On Improving Summarization Factual Consistency from Natural Language Feedback

Yixin Liu[*1], **Budhaditya Deb**[2], **Milagro Teruel**[2],
Aaron Halfaker[2], **Dragomir Radev**[1], **Ahmed H. Awadallah**[2]
[1]Yale University, [2]Microsoft Research
{yixin.liu, dragomir.radev}@yale.edu, {Budha.Deb, hassanam}@microsoft.com

Abstract

Despite the recent progress in language generation models, their outputs may not always meet user expectations. In this work, we study whether informational feedback in natural language can be leveraged to improve generation quality and user preference alignment. To this end, we consider *factual consistency* in summarization, the quality that the summary should only contain information supported by the input documents, as the user-expected preference. We collect a high-quality dataset, **DeFacto**, containing human demonstrations and informational natural language feedback consisting of corrective instructions, edited summaries, and explanations with respect to the factual consistency of the summary. Using our dataset, we study three natural language generation tasks: (1) *editing a summary* by following the human feedback, (2) *generating human feedback* for editing the original summary, and (3) *revising the initial summary* to correct factual errors by generating both the human feedback and edited summary. We show that DeFacto can provide factually consistent human-edited summaries and further insights into summarization factual consistency thanks to its informational natural language feedback. We further demonstrate that fine-tuned language models can leverage our dataset to improve the summary factual consistency, while large language models lack the zero-shot learning ability in our proposed tasks that require controllable text generation.

1 Introduction

While recent natural language generation (NLG) models (Radford et al., 2019; Lewis et al., 2020; Raffel et al., 2020; Brown et al., 2020) have made significant progress on the generation quality, they cannot always generate outputs that meet the user needs. For example, while state-of-the-art summarization systems can generate fluent and relevant summaries, recent work (Goyal and Durrett, 2021; Tang et al., 2022) have shown that they still make errors on fine-grained qualities such as *factual consistency*.[1] These errors can lead to serious risks to the intended users and make it difficult for them to trust the systems for their decision-making.

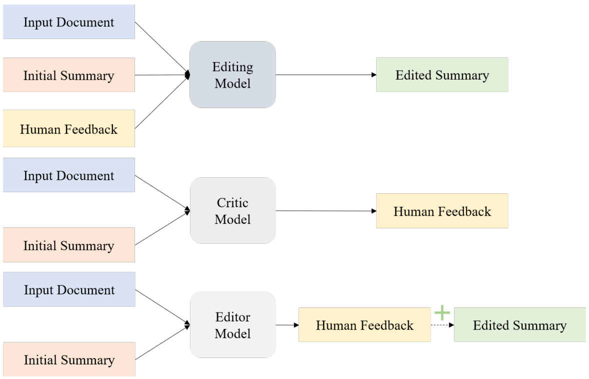


Figure 1: Three NLG tasks studied using our dataset. The ***Editing*** model aims to improve the initial system-generated summary given human feedback. The ***Critic*** model aims to predict human feedback according to a user-required quality. The ***Editor*** model aims to automatically correct factual errors by predicting both the human feedback and edited summary.

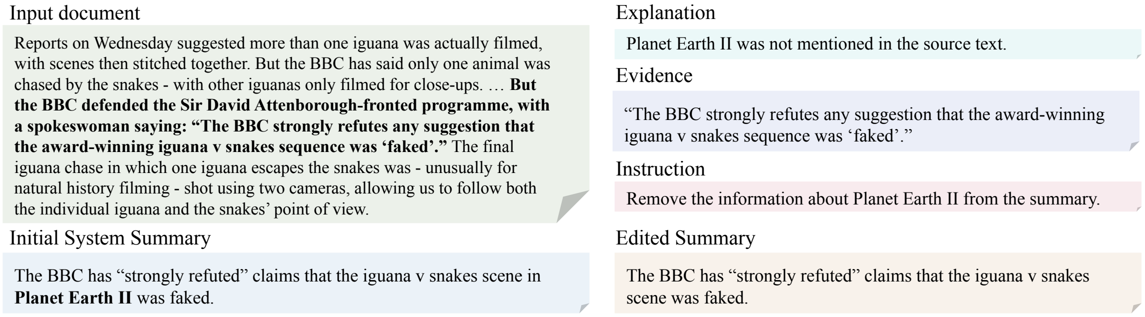
Such failures in satisfying the user needs have an *intrinsic* reason – the large benchmark datasets that are used to train NLG models are usually not collected according to pre-defined user needs, which results in a discrepancy between **model behaviors** and **user expectations**. For example, XSum (Narayan et al., 2018), one of the most commonly used summarization datasets, contains a large portion of reference summaries with *hallucinations*.[2] As a result, summarization models trained on XSum dataset generate many non-factual

[*] Most of the work was done while the first author was an intern at Microsoft Research.

[1] Following Goyal and Durrett (2021), we define factual consistency as the summary quality that *all the information of the summary can be supported by the source document*.

[2] Maynez et al. (2020) reports that around 76.9% reference summaries on the XSum dataset contains hallucinated contents that are *not* supported by the source documents.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 15144–15161
July 9-14, 2023



Input document

Reports on Wednesday suggested more than one iguana was actually filmed, with scenes then stitched together. But the BBC has said only one animal was chased by the snakes - with other iguanas only filmed for close-ups. … **But the BBC defended the Sir David Attenborough-fronted programme, with a spokeswoman saying: “The BBC strongly refutes any suggestion that the award-winning iguana v snakes sequence was ‘faked’.”** The final iguana chase in which one iguana escapes the snakes was - unusually for natural history filming - shot using two cameras, allowing us to follow both the individual iguana and the snakes’ point of view.

Initial System Summary

The BBC has “strongly refuted” claims that the iguana v snakes scene in **Planet Earth II** was faked.

Explanation

Planet Earth II was not mentioned in the source text.

Evidence

“The BBC strongly refutes any suggestion that the award-winning iguana v snakes sequence was ‘faked’.”

Instruction

Remove the information about Planet Earth II from the summary.

Edited Summary

The BBC has “strongly refuted” claims that the iguana v snakes scene was faked.

Figure 2: Data example in DeFacto. The initial summary contains a factual error about the name of the program *Planet Earth II*. The annotator provided an ***explanation*** about why the initial summary is not factually consistent, ***evidence*** (i.e., a sentence in the input document) to support their claims, ***instructions*** on how to correct the summary, and an ***edited summary*** (a demonstration) without the factual error.

contents, more than models trained on datasets such as CNN/DailyMail (Hermann et al., 2015) dataset (Goyal and Durrett, 2021). Unfortunately, it can be prohibitively expensive to collect new, large-enough datasets to train NLG models according to user needs, as they can be diverse, personal, and ever-changing over time.

Instead of aligning an existing NLG model to a specific user need, we explore adjusting *model outputs* according to the user needs through **human demonstrations and feedback**. Specifically, we investigate three scenarios (Fig. 1): (1) an ***Editing*** model that aligns initial system outputs to human demonstrations based on the user feedback; (2) a ***Critic*** model that predicts user feedback of initial system outputs according to the user requirements; (3) an ***Editor*** model that automatically aligns the initial system outputs to user needs by predicting both the user feedback and edited summary.

We choose ***factual consistency*** of system-generated summaries as the *user-required quality* to study the aforementioned application scenarios. To this end, we collect a high-quality, informational dataset containing human demonstrations and feedback. Specifically, the annotators are presented with initial system-generated summaries and asked to make changes to the summaries to make them factually consistent if they find errors in them. Apart from the **human-edited, factually consistent summaries**, the annotators are also required to provide **instructions** on how to change the initial summaries (i.e., if they find errors in them) and **explanation** on why the initial summaries are factually consistent or not. An example of our dataset is shown in Fig. 2. Using the collected dataset, we show that (1) the *Editing* model can effectively leverage human feedback to adjust the initial system outputs towards human demonstrations; (2) the *Critic* model can learn to generate meaningful feedback that can be used by the *Editing* model; (3) the *Editor* model can automatically correct factuality errors without explicit human intervention. Moreover, we find that the *Editor* model achieves better performance than the baseline model that only generates the edited summary, which indicates that natural language feedback can be beneficial for training models for the corresponding task.

Our contributions can be briefly summarized as: (1) we collect **DeFacto**,[3] a *high-quality dataset* containing human **De**monstrations and **F**eedback for improving f**act**ual c**o**nsistency of text summarization; (2) we conduct comprehensive analyses on the collected dataset, which provides further insights about factual consistency in text summarization, such as the relation between the type of factual errors and the type of editing operations; (3) we provide strong baseline models for the proposed three NLG tasks – summary editing (*Editing* model), feedback generation (*Critic* model), and automatic factuality error correction with feedback prediction (*Editor* model), which illustrates methods of leveraging natural language feedback for aligning model outputs with user expectations. (4) we present two case studies with large language models (LLMs) such as GPT-3.5 (Ouyang et al., 2022b), showing that LLMs still lack the *controllable* text generation ability in our proposed tasks.

2 The DeFacto Dataset

Our dataset, DeFacto, contains human demonstrations and feedback w.r.t. the factual consistency of system-generated summaries. We choose **XSum**

[3] We make the **DeFacto** dataset publicly available at `https://github.com/microsoft/DeFacto`.

dataset as the target dataset to conduct the data collection because it is the most commonly studied dataset for summarization factual consistency. For the system-generated summaries, we select **PEGASUS** (Zhang et al., 2020), a top-performing summarization model to generate summaries on both the validation and test set of the XSum dataset.

2.1 Annotation Process

Our annotation process follows the following steps:
(1) **Detect errors**: The annotator is required to evaluate a summary given the source document and **decide if the summary is factually consistent**.
(2) **Categorize errors**: If the annotator decides the summary *is not* factually consistent, they are required to **categorize the factual errors** in the summary as either *intrinsic* or *extrinsic*.[4] We note that both error detection and categorization are defined at the summary level.
(3) **Give explanation**: The annotator is required to **provide a natural language explanation** on why the summary is factually consistent or not.
(4) **Provide evidence**: The annotator is required to **select a sentence from the source document as evidence** to support their claims described in (3).
(5) **Write corrective instruction**: The annotator is required to **provide instructions** of how to correct the original summary if they think it is not factually consistent. To enforce uniformity and reduce the noise in the instructions, we provide six templates for the annotators corresponding to different operations: *Remove*, *Add*, *Replace*, *Modify*, *Rewrite*, and *Others*. The annotators need to fill in the templates to generate the instructions. The details of the templates are in Appendix A.1.
(6) **Correct summary**: Following the instruction in (5), the annotator is required to **edit the initial summary** to make it *factually consistent* with minimal, necessary modifications.
We provide annotated examples in Appendix A.2.

2.2 Data Collection

We conduct our data collection on Amazon Mechanical Turk[5] (MTurk) platform. The MTurk annotators need to pass a qualification test to be able to accept our assignments. The qualification test includes three actual annotation tasks, and we manually checked the correctness of the answers of the annotators and assigned them scores accordingly.

	Train	Val	Test	All
All	1000	486	1075	2561
w/ Errors	701	341	779	1821

Table 1: Numbers of data points in DEFACTO dataset. 71.1% of annotated summaries contain factual errors.

For the actual tasks, we collected one annotation for each example (i.e., a document-summary pair), and collected around 1000 examples on the test set and 1500 examples on the validation set. To estimate the inter-annotator agreement, we additionally collect two more annotations for 100 examples on the test set. We require the annotators to be located in the United States. Depending on the difficulty of the assignments, the annotators are compensated with 1.2 - 2.0 US dollars per assignment accordingly based on a $12/hour pay rate.

To check the inter-annotator agreement on steps (1) *Detect Errors* and (2) *Categorize Errors* in §2.1, we calculated the Krippendorff's alpha (Krippendorff, 2011), and found that the agreement score is 0.5552, 0.1899, 0.5260 for if the summary contains ***extrinsic*** factual errors, ***intrinsic*** factual errors and ***any*** factual errors respectively. For human-written ***explanation***, ***instructions***, and ***edited summary*** in step (3), (5), (6) in §2.1, we calculated the ROUGE (Lin, 2004) score among the answers provided by different annotators, and found the average ROUGE-1 F-score to be 30.52, 50.96, 71.77, respectively. Lastly, for the evidence in step (4), we consider two sentences as equivalent if the ROUGE-1 score between them is above 90. We found the match rate among different annotators to be 0.4403.

3 DEFACTO Analysis

With the collected annotations, we further split the data collected on the validation set of XSum dataset into a training set and a validation set for the following experiments. We perform data analyses with different aspects of the collected dataset. The basic dataset statistics are in Tab. 1. Out of all the examples, **71.1%** of them contain at least one factual error, **58.8%** of them contain *extrinsic errors*, **22.0%** of them contain *intrinsic errors*, and **9.63%** of them contain *both* types of errors.

[4] Following Goyal and Durrett (2021), we define **intrinsic errors** as errors that *arise as a result of misinterpreting information from the source article* and **extrinsic errors** as errors that *hallucinate new information or facts not present in the source article*.

[5] https://www.mturk.com/

	DAE	QAFactEval
Reference	0.6176	1.549
System-output	0.6904	1.826
Human-edited	**0.8975**	**2.540**

Table 2: Automatic factuality scores on the reference summaries, initial system outputs and human edited summaries. The human edited summaries in DEFACTO dataset have significantly higher ($p < 0.01$) factual consistency according to the automatic factuality metrics.

	R1	R2	RL
Ref. v.s. Sys.	48.01	25.54	40.45
Ref. v.s. Human	40.30	18.22	33.86
Sys. v.s. Human	75.79	66.17	74.89

Table 3: Textual similarity. **R1**, **R2**, **RL** stand for the ROUGE-1/2/L F1 scores respectively. **Ref.** denotes reference summaries, **Sys.** denotes initial system outputs, and **Human** denotes the human-edited summaries.

3.1 Edited Summary

For the edited summaries written by the annotators, we evaluate (1) their factual consistency; (2) their textual similarity with either the reference summaries or the initial outputs; (3) other aspects of their intrinsic quality (Grusky et al., 2018; Bommasani and Cardie, 2020).

Factual Consistency To evaluate the factual consistency, we use two automatic metrics, DAE (Goyal and Durrett, 2020) and QAFactEval (Fabbri et al., 2022a), which achieve strong performance on the XSum dataset (Tang et al., 2022).[6] The results are in Tab. 2, showing that the human-edited summaries are more factually consistent than both the reference summaries and initial system outputs.

Text Similarity For textual similarity, we compare human-edited summaries against both the reference summaries and initial system outputs in Tab. 3. We note two observations: (1) There is a high-degree similarity between the initial system outputs and human-edited summaries, indicating that the annotators only made small changes to the initial outputs. (2) Compared with the initial system outputs, the human-edited summaries have lower similarity with the reference summaries, which suggests that the reference summaries and initial system outputs may share similar factual errors, leading to higher textual similarity.

	Coverage	Novelty	Compression
Ref.	0.633	0.851	14.82
Sys.	0.699	0.788	17.84
Human	0.787	0.703	20.61

Table 4: Intrinsic evaluation of summary quality. *Coverage* is negatively correlated with abstractiveness while *Novelty* has a positive correlation. *Compression* is the ratio of the summary length against the article length.

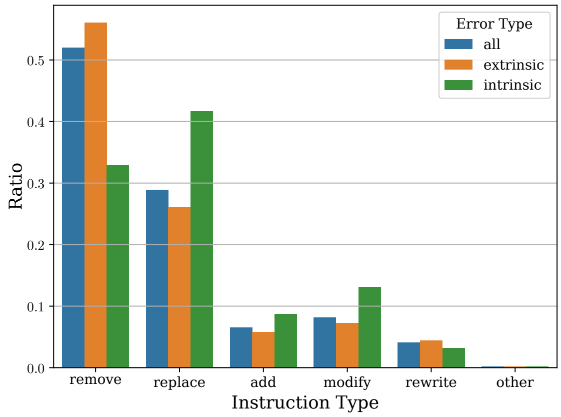


Figure 3: Distribution of six different types of instructions. *removing information* and *replacing information* are the most frequent types. *Extrinsic* errors are more likely to be corrected by *removing* while *Intrinsic* errors are more likely to be corrected by *replacing*.

Intrinsic Evaluation We evaluate two *intrinsic* summary qualities: the ***compression rate*** (Grusky et al., 2018) and the ***abstractiveness*** (Bommasani and Cardie, 2020). In particular, *compression rate* measures the length difference between the input text and the summary. And to evaluate *abstractiveness* we use two features, (1) Extractive Fragment Coverage (Grusky et al., 2018), which measures the extent to which the summary can be "copied" from the input text, (2) Novelty, which measures the ratio of words in the summary that are not in the input text.[7] The statistics in Tab. 4 suggest that the human-edited summaries are less abstractive than the initial system outputs and reference summaries. This finding is coherent with Xiao et al. (2022) which found that there exists a tradeoff between faithfulness and abstractiveness. However, we note that the decrease of abstractiveness can result from removing non-factual information from the summary, which is the most common operation for correct *extrinsic* errors, as we will show next.

[6] Automatic metric setting details are in Appendix B.

[7] More details can be found in Appendix C.1.

3.2 Instructions

The annotators need to provide instructions on how to make changes to the initial system outputs to correct factual errors. We find that the editing can take more than one instruction and the average number of instructions is 1.52. We show the distribution of the number of instructions in Appendix C.2. As for the distribution of instruction types (Fig. 3), we found that ***removing information*** and ***replacing information*** to be the most frequent operations. Interestingly, fine-grained analysis in Fig. 3 shows that *extrinsic* errors are more likely to be corrected by the *replacing operation* while *intrinsic* errors can require more diverse types of operations.

4 Summary Editing for Improving Factual Consistency

With DEFACTO, we propose a new NLG task: editing the initial summary based on human feedback.

4.1 Methods

We formulate the summary editing task as a sequence-to-sequence (Seq2Seq) (Sutskever et al., 2014) problem. Specifically, a Seq2Seq model g learns a mapping from an input sequence X to a target output sequence Y: $Y \leftarrow g(X)$.

For this specific task, the input X has three components: *input document*, *initial system-generated summary* and *human feedback*, while the target output is the *human-edited summary* (Fig. 1). The human feedback consists of the *instructions* and *explanation*. To concatenate the different components of the input sequence, a short "*prompt*" is appended at the beginning of each component, then the entire input sequence becomes: "Article: *input document*; Candidate: *initial system-generated summary*; Instruction: *human instructions*; Explanation: *human explanation*". While recent work (Sanh et al., 2022; Bach et al., 2022) has shown that prompt design can affect the model performance, for simplicity we use simple text snippets for the baseline models.

We instantiate the Seq2Seq model using a family of pre-trained Encoder-Decoder models, T5 (Raffel et al., 2020) and T0 (Sanh et al., 2022), which are widely used for transfer learning and few-shot learning where the data is scarce. To achieve better performance, the model is fine-tuned on the training set of DEFACTO using Maximum Likelihood Estimation (MLE) under the training paradigm of *teacher forcing* (Williams and Zipser, 1989). We note that we only used the subset of data in which the initial system output contains factual errors.

	R1	**R2**	**RL**	**DAE**	**QFE**
Sys.	75.98	66.32	75.05	0.704	1.837
Human.	100	100	100	0.905	2.550
D+S	77.04	67.96	76.03	0.835	2.248
S+I	87.48	81.72	86.16	0.857	2.289
D+S+I	88.74	83.16	87.48	0.904	**2.470**
D+S+E	81.83	74.10	80.36	0.899	2.437
D+S+I+E	**89.22**	**83.64**	**87.92**	**0.911**	2.465

Table 5: *Editing* model performance (T0pp) with different variants of input. **R1**, **R2**, **RL** stand for the ROUGE-1/2/L F1 scores calculated against the human-edited summary. **DAE** is the DAE (Goyal and Durrett, 2020) factuality metric while **QFE** is the the QAFactEval (Fabbri et al., 2022a) metric. **Sys.** denotes initial system outputs, and **Human** denotes the human-edited summaries. **D**, **S**, **I**, **E** stand for input **D**ocument, initial **S**ummary, human-written **I**nstructions, human-written **E**xplanation respectively. The combinations of **D**, **S**, **I**, **E** stand for different input variants.

4.2 Experiments

Implementation Details To initialize the *Editing* model, we use T5-3B and two variants of T0 models, T0-3B and T0pp.[8] We compare the model performance with different variants of input (e.g., with or without the human-written explanation). To evaluate the quality of the model output, we focus on two aspects: *textual similarity* with the human-edited summary, as evaluated by ROUGE (Lin, 2004), and *factual consistency* with the input document, as evaluated by **DAE** (Goyal and Durrett, 2020) and QAFactEval (**QFE**) (Fabbri et al., 2022a). The checkpoints are selected based on their performance on the validation set.

Experimental Results Tab. 5 shows the performance of fine-tuned T0pp with different input variants. We note the following observations: (1) Compared with the initial system-generated summaries, the *Editing* model is able to generate summaries more similar to the human-edited summaries and more factually consistent with the input document. (2) Both the human-written instructions and explanation can provide meaningful guidance to the *Editing* model, and the model with both of them as input (the **D+S+I+E** variant) achieves the best performance. (3) Without the input document, the *Editing* model (the **S+I** variant) can still improve

[8] T5-3B (`https://huggingface.co/t5-3b`), T0-3B (`https://huggingface.co/bigscience/T0_3B`), and T0pp (`https://huggingface.co/bigscience/T0pp`) have around 3, 3, and 11 billion parameters respectively.

	T0pp				T0-3B				T5-3B			
	R1	**R2**	**DAE**	**QFE**	**R1**	**R2**	**DAE**	**QFE**	**R1**	**R2**	**DAE**	**QFE**
D+S	*77.04*	*67.96*	*0.835*	*2.248*	76.10	66.66	0.821	2.168	75.99	66.35	0.784	2.063
S+I	*87.48*	*81.72*	*0.857*	2.289	87.30	81.00	0.852	*2.263*	87.59	81.50	0.844	2.237
D+S+I	*88.74*	*83.16*	*0.904*	2.470	88.36	82.16	0.894	*2.489*	86.42	80.56	0.876	2.411
D+S+E	*81.83*	*74.10*	*0.899*	2.437	79.85	71.41	0.902	***2.510***	79.09	71.20	0.877	2.373
D+S+I+E	***89.22***	***83.64***	***0.911***	2.465	88.69	82.44	0.899	*2.477*	87.03	80.77	0.865	2.375

Table 6: Performance with different variants of the *Editing* model. **R1**, **R2** stand for the ROUGE-1/2 F1 scores. **DAE** is from Goyal and Durrett (2020) while **QFE** is from Fabbri et al. (2022a). **D**, **S**, **I**, **E** stand for input **D**ocument, initial **S**ummary, human-written **I**nstructions, human-written **E**xplanation respectively, of which the combinations stand for different input variants. The best results with each input variant are *italicized*.

the initial system-generated summaries by following the instructions. However, taking the input document as part of the input helps the model (the **D+S+I** variant) to achieve better performance, especially for better factual consistency.

In Tab. 6, we compare the performance of T0pp, T0-3B and T5-3B with different kinds of inputs. We found that (1) the findings on T0pp (Tab. 5) are generalizable to T0-3B and T5-3B with few exceptions. (2) T0pp outperforms T0-3B across different input variants according to different automatic metrics except for the QAFactEval metric. (3) T0-3B generally outperforms T5-3B, likely thanks to the pre-training of T0 which is designed for performing zero-shot learning with instructions.

Human Evaluation We conduct a human evaluation on the quality of model-edited summaries. We ask the annotators two questions: (1) Are the generated summaries more factually consistent than the original summaries (yes/no); (2) Do the generated summaries follow the instructions (yes/partly/no). We randomly sampled 100 examples from the test set, and have each generated summary annotated by three MTurk annotators. The generated summaries are from the trained checkpoint of T0pp with the input containing input document, initial system-generated summaries, and human-written instructions. Under major voting (with ties ignored), we found that 97% of model-edited summaries are more factually consistent than the original system outputs, and 91% of them follow the provided human-written instructions.

4.3 Case Study of LLM Summary Editing

As a case study, we evaluate the zero-shot learning ability of GPT-3.5[9] for summary editing. We apply it to two settings, (1) editing without instructions and (2) editing by following instructions, in a zero-shot learning manner.[10] The results in Tab. 7 show that (1) GPT-3.5 is able to leverage the editing instructions; (2) Compared with the fine-tuned model (T0pp), GPT-3.5 can generate edited summaries with higher factual consistency but it is worse at maintaining the content similarity with the original summary, which suggests that it still struggles with *controllable* text generation.

Model	**Input**	**R1**	**R2**	**DAE**	**QFE**
Sys.	-	75.98	66.32	0.704	1.837
Human.	-	100	100	0.905	2.550
T0pp	**D+S**	77.04	67.96	0.835	2.248
T0pp	**D+S+I**	88.74	83.16	0.904	2.470
GPT-3.5	**D+S**	36.75	21.98	0.892	2.351
GPT-3.5	**D+S+I**	72.22	60.53	0.910	2.651

Table 7: Performance of GPT-3.5 for summary editing. **R1**, **R2** stand for the ROUGE-1/2 F1 scores calculated against the human-edited summary. **Sys.** denotes initial system outputs, and **Human** denotes the human-edited summaries. **D**, **S**, **I** stand for input **D**ocument, initial **S**ummary, human-written **I**nstructions.

5 Generating Feedback for Improving Factual Consistency

We investigate if it is possible to train a model to generate feedback from a given document and summary pair to correct factual errors, and we name the subsequent model as a *Critic* model.

5.1 Methods

Similarly to §4.1, we formulate the *Critic* model as a Seq2Seq model. The input sequence is a concatenation of the *input document* and the *initial system-generated summary* while the target output is the *human-written instructions* (Fig. 1).[11] We use

[9] OpenAI's `gpt-3.5-turbo-0301`: `https://platform.openai.com/docs/models/gpt-3-5`.

[10] The prompts we used can be found in Appendix D.2.

[11] While it is also possible to require the *Critic* model to generate the explanation, we choose human-written instructions as the target output as it works better at helping the *Editing* model to improve the initial outputs (§4.2).

	Rouge1	Rouge2	RougeL
T0pp	52.55	37.41	51.00
T0-3B	51.70	36.56	50.33

Table 8: *Critic* model performance with respect to the textual similarity between the system output and human-written instructions.

Method	Critic	R1	R2	DAE	QFE
Sys.	-	75.98	66.32	0.704	1.837
Human.	-	100	100	0.905	2.550
D+S	-	77.04	67.96	0.835	2.248
D+S+I	-	88.74	83.16	0.904	2.470
D+S+I*	T0pp	75.10	65.15	0.859	2.296
D+S+I*	T0-3B	73.01	62.15	0.859	2.278

Table 9: *Editing* model performance (T0pp) with the instructions generated by the *Critic* model. **R1**, **R2** stand for the ROUGE-1/2 F1 scores calculated against the human-edited summary. **Sys.** denotes initial system outputs, and **Human** denotes the human-edited summaries. **D**, **S**, **I** stand for input **D**ocument, initial **S**ummary, human-written **I**nstructions. **I*** stands for the instructions generated by the *Critic* model.

T0 as the startpoint to fine-tune the *Critic* model with MLE training on the subset of DEFACTO in which the initial summary contains factual errors.

5.2 Experiments

Experimental Results Tab. 8 shows the textual similarity between the instructions generated by the *Critic* model and the human-written instructions. To have a more intuitive understanding of the model performance, in Tab. 9 we evaluate the performance of the *Editing* model with the instructions generated by the *Critic* model. We found that (1) While the generated instructions cannot work as well as the human-written instructions, they are helpful to the *Editing* model to improve the factual consistency of the initial system-generated summaries. (2) Compared with the *Editing* model that only takes the input document and initial summary as input (**D+S**), the *Editing* model (**D+S+I***) that also uses the generated instructions achieves better performance with respect to factual consistency, but its outputs have lower textual similarity with the human-edited summaries. It indicates that the *Critic* model can generate useful instructions. Meanwhile, the lower textual similarity may result from the fact that there can be more than one way to correct the factual errors,[12] and the *Critic* model can generate instructions for a way of correction different from the human-edited summary.

Method	Critic	R1	R2	DAE	QFE
Sys.	-	75.98	66.32	0.704	1.837
Human.	-	100	100	0.905	2.550
D+S+I*	T0pp	75.10	65.15	0.859	2.296
D+S+I*	GPT-3.5	60.48	48.18	0.868	2.566
D+S+I*	GPT-4	63.60	51.15	0.860	2.604

Table 10: Case study of instruction generation with LLMs. Instructions generated by the *Critic* models are used to instruct the *Editing* model.

Human Evaluation To further evaluate the quality of model-generated instructions, we ask human annotators two questions: (1) Are the generated instructions equivalent to human-written instructions (yes/partly/no); (2) Are the generated instructions useful for correcting the factual errors (yes/partly/no). Similar to §4.2, we randomly sampled 100 examples from the test set, and have each generated instruction annotated by three MTurk annotators. For the first question, we found that the annotators think 24% of the generated instructions are *exactly* equivalent to the human-written instructions while 45% of them are *partly* equivalent. For the second question, we found that 39% of the generated instructions are useful for correcting factual errors while 31% of them are partly useful. As a result, we found that it is easier for the *Critic* model to generate useful instructions than generating instructions that are equivalent to human-written instructions. We hypothesize this is because there can be more than one way to edit the initial summary therefore the human-written instructions represent only one acceptable solution.

5.3 Case Study of LLM Critic

As a case study, we evaluate the zero-shot learning ability of GPT-3.5 and GPT-4[13] for instruction generation. The results in Tab. 10 show that, compared with fine-tuned models, instructions generated by both GPT-3.5 and GPT-4 lead the editing model to generate summaries that are more factual but less similar to the original summaries. This finding shows a similar trend as in §4.3, that LLMs in a zero-shot learning setting lack the ability of *con-*

[12] For example, one may choose to *remove* a factual error or *replace* the error with factual information when appropriate.

[13] OpenAI's `gpt-4-0314`: `https://platform.openai.com/docs/models/gpt-4`.

	Method	R1	R2	DAE	QFE
	Sys.	75.98	66.32	0.704	1.837
	Human.	100	100	0.905	2.550
T0pp	Editing	77.04	67.96	0.835	2.248
	Editor$_I$	78.01	**69.01**	0.804	2.108
	Editor$_E$	**78.46**	68.70	**0.867**	**2.309**
T0-3B	Editing	76.10	66.66	0.821	2.168
	Editor$_I$	**77.40**	**68.29**	0.808	2.112
	Editor$_E$	77.27	67.92	**0.838**	**2.241**
T5-3B	Editing	75.99	66.35	0.784	2.063
	Editor$_I$	**77.06**	**67.86**	**0.804**	2.106
	Editor$_E$	76.82	67.42	0.796	**2.114**

Table 11: *Editor* model performance. **R1**, **R2** stand for the ROUGE-1/2 F1 scores calculated against the human-edited summary. **Sys.** denotes initial system outputs, and **Human** denotes the human-edited summaries. *Editing* is the model in §4 with only the input document and initial system-generated summary as input. Editor$_I$ is the *Editor* model that generates both the instructions and edited summary, while Editor$_E$ is the one that generates both the explanation and edited summary.

trollable text generation. For example, GPT-3.5 responded with "No editing instructions needed" 23.9% of the time, despite being directly instructed to edit a factually inconsistent summary.[14]

6 Summary Editor with Feedback Generation and Editing

We define the third NLG task as to predict both the ***human feedback*** and the ***edited summary*** given the input document and initial system-generated summary. We name this model the ***Editor*** model because it needs to both evaluate the initial summary and make edits according to its own assessments.

6.1 Correcting Known Factual Errors

Similar to §4.1 and §5.1, we fine-tuned the pre-trained T0 and T5 models for our experiments. The two parts of the target output, the human feedback, and the edited summary are indicated by textual tags as specified in §4.1. We investigate two specific scenarios: (1) generating both the *instructions* and the edited summary; (2) generating both the *explanation* and the edited summary.

We present the experimental results in Tab. 11. Compared with the *Editing* model that takes only the input document and the initial system-generated summary as the input, the *Editor* models have better performance in textual similarity, and the one that generates the explanations also achieves higher factual consistency. The results suggest that learning to predict related information of a target generation task can be beneficial to the performance of language generation models, echoing the recent findings in chain-of-thought prompting (Wei et al., 2022; Huang et al., 2022; Jung et al., 2022).

System	R1	R2	RL	DAE	QFE
Pegasus	47.35	24.61	39.59	0.763	2.029
Human	41.94	19.49	34.97	0.905	2.550
CCGS	45.11	21.06	36.60	0.760	1.847
CLIFF	**46.40**	**23.38**	**38.38**	0.780	2.068
ReDRESS	43.50	19.77	35.28	0.830	2.065
FactPegasus	38.95	15.99	31.68	**0.882**	1.941
CompEdit	42.69	19.06	34.73	0.850	2.113
Editor	45.14	22.27	37.89	0.833	**2.250**

Table 12: *Editor* model performance (T0-3B) on the *entire* DEFACTO test set. **R1**, **R2**, **RL** stand for the ROUGE-1/2/L F1 scores calculated against the *reference* summary. **DAE** is the DAE (Goyal and Durrett, 2020) factuality metric while **QFE** is QAFactEval (Fabbri et al., 2022a). The initial system outputs are from **Pegasus**, and **Human** are the human-edited summaries.

6.2 Detecting and Correcting Factual Errors

While in the previous experiments the models are trained to edit the initial system outputs with known factual errors, the *Editor* model can also be used on *arbitrary* system outputs where it is required to edit the initial output *only* when it identifies factual errors in it. To this end, we use the entire DEFACTO in this experiment with the following modifications to the target output: (1) the target summary is set to the original system output when it contains no factual errors, and to the human-edited summary otherwise; (2) only explanations are used as part of the target output because it is always available.

We fine-tune T0-3B in this experiment and compare its results with several recently introduced summarization systems that are specifically trained to improve the summary factual consistency: (1) CCGS (Chen et al., 2021), (2) CLIFF (Cao and Wang, 2021), (3) ReDRESS (Adams et al., 2022), (4) FactPegasus (Wan and Bansal, 2022), (5) CompEdit (Fabbri et al., 2022b). More details about these systems can be found in Appendix D.1.

The results in Tab. 12 show that the *Editor* can achieve competitive performance compared with the baseline systems and yield a balanced performance between the *content similarity* with the reference summary and the *factuality consistency*. Since the *Editor* model is trained on much fewer data than

[14] Prompts and more details are in Appendix D.3.

the other systems, its strong performance indicates the effectiveness of utilizing human demonstrations and feedback for improving factual consistency.

7 Related Work

Factual Consistency in Text Summarization Factual consistency is an important quality of text summarization systems (Kryscinski et al., 2020; Maynez et al., 2020; Fabbri et al., 2021). Related work has proposed various methods of improving the factual consistency of summaries by (1) training abstractive summarization models with factuality metrics (Goyal and Durrett, 2021; Cao et al., 2022), (2) introducing new training objectives and multi-task learning for model training (Cao and Wang, 2021; Zhu et al., 2021; Aralikatte et al., 2021; Zhang et al., 2022; Xu and Zhao, 2022), (3) post-editing or re-ranking the initially generated summaries to improve the factual consistency (Cao et al., 2020; Chen et al., 2021; Balachandran et al., 2022; Fabbri et al., 2022b), (4) designing factuality-aware pre-training (Wan and Bansal, 2022).

To facilitate the evaluation of summarization models and automatic factuality metrics that evaluate the factual consistency of summaries, various benchmark datasets have been collected by the related work (Kryscinski et al., 2020; Wang et al., 2020; Huang et al., 2020; Fabbri et al., 2021; Goyal and Durrett, 2021; Pagnoni et al., 2021). In these benchmarks, system-generated summaries are evaluated by human annotators with either numerical quality scores, binary labels, or binary labels with fine-grained error taxonomies. In contrast, our dataset contains more detailed human feedback with *natural language descriptions* and provides error-free, human-edited summaries.

Neural Text Editing Neural text editing models (Malmi et al., 2022) are suitable for application scenarios where there is a significant textual overlap between the input and output sequences (Awasthi et al., 2019; Malmi et al., 2019; Stahlberg and Kumar, 2020; Mallinson et al., 2020; Reid and Zhong, 2021; Mallinson et al., 2022), such as grammatical error correction, text simplification, and text style transfer. Instead of autoregressive generation, text editing can also be achieved by predicting and performing edit operations (Stahlberg and Kumar, 2020; Mallinson et al., 2020) or through non-autoregressive text generation (Gu et al., 2019; Agrawal and Carpuat, 2022). Unlike most of the related work, we propose a text editing task that requires the editing models to follow the editing instructions. Faltings et al. (2021) introduces a similar dataset as ours containing single-sentence edits and the associated natural language commands crawled from Wikipedia. However, our dataset is different from theirs as we define a specific target quality, summary factual consistency, for the text edits and instructions.

Improving Neural Models through Human Feedback Leveraging human feedback to improve neural models has become a recent research focus. InstructGPT3 (Ouyang et al., 2022a) use human feedback to improve initial predictions from a GPT3 model for better user preference alignments. Madaan et al. (2021) propose the interactive MERCURIE system, where users interactively correct the explanations generated by a reasoning system. In Xu et al. (2022), a generic chatbot is continuously trained using various forms of human feedback including natural language comments. Schick et al. (2022) propose a collaborative language model, PEER, which imitates a draft writing process and interactively refines a language generation task through human feedback. For text summarization, prior works (Stiennon et al., 2020; Wu et al., 2021; Nguyen et al., 2022; Scheurer et al., 2022) have studied training summarization models through human feedback in the form of numerical scores of summary quality and thus different from natural language feedback used in our work.

8 Conclusions

Using summary factual consistency as a target quality, we study improving text generation with human demonstrations and feedback. We demonstrate the usages of human feedback in three proposed NLG tasks using the collected dataset, DeFacto, and show that human feedback can be used to improve summary factual consistency. We believe that our proposed tasks can be extended to other important text qualities beyond factual consistency, and utilizing natural language feedback for improving text generation can be a promising path for future work.

Acknowledgements

We thank the anonymous reviewers for their valuable feedback and helpful suggestions.

Limitations

The annotation task we proposed in this work, i.e., detecting factual errors in summaries and providing human demonstrations and feedback for correcting the identified errors, can be complicated and time-consuming. During our recruiting phase for MTurk annotators, we found that the ratio of annotators who were qualified after finishing the qualification test was relatively low. Therefore, it can be difficult to scale up the annotated dataset given the time and budget limitations. As a result, our dataset is of a relatively small scale and we only used one summarization dataset (XSum) and one base summarization model (Pegasus).

In this work, we view summary factual consistency as an example of user-expected quality to study leveraging natural language feedback for aligning system outputs with user preferences. However, user preferences can be diverse and personal and some user-expected output quality will be less well-defined and objective than summary factual consistency, which further increases the difficulty and ambiguity of data annotation and model evaluation. Therefore, it can be challenging to directly apply the methods we proposed in this work to such subjective quality aspects, and we leave it for future work to explore generalizing our methods to more diverse user expectations and preferences.

References

Griffin Adams, Han-Chin Shing, Qing Sun, Christopher Winestock, Kathleen McKeown, and Noémie Elhadad. 2022. Learning to revise references for faithful summarization. *arXiv preprint arXiv:2204.10290*.

Sweta Agrawal and Marine Carpuat. 2022. An imitation learning curriculum for text editing with non-autoregressive models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7550–7563, Dublin, Ireland. Association for Computational Linguistics.

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

Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. 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 4260–4270, Hong Kong, China. Association for Computational Linguistics.

Stephen Bach, Victor Sanh, Zheng Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-david, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Fries, Maged Al-shaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Dragomir Radev, Mike Tian-jian Jiang, and Alexander Rush. 2022. PromptSource: An integrated development environment and repository for natural language prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 93–104, Dublin, Ireland. Association for Computational Linguistics.

Vidhisha Balachandran, Hannaneh Hajishirzi, William W. Cohen, and Yulia Tsvetkov. 2022. Correcting diverse factual errors in abstractive summarization via post-editing and language model infilling.

Rishi Bommasani and Claire Cardie. 2020. Intrinsic evaluation of summarization datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8075–8096, Online. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

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

Meng Cao, Yue Dong, Jiapeng Wu, and Jackie Chi Kit Cheung. 2020. Factual error correction for abstractive summarization models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural*

Language Processing (EMNLP), pages 6251–6258, Online. Association for Computational Linguistics.

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.

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

Alexander Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022a. 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.

Alexander R. Fabbri, Prafulla Kumar Choubey, Jesse Vig, Chien-Sheng Wu, and Caiming Xiong. 2022b. Improving factual consistency in summarization with compression-based post-editing. *ArXiv*, abs/2211.06196.

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.

Felix Faltings, Michel Galley, Gerold Hintz, Chris Brockett, Chris Quirk, Jianfeng Gao, and Bill Dolan. 2021. Text editing by command. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5259–5274, Online. 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*, pages 3592–3603, Online. Association for Computational Linguistics.

Tanya Goyal and Greg Durrett. 2021. Annotating and modeling fine-grained factuality in summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1449–1462, 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.

Jiatao Gu, Changhan Wang, and Jake Zhao Junbo. 2019. *Levenshtein Transformer*. Curran Associates Inc., Red Hook, NY, USA.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1693–1701, Cambridge, MA, USA. MIT Press.

Dandan Huang, Leyang Cui, Sen Yang, Guangsheng Bao, Kun Wang, Jun Xie, and Yue Zhang. 2020. What have we achieved on text summarization? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 446–469, Online. Association for Computational Linguistics.

Fan Huang, Haewoon Kwak, and Jisun An. 2022. Chain of explanation: New prompting method to generate higher quality natural language explanation for implicit hate speech. *ArXiv*, abs/2209.04889.

Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. 2022. Maieutic prompting: Logically consistent reasoning with recursive explanations. *ArXiv*, abs/2205.11822.

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability.

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.

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.

Aman Madaan, Niket Tandon, Dheeraj Rajagopal, Yiming Yang, Peter Clark, Keisuke Sakaguchi, and Eduard H. Hovy. 2021. Improving neural model performance through natural language feedback on their explanations. *CoRR*, abs/2104.08765.

Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2022. Edit5: Semi-autoregressive text-editing with t5 warm-start.

Jonathan Mallinson, Aliaksei Severyn, Eric Malmi, and Guillermo Garrido. 2020. FELIX: Flexible text editing through tagging and insertion. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1244–1255, Online. Association for Computational Linguistics.

Eric Malmi, Yue Dong, Jonathan Mallinson, Aleksandr Chuklin, Jakub Adamek, Daniil Mirylenka, Felix Stahlberg, Sebastian Krause, Shankar Kumar, and Aliaksei Severyn. 2022. Text generation with text-editing models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorial Abstracts*, pages 1–7, Seattle, United States. Association for Computational Linguistics.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. 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 5054–5065, Hong Kong, China. 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.

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.

Duy-Hung Nguyen, Nguyen Viet Dung Nghiem, Bao-Sinh Nguyen, Dung Tien Tien Le, Shahab Sabahi, Minh-Tien Nguyen, and Hung Le. 2022. Make the most of prior data: A solution for interactive text summarization with preference feedback. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1919–1930, Seattle, United States. 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, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. 2022a. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155.

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

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.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Machel Reid and Victor Zhong. 2021. LEWIS: Levenshtein editing for unsupervised text style transfer. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3932–3944, Online. Association for Computational Linguistics.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Jérémy Scheurer, Jon Ander Campos, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. 2022. Training language models with language feedback.

Timo Schick, Jane Dwivedi-Yu, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave, and Sebastian Riedel. 2022. Peer: A collaborative language model.

Felix Stahlberg and Shankar Kumar. 2020. Seq2Edits: Sequence transduction using span-level edit operations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5147–5159, Online. 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.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.

Liyan Tang, Tanya Goyal, Alexander R. Fabbri, Philippe Laban, Jiacheng Xu, Semih Yahvuz, Wojciech Kryscinski, Justin F. Rousseau, and Greg Durrett. 2022. Understanding factual errors in summarization: Errors, summarizers, datasets, error detectors. *ArXiv*, abs/2205.12854.

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.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Ronald J. Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural Comput.*, 1(2):270–280.

Jeff Wu, Long Ouyang, Daniel M. Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. 2021. Recursively summarizing books with human feedback.

Yang Xiao, Jinlan Fu, Weizhe Yuan, Vijay Viswanathan, Zhoumianze Liu, Yixin Liu, Graham Neubig, and Pengfei Liu. 2022. DataLab: A platform for data analysis and intervention. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 182–195, Dublin, Ireland. Association for Computational Linguistics.

Jing Xu, Megan Ung, Mojtaba Komeili, Kushal Arora, Y-Lan Boureau, and Jason Weston. 2022. Learning new skills after deployment: Improving open-domain internet-driven dialogue with human feedback.

Wang Xu and Tiejun Zhao. 2022. Jointly learning guidance induction and faithful summary generation via conditional variational autoencoders. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2340–2350, Seattle, United States. Association for Computational Linguistics.

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

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Chenguang Zhu, William Hinthorn, Ruochen Xu, Qingkai Zeng, Michael Zeng, Xuedong Huang, and Meng Jiang. 2021. Enhancing factual consistency of abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 718–733, Online. Association for Computational Linguistics.

A Data Collection Details

We used the XSum dataset for our data collection. It is released under the Apache 2 license and contains news articles written in English.

A.1 Instruction Templates

We provide six templates for the annotators corresponding to different operations: *Remove*, *Add*, *Replace*, *Modify*, *Rewrite*, *Others*:

(1) ***Remove*** the information about __ from the summary.

(2) ***Add*** the information about __ to the summary.

(3) ***Replace*** the information about __ *with* the information about __.

(4) ***Modify*** the information about __ in the summary.

(5) ***Rewrite*** the summary *entirely* by __.

(6) ***Other*** instructions: __.

We note that sometimes it takes more than one instruction to edit the original summary.

A.2 Annotated Examples

We provide the following annotated examples.

Example 1

Original Summary: A Wirral biscuit factory is to close with the loss of 342 jobs.

Explanation: Location is in Moreton (Morton?), not Wirral, and 342 families will be affected but that technically doesn't translate to 342 jobs.

Instruction: Replace the information about Wirral with the information about Moreton. Replace the information about loss of 342 jobs with the information about affect 342 families.

Edited Summary: A Moreton biscuit factory is to close, affecting 342 families.

Example 2

Original Summary: Two teenage girls have appeared at Teesside Crown Court accused of murdering a woman in Middlesbrough.

Explanation: The Teesside Crown Court was not mentioned by name, only the Youth court. The woman was found in Stephen Street and not Midlesbrough.

Instruction: Replace the information about Middlesbrough with the information about Stephen Street. Replace the information about Teesside Crown Court with the information about Teesside Youth Court.

Edited Summary: Two teenage girls have appeared at Teesside Youth Court accused of murdering a woman in Stephen Street.

Example 3

Original Summary: Michael O'Halloran believes St Johnstone manager Tommy Wright will get the best out of him following his release by Rangers.

Explanation: the first name info in summary is not found in the source text. St Johnstone info is also not mentioned in the source.

Instruction: Remove the information about the first names of both people from the summary. Remove the information about Wright being the St Johnstone manager from the summary.

Edited Summary: O'Halloran believes Wright will get the best out of him following his release by Rangers.

Example 4

Original Summary: Aberdeen's Royal Concert Hall is to sell off hundreds of items of memorabilia as part of building work.

Explanation: The source text doesn't state the name of the concert hall.

Instruction: Replace the information about Aberdeen's Royal Concert Hall with the information about Aberdeen Performing Arts.

Edited Summary: Aberdeen Performing Arts is to sell off hundreds of items of memorabilia as part of building work.

Example 5

Original Summary: Lancashire County Council's decision to stop composting waste has been criticised as "catastrophic".

Explanation: The original summary strongly implies that the decision to stop composting was "catastrophic" but the original text strongly implies that the composting program itself was a catastrophic failure versus stopping the program.

Instruction: Replace the information about the stoppage of the composting program being catastrophic with the information about how the composting program was a catastrophic failure.

Edited Summary: Lancashire County Council's decision to stop composting waste shows the program has been a "catastrophic" failure.

Example 6

Original Summary: Alex Goode and Ollie Devoto have been called up to England's Six Nations training squad.

Explanation: The source text does not mention the name of England's training squad.

Instruction: Remove the information about the name of England's training squad from the summary.

Edited Summary: Alex Goode and Ollie Devoto have been called up to England's training squad.

B Factuality Metrics Setting

We use two factuality metrics, DAE (Goyal and Durrett, 2020) and QAFactEval (Fabbri et al., 2022a), in our experiments. For DAE, we transfer its *dependency-level* predictions into *summary-level* scores. Specifically, following the notation of Goyal and Durrett (2021), we use $d(S)$ to denote the dependency-parse of a summary S. For each arc a in $d(S)$, the DAE metric predicts a label y_a representing if this dependency is entailed by the input document ($y_a = 1$ means entailment.) Then, we define a summary-level factuality score f_S using the predictions:

$$f_S = \frac{\sum_a y_a}{|d(S)|}. \quad (1)$$

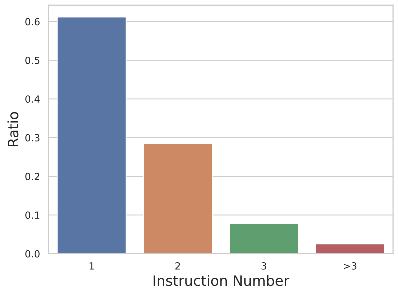


Figure 4: Number of instructions to correct a factually inconsistent system-generated summary.

For QAFactEval, we directly use its prediction scores since they are summary-level scores.

C DEFACTO Analysis

C.1 Intrinsic Summary Evaluation

In §3.1, we use three metrics to evaluate the summary's intrinsic quality.

(1) Compression rate (Grusky et al., 2018) is defined as the ratio of the number of words in the input document D and in the summary S:

$$\text{COMPRESSION}(D,S) = \frac{|D|}{|S|}. \qquad (2)$$

(2) Extractive Fragment Coverage (Grusky et al., 2018) is defined with *extractive fragments* (Grusky et al., 2018), $F(D,S)$, which is a set of word sequences shared between the input document D and the summary S. Then, the Extractive Fragment Coverage is defined as

$$\text{COVERAGE}(D,S) = \frac{1}{|S|}\sum_{f\in F(D,S)} |f|. \qquad (3)$$

(3) We define a word-level novelty between the input document D and the summary S as

$$\text{NOVELTY} = 1 - \frac{|D \cap S|}{|S|}. \qquad (4)$$

C.2 Human-written Instructions

We find that it can take more than one instruction to perform the summary editing. Fig.4 shows the distribution of the number of instructions.

D Experimental Details

We use T5-3B and two variants of T0 models, T0-3B and T0pp in our experiments.[15] For the 3B models, it takes one 40GB GPU to train the model, and the training time is around 8 hours. For the 11B models, it takes eight 32GB GPUs to train the model, and the training time is around 20 hours. All the experiments converged in 50 epochs.

D.1 Baseline Summarization Systems

In §6.2, we compare the performance of ***Editor*** model with the following summarization systems:
(1) CCGS (Chen et al., 2021), which is based on contrastive candidate generation and selection.
(2) CLIFF (Cao and Wang, 2021), which is trained with contrasting learning and synthetically generated contrastive examples.
(3) ReDRESS (Adams et al., 2022), which is a summary post-editor that learns to remove factual errors through contrastive learning.
(4) FactPegasus (Wan and Bansal, 2022), which is pre-trained and fine-tuned with factual-consistency-aware training objectives.
(5) CompEdit (Fabbri et al., 2022b), which is a compression-based post-editing model that removes the non-factual entities from the original summary by performing summary compression.

D.2 Setting of LLM Case Study for Summary Editing

We use GPT-3.5 for the summary editing experiment with LLMs. To ensure stable results, we set the sampling temperature to 0. The prompt for summary editing *without* instructions is as follows:

> You will be given an article and a summary of the article, which is not factually consistent with the article. That is, the summary contains information that is not supported by the article.
>
> Your task is to edit the summary to make it factually consistent with the article. The correction should preserve most of the summary and only adapt it. Please only make the necessary changes to the summary. However, if you find all the information in the summary is not correct, please write a new summary of the entire article instead.
>
> The edit operations are: 1. Remove Information, 2. Add Information, 3. Replace Information, 4. Modify Information, 5. Rewrite Summary 6. Others.
>
> The summary should contain only one sentence. Please keep the style of the

[15] T5-3B (`https://huggingface.co/t5-3b`), T0-3B (`https://huggingface.co/bigscience/T0_3B`), and T0pp (`https://huggingface.co/bigscience/T0pp`) have around 3, 3, and 11 billion parameters respectively.

> summary unchanged, and the length of the summary should be similar before and after your edits.
>
> Article: {{Article}}
>
> Summary: {{Summary}}
>
> Please edit the summary accordingly:

The prompt for summary editing *with* instructions is as follows:

> You will be given an article and a summary of the article, which is not factually consistent with the article. That is, the summary contains information that is not supported by the article.
>
> You will be given instructions about how to edit the summary to make it factually consistent with the article. Your task is to follow the instructions and edit the summary accordingly. The correction should preserve most of the summary and only adapt it. Please only make necessary changes to the summary, and keep the summary length close to the original length. The summary should contain only one sentence.
>
> Article: {{Article}}
>
> Summary: {{Summary}}
>
> Instructions: {{Instruction}}
>
> Please edit the summary accordingly:

D.3 Setting of LLM Case Study for Instruction Generation

The prompt we used for instruction generation is as follows:

> You will be given an article and a summary of the article, which is not factually consistent with the article. That is, the summary contains information that is not supported by the article.
>
> Your task is to generate instructions for editing the summary to make it factually consistent with the article. The correction should preserve most of the summary and only adapt it. Please only make the necessary changes to the summary.
>
> The edit operations are: 1. Remove Information, 2. Add Information, 3. Replace Information, 4. Modify Information, 5. Rewrite Summary 6. Others. Please note that “Remove Information” and “Replace Information” should be the majority of the operations. “Add Information” comes next.
>
> The summary should contain only one sentence. Please keep the style of the summary unchanged, and the length of the summary should be similar before and after the editing.
>
> Example:
>
> Summary: A coalition of US civil rights groups has called on Facebook to do more to protect black people from racist abuse on the social network, saying the site is "racially biased".
>
> The summary is not factually consistent because the source text does not contain the information called on Facebook to do more to protect black people, which is stated in the summary.
>
> Instructions: Replace the information about the claim of the coalition with information about better moderation on the platform.
>
> More instruction examples (summaries omitted):
>
> - “Instruction 1”
> - “Instruction 2”
> - ...
>
> Input:
>
> Article: {{Article}}
>
> Summary: {{Summary}}
>
> Please generate the editing instructions for the summary to make it factually consistent with the article:

As discussed in §5.3, LLMs still lack the ability of *controllable* instruction generation. For example, GPT-3.5 responded with “No editing instructions needed” 23.9% of the time, despite being directly instructed to edit a factually inconsistent summary. Conversely, GPT-4 only made such mistakes in 1.8% of examples.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 9

☑ A2. Did you discuss any potential risks of your work?
Section 9

☑ A3. Do the abstract and introduction summarize the paper's main claims?
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 2-6.

☑ B1. Did you cite the creators of artifacts you used?
Section 1

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Yes. We described the license of XSum dataset in the Appendix. We will add descriptions of the license for the dataset we create upon acceptance.

☒ 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)?
No, due to time constraints, we will specify the data release license upon acceptance. Our use case is research-based and consistent with the underlying licenses.

☒ 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 benchmark datasets that have been widely used so we expect the risk of their containing offensive or personal information is relatively low.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
In Appendix A.

☑ 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 2.

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 4, 5, 6.

☑ 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 4 and Appendix D

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
In Section 4, 5, 6, Appendix D, and we provide the hyperparameter settings in the training scripts, which is included in the supplementary material.

☒ 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?
We only report the result one time since we only report the baseline model results for our collected dataset.

☑ 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.)?
Appendix B.

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

Section 2.

☑ 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 A and supplementary materials.

☑ 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 2.

☒ 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?
The annotators had signed consent before they accepted the annotation asks. Due to anonymity concerns, we will release the details of the consent upon acceptance.

☒ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
We use standard benchmark datasets that have been widely used so we expect the risk of their containing offensive or personal information is relatively low.

☑ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Section 2.