Factually Consistent Summarization via Reinforcement Learning with Textual Entailment Feedback

Paul Roit[β γ *] Johan Ferret[γ *] Lior Shani[γ *]

Roee Aharoni[γ] Geoffrey Cideron[γ] Robert Dadashi[γ] Matthieu Geist[γ]
Sertan Girgin[γ] Léonard Hussenot[γ] Orgad Keller[γ] Nikola Momchev[γ]
Sabela Ramos[γ] Piotr Stanczyk[γ] Nino Vieillard[γ]

Olivier Bachem[γ] Gal Elidan[γ] Avinatan Hassidim[γ] Olivier Pietquin[γ] Idan Szpektor[γ]

[β]Bar-Ilan University [γ]Google Research
{plroit,jferret,liorshani}@google.com

Abstract

Despite the seeming success of contemporary grounded text generation systems, they often tend to generate factually inconsistent text with respect to their input. This phenomenon is emphasized in tasks like summarization, in which the generated summaries should be corroborated by their source article. In this work we leverage recent progress on textual entailment models to directly address this problem for abstractive summarization systems. We use reinforcement learning with reference-free, textual-entailment rewards to optimize for factual consistency and explore the ensuing trade-offs, as improved consistency may come at the cost of less informative or more extractive summaries. Our results, according to both automatic metrics and human evaluation, show that our method considerably improves the faithfulness, salience and conciseness of the generated summaries.

1 Introduction

Recent advancements in abstractive summarization systems (Zhang et al., 2019; Liu et al., 2022b) are often impeded by their tendency to output information that is either contradicting or unsupported by their input article, often termed as "hallucinations" or factual inconsistency (Falke et al., 2019; Maynez et al., 2020; Pagnoni et al., 2021). While these systems produce highly relevant and coherent text, this lack of factual consistency often limits their wide-spread adoption in real-world applications. An example is depicted in Figure 1, where the highlighted statement in the summary, while plausible, has no support in the input article.

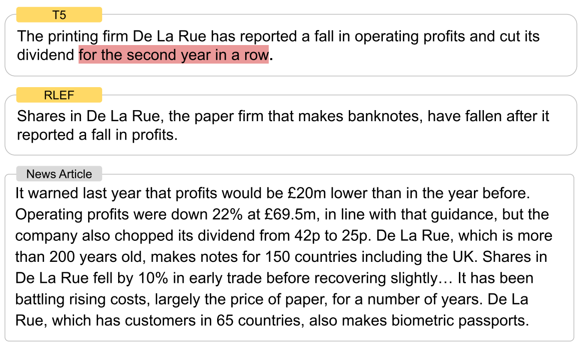

Figure 1: Summaries produced by multiple methods from a news article in the XSum dataset. Hallucinations or contradictions are highlighted in red. Note how the T5 generated summary mentions that there is a fall in operating profits *for the second year in a row*, while the article only discusses a recent decline in earnings and a warning made in the previous year.

Since widely-used metrics such as ROUGE (Lin, 2004) were shown to be inefficient for detecting hallucinations, many recent research efforts introduced novel automatic metrics for measuring factual consistency (Kryscinski et al., 2020; Goyal and Durrett, 2020; Scialom et al., 2021, inter alia). We propose to leverage these automatic metrics within a *reinforcement learning* (RL) framework at training time. Specifically, we apply *textual entailment* assessment (a.k.a. *natural language inference*, or NLI; Dagan et al., 2005; Bowman et al., 2015) between the source article and the generated summary as a reward.

Our reward is based on the well studied textual entailment task (Pavlick and Kwiatkowski, 2019; McCoy et al., 2019; MacCartney and Manning, 2007, inter alia), for which there are many publicly available datasets (Nie et al., 2020; Liu et al.,

*Equal contribution

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 6252–6272
July 9-14, 2023 ©2023 Association for Computational Linguistics

2022a). While these NLI datasets are not specific to summarization, it was shown that classifiers trained on these datasets perform well in detecting factual inconsistencies in summarization and other generative tasks (Honovich et al., 2022). Because faithful summaries must be textually entailed from the corresponding input documents, using such a reward explicitly should guide a summarization model towards generating more factually consistent summaries. Yet, a high-quality summary should also be coherent and contain relevant information (Fabbri et al., 2021), aspects which may not be captured by entailment alone. Moreover, a reward that is based only on entailment raises the risk of degenerate solutions, leading to either highly extractive (Ladhak et al., 2022) or less informative summaries ("reward hacking"; Amodei et al., 2016; Skalse et al., 2022; Pan et al., 2022).

To address these issues, we propose Reinforcement Learning with Entailment Feedback (RLEF): Start with a model trained to produce summaries with the conventional cross-entropy objective, and further fine-tune it using RL with an entailment-based reward. Throughout the RL procedure, we constrain the candidate models to stay close to the initial model. This way, while the model is being corrected for higher consistency, it also retains other summarization capabilities that were learnt with the maximum-likelihood (MLE) objective. In this work we explore the consistent vs. informative trade-off in our RL-based summaries w.r.t. various aspects including model scale, regularization and decoding strategies. We find those aspects to be highly important and interdependent for the final model performance, highlighting the importance of carefully tuning them.

Our work stands in contrast to two prior RL-based approaches. The first approach induces a reward function from human feedback that encompasses various task-specific requirements into a single value (Böhm et al., 2019; Stiennon et al., 2020). Collecting such feedback is expensive and requires dedicated data collection for each target task. In contrast, we use readily-available models and datasets for the reward, which address a specific aspect of generation that is generic across many different tasks. Other works modeled the reward using different similarity functions between the *reference* and the generated summaries (Pasunuru and Bansal, 2018; Gunasekara et al., 2021), thus requiring reliable reference data. Instead, our

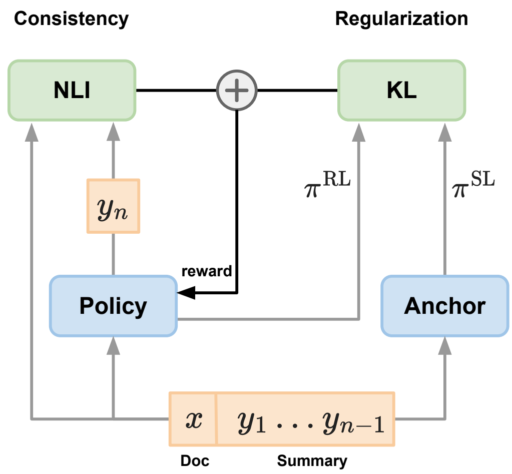

Figure 2: RLEF training loop: (Left) given an input document, the policy generates a summary to be scored by the NLI model for consistency; (Right) given a document and the current generated summary, the KL distance between the RL and anchor model policies is used for regularization; Both scores are combined for training the policy. The black lines represents reward feedback to the model.

reward function evaluates the generated output only w.r.t. the *input*, enabling to train using RL on data without reference summaries. We evaluated our approach on the widely used XSum (Narayan et al., 2018a) dataset, using both automated metrics and human raters. The results show considerable improvements over strong baselines for factual consistency, salience, and conciseness of the generated summaries.

2 Method

We would like to increase factual consistency using an entailment-based reward, while retaining the high salience and coherence that current summarization models already obtain. To achieve this, we propose to initialize an RL policy with a summarization model trained on supervised data (the *anchor model*). From there, in each RL-based training step we update the parameters according to two signals: an entailment reward and a regularization term grounded on the anchor model. During RL training, the entailment reward directs the model towards increased faithfulness, while the regularization term keeps the model from drifting to degenerate solutions and "forgetting" how to summarize. The process is illustrated in Figure 2.

2.1 RLEF: RL from Entailment Feedback

Problem Formulation. We denote the input document and output summary as x, y respectively. Let $\mathbb{V}$ denote the input and output vocabulary, and $y_{:n} = (y_1, ..., y_n)$ denote the generated summary up to the n-th token. We define the token-wise generative summarization process as a deterministic Contextual Markov Decision Process (CMDP, Hallak et al. 2015) with observable context, where the *context* is the input text x, the *state* at the n-th token generation is the sequence generated thus far $y_{:n-1}$, and the action space is defined over the vocabulary $\mathbb{V}$. A policy $\pi(\cdot \mid y_{:n-1}, x)$, is a probability distribution over all tokens in $\mathbb{V}$, conditioned on the context and state. We note that following this formulation, the policy is identical to a token-level auto-regressive language model (Bengio et al., 2003). The RL objective is to find the optimal policy, which maximizes the cumulative reward signal.

Rewards. We use an NLI classification model as a factual consistency reward signal. Since the model is trained to evaluate complete utterances and expects as input a *grammatical* premise (document) / hypothesis (summary) pair, we use sequence-level rewards and define the token-level NLI reward to be zero on every token except for the end-of-sequence (EOS) token. For the EOS token we set the reward to be the log-probability for an "entailment" decision according to the NLI classifier, using x as the premise and $y_{:n}$ as the fully generated hypothesis:

$$r^{\text{NLI}}(y_n; y_{:n-1}, x) = \begin{cases} \text{NLI}(y_{:n}; x) & y_n = \texttt{[EOS]}; \\ 0 & \text{otherwise,} \end{cases}$$

where [EOS] is an end-of-sequence symbol, and $\text{NLI}(y_{:n}; x) = \log \Pr(\text{entailment} \mid y_{:n}, x)$.

To retain the summarization capabilities of the anchor model, we use Kullback-Leibler (KL) regularization to keep the RL-based policy close to the supervised anchor policy (Jaques et al., 2017):

$$r^{\text{KL}}(y_n; y_{:n-1}, x) = \log \frac{\pi^{\text{SL}}(y_n \mid y_{:n-1}, x)}{\pi^{\text{RL}}_{\theta}(y_n \mid y_{:n-1}, x)} \,.$$

This term is added to the NLI reward, producing the final token-level reward:

$$\begin{aligned} r(y_n; y_{:n-1}, x) = (1-\alpha) r^{\text{NLI}}(y_n; y_{:n-1}, x) \\ + \alpha r^{\text{KL}}(y_n; y_{:n-1}, x) \,. \quad (1) \end{aligned}$$

The hyperparameter α enables controlling the trade-off between enforcing faithfulness through the reward and remaining close to the anchor policy.

Training Algorithm. We train the policy to optimize for the rewards defined in Equation (1) using an on-policy actor-critic policy gradient (PG) approach. Since we keep proximity to the anchor model via the KL penalty reward, the algorithm can be considered a regularized PG algorithm, similarly to works by Geist et al. (2019); Shani et al. (2020); Abdolmaleki et al. (2018); Tomar et al. (2022); Vaswani et al. (2021); see Appendix C for a detailed formulation. Specifically, two models are learned: a policy (the generation model) and the expected value of the policy (the value network). We use the supervised model to initialize the parameters of both models, with the exception that the last layer of the value network outputs single scalars instead of a distribution over the vocabulary.

The RL training process consists of the following stages: (1) Generating summaries with the current policy and (2) Scoring the summaries using the reward signal. Then, (3) Policy and value networks are trained, jointly: the policy is trained via the PG loss while using the value for generalized advantage estimation (GAE, Schulman et al. (2016)); the value is trained via standard bootstrapping, using the GAE predictions. Notably, this process does not require reference summaries for learning the policy. More details regarding the algorithm and losses can be found in Appendix A.

2.2 Decoding at Inference Time

As a direct consequence of RL training, the model explicitly learns to generate tokens with the goal of maximizing the long-term sequence reward. This is in contrast to MLE-based training, where the model learns to generate each token myopically, requiring heuristic decoding strategies such as beam-search to plan ahead. As a result, we can use the more efficient temperature sampling instead of beam-search when decoding from an RL-trained policy.[1]

3 Experimental Design

3.1 Data

We focus on XSum (Narayan et al., 2018a), an abstractive summarization dataset that poses challenges around factual consistency. XSum is compiled from 200K web-scraped BBC news articles, where the lead (introductory) sentence in every article is taken as the summary, and the rest of the

[1] We found that temperature sampling is sufficient for RL, while beam-search is required to improve the supervised policy.

sentences are taken as the source document.

Due to this formulation, XSum summaries may contain additional information that was not repeated in the rest of the sentences. Indeed, prior work found that only 20% of the reference summaries in XSum are entailed from their source document (Maynez et al., 2020), and that summarization systems trained on XSum are likely to generate factually inconsistent summaries. For this reason we find XSum suitable for our experiments, as we would like to see if the RL-based reward could alleviate the factual inconsistencies that supervised models learn to generate based on this data.

We also experiment on two additional datasets to compare to prior work. The TL;DR dataset (Völske et al., 2017), using the same cleaned version provided by Stiennon et al. (2020), which contains 120K Reddit posts and their short summaries, and the CNN/DM (Nallapati et al., 2016) dataset. The latter contains 200K news articles and their bullet-point highlights, which are mostly copied excerpts from article sentences. In this work we focus on abstractive summarization, and therefore evaluate our methods on CNN/DM with models trained, both supervised and reinforced, over TL;DR.

3.2 Entailment Model

In this work we focus on combining an existing entailment model as a reward in an RL framework. We employ the NLI classifier from Honovich et al. (2022) across our study as a reward as well as for evaluation and data labelling for baseline methods. It was trained over the ANLI dataset (Nie et al., 2020) with the T5-XXL architecture. The classifier produces the characters '1' or '0' as its output for *binary* entailment and non-entailment decisions, respectively. We pose the source document as the premise and the predicted summary as the hypothesis, and use the log-probability of the decoded character '1' conditioned on the input as our reward.[2] We leave improvements to the underlying factual consistency models for future efforts. See Section 6 for more discussion about different factual consistency models.

3.3 Baseline Methods

SL. Our supervised learning baseline is obtained by fine-tuning a T5 model on document-summary pairs. We use the T5X framework (Roberts et al., 2022) for fine-tuning with batch size of 32 and keep the other hyperparameters to their default values (see Appendix A for details). Fine-tuning is stopped once the model converges in terms of ROUGE on the validation set. This supervised baseline will also be used as the initialization checkpoint of our RL methods. Decoding a summary using this model is implemented using beam search.

Filtered. Similar to the SL approach, with the distinction that we filter out training data where the summaries are not entailed by the input document according to our NLI model. This filtering leaves 60% of the original XSum training set. We train the model similarly to the SL model, and evaluate on the full validation and test splits, without filtering.

CTRL. Inspired by Filippova (2020); Rashkin et al. (2021b), we train the model on the full training set to explicitly differentiate between generating faithful and unfaithful summaries: each training document is prepended with a phrase indicating if the target summary is entailed or not according to our NLI model. At inference, since we aim to produce consistent summaries, each document is always prepended with the phrase denoting an entailing summary, and continue decoding the summary using beam search. Other parameters are similar to the SL method.

FactPegasus. Wan and Bansal (2022) employ a tailored pre-training setup similar to PEGASUS (Zhang et al., 2019) that also takes factual consistency into account, and combine it with data pre-processing, and contrastive learning to generate more faithful summaries.

CLIFF. Cao and Wang (2021) propose a contrastive learning objective that distinguishes between reference and heuristically created noisy summaries.

RLHF. Stiennon et al. (2020) uses an RL approach with a reward model that learns from human comparisons of summaries. They iteratively add new feedback from humans for summaries generated by the current policy, and re-train the reward model. We use their publicly released samples of the TL;DR validation set and the CNN/DM test set.

3.4 Proposed Models

We train two flavors of RL-based models. The first, RLEF_L, gives a lower weight to the regularization

[2] We use an empirically validated classification threshold of 0.5 for entailment decisions.

reward by setting $\alpha = 0.1$ and the sampling temperature to 1. The second model, RLEF_H, gives a higher weight to the regularization reward with $\alpha = 0.2$ and a sampling temperature of 0.3. We altered both the α values and the sampling temperatures since we saw that both parameters affect the trade-off between factual consistency, as measured by the NLI model, and lexical similarity, as measured by ROUGE (see Figure 3). For additional implementation details see Appendix A.

3.5 Automatic Evaluation Metrics

We report the common lexical n-gram overlap evaluation metrics and a set of factual consistency metrics, as the former were shown to be ill-suited for detecting unfaithful outputs (Falke et al., 2019; Pagnoni et al., 2021).

For factual consistency, we report NLI, which is the percent of entailed summaries according to our NLI classifier, and the Q^2 score (Honovich et al., 2021). Q^2 is similar to QAGS (Wang et al., 2020) and QuestEval (Scialom et al., 2021) but was shown to work better on XSum data (Honovich et al., 2022) with higher correlation with human judgements.

When optimizing for faithfulness, an RL policy may resort to less abstractive summaries that are copied verbatim from the source (Ladhak et al., 2022), or less informative ones with a reduced level of detail. To explicitly measure these attributes in a summary, we report *extractiveness* metrics: COVERAGE and DENSITY (Grusky et al., 2018), where the first measures the percent of summary tokens that also appear in the document, while the second measures a quantity similar to the average length of extractive spans in the summary. Finally, we report the average summary length[3] (LENGTH).

3.6 Manual Evaluation Protocol

We asked human evaluators to rate a sample of the XSum test-set from several selected methods. Each summary was evaluated by 3 different raters. Inspired by Fabbri et al. (2021), we pose 4 questions outlining comprehensibility, attribution, salience and conciseness (see example in Figure 5 in the appendix). To get conclusive results, similarly to Rashkin et al. (2021a) we request binary yes/no answers and ask to answer "No" for any slight deviation from the desired property. For unfaithful summaries, the evaluator also provides the offending phrase. Our evaluator pool consists of 11 workers that successfully completed a short training round of 10 examples (for details, see Appendix B).

[3] We use `SequenceMatcher::get_matching_blocks` from the python standard library to compute the set of extractive spans. Texts are tokenized with NLTK (Loper and Bird, 2002).

		Faithfulness		ROUGE			Extractiveness		
Size	Method	NLI	Q^2	1	2	L	Coverage	Density	Length
XXL	SL	63.93	41.08	**45.32**	22.77	37.56	68.93	0.79	**21.69**
	Filtered	74.54	43.01	43.84	21.36	36.24	69.21	0.81	20.74
	CTRL	71.64	43.26	45.19	22.70	37.57	69.83	0.82	20.94
	RLEF_L	**94.66**	**54.84**	41.77	19.95	34.75	75.03	0.98	17.72
	RLEF_H	83.17	48.40	44.8	22.37	37.29	72.08	0.91	20.14
Base	SL	52.44	36.16	39.84	17.77	32.63	71.77	0.87	20.52
	RLEF_L	**79.90**	**46.70**	38.13	16.47	31.33	76.06	1.06	17.72
	CLIFF	68.16	45.71	**45.17**	**23.32**	**37.61**	73.37	1.21	20.86
	FactPegasus	62.01	42.69	37.16	15.13	30.36	**78.33**	**1.42**	18.47

Table 1: Automatic evaluation results, XSum test set. RLEF with various regularization patterns vs. baseline methods. Highest values are in bold. Due to stability issues in T5-Base RL-training (see Section 5), $T = 0.3$ was used.

		Faithfulness		ROUGE			Extractiveness		
Test set	Method	NLI	Q^2	1	2	L	Coverage	Density	Length
TL;DR	SL	94.11	74.34	**36.75**	**14.87**	**29.13**	91.40	3.86	27.69
	RLEF_L	**99.39**	**77.55**	36.58	14.81	29.12	92.89	4.14	26.57
	RLHF-6B	94.56	74.19	33.68	11.86	25.49	89.22	3.56	**37.12**
CNN/DM (transfer)	SL	92.53	69.52	31.72	11.85	27.42	94.67	**5.5**	30.14
	RLEF_L	**95.00**	**71.08**	31.28	11.79	27.20	**95.24**	5.32	28.16
	RLHF-6B	91.48	70.42	**32.51**	**11.93**	**27.85**	93.10	4.85	**32.73**

Table 2: Automatic evaluation results for TL;DR and CNN/DM test sets. Highest values are in bold. RLEF models are based on T5-XXL. For CNN/DM (transfer) we employ the RLEF and SL models trained on TL;DR and predict summaries on the CNN/DM test-set, similarly to the *transfer* setting in Stiennon et al. (2020). For RLHF, we use the publicly available predictions of their human feedback model in the *transfer* setting.

4 Results

Automatic Evaluation. Table 1 presents the automatic evaluation results on the XSum test set, comparing the supervised baselines to the two RL-based models (RLEF_L, RLEF_H).

The table shows that the RL-based models achieve the highest entailment scores as measured by the NLI and Q^2 metrics. Notably, the RL approach is the most effective approach to utilize the NLI signal, scoring favorable compared to supervised baselines Filtered and CTRL, which leverage the same signal.

Analyzing ROUGE reveals the trade-off between the entailment and other summarization traits. Without strong regularization, RLEF_L scores highest on entailment but lower on ROUGE, indicating that in order to reach higher factual consistency, the model pushed farther away from the supervised starting point. The more strongly reg-

Size	Method	Comprehension	Attribution	Salience	Conciseness
XXL	SL	**99.0** ± 1.1	27.3 ± 5.0	61.6 ± 5.5	35.0 ± 5.4
	Filtered	96.3 ± 2.1	31.3 ± 5.2	61.3 ± 5.5	34.3 ± 5.3
	RLEF_L	98.7 ± 1.3	**56.6** ± 5.6	**78.0** ± 4.7	**61.0** ± 5.5
	RLEF_H	98.0 ± 1.5	39.0 ± 5.5	70.6 ± 5.1	45.3 ± 5.6
Base	RLEF_H	96.0 ± 2.2	**38.3** ± 5.5	**64.3** ± 5.4	**44.3** ± 5.6
	CLIFF	**99.3** ± 0.9	28.3 ± 5.1	58.3 ± 5.6	33.3 ± 5.3
XSum	reference	99.3 ± 0.9	23.6 ± 4.8	62.6 ± 5.4	30.3 ± 5.2

Table 3: Human evaluation results over 100 test set samples, each summary rated by 3 workers, results are micro-averaged. Each value corresponds to a percentage of positive answers per category with 95% confidence intervals around the sample proportion. Highest values for each model size are in bold.

ularized RLEF_H achieves a ROUGE score on par with the CTRL and SL baselines, suggesting that our KL-regularization prevented the policy from drifting.

Looking at extractiveness, the Density metric suggests that RL policies do not resort to copying text, and the increased Coverage implies that they tend to use more terms from the document, suggesting fewer hallucinations. Lower ROUGE scores *may* hint at lower quality summaries for the less regularized entailment model, yet the other metrics actually point at higher conciseness. We next present our human evaluation to shed light on these differences, and analyze whether the improvement in entailment is also captured by human readers, and whether the lexical divergence from the reference summary affects has implications on salience or conciseness.

Human Evaluation. The results of our human evaluation are detailed in Table 3. Our raters fully agreed on 60% of the examples regarding attribution. From attribution (factual consistency) perspective, the results strengthen the evidence that the RL approach is superior to other methods by a large gap. Interestingly the XSum reference summaries scored lowest with 23.6%, showing that they are ill-suited to serve as faithful references for ROUGE and similar reference-based metrics. Notably, the human attribution evaluation was much stricter than the NLI metric, with much lower scores for all models, and we analyze this discrepancy in Section 5.

Surprisingly, the RLEF models outperforms all other models also on Salience and Conciseness. Specifically, the less regularized RLEF_L learned to generate not only the most factually consistent summaries but also to improve on Salience and Conciseness, indicating that they are correlated w.r.t human quality perception.

Comparison with RLHF. We applied our RL approach on the TL;DR dataset. We used the same input format and data split as in Stiennon et al. (2020) for both the supervised and RL training processes. For the supervised model (SL) we used hyper-parameters identical to our previous experiments (see Appendix A) except for a batch size of 128 and learning rate of 2e-4.

We compared our results using automated metrics with the RLHF approach (Stiennon et al., 2020). This approach is also based on the T5 model and uses a similar RL setup, yet it employs a reward model trained on *task-specific human preferences* and applying a KL-based anchor. The results, detailed in Table 2, show that RLEF achieves higher entailment scores in both NLI and Q^2 metrics, while our supervised model is on par with RLHF. We also note that RLHF produces noticeably different and longer summaries compared to our supervised baseline, while RLEF maintains similar length and ROUGE to the supervised baseline.

We also compared the two approaches in a transfer learning setting, where we predicted a summary on a different dataset (CNN/DM) using models trained on TL;DR. The results show similar trends, with higher entailment score for RLEF. These results hint at the benefit of utilizing a general NLI reward function, which managed to outperform the domain-specific RLHF reward both on the source domain and on a transfer setting.

5 Analysis

Regularization and Sampling Temperature. Figure 3 describes an ablation experiment where we vary the regularization α and the decoding temperature and measure the effect on different automatic metrics. Higher sampling temperature correlates with higher entailment and lower ROUGE scores. We conjecture that this is since higher temperature generates more diverse summaries, which amplifies exploration away from the original gold references. A similar phenomenon is observed when considering token length, as lower temperature policies produce summaries closer in length to the data-mean than their higher temperature counterparts.

As for the regularization coefficient α, we observe the expected trade-off: lower regularization (smaller α) leads to higher entailment (NLI), lower similarity to the supervised summary (ROUGE),

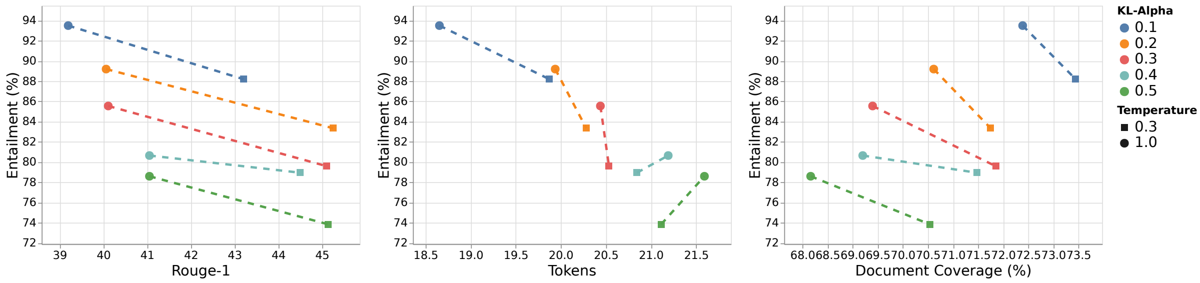

Figure 3: Trade-offs between entailment and Rouge-1, Summary Token Length and Document Coverage as measured over the XSum validation set. Setups differ in KL-regularization (color) and sampling temperature (dot shape), model architecture is fixed to T5-XXL.

and higher Coverage. These may be explained by removal of external hallucinations that often use vocabulary terms that are unrelated to the document.

Surprisingly, in each KL setting, the lower temperature policy favors more document-aligned terms (perhaps for their higher initial probability), yet this is not reflected in the NLI metric, that stays lower than its higher-temperature counterpart. We also observe that the summaries get shorter with less regularization, as the policy learns to mention fewer details as a way to alleviate generating inconsistencies.

Model Size. We tested our approach with different model sizes to study the effect of scale in the RLEF_H setup. We compared T5-Base (220M parameters), T5-Large (770M) and T5-XXL (11B), using the same hyper-parameters for all three models. Figure 4 shows the entailment rate on the XSum validation set during RL-finetuning. For all model sizes, our approach improved the entailment ratio over the supervised model by a large margin.

However, while the Large and XXL models changes the average summary length only slightly, the Base model completely degenerates, “hacking” the NLI reward by generating summaries that are half as short as the reference. This suggests that higher-capacity models are essential to prevent reward hacking, perhaps due to two possible reasons. First, the larger policies have higher generalization capabilities overall and can better accommodate different rewards, such as entailment and summarization regularization in our case. Second, since the anchor model uses the same architecture, the higher capacity anchor model is more robust to changes in the summary and produces lower scores for less informative or more extractive summaries.

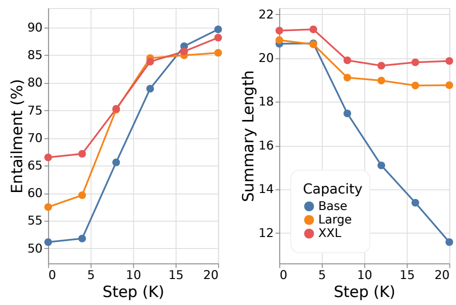

Figure 4: Entailment ratio and summary length during RL training for different model sizes.

5.1 Manual Analysis.

To gain more insight into the inner workings of RLEF, we propose two manual inspections about the types of changes being induced by the policy, and analysis of attribution errors found by our human evaluation procedure.

Changes to the summary during RL training. We study the changes that the RLEF_H policy induces on a summary during RL training, focusing on the changes that cause a flip in entailment decision. We sample 200 documents from the validation set for which we obtain the predicted summary at different checkpoints throughout the RL training process in 4K steps intervals. We apply the NLI classifier for each document and summary list, and select 60 examples for which the NLI decision has flipped between any pair of consecutive checkpoints, and study what changes have been made to the summary that caused the flip. Notably, most flips occur only once during training, and from the non-entailed to the entailed decision. Examples are shown in Table 4 together with our categorization of the changes, with some summaries morphing in more than one way. We notice that for summaries produced by RLEF_H most changes are local, meaning that the main predicate clause and the

	Summary before the NLI flip	Summary after the NLI flip	NLI	Description
1	Two astronauts who spent a year living on the International Space Station have landed in Florida.	Two astronauts who spent a year living on the International Space Station have returned to Earth.	✓	Abstractive Rephrasing
2	Afghan forces have repelled an advance by Taliban fighters on the northern city of Kunduz, officials say.	Afghan forces have been battling Taliban insurgents in the northern city of Kunduz.	✓	Abstractive Rephrasing
3	A senior Nigerian military official has said militant Islamist group Boko Haram is no longer a threat, after a mosque attack that left at least 82 people dead.	A senior Nigerian official has denied that Islamist militant group Boko Haram was behind a mosque attack in the north in which more than 100 people were killed.	✓	Claim Change
4	Two people have been arrested on suspicion of manslaughter after a three-year-old boy died at a water park.	Two people have been arrested after a four-year-old boy died at a water park.	✓	Argument Omission
5	Bolton Wanderers manager Lee Trotter has apologised after he and striker Aaron Lennon swore at fans on live television.	Bolton Wanderers manager Lee Trotter has apologised after he and team-mate Gary Caldwell swore at fans on live television.	✗	Argument Change

Table 4: Examples of summaries for the same document on consecutive checkpoints during RL training, before and after the NLI classification of the summary has flipped. The summaries maintain a stable main structure which enables manual inspection. The main changes are highlighted in gray, the NLI column depicts the entailment decision for the latter summary, and the description specifies the type of the semantic change.

core participants remain the same throughout most checkpoints. We classified 13 out of 60 examples as abstractively rephrased, where a specific detail is replaced with a broader description, e.g. *returned to earth* instead of *landed in Florida* (ex. 1). However, we also found that 27 examples contained argument omissions, where verbal arguments or noun modifiers with typically non-core semantic roles (Palmer et al., 2005) are removed (e.g. Locative or Temporal descriptions). See for example the "Cause for arrest" omission in ex. 5. Such omissions keep the information regarding the main participants intact, while lowering the risk of errors around non-core details. Other changes included claim changes (16 cases) where a predicate has been replaced (see ex 3), argument replacements (8 cases), and other non-specific alterations.

Attribution error analysis. We analyzed attribution errors from the human evaluation of our best policy, RLEF_L, aggregated by majority vote. We inspect the offending phrase supplied by the evaluator for 39 out of 100 examples that are found to be non-attributable. 28 are considered as a local hallucination, mostly confirming to addition of personal names, numbers, places, and roles that did not appear in the article. For example, an article mentioned *Kevin O'Malley* without alluding to his job title, while the summary referred to him as the *Irish Ambassador*. While Kevin O'Malley was indeed an Irish ambassador, the model should not add such details if they are not explicitly mentioned in the article. Since most of these examples were found as entailing by our reward, this may point at issues with the NLI model that are due to knowledge conflicts between its parametric and contextual knowledge (Neeman et al., 2022). The rest of the examples include 5 contradictions and 5 major hallucinations.

6 Related Work

RL for text generation. RL has been applied to many text generation tasks like neural machine translation (Wu et al., 2018; Leblond et al., 2021), extractive summarization (Narayan et al., 2018b; Wu and Hu, 2018; Gao et al., 2019; Arumae and Liu, 2019), abstractive summarization (Chen and Bansal, 2018) and others (Bahdanau et al., 2017; Welleck et al., 2019; Bai et al., 2022a; Ouyang et al., 2022; Bai et al., 2022b).

Specifically for summarization, prior RL approaches used different reference-based metrics as a reward function. In Pasunuru and Bansal (2018), two reward signals are measured between the generated and reference summaries: lexical overlap (ROUGE) to gauge salience and an entailment score to measure factual consistency. Gunasekara et al. (2021) employed a similar approach with question-answering, they produced QA pairs conditioned on the generated summary to detect inconsistencies with the reference, and another set of QAs conditioned on the reference to measure salience. Additionally, Nan et al. (2021) proposed QUALS, a more computationally efficient QA approach, that was used in a contrastive learning setting. While their approach could be used without comparing outputs to reference summaries, they observed that adding such comparisons with the reference is essential for the stability of their method. We note that for some datasets, reference summaries are likely to contain factual errors (Maynez et al., 2020), decreasing the effectiveness of reference-based rewards.

Other RL methods, instead of explicitly defining the quality of a summary suggest to model it directly from human feedback (Böhm et al., 2019; Ziegler et al., 2019; Wu et al., 2020; Stiennon et al., 2020). This technique can prevent errors due to references that are misaligned with human judgment. While it is a promising approach, it also requires acquiring task-specific annotation, which can be labor-intensive.

Another hybrid approach interleaves a cross-entropy objective with policy gradients (Pang et al., 2021) in multi-document summarization (MDS). They use an in-domain NLI model, for which they annotate their MDS dataset with entailment decisions. To stabilize their policy they employ an additional GAN-like training regime and add a discriminator loss between generated and reference summaries to their reward.

Trade-offs in consistency models. The choice of which factual consistency approach to use has interesting consequences for the RL setup. Our work employs a binary NLI decision that does not point towards the specific inconsistent parts in the output summary. Consequently, the reward is assigned to the final token of the summary, leaving proper credit assignment to the RL algorithm. Other methods, specifically those based on question-answering (Durmus et al., 2020; Wang et al., 2020; Honovich et al., 2021) can frame misaligned answers in the generated summary and assign the reward explicitly to the offending tokens. However, these QA-QG based methods may be much slower to compute. Our reward requires a single forward pass using a transformer model over the document-summary pair, in comparison, QA-QG approaches require generating answer candidates, questions, answers from both sources and computing answer alignment. Some of this complexity is remedied by generating jointly questions-and-answers (Nan et al., 2021), but it still requires a lengthy decoding of QA pairs. A different NLI-based approach decomposes the document and summary into smaller blocks of sentences (Laban et al., 2022) and aggregates the final decision over a matrix of block-level NLI scores. Such approach could aid the RL algorithm with credit assignment when generating long summaries. In practice, the abstractive summarization datasets in this study use short single sentence summaries.

7 Conclusions and Future Work

We propose to leverage NLI models as a ready-made, reference-free reward signal for RL training of factually consistent abstractive summarization models. Our experiments and thorough analysis with automatic and human evaluation show promising results for this approach, with our RL approach outperforming all baselines on factual consistency, while maintaining and even improving on other desired summarization attributes as well.

In future work, we would like to extend this approach to other grounded generation tasks, like knowledge-driven dialog. In addition, we find it interesting to explore additional reference-free reward models for other summarization attributes (or for other tasks). Then, an important research direction would be to understand how to properly adapt our method to work with multiple such rewards.

Limitations

While our approach shows promising results in both automatic and human evaluation, it relies on two significant pillars: a strong entailment model and a strong initial summarization model. The NLI model implicitly encodes the biases and other data regularities that were part of the NLI training set into the generated summaries of our policy. This is well demonstrated by the gap between human attribution judgements and the automatic NLI metric. Our RL policies cannot improve on factual consistency errors if they are undetectable by the NLI reward. Hopefully, as NLI capabilities get better, so will the efficacy of RLEF and the abilities to automatically flag hallucinations and contradictions.

Secondly, a strong summarization model is essential for our method in two ways: as an initialized starting point for RL exploration and as an anchor point to a policy. While our RL training does not require any reference data and opens the possibility to use more un-summarized documents, it would probably not succeed as well without initializing from a high-quality supervised model.

Another limitation is that our experiments suggest that model size is important when using RLEF (Figure 4): both our summarization and NLI models are 11B parameters models. We believe it is important to further understand how to make our approach more robust to smaller models, to increase its computational efficiency and availability.

Ethics Statement

Our work aims at solving the ethical issue of addressing misinformation in automated text generation tasks. Yet, adopting automatic summarization by real users can amplify misinformation in cases where the model still makes an error or when the input text itself is not trustworthy. As we stated in the limitations, our trained models heavily rely on other predictive models and therefore carry the biases of their training data, and may implicitly encode these into our generative process. Therefore, we believe that to reach real-world use, not just our method should be scrutinized but also the NLI and summarization datasets that were used to train these models. Thus, such methods should be used with caution and combined with other techniques to ensure humans are capable of judging the validity of the information generated by the model.

Acknowledgements

We would like to thank our anonymous reviewers for their thorough comments and insightful suggestions.

References

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. 2018. Maximum a posteriori policy optimisation. *arXiv preprint arXiv:1806.06920*.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*.

Kristjan Arumae and Fei Liu. 2019. Guiding extractive summarization with question-answering rewards. In *Proceedings of NAACL*.

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. An actor-critic algorithm for sequence prediction. In *International Conference on Learning Representations*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022b. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. In *Journal of machine learning research*.

Florian Böhm, Yang Gao, Christian M. Meyer, Ori Shapira, Ido Dagan, and Iryna Gurevych. 2019. Better rewards yield better summaries: Learning to summarise without references. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3110–3120, Hong Kong, China. Association for Computational Linguistics.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Shuyang Cao and Lu Wang. 2021. CLIFF: Contrastive learning for improving faithfulness and factuality in abstractive summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6633–6649, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of ACL*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, MLCW'05, page 177–190, Berlin, Heidelberg. Springer-Verlag.

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 R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, Florence, Italy. Association for Computational Linguistics.

Katja Filippova. 2020. Controlled hallucinations: Learning to generate faithfully from noisy data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 864–870, Online. Association for Computational Linguistics.

Yang Gao, Christian M Meyer, Mohsen Mesgar, and Iryna Gurevych. 2019. Reward learning for efficient reinforcement learning in extractive document summarisation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. 2019. A theory of regularized markov decision processes. In *International Conference on Machine Learning*, pages 2160–2169. PMLR.

Tanya Goyal and Greg Durrett. 2020. Evaluating factuality in generation with dependency-level entailment. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3592–3603, Online. Association for Computational Linguistics.

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

Chulaka Gunasekara, Guy Feigenblat, Benjamin Sznajder, Ranit Aharonov, and Sachindra Joshi. 2021. Using question answering rewards to improve abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 518–526.

Assaf Hallak, Dotan Di Castro, and Shie Mannor. 2015. Contextual markov decision processes. *arXiv preprint arXiv:1502.02259*.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. True: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3905–3920, Seattle, United States. Association for Computational Linguistics.

Or Honovich, Leshem Choshen, Roee Aharoni, Ella Neeman, Idan Szpektor, and Omri Abend. 2021. Q2: Evaluating factual consistency in knowledge-grounded dialogues via question generation and question answering. *arXiv preprint arXiv:2104.08202*.

Ronald A Howard. 1960. *Dynamic programming and markov processes.* John Wiley.

Natasha Jaques, Shixiang Gu, Dzmitry Bahdanau, José Miguel Hernández-Lobato, Richard E Turner, and Douglas Eck. 2017. Sequence tutor: Conservative fine-tuning of sequence generation models with kl-control. In *International Conference on Machine Learning*, pages 1645–1654. PMLR.

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

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.

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

Rémi Leblond, Jean-Baptiste Alayrac, Laurent Sifre, Miruna Pislar, Jean-Baptiste Lespiau, Ioannis Antonoglou, Karen Simonyan, and Oriol Vinyals. 2021. Machine translation decoding beyond beam search. In *Proceedings of EMNLP*.

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.

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022a. Wanli: Worker and ai collaboration for natural language inference dataset creation. In *Proceedings of EMNLP Findings*.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022b. BRIO: Bringing order to abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2890–2903, Dublin, Ireland. Association for Computational Linguistics.

Edward Loper and Steven Bird. 2002. NLTK: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pages 63–70, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Bill MacCartney and Christopher D. Manning. 2007. Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 193–200, Prague. Association for Computational Linguistics.

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

Computational Linguistics, pages 1906–1919, Online. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

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

Feng Nan, Cicero Nogueira dos Santos, Henghui Zhu, Patrick Ng, Kathleen McKeown, Ramesh Nallapati, Dejiao Zhang, Zhiguo Wang, Andrew O. Arnold, and Bing Xiang. 2021. Improving factual consistency of abstractive summarization via question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6881–6894, Online. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018a. 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.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018b. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1747–1759, New Orleans, Louisiana. Association for Computational Linguistics.

Ella Neeman, Roee Aharoni, Or Honovich, Leshem Choshen, Idan Szpektor, and Omri Abend. 2022. Disentqa: Disentangling parametric and contextual knowledge with counterfactual question answering. *arXiv preprint arXiv:2211.05655*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. 2022. The effects of reward misspecification: Mapping and mitigating misaligned models. In *International Conference on Learning Representations*.

Richard Yuanzhe Pang, Adam Lelkes, Vinh Tran, and Cong Yu. 2021. AgreeSum: Agreement-oriented multi-document summarization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3377–3391, Online. 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.

Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694.

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. 2021a. Measuring attribution in natural language generation models. *CoRR (arXiv preprint)*.

Hannah Rashkin, David Reitter, Gaurav Singh Tomar, and Dipanjan Das. 2021b. Increasing faithfulness in knowledge-grounded dialogue with controllable features. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 704–718, Online. Association for Computational Linguistics.

Adam Roberts, Hyung Won Chung, Anselm Levskaya, Gaurav Mishra, James Bradbury, Daniel Andor, Sharan Narang, Brian Lester, Colin Gaffney, Afroz Mohiuddin, Curtis Hawthorne, Aitor Lewkowycz, Alex Salcianu, Marc van Zee, Jacob Austin, Sebastian Goodman, Livio Baldini Soares, Haitang Hu, Sasha Tsvyashchenko, Aakanksha Chowdhery, Jasmijn Bastings, Jannis Bulian, Xavier Garcia, Jianmo Ni, Andrew Chen, Kathleen Kenealy, Jonathan H. Clark, Stephan Lee, Dan Garrette, James Lee-Thorp, Colin Raffel, Noam Shazeer, Marvin Ritter, Maarten Bosma, Alexandre Passos, Jeremy Maitin-Shepard, Noah Fiedel, Mark Omernick, Brennan Saeta, Ryan Sepassi, Alexander Spiridonov, Joshua Newlan, and Andrea Gesmundo. 2022. Scaling up models and data with `t5x` and `seqio`. *arXiv preprint arXiv:2203.17189*.

John Schulman, Xi Chen, and Pieter Abbeel. 2017a. Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2016. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017b. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, Alex Wang, and Patrick Gallinari. 2021. QuestEval: Summarization asks for fact-based evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6594–6604, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lior Shani, Yonathan Efroni, and Shie Mannor. 2020. Adaptive trust region policy optimization: Global convergence and faster rates for regularized mdps. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5668–5675.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR.

Joar Max Viktor Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. 2022. Defining and characterizing reward gaming. In *Advances in Neural Information Processing Systems*.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan J. Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize from human feedback. *ArXiv*, abs/2009.01325.

Manan Tomar, Lior Shani, Yonathan Efroni, and Mohammad Ghavamzadeh. 2022. Mirror descent policy optimization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Sharan Vaswani, Olivier Bachem, Simone Totaro, Robert Mueller, Matthieu Geist, Marlos C Machado, Pablo Samuel Castro, and Nicolas Le Roux. 2021. A functional mirror ascent view of policy gradient methods with function approximation. *arXiv preprint arXiv:2108.05828*.

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.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics.

Sean Welleck, Kianté Brantley, Hal Daumé Iii, and Kyunghyun Cho. 2019. Non-monotonic sequential text generation. In *International Conference on Machine Learning*, pages 6716–6726. PMLR.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256.

Jeff Wu, Long Ouyang, Daniel M Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. 2020. Recursively summarizing books with human feedback. In *Advances in Neural Information Processing Systems*.

Lijun Wu, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018. A study of reinforcement learning for neural machine translation. In *Proceedings of EMNLP*.

Yuxiang Wu and Baotian Hu. 2018. Learning to extract coherent summary via deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *ArXiv*, abs/1912.08777.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Experimental Details

RL Algorithm Details. We use an actor-critic on-policy PG algorithm with a learned value function V_ψ and a parameterized policy π_θ to maximize the RL objective. The policy gradient w.r.t. to the regularized reward $r(y_t; y_{:t-1}, x)$ defined in Equation (1) is

$$\nabla_\theta J(\theta) = \mathbb{E}_{x,y\sim\pi_\theta}\Big[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(y_t|y_{:t-1}, x) G_t^\alpha\Big],$$

where for brevity we denote $G_t^\alpha = \sum_{t'=t}^{T} r(y_t; y_{:t-1}, x)$, the accumulated regularized return. For more details on the derivation of this expression, and framing the regularized objective as an RL problem, we refer the reader to Appendix C.

We use the value V_ψ as a baseline, a state-dependent function that can be subtracted in the policy gradient without changing it. This leads to the following equivalent policy gradient

$$\nabla_\theta J(\theta) = \mathbb{E}_{x,y\sim\pi_\theta}\Big[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(y_t|y_{:t-1}, x)\times \big(G_t^\alpha - V_\psi(y_{:t-1}, x)\big)\Big]$$
$$= \mathbb{E}_{x,y\sim\pi_\theta}\Big[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(y_t|y_{:t-1}, x) A_\psi^{\mathrm{GAE}}(y_{:t})\Big]$$

where A_ψ is termed the advantage function. Applying this PG can be regarded a variant of the REINFORCE (Williams, 1992) algorithm with a baseline. In practice, we replace the advantage in the expression above by *generalized advantage estimation* (GAE, Schulman et al., 2016), which allows to better control the bias-variance trade-off via the λ parameter:

$$A_\psi^{\mathrm{GAE}}(y_t; y_{:t-1}, x) = \sum_{t'=t}^{+\infty} (\gamma\lambda)^{t'-t} \times \Big(r(y_{t'}; y_{:t'-1}, x) + \gamma V_\psi(y_{:t'}, x) - V_\psi(y_{:t'-1}, x)\Big).$$

Finally, the above policy gradient definition leads to the following per-example loss for learning the policy π_θ,

$$\mathcal{L}^\pi(\theta)(y_{:t}, x) = A_\psi^{\mathrm{GAE}}(y_{:t}, x) \log \pi_\theta(y_t|y_{:t-1}, x),$$

where the gradients are only propagated here w.r.t. the policy parameters.

The value V_ψ itself is learned via regression towards the return estimate induced by GAE, which is equivalent to minimizing the GAE advantage:

$$\mathcal{L}^V(\psi)(y_{:t}, x) = \big(A_\psi^{\mathrm{GAE}}(y_{:t}, x)\big)^2.$$

We now describe more intricate implementation details and hyper parameter choices.

RL Implementation Details. Given that we operate in the finite horizon setting, we naturally set the discount factor γ to 1. Similarly to the PPO algorithm (Schulman et al., 2017b), we normalize the advantages in a given batch of data so that they approximately follow a standard normal distribution. We also normalize the value loss by dividing it by the variance of the batch returns. An important difference between our implementation and the standard (regularized) PG implementation is that instead of treating KL penalties along a given sequence as immediate rewards, we accumulate those and treat the resulting quantity as a sequence-level penalty. We found this to lead to more stability in the RL procedure.

Unlike the conventional RL setting where both the policy and value are randomly initialized, in our case the policy is already fine-tuned to solve the required task. Thus, to make the value function accurate w.r.t. the already initialized policy, we observed that we needed a small number of iterations before the value estimation is sufficiently accurate to avoid detrimental policy gradients. To do so, we run RL fine-tuning for 20K steps, with a warmup of 5K steps for the value network. We also noticed that it was beneficial to use distinct values for the policy and value learning rates, so we decouple them in practice.

Optimization. We use Adafactor (Shazeer and Stern, 2018) with a learning rate warmup phase: the learning rate is linearly annealed from zero to the specified asymptotic value.

Hyperparameter Search. We noticed that the optimal value of the policy and value learning rates are highly correlated. Hence, we propose a decoupled hyperparameter search: we start by finding a suitable value learning rate by keeping the policy fixed. We then follow a standard grid search to find suitable values for the remaining hyperparameters including the policy learning rate, temperature and the regularization coefficient α. Specifically, in our hyperparameter sweep we used temperatures $[0.1, 0.3, 1.0]$ and α values between 0.1 and 0.8

Hyperparameter	Value
γ	1
GAE λ	0.95
Batch size	32
Temperature	0.3 / 1.0
Regularization α	0.2 / 0.1
LR warmup period	2000
Policy update delay	5000
Policy LR	1e-5
Value LR	1e-5

Table 5: Hyperparameters for the RL fine-tuning procedure of RLEF. When denoted left/right, left refers to the hyper parameter used for RLEF_L and right for RLEF_H.

with a grid size of 0.1. Thus overall, our main sweep for the XXL model consisted of 24 runs of $20K$ iterations.

We list all the hyperparameters used (unless different values are mentioned in the text) in Table 5. For the learning rate warmup and policy update delay, note that the number of steps reported correspond to gradient steps of the RL fine-tuning procedure.

SL Implementation details. For the SL models, we decode summaries with beam search with a beam width of 4 and a brevity penalty of 0.6. For training we use the same optimizer with base learning rate of 0.001, batch size of 32, and a dropout rate of 0.1.

Resources. We used TPU-v4 chips to train all the models mentioned. Each of our T5-XXL based RLEF experiment ran for approximately 17 hours on 64 TPU chips. Furthermore, our main hyper parameters sweep included 24 such experiments, accounting for 1088 TPU-days.

B Evaluator Demographics, UI and Instructions

We employed full-time hourly workers to rate the summary quality. Our raters consist of native English speakers, nationals from the U.S. and U.K. that hold graduate (70%) and high-school (30%) diplomas. We supplied them with 2 pages of instructions and additional examples, and conducted an initial pilot study and training batch before proceeding to rate the summaries. The UI that we used is displayed in Figure 5. In what follows we attach the guidelines presented to the raters in the human evaluation described in Section 3.6. The guidelines are loosely based on Rashkin et al. (2021a).

B.1 Guidelines

In this task you will be presented with a news article and multiple summaries of the article, and you are asked to evaluate the summary quality. You will rate each summary with 4 yes/no questions. These questions ask if the summary is: Comprehensible and understandable. Attributable (supported) by the article - no contradicting or unattested information. Captures the main idea(s) behind the article. Concise - does not contain additional details beyond the key information in the article. Read carefully the text and the summary. The summaries may appear very fluent and well-formed, but contain slight inaccuracies that are not easy to discern at first glance.

Q1: Comprehensibility. An incomprehensible summary is not understandable due to significantly malformed phrases and sentences that are difficult to comprehend or make sense of. If there is any part of the summary that is unclear or hard to understand or malformed (e.g., partially cut-off or contains strange characters), select "No, not fully comprehensible". Summary When you leave it late, you leave it late is adding interest to your pension money as a result of the financial crisis. o, not fully comprehensible

Q2: Attributable (Supported) by the article. A fully supported summary contains information that can be found in the source article. No information in the summary is unattested when compared against the source news article. In other words, if you can say that “According to the news article. . . ” with the summary following this phrase, you should answer, “Yes, it is attributable.” If some key details in the summary are not supported by the article (e.g. missing from the article), inaccurately represent the information in the article, or contradicted by the article, then please mark “No, not fully attributable.”

Q3: Main Idea. A main idea captures a fact or theme that is central to the article’s discussion. It should involve the people, locations, or events that the article focuses on. If a main idea was removed from the original article, it would change the meaning, focus, or argument of the article. Note that

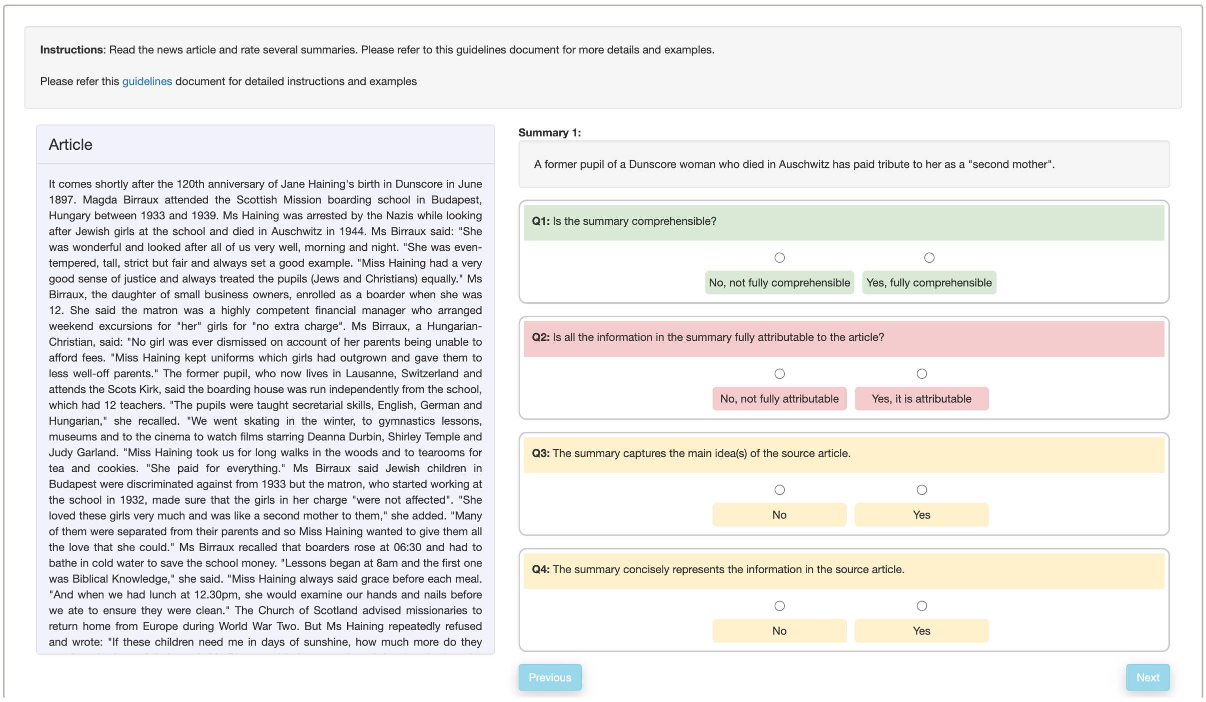

Figure 5: The evaluator user interface that we have used to rate summaries.

this question is NOT asking whether the summary includes ONLY main ideas.

In Q3, to the best of your ability try to distinguish between the following cases, some may be more rare than others:

- The summary is fully supported (yes to Q2) and captures the main idea (yes to Q3).
- The summary is fully supported (yes to Q2), but ignores the central point of the document (No in Q3).
- The summary contradicts the document in minor details or hallucinates some information (No to Q2), but the idea behind the document is mostly captured even if some details are incorrect (Yes to Q3).
- The summary contradicts the document in key details (No to Q2) to the level where the main idea is unrecoverable or largely missed (No to Q3).

Q4: Conciseness. A summary is concise if it includes only the necessary details and the important information in the article. If it includes any details which are not central to the article, it should be marked as "No, it is not concise". A summary may be concise even if some details are contradicting (i.e. you marked “No, it is not fully attributable” in Q2) as long as those were part of the main idea of the article.

In Q4 we are trying to find if the summary contains substantial information that does not belong to the main idea. If some minor details in the summary are contradicting, yet they are part of the main idea, then this summary is still concise, the system made an error of attribution, but not of overgeneration.

C Fine-Tuning Language Models with Reinforcement Learning

C.1 Language Generation as a Contextual Markov Decision Problem

In this appendix, we explain the connection between arbitrary language generation tasks and the Markov Decision Process (MDP) framework (Howard, 1960) which is widely used in RL. We recall that an MDP M is a tuple $M = (\mathcal{S}, \mathcal{A}, \gamma, r, P)$, where $\mathcal{S}$ is a state space, $\mathcal{A}$ is an action space, $\gamma \in [0, 1]$ is a discount factor, $r : \mathcal{S} \times \mathcal{A} \to [-r_{max}, r_{max}]$ is a bounded reward function and $P : \mathcal{S} \times \mathcal{A} \to \Delta_{\mathcal{S}}$ is a transition kernel. Δ_{χ} denotes the standard simplex over χ. We represent sequential decision-making strategies as policies $\pi : \mathcal{S} \to \Delta_{\mathcal{A}}$. At any point in time t, a policy π interacts in an MDP by observing the current state s_t, selecting an action $a_t \sim \pi(\cdot|s_t)$, and accordingly receiving a reward $r_t = r(s_t, a_t)$, before observing a new state $s_{t+1} \sim P(\cdot|s_t, a_t)$. We define the return as the discounted sum of rewards in one episode of interaction: $G_t = \sum_{t'=t}^{T} \gamma^{t'} r_{t'}$, where T is called the horizon and is potentially infinite. We now introduce Contextual MDPs (CMDPs) (Hallak et al., 2015). They model the fact that a fixed context is available and determines the nature of rewards and dynamics. Formally, a CMDP is a tuple $M_c = (\mathcal{C}, f_M)$, where $\mathcal{C}$ is a context space and $f_M : c \in \mathcal{C} \to M$ is a function that maps a context to the corresponding MDP.

Any language generation task can be seen as the following interactive process: a language model observes the current state $s_t = y_{:t-1}$ and context $c = x$, that is both the input text x and the text generated so far $y_{:t-1}$, and selects a token $a_t = y_t$. Thus, we can view any language generation task as a CMDP $M_c = (\mathcal{C}, f_M)$ with $f_M(c) = (\mathcal{S}, \mathcal{A}, \gamma, r(\cdot\,; c), P(\cdot\,; c))$, with the policy π being the language model itself. The state space $\mathcal{S}$ is the set of all potential generations (either complete or incomplete). We suppose that the maximum length of generated text T, which is equivalent to the horizon, and that of the input text T_c are finite, which is a common assumption in NLP. Accordingly, if we note $\mathbb{V}$ the vocabulary (the set of all admissible tokens), we have $\mathcal{S} = \cup_{i=0}^{T} \mathbb{V}^i$. Similarly, we have the context space $\mathcal{C} = \cup_{j=0}^{T_c} \mathbb{V}^j$. The action space $\mathcal{A}$ is the set of tokens that the policy can output at any point in time, that is the vocabulary, hence $\mathcal{A} = \mathbb{V}$. The discount factor γ is arbitrary and can be set to 1 given that the horizon is supposedly finite. The reward function r is also arbitrary, but in the case of interest exposed in the main text we set it to:

$$r(s_t, a_t; c) = \begin{cases} \mathrm{NLI}(y_{:t}; x) \text{ if } y_t = \text{[EOS]} \text{ or } t = T, \\ 0 \text{ otherwise.} \end{cases}$$

Finally, and most importantly, the transition kernel is deterministic:

$$P(s_{t+1}|s_t, a_t; c) = \begin{cases} 1 \text{ if [EOS]} \in s_t \text{ and } s_{t+1} = s_t, \\ 1 \text{ if [EOS]} \notin s_t \text{ and } s_{t+1} = y_{:t}, \\ 0 \text{ otherwise.} \end{cases}$$

Indeed, any state that contains an [EOS] token can be considered an absorbing state.

C.2 Language Generation From a Pre-Trained Model as a Regularized Markov Decision Problem

While the previous formalism applies to all language generation tasks, we now describe a formalism that specifically applies to the language generation task that is explored in the main text: language generation when a pre-trained model is available. It models the fact that we want generated text to be likely according to the pre-trained model, which we call *anchor model* in what is next. We note the corresponding policy π^{anchor}. We consider the following reward function:

$$r(s_t, a_t) = (1 - \alpha) r(s_t, a_t) + \alpha r^{\text{KL}}(s_t, a_t),$$

with the regularization term:

$$r^{\text{KL}}(s_t, a_t) = \log \pi^{\text{anchor}}(a_t|s_t) - \log \pi(a_t|s_t),$$

where r is the reward function defined previously and α is a scalar controlling the regularization strength. We recall that the Kullback-Leibler (KL) divergence between the current policy and the anchor policy has the expression:

$$\mathrm{KL}(\pi||\pi^{\text{anchor}})(s_t) = \\ - \mathbb{E}_{a_t \sim \pi} \Big[\log \pi^{\text{anchor}}(a_t|s_t) - \log \pi(a_t|s_t) \Big].$$

Hence, the regularization term is an unbiased estimator for the KL divergence between current and anchor policies. Intuitively, it encourages the learned policy to keep a distribution that is close to the distribution over tokens induced by the anchor policy (the fine-tuned model). Since the learned policy evolves along training, the reward function we described is non-stationary, that is the reward for a given state-action pair (s, a) changes with the policy π. Hence, the modified MDP is best viewed as a regularized MDP (Geist et al., 2019). We define the KL regularizer as $\Omega(\pi) = \mathrm{KL}(\pi||\pi^{\text{anchor}})$, which is a strongly convex function. We can show that this formalism is equivalent to the MDP with the non-stationary reward function described above.

C.3 Defining the Reinforcement Learning Objective

In this section, we show that the regularized reward defined in Equation (1) can be used together with any PG based algorithm. To do that, we show that for any MDP (see Appendix C.1), the policy gradient can be easily re-derived for our regularization scheme when using parameterized policies. This repeats the derivations in Schulman et al. (2017a); Geist et al. (2019). We denote trajectories $\tau = \{s_0\} \cup \{a_t, s_{t+1}\}_{t=0}^{T-1}$. By a slight abuse of notations we denote the probability of a given trajectory under the policy π as $\pi(\tau)$, that we can decompose as $\pi(\tau) = P(s_0)\prod_{i=0}^{T-1}\pi(a_i|s_i)P(s_{i+1}|s_i,a_i)$. We also denote G_t as the return of a trajectory starting from time-step t. Now, denote a parameterized policy π_θ, and define the standard RL objective,

$$\begin{aligned} J(\theta) &= \mathbb{E}_{\tau\sim\pi_\theta}\Big[\sum_{t=0}^{T} r(s_t,a_t)\Big], \\ &= \mathbb{E}_{\tau\sim\pi_\theta}[G_0]. \end{aligned}$$

The goal of RL is to find a parameterization θ^* that maximizes the following objective:

$$\theta^* \in \arg\max_\theta J(\theta).$$

The policy gradient theorem states that

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\Big[\sum_{t=0}^{T}\nabla_\theta \log \pi_\theta(a_t|s_t)G_t\Big].$$

We now place ourselves in the specific regularized MDP defined in Equation (1) and Appendix C.2, with the reward regularization scheme, $r^\alpha(s,a) = (1-\alpha)r(s,a) + \alpha \log \frac{\pi^{\text{anchor}}(a|s)}{\pi_\theta(a|s)}$. Define the RL objective of interest, which adds a regularization term to the reward function,

$$\begin{aligned} J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\Big[&\sum_{t=0}^{T}(1-\alpha)r(s_t,a_t) \\ &+ \alpha \log \frac{\pi^{\text{anchor}}(a_t|s_t)}{\pi_\theta(a_t|s_t)}\Big]. \end{aligned}$$

For $r(s_t,a_t)$, we repeat standard steps to re-derive the corresponding policy gradient. However, we need to have a separate treatment for the KL regularization reward $\log \frac{\pi^{\text{anchor}}(a_t|s_t)}{\pi_\theta(a_t|s_t)}$, as it explicitly depends on θ. We have:

$$\begin{aligned} &\nabla_\theta \mathbb{E}_{\tau\sim\pi_\theta}\Big[\sum_{t=0}^{T}\log \frac{\pi^{\text{anchor}}(a_t|s_t)}{\pi_\theta(a_t|s_t)}\Big] \\ &= -\nabla_\theta \mathbb{E}_{\tau\sim\pi_\theta}\Big[\sum_{t=0}^{T}\log \frac{\pi_\theta(a_t|s_t)}{\pi^{\text{anchor}}(a_t|s_t)}\Big] \\ &= -\nabla_\theta \sum_\tau \pi_\theta(\tau)\sum_{t=0}^{T}\log \frac{\pi_\theta(a_t|s_t)}{\pi^{\text{anchor}}(a_t|s_t)}, \\ &= -\sum_\tau \nabla_\theta\Big(\pi_\theta(\tau)\sum_{t=0}^{T}\log \frac{\pi_\theta(a_t|s_t)}{\pi^{\text{anchor}}(a_t|s_t)}\Big), \\ &= \underbrace{-\sum_\tau \nabla_\theta\pi_\theta(\tau)\sum_{t=0}^{T}\log \frac{\pi_\theta(a_t|s_t)}{\pi^{\text{anchor}}(a_t|s_t)}}_{A} \\ &\quad \underbrace{-\sum_\tau \pi_\theta(\tau)\nabla_\theta\sum_{t=0}^{T}\log \frac{\pi_\theta(a_t|s_t)}{\pi^{\text{anchor}}(a_t|s_t)}}_{B}. \end{aligned}$$

We keep A as is and show that B is equal to 0:

$$\begin{aligned} B &= \sum_\tau \pi_\theta(\tau)\nabla_\theta\sum_{t=0}^{T}\log \frac{\pi_\theta(a_t|s_t)}{\pi^{\text{anchor}}(a_t|s_t)} \\ &= \sum_\tau \pi_\theta(\tau)\sum_{t=0}^{T}\nabla_\theta\log \frac{\pi_\theta(a_t|s_t)}{\pi^{\text{anchor}}(a_t|s_t)}, \\ &= \sum_\tau \pi_\theta(\tau)\nabla_\theta\sum_{t=0}^{T}\log \pi_\theta(a_t|s_t), \\ &= \sum_\tau \pi_\theta(\tau)\nabla_\theta\log \pi_\theta(\tau), \\ &= \sum_\tau \nabla_\theta\pi_\theta(\tau), \\ &= \nabla_\theta\sum_\tau \pi_\theta(\tau), \\ &= 0. \end{aligned}$$

By putting all the pieces together we get the expression of the policy gradient for the modified RL objective:

$$\begin{aligned} &\nabla_\theta J(\theta) \\ &= \mathbb{E}_{\tau\sim\pi_\theta}\Big[\sum_{t=0}^{T}\nabla_\theta \log \pi_\theta(a_t|s_t)\sum_{t'=t}^{T} r^\alpha(s_t,a_t)\Big], \end{aligned}$$

Denoting G_t^α, the return of the trajectory when using r^α, this can be rewritten as,

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\Big[\sum_{t=0}^{T}\nabla_\theta \log \pi_\theta(a_t|s_t)G_t^\alpha\Big].$$

Note that we recovered the standard policy gradient for the regularized reward r^α (and corresponding return G^α). This means that by treating r^α as the reward we can use any policy gradient method, to solve the new objective. Because this holds for any MDP, it holds for the specific MDP defined in Appendix C.1 for the summarization task. To see how this is concretely used in our approach to construct the PG losses, we refer the reader to Appendix A.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
We discuss the limitations of our work in section 8 Limitations.

☑ A2. Did you discuss any potential risks of your work?
We discuss potential risks in our ethics statement.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 2-3

☑ B1. Did you cite the creators of artifacts you used?
We cite the dataset creators in section 3.

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 3. We use well-known and publicly released datasets, with CC-BY-4 or MIT open licenses

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
The intended use of the summarization datasets we employed is to advance research in summarization, as we did.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
We use standard, well-known and publicly available datasets. Some of the content has been filtered by the creators of the datasets to remove problematic content. We publish only aggregated numerical rating data collected by our evaluators.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 3, we briefly mention the content type of each dataset.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 3, briefly, as we use the standard train validation and dev splits or splits used in earlier works (and we explicitly mention and cite those).

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C ☑ Did you run computational experiments?

Section 3 and appendix A.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 3 and Appendix A

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Sections 2-3, Appendices

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Sections 4-5. We report results of a single run for automatic metrics on mainstream dataset splits. Human evaluation results have confidence intervals.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 3, Appendix A

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Section 3-4, Appendix B

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Appendix B

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Section 3, Appendix B

☒ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
We hired full-time annotators and explained the essence of their work is for research purposes in NLP.

☑ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
A similar protocol was determined in previous studies.

☑ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Section 3, Appendix B