

Improving Factual Consistency in Summarization with Compression-Based Post-Editing

Alexander R. Fabbri Prafulla Kumar Choubey Jesse Vig
Chien-Sheng Wu Caiming Xiong

Salesforce AI Research

{afabbri, pchoubey, jvig, wu.jason, cxiong}@salesforce.com

Abstract

State-of-the-art summarization models still struggle to be factually consistent with the input text. A model-agnostic way to address this problem is post-editing the generated summaries. However, existing approaches typically fail to remove entity errors if a suitable input entity replacement is not available or may insert erroneous content. In our work, we focus on removing extrinsic entity errors, or entities not in the source, to improve consistency while retaining the summary’s essential information and form. We propose to use sentence-compression data to train the post-editing model to take a summary with extrinsic entity errors marked with special tokens and output a compressed, well-formed summary with those errors removed. We show that this model improves factual consistency while maintaining ROUGE, improving entity precision by up to 30% on XSum, and that this model can be applied on top of another post-editor, improving entity precision by up to a total of 38%. We perform an extensive comparison of post-editing approaches that demonstrate trade-offs between factual consistency, informativeness, and grammaticality, and we analyze settings where post-editors show the largest improvements.

1 Introduction

Text summarization aims to compress a long document(s) into a short and fluent form that preserves salient information. State-of-the-art models, however, are often not factually consistent with the input they are conditioned on (Maynez et al., 2020; Fabbri et al., 2022). While recent modeling techniques have been proposed to improve factual consistency (Nan et al., 2021; Kang and Hashimoto, 2020; Aralikkatte et al., 2021), a model-agnostic approach is to post-edit the summaries.

Prior work in post-editing for factual consistency has focused on swapping inconsistent entities with those in the input (Dong et al., 2020; Lee et al., 2022), including by reranking the entity-replaced

Source: The oil price collapse sent global markets reeling throughout 2015. ... Brent crude oil was up 3% at \$37.60 per barrel for the day but down 35% over the year.	
System	Summary
Original	Wall Street markets closed lower on the last trading day of 2015 as oil prices languished at \$28 a barrel for much of the year.
Entity Swap	No Change
Rewrite	Wall Street markets closed lower on the last trading day of 2015 as oil prices languished at \$37.60 per barrel ... but remain down 35% over the year.
Delete	Wall Street markets closed lower on 2015 as oil prices languished at a for much of the year.
Compress (ours)	Wall Street markets closed lower as oil prices languished for much of the year.

Table 1: Example of post-editing approaches. Entity swap (Chen et al., 2021) does not modify the errors (red) if a suitable entity replacement is not found, revision-based editing (Adams et al., 2022) performs more extensive changes, deletion (Wan and Bansal, 2022) removes errors but affects grammaticality (orange), while ours returns a well-formed summary through compression.

summaries (Chen et al., 2021), autoregressive approaches that learn to rewrite and remove perturbations from the input (Cao et al., 2020; Zhu et al., 2021; Adams et al., 2022), or deletion-based editing of references (Wan and Bansal, 2022). We provide example outputs of these approaches that demonstrate potential downsides in Table 1. For summaries in datasets such as XSum (Narayan et al., 2018) that contain extrinsic entity errors in up to 70% of reference and generated summaries (Maynez et al., 2020), a *suitable entity replacement is often not available* from the source.

To address such errors, we propose a post-editing model that performs *controlled compression*: given input text and a set of (factually incorrect) entities, it generates a compressed output with the specified entities removed, as demonstrated at the bottom of Table 1. To synthesize a training set for this model, we use existing sentence compression data to first train a perturber model that maps a compressed sentence and a list of entities to an uncompressed sentence containing those entities. We then apply

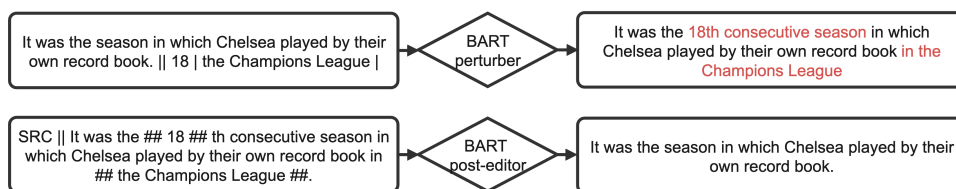


Figure 1: Overview of our compression-based approach. A perturber model is trained to produce an uncompressed sentence with inserted entities, conditioned upon the compressed sentence and an appended list of entities. The perturber is applied to a dataset’s reference summaries to create data to train a post-editor that takes the document (SRC) and a summary with entities to remove surrounded by special tokens, and removes the erroneous words.

the perturber model to insert entities on a subset of the target dataset whose summaries contain only named entities also found in the input. Our post-editor is then trained in the reverse direction, conditioning upon the input article and the perturbed reference summary, with entities to remove marked with special tokens, to produce the compressed summary, as illustrated in Figure 1.

Our contributions are the following: 1) We propose a compression-based method for summary post-editing that removes extrinsic entity errors, improving entity precision with respect to the input by up to 25% along with improvements in other factual consistency metrics while retaining informativeness according to automatic analysis. 2) We show that this method can be applied on top of a rewriting-based post-editor, improving entity precision by up to 38% overall with a small decrease in ROUGE score. 3) We perform an extensive comparison of prior post-editing methods across two datasets and six summarization models to better understand the trade-offs between factual consistency, informativeness, and grammaticality. Models are made publicly available: <https://github.com/salesforce/CompEdit>.

2 Methodology

2.1 Prior Post-Editors

SpanFact We implement a variation of the autoregressive model from Dong et al. (2020). We train a BART-large (Lewis et al., 2020) model to take the source document and the summary with entity slots masked and fill in the masked slots. We also train the model on the data subset whose summaries contain only named entities found in the input, which we call **SpanFact-c**.

CCGS We apply the model from Chen et al. (2021) that generates candidate summaries by enumerating all ways to replace summary entities with similar-typed input entities and training BART with a clas-

sification layer to re-rank these summaries.

ReDRESS We apply the approach from Adams et al. (2022) for revising clinical reference summaries to news summarization. The approach consists of two stages: 1) a perturber learns to corrupt a summary by using entity swaps between the input and a retrieved set of entities, span deletion, and shuffling as training data. 2) The perturber is then applied to the reference summaries to create training samples for a reviser that learns to remove errors through contrastive learning. In contrast, our approach focuses on more controlled perturbations and revisions to remove and compress rather than rewrite, which may insert errors.

FactPegasus We apply the deletion-based corrector component of Wan and Bansal (2022). This method removes extrinsic entity error tokens and surrounding words based on manually-defined rules over the dependency parse of the summary, which may introduce grammatical errors.

2.2 Compression-Based Post-Editor

We train a post-editor in two steps on sentence compression data from Filippova and Altun (2013), where the uncompressed sentence can be viewed as the summary containing information not present in the compressed version. Example input and output for these two steps are shown in Figure 1. We first train a BART-large **perturber** model conditioned upon a compressed sentence and entities present in the longer sentence but not present in the compressed one to produce a longer sentence. In other words, we maximize the following probability: $P(\text{Uncompressed} \mid \text{Compressed}, \text{Ents})$. Then, for a given data point in the summarization dataset, we select each of one, two, and three entities from the source not present in the reference and condition upon those entities and the reference summary to produce a longer, perturbed version containing those entities. Varying the number of en-

Model	E- P_{src}	BS- P_{src}	D_{arc}	QAFE	E- R_{ref}	BS-F1 $_{ref}$	R1	R2	RL	R1-c	CoLA	Edit%
BART	61.59	41.69	82.80	1.960	55.02	48.88	45.20	21.95	36.95	43.59	98.83	-
BART-c	80.42	44.39	89.53	2.161	41.47	44.83	41.10	17.63	33.10	41.32	98.81	-
SpanFact (Dong et al., 2020)	61.96	41.70	82.90	1.947	54.51	48.77	45.10	21.88	36.88	43.54	98.77	40.53
SpanFact-c	74.26	42.82	85.85	1.978	45.20	47.32	43.44	19.80	35.42	43.45	98.44	60.27
CCGS (Chen et al., 2021)	64.80	41.93	83.44	1.938	53.22	48.56	44.89	21.61	36.71	43.57	98.54	15.07
ReDRESS (Adams et al., 2022)	75.42	44.90	88.54	2.168	47.37	46.98	43.30	20.12	35.29	42.81	98.79	61.82
FactPegasus (Wan and Bansal, 2022)	98.71	42.82	90.82	2.082	35.76	44.39	41.98	18.30	34.45	43.98	93.25	67.05
CompEdit (ours)	80.02	43.03	88.35	2.124	44.49	47.00	43.45	20.29	35.63	43.61	98.61	54.31
ReDRESS + CompEdit	85.07	45.39	90.61	2.224	42.20	45.97	42.32	19.19	34.53	42.67	98.66	73.36

Table 2: Baseline and post-editing automatic results for factual consistency, relevance, and grammaticality metrics on XSum. The top two scores from the post-editors in each column are highlighted.

tities inserted mirrors differing levels of editing required. Note that while the entities inserted are not extrinsic errors since they appear in the source, they are inserted out-of-context since the perturber does not condition upon the source document. We then train our BART-large model **post-editor** model, which we call **CompEdit**, to produce the original reference summaries conditioned upon the source document and these perturbed summaries, with special tokens surrounding the entities that should be removed. In this setup, the post-editor learns to remove related, out-of-context entities during training. We only train on references for which all named entities were found in the source and thus maximize the following: $P(\text{Gold summary} | \text{source, special-token perturbed summary})$. While a post-editor could be trained directly from sentence compression data or on summarization data without access to the source, such models resulted in a degradation of summary salience. During inference, we surround extrinsic entity errors, as determined by named entity overlap with the input, with special tokens to signal the model to remove them.

3 Experiments

3.1 Settings

We evaluate the above approaches on the XSUM (Narayan et al., 2018) and CNN/DM (Hermann et al., 2015) datasets. We apply BART-large as the base summarization model, and post-editing models are applied to this base model’s output. We also report results for BART-large trained on a data subset in which all summary named entities are found in the source, which we call **BART-c**. All SpanFact and CompEdit models are trained for 10 epochs with a batch size of 64, with the best checkpoint chosen according to the highest average of ROUGE-1/2/L (Lin, 2004) on the validation set. Additional dataset and modeling details, including

Model	E- P_{src}	BS- P_{src}	D_{arc}	R1	R1-c
UniLM	60.63	42.45	0.83	42.13	40.74
FASumFC	60.66	42.40	0.83	42.17	40.78
ReDRESS	76.01	46.28	0.90	40.73	40.33
FactPegasus	99.06	43.28	0.91	39.26	41.05
CompEdit (ours)	80.72	43.86	0.89	40.79	40.93
ReDRESS + CompEdit	78.44	45.43	0.92	39.17	39.46
BottomUp	43.31	31.38	0.41	26.90	26.28
FASumFC	44.76	32.09	0.43	28.19	27.50
ReDRESS	82.11	45.70	0.85	31.50	31.62
FactPegasus	91.35	31.65	0.49	26.78	26.71
CompEdit (ours)	79.74	34.13	0.54	27.65	28.40
ReDRESS + CompEdit	83.66	44.30	0.87	31.67	32.03

Table 3: Comparison of post-editors with FASumFC (Zhu et al., 2021) on UniLM (Dong et al., 2019) and BottomUp (Gehrmann et al., 2018) outputs on XSum.

CNN/DM results, are found in the Appendix.

To show the generalizability of our results to post-editing models other than BART, we test the best post-editors on UniLM (Dong et al., 2019) and BottomUp (Gehrmann et al., 2018) outputs (additional models in the Appendix). Zhu et al. (2021) released the output of their post-editor **FASumFC** on these outputs but not their trained model. This post-editor is a seq2seq denoising model trained with entity swaps for inserting inconsistencies and backtranslation to create paraphrases.

3.2 Automatic Evaluation

We evaluate using standard ROUGE-1/2/L (**R-1/2/L**) and include a variation called **R1-c** that evaluates R1 on the reference summaries with entities not found in the input removed from the summary. We include the percentage of the base model summaries that are edited by the post-editor (**Edit%**) and the following metrics:

E- P_{src} (**E- R_{ref}**) measures the percentage of entities in the generated summary (reference) present in the input (generated summary). E- P_{src} as a metric performs on par with model-based, token-level metrics (Zhou et al., 2021; Cao et al., 2022). The subset of data without named entity errors has an E- P_{src} of 100.

Model	Consistent	Inconsistent
Unedited summaries	441	62
Cao et al. (2020)	447	56
Lee et al. (2022)	446	57
ReDRESS + CompEdit	473	30

Table 4: Human evaluation of the number of factually consistent and inconsistent summaries after post-editing on the FactCC (Kryscinski et al., 2020) test dataset.

BS-P_{src} (**BS-F1_{ref}**) represents the BERTScore (Zhang et al., 2020) precision (F1) w.r.t. the source article (reference summary).

D_{arc} measures the percentage of dependency arcs in summary entailed by the source article using the model from Goyal and Durrett (2021).

QAFE is the QAFactEval question answering-based consistency metric (Fabbri et al., 2021).

CoLA To evaluate grammaticality, we apply a RoBERTa-large (Liu et al., 2019) model from Krishna et al. (2020) trained on the CoLA dataset (Warstadt et al., 2019), which includes sentences and labels for their grammatical acceptability.

3.3 Results

Results of applying post-editing models to BART on XSum are shown in Table 2, and examples, and results on CNN/DM, are in the Appendix.

We note that BART-c improves across all factual consistency metrics compared to BART at the cost of ROUGE and other informativeness metrics. We find that SpanFact does not improve factual consistency, as this model is trained to fill in entity masks on the original, noisy dataset, but SpanFact-c improves factual consistency. While CCGS does improve factual consistency slightly, the model only edits a small percentage of the summaries (Edit% of about 15). Other post-editors make a more realistic number of edits, considering that over 70% of XSum reference summaries may contain factual inconsistencies (Maynez et al., 2020).

ReDRESS performs better than SpanFact-c on factual consistency and also E-R_{ref}, which aligns with its rewriting objective that can insert semantically relevant entities. FactPegasus improves entity precision by definition (it is not 100 due to differences in entity processing), but it is the only post-editor that decreases grammaticality by a noticeable margin. Our CompEdit model improves factual consistency, and we see further improvements when applying our model on top of the ReDRESS corrector; ReDRESS as a first-stage corrector can insert suitable replacement entities while

the second-stage compressor may remove any remaining entity errors. CompEdit does not remove all entity errors in the summaries, and a qualitative inspection revealed that common world knowledge tokens, such as names of world leaders, were often left unchanged. Recent work has also noted the presence of extrinsic world knowledge errors (Cao et al., 2022), and we leave a larger study of such artifacts to future work. We note that vanilla ROUGE scores do decrease along with improvements in factual consistency. However, R1-c actually shows improvements when applying FactPegasus or CompEdit, showing that much of the loss in vanilla R1 is due to factual inconsistencies in the references.

We show the results of applying the above post-editors on non-BART models in Table 3. We find large improvements over Zhu et al. (2021). We see a gain in R1-c and entity precision when applying CompEdit to the pretrained UniLM, showing the benefits of our compression-based approach when the underlying model contains relatively high-quality summaries. ReDRESS shows a large performance increase on BottomUp; its ability to completely rewrite the summary allows it to improve the non-pretrained, lower-quality summaries that require editing beyond just compression. There is also a benefit from the combination of ReDRESS and CompEdit in terms of entity precision for the BottomUp model, which has been shown to contain a high proportion of entity errors and inconsistencies (Huang et al., 2020; Pagnoni et al., 2021).

Finally, we show the manually-annotated results of post-editing a non-pretrained model from the CNN/DM-based FactCC (Kryscinski et al., 2020) test dataset in Table 4. We compare with previously-reported results: an autoregressive post-editor trained on entity swaps and backtranslated paraphrases (Cao et al., 2020), and a method that substitutes entities from retrieved source sentences (Lee et al., 2022). As extrinsic entity errors, for which CompEdit and FactPegasus are designed, constitute only a small proportion of errors in this dataset, we only include ReDRESS + CompEdit. CompEdit was applied to the ReDRESS outputs containing extrinsic named entity errors, 2.5% of edited summaries, all of which were then labeled factually consistent. We see a large improvement in consistent summaries and similar trends in the benefit of ReDRESS and adding CompEdit as the above results on the non-pretrained BottomUp model.

4 Conclusion

In this work, we propose a sentence-compression-based post-editing model to improve factual consistency in summarization while maintaining the informativeness and grammaticality of the resulting summaries. We show that this model can be used in tandem with a post-editor that performs extensive rewriting for further improvement, especially in pretrained models and datasets with a high proportion of entity errors. In future work, we plan to build on these models by studying the role of dataset artifacts in error correction and addressing unfixable summaries.

5 Limitations

We train and test our models on publicly available news summarization datasets. Political and gender biases may exist in these datasets, and thus models trained on these datasets may propagate these biases. Furthermore, while our models are not language-specific, we only study English summarization.

When used as intended, these post-editing models can help eliminate some factual inconsistencies in model summaries. However, these models do not remove all factual inconsistencies, and thus much care should be taken if one wants to use such models in a user-facing setting.

The experiments in this paper make use of A100 GPUs. We used up to 8 GPUs per experiment, and the experiments may take up to a day to run. Multiple experiments were run for each model, and future work should experiment with distilled models for more lightweight training.

References

- Griffin Adams, Han-Chin Shing, Qing Sun, Christopher Winestock, Kathleen McKeown, and Noémie Elhadad. 2022. [Learning to revise references for faithful summarization](#).
- Rahul Aralikkatte, 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.
- 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.
- Sihao Chen, Fan Zhang, Kazuo 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.
- James Clarke and Mirella Lapata. 2008. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research*, 31:399–429.
- Mingkai Deng, Bowen Tan, Zhengzhong Liu, Eric Xing, and Zhiting Hu. 2021. [Compression, transduction, and creation: A unified framework for evaluating natural language generation](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7580–7605, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. [Unified language model pre-training for natural language understanding and generation](#). In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13042–13054.
- Yue Dong, Shuohang Wang, Zhe Gan, Yu Cheng, Jackie Chi Kit Cheung, and Jingjing Liu. 2020. [Multi-fact correction in abstractive text summarization](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9320–9331, Online. Association for Computational Linguistics.
- Alexander Fabbri, Wojciech Kryscinski, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2022. [Summeval: Re-evaluating summarization evaluation](#). *Transactions of the Association for Computational Linguistics*, 9(0):391–409.
- Alexander R. Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2021. [Qafacteval: Improved qa-based factual consistency evaluation for summarization](#). *ArXiv preprint*, abs/2112.08542.

- Katja Filippova and Yasemin Altun. 2013. [Overcoming the lack of parallel data in sentence compression](#). In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1481–1491, Seattle, Washington, USA. Association for Computational Linguistics.
- Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. [Convolutional sequence to sequence learning](#). In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252. PMLR.
- Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. [Bottom-up abstractive summarization](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium. 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.
- Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. [Teaching machines to read and comprehend](#). In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1693–1701.
- 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.
- Daniel Kang and Tatsunori B. Hashimoto. 2020. [Improved natural language generation via loss truncation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 718–731, Online. Association for Computational Linguistics.
- Kalpesh Krishna, John Wieting, and Mohit Iyyer. 2020. [Reformulating unsupervised style transfer as paraphrase generation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 737–762, Online. Association for Computational Linguistics.
- Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. [Evaluating the factual consistency of abstractive text summarization](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.
- Hwanhee Lee, Cheoneum Park, Seunghyun Yoon, Trung Bui, Franck Dernoncourt, Juae Kim, and Kyomin Jung. 2022. [Factual error correction for abstractive summaries using entity retrieval](#).
- 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.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized bert pretraining approach](#). *ArXiv preprint*, abs/1907.11692.
- 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.
- Feng Nan, Ramesh Nallapati, Zhiguo Wang, Cicero Nogueira dos Santos, Henghui Zhu, Dejie Zhang, Kathleen McKeown, and Bing Xiang. 2021. [Entity-level factual consistency of abstractive text summarization](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2727–2733, 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.
- 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.
- David Wan and Mohit Bansal. 2022. [Factpegasus: Factuality-aware pre-training and fine-tuning for abstractive summarization](#).

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. [Neural network acceptability judgments](#). *Transactions of the Association for Computational Linguistics*, 7:625–641.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. [Bertscore: Evaluating text generation with BERT](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Francisco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. 2021. [Detecting hallucinated content in conditional neural sequence generation](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1393–1404, Online. Association for Computational Linguistics.

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 Additional Results

Results of applying post-editing models to BART on CNN/DM are shown in Table 5. BART-c decreases performance in factual consistency on CNN/DM, perhaps due to the reduction in training data points; CNN/DM is largely extractive and encourages factually consistent as-is. This is reaffirmed by the small gap between SpanFact and SpanFact-c and the generally very high scores on CNN/DM, which leave less room for improvements compared to the XSum dataset. We also see a lower Edit% on CNN/DM, which aligns with Pagnoni et al. (2021) where 27% of BART summaries on CNN/DM contained a factual inconsistency.

In Table 6, we provide model outputs that illustrate the characteristics of the post-editors studied.

We show the results of applying post-editors on additional non-BART models in Table 7. We find

similar trends as the post-editors applied to other non-pretrained in the main text.

Additionally, we trained a post-editor only on sentence compression data; the model produces the compressed sentence (summary) given an uncompressed sentence (summary), with entities to remove marked with special tokens. This model applied to BART XSum summaries provided similar entity precision but resulted in about a 1.2 drop in ROUGE-1 and a five-point drop in entity recall on the validation set, so we did not include this model.

B Additional Model Details

In this section, we provide additional details for our model, baselines, and metrics. To encourage retaining essential information, we filter data points from Filippova and Altun (2013) in which the compressed sentences contain less than 75% the number of tokens in the longer sentence. We compared the effect of sentence compression data on downstream post-editor performance, experimenting with higher compression ratios as well as other sentence compression datasets (Clarke and Lapata, 2008), but these models resulted either in lower ROUGE performance or a decrease in entity precision on the validation set. To create training examples for our post-editor, we inserted one, two, and three entities into the references by applying our perturber. The size of the subsets of reference summaries entity precision of 100 on (training, validation) is (52k, 2.9k) for XSum and (160k, 6.5k) for CNN/DM. We then sample 200k data points to train on a dataset of size similar to XSum and found that doubling the size of this data did not give further improvements.

We experimented with other approaches for marking extrinsic errors such as model-based approaches Zhou et al. (2021) and Deng et al. (2021), but the entity overlap approach performed better and is more interpretable.

We use the following inference parameters: (beam size, min generation length, max generation length, length penalty) for XSum = (6, 11, 62, 1.0) and CNN/DM = (4, 40, 140, 2.0). We retrain BART-large on XSum, and for CNN/DM we run inference from the fairseq/bart-large-cnn¹ checkpoint from the transformers library (Wolf et al., 2020).

For ReDRESS, we train BART-large rather than the BART-base model used in the original paper for

¹<https://huggingface.co/facebook/bart-large-cnn>

Model	E-P _{src}	BS-P _{src}	D _{arc}	QAFE	E-R _{ref}	BS-F1 _{ref}	R1	R2	RL	R1-c	CoLA	Edit%
BART	98.50	71.10	98.57	4.550	58.93	43.55	44.05	21.12	41.02	44.11	95.68	-
BART-c	97.60	68.59	97.88	4.405	58.00	43.70	44.09	20.89	41.11	44.14	95.11	-
SpanFact (Dong et al., 2020)	98.46	71.06	98.49	4.545	58.32	43.45	43.97	21.02	40.95	44.03	95.60	28.29
SpanFact-c	98.83	71.07	98.53	4.544	58.19	43.41	43.95	20.99	40.93	44.01	95.50	32.36
CCGS (Chen et al., 2021)	98.18	71.04	98.49	4.466	58.00	43.48	43.98	21.03	40.96	44.04	95.48	14.93
ReDRESS (Adams et al., 2022)	99.03	71.32	98.66	4.557	58.81	43.53	43.98	21.08	40.96	44.05	95.66	30.13
FactPegasus (Wan and Bansal, 2022)	99.94	71.10	98.68	4.543	58.40	43.44	43.98	21.03	40.96	44.12	95.21	13.44
CompEdit (ours)	99.10	71.13	98.75	4.554	58.61	43.50	44.02	21.08	41.01	44.11	95.52	27.93
ReDRESS + CompEdit	98.95	71.24	98.74	4.555	58.49	43.44	43.96	21.04	40.95	44.04	95.42	44.53

Table 5: Baseline and post-editing automatic results for factual consistency, relevance, and grammaticality metrics on CNN/DM. The top two scores from the post-editors in each column are highlighted.

Source Article	System	Summary
Kevin Patrick Dawes, 33, was abducted in 2012 as he entered Syria. ... The Czech government, which represents US diplomatic interests in Syria, helped secure his release.	BART	A US contractor who was held hostage in Syria for more than three years has been freed, the US State Department says.
	ReDRESS, (w/CompEdit)	A US contractor who was held hostage in Syria has been freed by Czech authorities , the US State Department says.
	FactPegasus, CompEdit	A US contractor who was held hostage in Syria has been freed, the US State Department says.
The victims were farming in an area... close proximity to Mount Sinabung. The volcano was still spewing ash on Sunday. ... Rescue teams are still scouring the area, looking for more victims who may have been killed or badly burned by the hot gas and ash clouds released in the eruption.	BART	Rescue workers in Indonesia are searching for more than 20 people feared dead after the Mount Sinabung volcano erupted on Saturday .
	ReDRESS	Rescue workers in Indonesia are searching for more victims who may have been killed when the Mount Sinabung volcano erupted on Saturday .
	ReDRESS + CompEdit	Rescue workers in Indonesia are searching for more victims who may have been killed when the Mount Sinabung volcano erupted.

Table 6: Example BART and post-editor outputs showing the capacity of ReDRESS to insert related entities and for CompEdit to remove errors. A comma separating two systems indicates that the two return the same summary.

FASum	71.01	37.17	0.71	30.27	29.90
FASumFC	68.59	37.03	0.71	30.19	29.83
ReDRESS	83.26	44.56	0.88	32.76	32.91
FactPegasus	97.81	37.03	0.77	29.52	30.01
CompEdit (ours)	89.32	37.99	0.77	30.54	30.63
ReDRESS + CompEdit	83.04	43.17	0.89	32.49	32.88
TConvS2s	70.58	36.93	0.72	31.45	31.00
FASumFC	69.53	36.57	0.73	32.43	31.95
ReDRESS	83.49	45.17	0.88	33.59	33.71
FactPegasus	97.56	37.44	0.77	30.57	31.08
CompEdit (ours)	88.08	38.80	0.79	32.04	32.06
ReDRESS + CompEdit	83.02	43.62	0.90	33.17	33.55

Table 7: Comparison of post-editors with FASumFC (Zhu et al., 2021) on FASum (Zhu et al., 2021) and TConvS2S (Gehring et al., 2017) outputs on XSum.

consistency with the above models. A key component of this post-editor is the over-generation and reranking of the edited summaries; we rerank according to entity precision over the input. We also vary how entities are matched in the ReDRESS retrieval component and use exact string matching to match how we filtered the clean data subset.

For CCGS, we apply the XSum-trained reranker to both XSum and CNN/DM.

For DAE, we apply the DAE_xsum_human_best model.² We provide our BERTScore run hash.³

²<https://drive.google.com/file/d/1TGS0RmS1sY1FyU52LxWfPA3Jiu0d--M/>

³bert-base-uncased_L8_no-idf_version=0.3.9(hug_trans=4.11.3)-rescaled