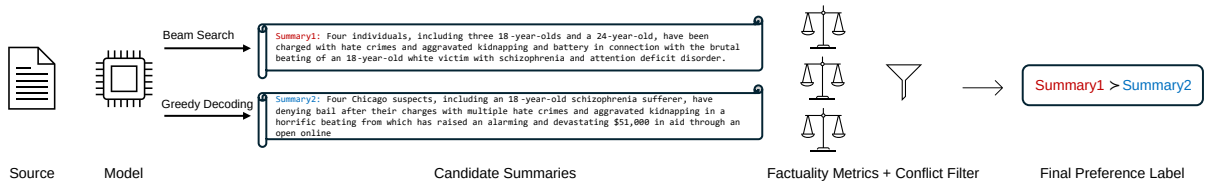


Figure 1: Our method only requires source documents to build a preference dataset.

ing on human annotations or reference summaries. We adopt the sampling method from Choi et al. (2024), 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 their work, which paired diverse samples together, our approach ensures that summaries in a pair are lexically similar. This lexical similarity minimises confounding stylistic or structural differences, allowing the model to focus specifically on factual distinctions, which facilitates the factuality improvement during training.

With the generated summary pairs, we use an ensemble of factuality metrics to score them and derive preference labels from the scores. To mitigate 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 annotations, 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, in contrast with previous work (Roit et al., 2023) that requires an LLM-sized reward model to stabilise the training, we never encounter catastrophic forgetting in our experiments, demonstrating the stability and effectiveness of our proposed method on various language models. Our method consistently demonstrates factuality improvements on different model architectures, including BART (Lewis et al., 2020), GPT-J (Wang and Komatsuzaki, 2021), LLaMA-3 (AI@Meta, 2024), and DeepSeek-R1 (DeepSeek-AI, 2025), 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 threefold:

- We present a novel, fully automated preference learning pipeline for optimising summarisation factuality, which not only improves LLMs' factuality scores but also elevates the factuality performance of smaller models to the same level.
- We show a promising way to adapt multiple existing factuality metrics into training targets. By leveraging preference learning and filtering cases with high disagreement, we improve the reliability of the training data, leading to more robust training in practice.
- We analyse the contribution of lexical similarity between summary pairs and conclude that, with sufficiently accurate preference annotations, similar 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 Classical similarity-based metrics such as ROUGE (Lin, 2004) and BLEU (Papineni et al., 2002) reported the n-gram overlapping ratio between the system and the reference summaries. Subsequently, BERTScore (Zhang et al., 2020) replaced the exact word matching with embedding-based cosine similarity to enhance robustness to lexical and syntactic variation. This idea was then extended to sentence embedding similarity by Ye et al. (2024), who found that using

the source document as reference could shrink the performance gap between similarity-based metrics and other methodologies.

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.

Entailment-based metrics These methods assess whether the source document entails the summary using natural language inference (NLI) models. Early approaches that simply concatenated the entire document and the summary as input to an NLI model often underperformed. Recent methods 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 for 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 (AI@Meta, 2024). More recently, DeepSeek-R1 have demonstrated that RL can also facilitate emergent capabilities, such as reasoning (DeepSeek-AI, 2025). However, this depends on sparse rule-based rewards that may be difficult to learn from. While RLHF (Ouyang et al., 2022) 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 on factuality (Wang et al., 2024; Augenstein et al., 2023).

To avoid the limitations and costs of human annotation, Choi et al. (2024) proposed to label summary pairs using simple rules. It over-simplifies the problem and can introduce noise into the labels. Therefore, we propose instead to use a combination of existing evaluation metrics that directly target factual consistency. Our experiments suggested solid gains in factuality compared to their approach.

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 (AI@Meta, 2024; DeepSeek-AI, 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 towards 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_i, we obtain score $S_{m_i}(\mathbf{y}, \mathbf{x})$ 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 $P_{m_i}(\mathbf{y}_1, \mathbf{y}_2, \mathbf{x}) = \text{sign}(S_{m_i}(\mathbf{y}_1, \mathbf{x}) - S_{m_i}(\mathbf{y}_2, \mathbf{x}))$

3. The conflict filter checks $\{P_{m_i}(\mathbf{y}_1, \mathbf{y}_2, \mathbf{x})\}$ and only keeps 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 XSUM (Narayan et al., 2018) and TL;DR (Völske et al., 2017). Both datasets require the summarisation of long news 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. Numbers in parentheses refer to the test split while other numbers are for the training split. Length refers to the total number of

words in the text. Compression Ratio is computed between source length and summary length.

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.

For evaluation, we apply the same automatic metrics as in the previous work (Choi et al., 2024) to ensure a fair comparison. AlignScore (Zha et al., 2023) and FactCC (Kryściński et al., 2019) reflect factuality, while ROUGE-L (Lin, 2004) and BARTScore (Yuan et al., 2021) check the coherence. The definitions of these metrics are in Appendix A. 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 to understand the impact of our training pipeline, again using ChatGPT to categorise mistakes.

4.1.2 Language Model Selection

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 (AI@Meta, 2024), and DeepSeek-R1-Distill-Qwen-7B (DeepSeek-AI, 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.

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.

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 maths 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 study how decoding strategies influence factual consistency on our selected datasets 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 (RS#1 and RS#2). 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. Their definitions are in Appendix C. 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 TL;DR.

4.3 Baselines

We compare our proposed approach with three baselines: supervised fine-tuning (SFT), reinforcement 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 is often applied after SFT, using human preference rankings to optimise via RL rather than direct supervision.

We reuse the official SFT checkpoints for BART on XSUM[2] and RLHF checkpoints for GPT-J[3] on TL;DR. For RLHF on the other models, we perform training using the TRL[4] library, with the trl-lib/tldr-preference dataset[5], 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 their study. Our proposed method overcomes this by using multiple computationally efficient metrics to annotate generated summaries, allowing greater resilience to input similarity and better utilisation of summaries from various decoding strategies.

4.4 Experimental Results

We built two datasets for each model, using (BS#1,BS#2) and (BS#1,Greedy) respectively. Table 4 presents the better factuality performance between the two settings for MPO and our approach individually. 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.

Dataset	Model	Method	AlignScore(Δ)	BARTScore	FactCC	ROUGE-L
XSUM	BART	SFT	61.9(\)	-3.69	65.7	**36.4**
		MPO	62.0(+0.1)	-3.67	76.0	33.5
		Ours	**86.6(+24.7)**	**-3.63**	**83.3**	33.9
	GPT-J	SFT	59.7(\)	-3.63	62.4	**25.0**
		MPO	53.5(-6.2)	-3.90	73.8	23.6
		Ours	**75.8(+16.1)**	**-3.57**	**80.6**	22.3
	LLaMA	SFT	86.1(\)	-3.48	75.1	**19.2**
		MPO	79.8(-6.3)	-3.49	76.0	18.8
		Ours	**88.7(+2.6)**	**-3.47**	**77.4**	18.3
	DeepSeek	SFT	82.5(\)	-3.34	75.9	**14.8**
		MPO	81.3(-1.2)	-3.39	78.8	12.5
		Ours	**83.2(+0.7)**	**-3.23**	**80.5**	14.0
TL;DR	BART	SFT	84.9(\)	-3.48	43.2	**25.8**
		RLHF	73.1(-11.8)	-3.95	41.5	22.6
		MPO	88.1(+3.2)	-3.40	56.9	24.2
		Ours	**94.2(+9.3)**	**-3.33**	**65.6**	22.4
	GPT-J	SFT	89.6(\)	-3.69	30.9	**26.8**
		RLHF	81.5(-8.1)	-3.59	34.1	23.4
		MPO	92.3(+2.7)	-3.53	37.5	23.7
		Ours	**93.8(+4.2)**	**-3.44**	**46.0**	22.3
	LLaMA	SFT	91.4(\)	-3.81	73.7	15.6
		RLHF	90.2(-1.2)	-3.78	64.1	**18.3**
		MPO	86.4(-5.0)	-3.78	81.3	15.4
		Ours	**93.5(+2.1)**	**-3.74**	**84.1**	15.1
	DeepSeek	SFT	89.1(\)	-3.79	66.7	**15.8**
		MPO	89.7(+0.6)	-3.77	72.5	15.1
		Ours	**90.9(+1.8)**	**-3.69**	**75.8**	15.1

Table 4: Evaluation results on the two datasets. Δ refers to the performance difference over SFT results. The best results are highlighted in **bold**.

Our approach consistently outperforms all three baselines on AlignScore, FactCC and BARTScore, bringing positive effects to all models across both datasets, and the largest improvements across all models. RLHF and MPO sometimes decreased the factuality, specifically for LLaMA on both datasets.

For ROUGE-L, we found the same trade-off between it and the factuality performance as in Choi et al. (2024). ROUGE is computed between the

[2] https://huggingface.co/facebook/bart-large-xsum

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

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

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

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

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 5: The win rates against baselines, judged by ChatGPT for overall summary quality.

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 maintaining overall quality compared 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 an AlignScore increase of 24.7 on XSUM and 9.3 on TL;DR. Although LLMs had less headroom for the factuality improvement, our method still managed to increase their scores marginally. 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. 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 would be overlooked by the scoring metrics, resulting in a noisy training signal.

4.5 Overall Quality Evaluation

To gain a better understanding on 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. Some win rates against RLHF are not available due to the availability of the human preference dataset.

Table 5 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 reasons for its selections, 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 D). 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 6. 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 because contradicting labels may appear in different batches, thereby adding noise during training.

5.2 Similarity of Summary Pairs

Taking the training outcome of BART on XSUM as the example, we 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, as also shown in Table 6. Highly similar summary pairs help the model focus on subtle fac-

Dataset	Model	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 6: AlignScore of language models fine-tuned by different training settings using our approach on the two datasets. The best results are highlighted in **bold**.

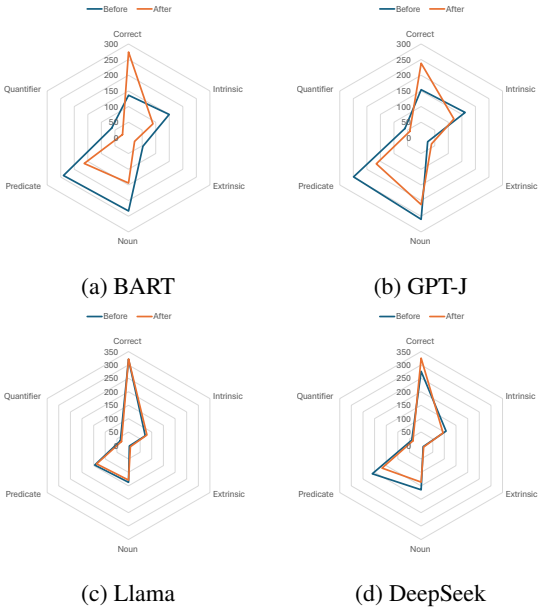
tual consistency differences, but we speculate that there could exist a threshold. The (BS#1,Greedy) strategy is competitive with (BS#1,BS#2) overall in Table 6, suggesting that an average similarity ~ 0.7 may be sufficient.

Taking BART as an example, in Table 7, we further investigated the effect of less similar summary pairs (BS#1,RS#1), i.e., the best setting for MPO, to which we applied the same preference label generation process. Using our method to fine-tune with these labels still improved factuality but to a lesser degree than the similar pairs (BS#1,BS#2) and (BS#1,Greedy). Although MPO was able to obtain the largest improvement on (BS#1,RS#1) pairs, our method still outperformed it, validating the effectiveness of our method on reducing the noise in the dataset. Furthermore, we observe the same degeneration mentioned in Choi et al. (2024) on the similar pairs (BS#1,BS#2) and (BS#1,Greedy). We show the evaluation accuracy curve during training in Appendix E, 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 and our labelling step using automated metrics contribute to the final improvement of our approach.

Decoding Strategy	Pair Similarity	Method	AlignScore
-	-	SFT	61.9
(BS#1, BS#2)	94.0	MPO	62.0
		Ours	86.6
(BS#1, Greedy)	82.6	MPO	36.3
		Ours	86.1
(BS#1, RS#1)	34.9	MPO	66.4*
		Ours	72.0

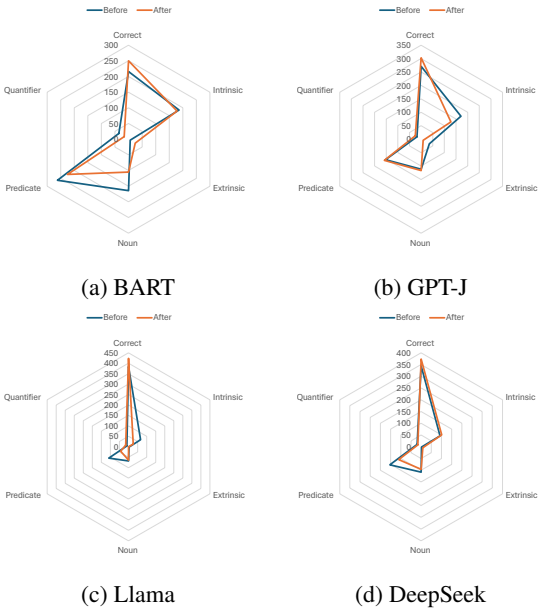
Table 7: The effect of using different decoding strategies to generate summary pairs for training BART on XSUM. * indicates the result cited from Choi et al. (2024).



(a) BART (b) GPT-J

(c) Llama (d) DeepSeek

Figure 2: Error frequencies on XSUM.



(a) BART (b) GPT-J

(c) Llama (d) DeepSeek

Figure 3: Error frequencies on TL;DR.

5.3 Disagreement Analysis

We looked into the rates of each model triggering the disagreement filter on the two datasets. In practice, 1000 summaries pairs were generated to obtain the preference labels. Table 8 below shows that at least 60% of the data was retained for training after the filtering process.

Model	XSUM	TL;DR
BART	28.6	29.8
GPT-J	31.8	27.3
LLaMA	37.3	31.4
DeepSeek	34.7	28.9

Table 8: The percentage of data that triggers disagreement filter in our experiments.

5.4 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.

Figures 2 and 3 show that the error frequencies of *Noun*, *Predicate*, and *Quantifier* types mostly decreased. Consequently, our approach achieved many more *Correct* summaries 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 baselines and boosts the factuality performance of smaller models to a level comparable 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 we 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.

We also did not include results for RL with the scalar metric scores used directly as the reward signal. We explored this method in our early investigation and found that it requires both the model and the metric backbone to be large and extremely computationally costly, otherwise catastrophic forgetting became a very common problem during training. Although we did not have the comparison against this line of work, we believe that our method provides a more stable training paradigm in practice, as we never encountered the catastrophic forgetting problem in our experiments.

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 may 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.

Acknowledgments

The authors acknowledge the use of resources provided by the Isambard-AI National AI Research Resource (AIRR). Isambard-AI is operated by the University of Bristol and is funded by the UK Government's Department for Science, Innovation and Technology (DSIT) via UK Research and Innovation; and the Science and Technology Facilities Council [ST/AIRR/I-A-I/1023]. The financial support for Yuxuan Ye is provided by the programme of the China Scholarship Council (No. 202108060154). Raul Santos-Rodriguez is supported by the UKRI Turing AI Fellowship EP/V024817/1.

References

AI@Meta. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

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. 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*.

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.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Evaluating the factual consistency of abstractive text summarization. *arXiv preprint arXiv:1910.12840*.

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.

Wanqi Xue, Bo An, Shuicheng Yan, and Zhongwen Xu. 2024. Reinforcement learning from diverse human preferences. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, IJCAI '24.

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*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in neural information processing systems*, 34:27263–27277.

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 Evaluation Metrics

For a given source document D, we denote its generated summary as S. The length of the source document is denoted $|D|$ and the summary length is denoted $|S|$.

AlignScore breaks both source document and summary into certain length chunks. We denote the sets of chunks as $\{D_j\}$ and $\{S_i\}$. It then aggregates the entailment scores obtained on these text chunk pairs. The backbone NLI classifier is trained for predicting the alignment degree in sentence pairs.

$$\text{AlignScore}(S, D) = \frac{1}{|\{S_i\}|} \sum_{s \in \{S_i\}} \max_{d \in \{D_j\}} \text{Alignment}(s, d)$$

FactCC provides a coarse-grained insight into whether the generated summary is entailed by the source document. The backbone is an NLI model based on BERT, so the equation can be written as

$$\text{FactCC}(S, D) = \text{NLI}(S; D)$$

BARTScore is essentially the weighted log-likelihood computed using BART. It can be written as

$$\text{BARTScore}(S, D) = \sum_{t=1}^{|S|} \omega_t \log p(S_t | S_{<t}, D, \Theta)$$

where S_t is each token in S and Θ is the trained weights of BART.

ROUGE computes the n-gram recall between the candidate and the target text. In this paper, we use ROUGE-L, which match the longest common subsequence (LCS) that appear in both summaries. The equation can be expressed as

$$\text{ROUGE-L}(S, D) = \frac{\text{LCS Length}}{\text{Reference Summary Length}}$$

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 4. *{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 the completion.

```
Document: {doc}

Summary:
```

Figure 4: Prompt for GPT-J.

Figure 5 and 6 present the prompts we used to generate summaries using LLaMA and DeepSeek on the two datasets. The only difference in the prompt is that we indicate that the source documents are reddit posts in TL;DR and news documents in XSUM.

```
You are a useful AI assistant that helps people to summarize [reddit posts/news
documents]. Summarize the given post into a single sentence:

Document: {doc}

Summary:
```

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

DeepSeek requires a special token <*think*> to trigger the thinking process, as shown in Figure 6. Following the prompt, it generates a chain-of-thought that ends with <*/think*> before generating the final output. Therefore, we take all the output

after *</think>* as the final summary for the metrics to score.

```
You are a useful AI assistant that helps people to summarize[reddit posts/news
documents]. Think first and then summarize the given post into a single sentence.

Document: {doc}

<think>
```

Figure 6: Prompt for DeepSeek to generate summaries on the two datasets.

B.2 Prompt for ChatGPT Evaluation

We use a similar prompt to the previous work (Choi et al., 2024) for ChatGPT to compare two summaries, as described in Figure 7. *{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.

```
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 7: Prompt for ChatGPT win rate evaluation.

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 8. *{source}* and *{summary}* represent the source document and the summary to analyse.

```
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 8: Prompt for ChatGPT inconsistency type analysis.

C Scoring Metric Definitions

For a given source document D, we denote its generated summary as S. Their sentence collections are marked as $\{D_j\}$ and $\{S_i\}$ respectively. SBERTScore is defined as below.

$$\text{SBERTScore}(S, D) = \frac{1}{|\{S_i\}|} \sum_{s \in \{S_i\}} \max_{d \in \{D_j\}} \cos\text{sim}(s, d)$$

SummaC firstly computes NLI scores on sentence pairs to get a score matrix A, such that

$$A_{ij} = \text{NLI}(s, d) \quad s \in \{S_i\}, d \in \{D_j\}$$

It then maps A into a score frequency matrix $H = \text{bin}(A)$, where it bins the NLI scores into h evenly spaced bins for each summary sentence. Then a convolutional layer is trained to aggregate H to obtain the final score.

$$\text{SummaC}(S, D) = \text{Conv}(H)$$

D ChatGPT Win Rate Reason Analysis

We print out the common words that appeared in the reasons given by ChatGPT for choosing SFT and RLHF summaries over ours in Figure 9. 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 fewer details.

Figure 9: Word cloud showing frequency of terms in the reasons generated by ChatGPT for preferring SFT and RLHF summaries over those produced by our approach.

E Evaluation Accuracy Curve during Training

Figure 10 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 from the rejected summary so that it gains better accuracies during training.

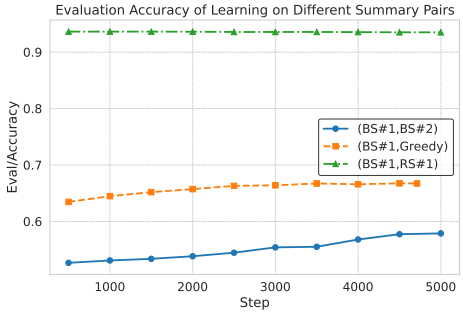


Figure 10: Evaluation accuracies over pairwise labels during DPO training for BART on XSUM.