Length Does Matter: Summary Length can Bias Summarization Metrics

Xiaobo Guo and **Soroush Vosoughi**
Department of Computer Science
Dartmouth College
Hanover, New Hampshire
{xiaobo.guo.gr, soroush.vosoughi}@dartmouth.edu

Abstract

Establishing the characteristics of an effective summary is a complicated and often subjective endeavor. Consequently, the development of metrics for the summarization task has become a dynamic area of research within natural language processing. In this paper, we reveal that existing summarization metrics exhibit a bias toward the length of generated summaries. Our thorough experiments, conducted on a variety of datasets, metrics, and models, substantiate these findings. The results indicate that most metrics tend to favor longer summaries, even after accounting for other factors. To address this issue, we introduce a Bayesian normalization technique that effectively diminishes this bias. We demonstrate that our approach significantly improves the concordance between human annotators and the majority of metrics in terms of summary coherence[1].

1 Introduction

Text summarization aims to condense lengthy documents into concise, coherent, and human-readable texts while preserving essential ideas. Deep learning and large-scale datasets have substantially advanced this field, as demonstrated by models (Sutskever et al., 2014; Lewis et al., 2020; Raffel et al., 2020) and datasets (Nallapati et al., 2016; Narayan et al., 2018; Koupaee and Wang, 2018).

Progress in this field hinges on reliable metrics for evaluating generated summary quality. While human annotations are considered the gold standard, they can be costly, time-consuming, and subjective. Automated evaluation metrics address these challenges, but assessing summary quality remains complex. Efforts to improve text summarization metrics are ongoing (e.g., (Cohan and Goharian, 2016; Koto et al., 2021; Pagnoni et al., 2021)). However, biases concerning generated summary length remain underexplored. If certain metrics favor shorter or longer summaries, evaluations may be flawed, as they should focus on quality, not extraneous factors.

In this paper, we investigate the impact of generated summary length on 14 distinct metrics. Our experiments reveal that: 1) the length of generated summaries affects most metrics to varying extents; 2) although some discrepancies result from the correlation between quality and summary length, they persist even after controlling for quality; 3) these effects are consistent across datasets, but the magnitude of the effects may differ; 4) metrics based on gram overlap (e.g., Rouge) are more inclined to assign higher scores to longer summaries than metrics based on word/sentence embeddings (e.g., BERTScore). In response to these findings, we propose a Bayesian normalization strategy to diminish the influence of summary length on metrics. We demonstrate that our approach significantly improves alignment with human annotators in terms of summary coherence for the majority of metrics.

2 Related Work

Numerous studies have explored the limitations of automatic metrics for text summarization, addressing issues like weak correlation with human judgment, poor adaptability to diverse corpora, and failure to capture linguistic nuances. For instance, Fabbri et al. (2021) found a modest correlation between human judgment and most metrics, while Cohan and Goharian (2016) showed Rouge metrics struggle with varied terminology and paraphrasing. Cross-language studies (Koto et al., 2021) demonstrated suboptimal performance for some metrics on non-English corpora. Additionally, Maynez et al. (2020) and Pagnoni et al. (2021) critiqued automatic metrics for not detecting factual inconsistencies.

A fundamental characteristic of a robust metric for this task is the capacity to consistently rate summaries of equal quality, even when they differ along other dimensions. Prior research, such as

[1] The code is available at SLDbias

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 15869–15879
December 6-10, 2023 ©2023 Association for Computational Linguistics

Yuan et al. (2021); Fabbri et al. (2021), highlighted variable average scores for some metrics depending on reference/generation lengths. However, they neither proposed debiasing methods nor extended their focus beyond ROUGE and BARTScore.

Sun et al. (2019) addressed length bias by comparing generated text with a randomly chosen, length-controlled extractive summary. Although this method can partially mitigate length bias, it is limited in applicability to various summarization techniques and has only been tested on the ROUGE family of metrics. Nonetheless, we benchmark our method against theirs, demonstrating consistent superiority across multiple summarization approaches and metrics.

3 The Impact of Length on Metrics

Here, we investigate the effect of generated summary length on 14 summarization metrics, including both lexical-based and embedding-based metrics (Sai et al., 2022). We conduct experiments using three popular datasets: CNN/Daily Mail corpus (CNN/DM) (Nallapati et al., 2016), WikiHow (Koupaee and Wang, 2018), and webis-tldr-17-corpus (Web-tldr) (Völske et al., 2017), with summaries generated by three models: BART (Lewis et al., 2020), Longformer (Beltagy et al., 2020), and T5 (Raffel et al., 2020). More details can be found in Appendix A.

To isolate the impact of generated summary length, we control for the source article and reference summary lengths (Appendix B shows that both factors also bias metrics). Using equal frequency binning, we categorize samples based on source-article and reference lengths (10 buckets each, totaling 100 buckets). Within each bucket, we compute the mean score for each five percentile of the generated summary length distribution, then average the scores across the 100 buckets to calculate the mean score for that percentile. As our study focuses on percentile rankings, the varying distribution of generated summary lengths is not a concern. For clarity, we shift scores by the minimum score for each metric in our figures.

Figure 1 shows the trend for BART (results for other models in Appendix D). Even after accounting for source article and reference lengths, most metrics are influenced by generated summary lengths to varying degrees. We also observe that the trends are generally consistent across datasets, with the exception of BERTScore, which exhibits divergent behavior among different datasets.

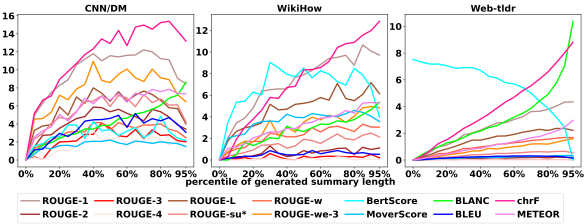


Figure 1: Average metric values for varying generated summary lengths, controlling for the length of the source article and reference. Note that scores have been shifted by the minimum value, resulting in the lowest score being 0. For clarity, scores have been rescaled from a 0-1 range to a 0-100 range.

Next, we control for summary quality to eliminate any correlations between generated summary length and quality by randomly shuffling generated summaries among samples, thus randomizing reference-summary pairings. We repeat our experiments with this modified dataset and display the results for BART in Figure 2 (see Appendix D for other models). While weaker, the trends persist for several metrics, indicating that metric length preference primarily drives these trends. Interestingly, BERTScore exhibits an opposite trend, potentially due to its reliance on text semantics, which may become diluted in longer text.

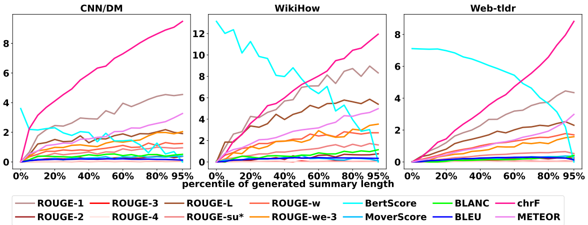


Figure 2: Repeated experiments with randomly shuffled generated summaries (using 3 different random seeds).

4 Modeling the Effect of Length

We quantitatively examine the impact of generated summary length on various metrics using a Bayesian network, considering four variables: source article length, reference summary length, generated summary length, and score. The Bayesian network structure is shown in Figure 3.

In our experiments, we manually design the structure and provide the Bayesian network with the joint distribution of all variables based on text samples from our data. Once trained, the network can be employed to calculate the expected score $\hat{s}$, given the source article length (l_a), reference length (l_r), and generated summary length (l_g), using the following equation:

$$\hat{s} = \sum_{s \in S} P(s, l_a, l_r, l_g) * s \quad (1)$$

where S represents the set of all possible scores and $P(s, l_a, l_r, l_g)$ denotes the joint probability of these four variables. We opt for a Bayesian network over simpler linear models due to the non-linear relationship between variables, as shown in Figures 1 and 2, and discussed in Section 5.

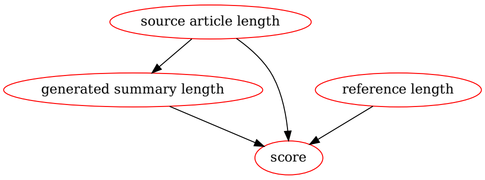


Figure 3: The structure of the Bayesian network for the quantitative analysis.

We discretize variables using equal frequency binning for source article length (10 buckets), reference length (10 buckets), generated summary length (10 buckets), and equal width binning for the score (1000 buckets, 0.1 each). A sample is represented as (l_a, l_r, l_g, s), where s is the discretized score, and l_a, l_r, and l_g denote the source article, reference, and generated summary percentile ranks, respectively. We train 14 Bayesian networks, one for each metric. Each network predicts the score for a specific metric given the length of the source article, reference, and generated summary based on the joint probability of the four variables.

We use the trained networks to quantify the effect of generated summary length on each metric's score. For each sample, we keep l_a and l_r constant, increment l_g by one (ten percentile)[2], and input the updated values into the Bayesian networks to predict new scores for each metric. We average the difference between these new scores and the original corresponding scores across all samples, yielding the mean score difference due to a ten-percentile increase in generated summary length.

We experiment with all 14 metrics across three datasets and three models, using both shuffled and non-shuffled data. Table 1 shows most metrics exhibit consistent directionality of the effect across multiple datasets, with varying effect sizes depending on datasets and models. This aligns with our earlier qualitative experiments. In shuffled data experiments, all metrics except BERTScore show increased predicted scores in at least one dataset as generated summary length increases, while BERTScore decreases.

Trends remain consistent across different models, with only one instance of disagreement (ROUGE-3 for the Web-tldr dataset). Comparing lexical-based and embedding-based metrics, we find embedding-based metrics are generally less prone to score increases with increasing generated summary length. Non-shuffled experiments display higher sensitivity, on average, to generated summary length, suggesting other unknown factors may also influence scores. A limitation of our work is the presence of such unaccounted-for factors.

Our experiments in Sections 3 and 4 conclude that: 1) Controlling for source article length, reference summary length, generated summary quality, model, and dataset, generated summary length appears to bias most metrics, with longer summaries resulting in higher scores in a non-linear relationship; 2) Trends are generally consistent across datasets and models, but effect sizes vary; 3) The relationship between generated summary length and score can be partially attributed to metrics' preference for longer text, but not all variance is accounted for, indicating other unknown variables contribute to the relationship.

5 Reducing Length Bias

Here, we explore strategies for mitigating the metrics' bias concerning the length of generated summaries and generating a length-adjusted score. Intuitively, if all generated summaries had the same length as their corresponding reference summaries, the length would no longer be a confounding factor. Since we do not want to control the length of the generations, we adjust the scores for a generated summary post hoc. Each metric is adjusted based on the corresponding Bayesian network's predicted score for a generated summary with the same length as the sample's reference length while keeping every other variable assignment the same as the sample. We employ a Bayesian network to better capture the non-linear relationships between our variables.

The Bayesian network used in this section is identical to the previous network, except for

[2] We exclude data in the last 10th percentile, as its l_g cannot be increased further.

		Not-Shuffled			Shuffled		
		CNN/DM	WikiHow	Web-tldr	CNN/DM	WikiHow	Web-tldr
Embedding	BERTScore	0.27	0.26^	-1.10	-0.42	-1.05	-1.12
	BLANC	0.91	0.45	0.34	0.04	0.03	0.04
	MoverScore	0.18	0.70	0.00^	0.04	0.15	0.01
Lexical	BLEU	0.39	0.75	0.01	0.01	0.19	0.01
	chrF	1.33	1.23	0.77	0.95	0.63	0.82
	METEOR	0.72	0.84	0.23	0.33	0.47	0.23
	ROUGE-1	0.72	1.43	0.32	0.46	0.86	0.30
	ROUGE-2	0.39	1.10	0.00^	0.03	0.44	0.02
	ROUGE-3	0.31	0.85	0.00	0.00	0.27	0.00^
	ROUGE-4	0.25	0.77	0.00	0.00	0.27	0.00
	ROUGE-L	0.39	1.17	0.14^	0.18	0.61	0.16
	ROUGE-su*	0.45	1.07	0.05	0.10	0.39	0.04
	ROUGE-w	0.26	0.77	0.08	0.12	0.37	0.12
	ROUGE-we-3	0.61	1.62	0.10	0.19	0.88	0.12

Table 1: The predicted difference in performance when increasing the generated summary length by 10% percentile across the entire test samples using the Bayesian network. The performance is calculated as the mean of the results from the three models. A ^ indicates that the trend of at least one model differs from the others. For clarity, the scores have been rescaled from a 0-1 range to a 0-100 range.

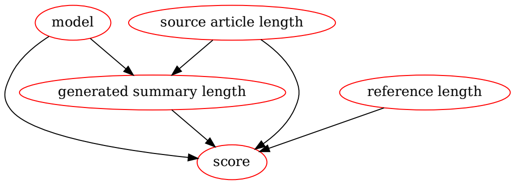


Figure 4: The structure of the Bayesian network for the score adjustment experiments.

one additional variable, the model (M), which is a categorical variable corresponding to the type of model (Figure 4). For a given sample, we use m, l_a, l_r, l_g, to predict s twice: once regularly ($\hat{s_g} = Bayesian(m, l_a, l_r, l_g)$) and the other where we set $l_g = l_r$ ($\hat{s_r} = Bayesian(m, l_a, l_r, l_r)$). We calculate the difference caused by the generated summary length as $s * \frac{\hat{s_g} - \hat{s_r}}{\hat{s_r}}$. As discussed earlier, the quality of the summaries (and other factors) is partially responsible for this difference. Thus, we use an adjustment scale α to control the portion of the difference that must be adjusted. Therefore, the difference that needs to be removed is $\alpha * s * \frac{\hat{s_g} - \hat{s_r}}{\hat{s_r}}$. Our length-adjusted score is:

$$s_{adj} = s - \alpha * s * \frac{\hat{s_g} - \hat{s_r}}{\hat{s_g}} \qquad (2)$$

5.1 Evaluation

We conduct experiments on 14 models using generated summaries from original papers (see Appendix E.1). Our evaluation utilizes system-level analysis (Louis and Nenkova, 2013), correlating quality rankings of summaries based on human judgment and automatic metrics. We leverage human judgment results from Fabbri et al. (2021), encompassing 100 samples. For each model, we consider the mean human scores of all 100 samples as the human-annotated score and the mean score of each automatic metric as that metric's score for the model. Models are ranked using human and automatic metric scores, and Kendall's τ calculates rank correlation. We adopt human annotation "coherence" scores as a holistic summary quality measure Fabbri et al. (2021).

Our experiments employ two baselines: Random extractive summaries Normalization (RN) (Sun et al., 2019) and Linear Regression (LR). RN mitigates generation length influence by comparing performance against randomly chosen length-controlled extractive summaries. It generates summaries with the same mean length, calculating performance based on the ground truth and normalizing the mean performance of the tested model. LR estimate scores $\hat{s_g}$ and $\hat{s_r}$, incorporating model, source article length, reference length, and generated summary length to compute s_{adj} using Eq. 2. Figure 5 presents adjustment experiments' results for varying αs.

Figure 5 illustrates the improvement of the adjusted score using the Bayesian network and linear regression for each metric, as well as the mean improvement ("MEAN of All") across 14 metrics at each given α. The score is measured by Kendall's τ between the human- and metric-ranked models.

Figure 5 illustrates the improvement in adjusted scores using the Bayesian network and linear regression for each metric and the mean improvement across 14 metrics at each given α. Kendall's τ measures the score between human- and metric-ranked models. Key observations include: 1) The "MEAN of All" shows the Bayesian network-based adjusted

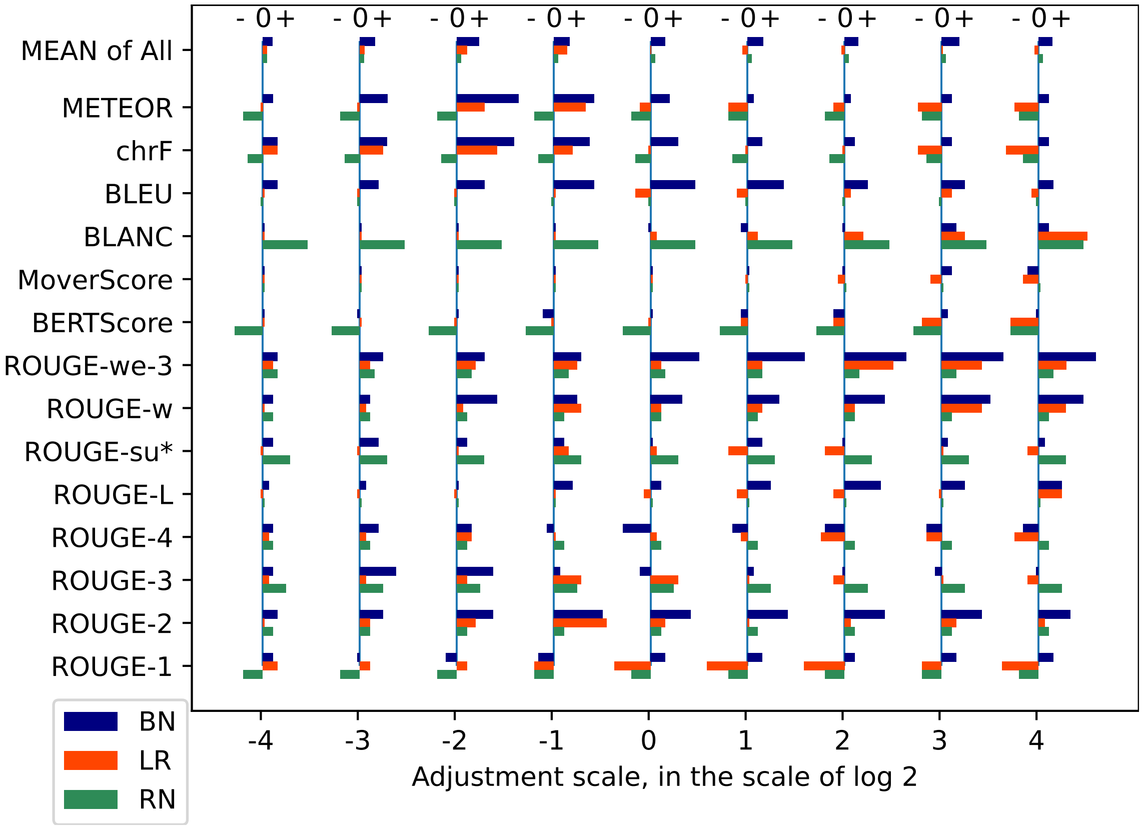


Figure 5: The improvement of metrics adjusted using a Bayesian network (BN), linear regression (LR) with different adjustment scales, α, and the Random Extractive Summaries Normalization (RN) (Sun et al., 2019). "MEAN of All" represents the mean improvement of all 14 metrics at a given α. "+" indicates better performance than the unadjusted score, while "-" signifies worse performance than the unadjusted score.

score outperforms LR and RN baselines for all αs, with a mean improvement of 0.1 at $\alpha = 2^{-2}$. 2) Overall, our Bayesian-network-based adjusted score is better or equally correlated with human judgment for the majority (106 out of 126) of metrics, except for BLANC and certain αs of ROUGE. 3) Smaller αs result in more stable adjustments, with $\alpha = 2^{-4}$ showing no decrease in Kendall's τ.

Table F1 in the Appendix shows the best performance for all methods from Figure 5. Optimal performance is achieved at $\alpha = 2^{-2}$ for our Bayesian network (BN) and $\alpha = 2^{-1}$ for linear regression (LR). Unadjusted score (Ori) and random extractive summaries normalization (RN) are not α-dependent. Our method outperforms baselines, achieving the best performance in 11 out of 14 metrics and demonstrating the highest stability, with a decrease for only one metric. In contrast, RN decreases in 5 metrics and LR in 2 metrics.

These results suggest that adjusting automatic metrics' scores to minimize generated summary length influence improves correlation with human assessments of summary coherence. However, not all metrics consistently improve. The Bayesian network's superior performance compared to the linear model supports the hypothesis that a probabilistic model is better suited to capture the relationship between generated summary length and metrics.

6 Conclusions & Future Work

This paper investigates the relationship between generated summary length and 14 summarization metrics. Our findings reveal a correlation between generated summary length and metric scores, even after accounting for generation quality, source article length, reference summary length, model, and dataset. We propose a Bayesian-network-based approach to adjust metric scores based on generated summary length, resulting in an improved agreement between model-generated and coherence-based human rankings for most metrics.

Future work could examine new metrics for summarization tasks, such as conditioning on target length or measuring "information density" rather than absolute information.

7 Ethical Considerations & Limitations

We do not anticipate any significant ethical concerns arising from our work. However, we utilize a publicly available dataset of Reddit posts, which may contain offensive or sensitive content. Caution should be exercised when working with this dataset.

When implementing the findings and the proposed method from our paper in summarization tasks, several potential limitations should be considered:

First, we view effective summarization as striking a balance between brevity and completeness. In most instances, an ideal summary is concise yet comprehensive, and our proposed approach aims to address this balance by mitigating the influence of length bias on summary quality assessments. However, in specific domains such as legal and medical fields, longer summaries may be more acceptable if they provide a higher level of completeness.

Second, our paper presents a method that can be applied to the majority of metrics to reduce length bias instead of proposing a new metric. We pursued this approach for two reasons: 1) Our experiments demonstrated that numerous metrics are affected by summary length, and our goal was to minimize this influence while maintaining the usability of these metrics. 2) Creating a new metric to supplant the majority of existing metrics is difficult, as they each address different facets of summarization quality. Nonetheless, our proposed adjustment method implies that all adjusted metrics might still be influenced by other factors that affect the unadjusted metrics. As a result, researchers should weigh the benefits and limitations of the unadjusted metrics when employing the adjusted metrics.

References

Abhaya Agarwal and Alon Lavie. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. *Proceedings of WMT-08*.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686.

Arman Cohan and Nazli Goharian. 2016. Revisiting summarization evaluation for scientific articles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 806–813.

Alexander Richard 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.

Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2018. Soft layer-specific multi-task summarization with entailment and question generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 687–697.

Wan-Ting Hsu, Chieh-Kai Lin, Ming-Ying Lee, Kerui Min, Jing Tang, and Min Sun. 2018. A unified model for extractive and abstractive summarization using inconsistency loss. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 132–141.

Yichen Jiang and Mohit Bansal. 2018. Closed-book training to improve summarization encoder memory. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4067–4077.

Fajri Koto, Jey Han Lau, and Timothy Baldwin. 2021. Evaluating the efficacy of summarization evaluation across languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 801–812.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*.

Wojciech Kryściński, Romain Paulus, Caiming Xiong, and Richard Socher. 2018. Improving abstraction in text summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1808–1817.

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.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Annie Louis and Ani Nenkova. 2013. Automatically assessing machine summary content without a gold standard. *Computational Linguistics*, 39(2):267–300.

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.

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

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

Jun Ping Ng and Viktoria Abrecht. 2015. Better summarization evaluation with word embeddings for rouge. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1925–1930.

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.

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

Ramakanth Pasunuru and Mohit Bansal. 2018. Multi-reward reinforced summarization with saliency and entailment. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 646–653.

Maja Popović. 2015. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395.

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

Ananya B Sai, Akash Kumar Mohankumar, and Mitesh M Khapra. 2022. A survey of evaluation metrics used for nlg systems. *ACM Computing Surveys (CSUR)*, 55(2):1–39.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.

Simeng Sun, Ori Shapira, Ido Dagan, and Ani Nenkova. 2019. How to compare summarizers without target length? pitfalls, solutions and re-examination of the neural summarization literature. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 21–29.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Oleg Vasilyev, Vedant Dharnidharka, and John Bohannon. 2020. Fill in the blanc: Human-free quality estimation of document summaries. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 11–20.

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

Yuxiang Wu and Baotian Hu. 2018. Learning to extract coherent summary via deep reinforcement learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, pages 5602–5609.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M Meyer, and Steffen Eger. 2019. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. 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 563–578.

Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018. Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–663.

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

A Datasets, Metrics, and Models

A.1 Datasets

We conducted experiments on widely-used summarization datasets: CNN/Daily Mail (CNN/DM) (Nallapati et al., 2016), WikiHow (Koupaee and Wang, 2018), and Webis-tldr-17-corpus (Web-tldr) (Völske et al., 2017). vbnet

CNN/DM comprises articles from CNN and DailyMail, with human-generated abstractive summaries from their respective websites. The dataset contains 286,817 training, 13,368 validation, and 11,487 testing samples. We use the non-anonymized version, released under an MIT License.

WikiHow is sourced from the online WikiHow knowledge base, with summary sentences extracted from bold lines in each paragraph. The dataset includes 168,128 training, 6,000 validation, and 6,000 testing samples, and is under a CC-BY-NC-SA license.

Web-tldr contains approximately 4 million content-summary pairs extracted from Reddit between 2006-2016. We apply a 70/10/20 train/validation/test split. The dataset is under a CC BY 4.0 license.

A.2 Metrics

We describe the metrics used in our experiments.

A.2.1 Embedding-based Metrics

- **BERTScore (Zhang et al., 2019)** computes similarity scores between tokens in reference and generated summaries, aligning tokens greedily based on the cosine similarity between BERT embeddings.

- **BLANC (Vasilyev et al., 2020)** is a reference-less metric assessing improvements in language comprehension tasks when a pre-trained language model has access to a document summary.

- **MoverScore (Zhao et al., 2019)** calculates semantic distance between a summary and reference text using Word Mover's Distance.

A.2.2 Lexical-based Metrics

- **BLEU (Papineni et al., 2002)** measures n-gram overlap between a generated summary and reference text with a shortness penalty.

- **chrF (Popović, 2015)** calculates character n-gram F-score overlap between generated and reference texts.

- **METEOR (Agarwal and Lavie, 2007)** aligns generated summaries and reference sentences using unigram mapping based on surface and stemmed forms, computing precision and recall as a harmonic mean.

- **Rouge (Lin, 2004)** estimates summary quality based on overlapping textual units (n-grams, word sequences) between generated and reference summaries.

- **ROUGE-WE (Ng and Abrecht, 2015)** extends ROUGE with soft lexical matching using Word2Vec embedding cosine similarity.

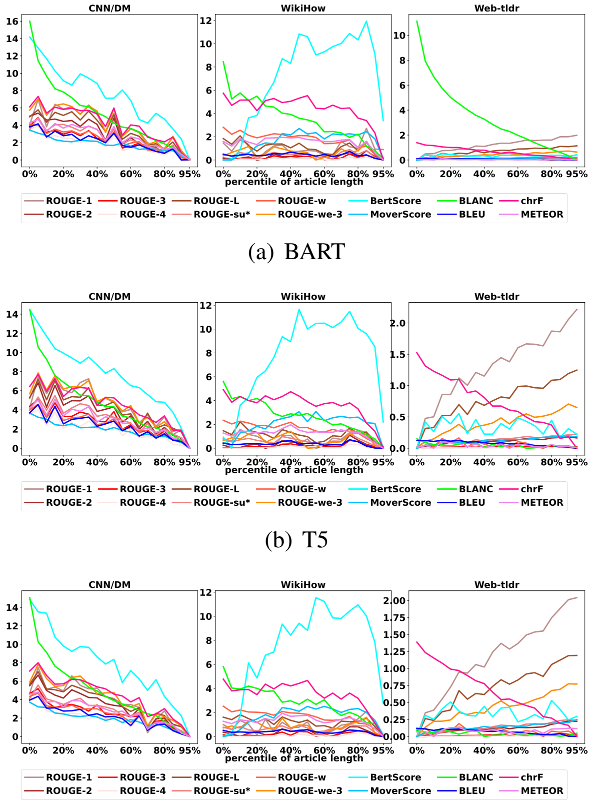
A.3 Details of the Models

Currently, encoder-decoder and decoder-only models with attention mechanisms dominate summarization tasks. Thus, we employ two encoder-decoder attention models, BART (Lewis et al., 2020) and Longformer (Beltagy et al., 2020), and one decoder-only attention model, T5 (Raffel et al., 2020), across all datasets.

B Impact of Source Article and Reference Length

We investigate the influence of source articles and reference lengths on generated summary length, as reported in the main paper. Our procedure is similar but only controls for the length of generated summaries. Findings are displayed in Figures B1 and B2, for source article and reference lengths, respectively.

Comparing model trends on the same dataset reveals that the impact of source article length is mostly consistent across models (except for BLANC on Web-tldr). While correlations between source article length and metrics exist, no uniform trends are observed across datasets. For example, all metrics decrease as source article length increases for CNN/DM dataset. Conversely, WikiHow dataset trends vary by metric, exhibiting increases (e.g., BLANC), decreases (e.g., BERTScore), or no clear patterns. Web-tldr exhibits trends similar to WikiHow. The effect of reference length on metrics yields comparable conclusions.



(a) BART

(b) T5

(c) LongFormer

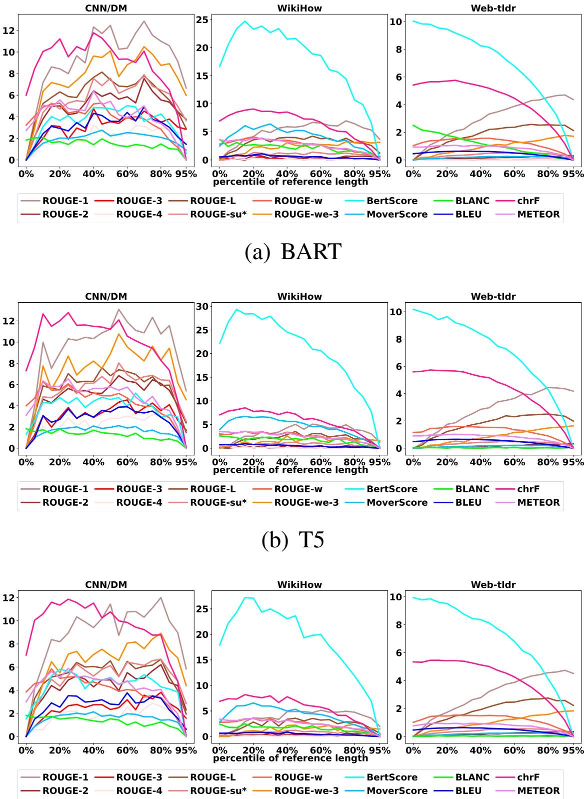
Figure B1: The relation between source article length and different metrics when controlling for the length of generated summaries.

These results emphasize the significance of accounting for source article and reference lengths in our primary analysis presented in the main paper.

C Experiment Settings

C.1 Computing Infrastructure

In our experiments, we utilized 2 Lambda machines with 250 GB of memory, 4 RTX 6000 GPUs, and 64 CPU cores. The operating system of the machine is Ubuntu 20.04. Our experiments are conducted with Python 3.8.10. The CUDA version is 11.2 and the GPU Driver Version is 460.73. The details about the packages can be seen in the 'requirements.txt' file in the supplementary material.



(a) BART

(b) T5

(c) LongFormer

Figure B2: The relation between reference length and different metrics when controlling for the length of generated summaries.

C.2 Hyperparameters and Random Seed

In our experiments, all random seeds are set to 0. We utilize the "Hugging Face" implementation for fine-tuning the language models. During the fine-tuning, because of the limits of the GPU memory, we set the batch size to 16. The training epoch is set to be 10 with early stopping settings. All the other hyperparameters for the training process are set to be the default value of the package.

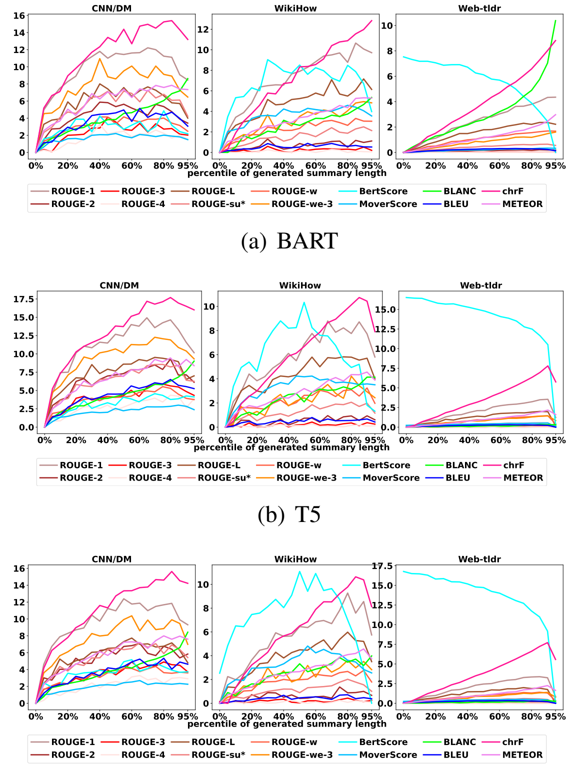
For the summarization metrics, we utilize the package SummEval by Fabbri et al. (2021), and follow all the instructions from that paper.

C.3 Steps for Reproducing our Results

As part of the supplementary material, we have included the code for reproducing our results. Please follow the "readme.md" file to reproduce the results.

D Results of Experiments in Section 3 for Other Models

Figure D1, shows the results of the experiments in Section 3 for all three models. We observe that the trends of a single metric on one dataset are consistent across models.



(a) BART

(b) T5

(c) LongFormer

Figure D1: The average values reported by different metrics for different generated summary lengths with different models, when controlling the length of source article and reference. Note that scores have been minimum-shifted so that the lowest score will be 0. For clarity, the scores have been rescaled from 0-1 to 0-100.

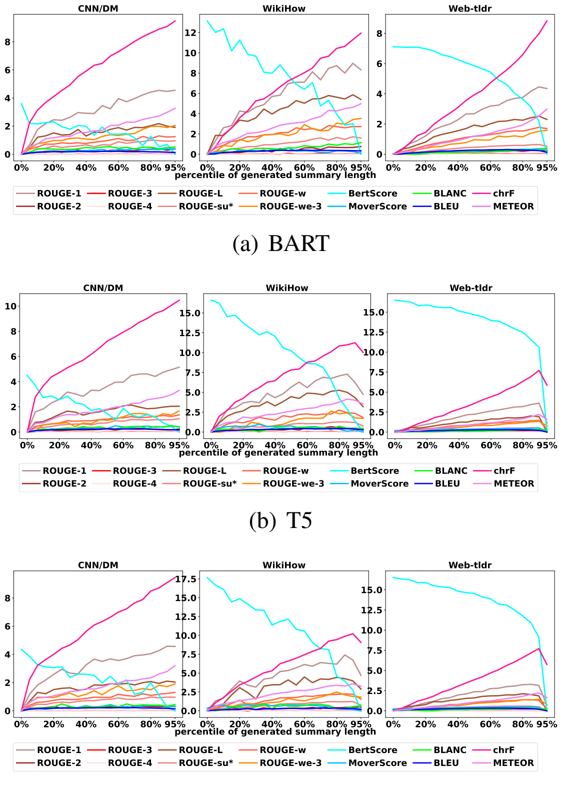
Figure D2, shows the same results for the shuffled experiments. Similarly, we observe that the patterns are consistent across models.

E Additional Details for the Score Adjustment Experiments

E.1 Model Details

For the experiments on reducing length bias in Section 5 of the main paper, we utilize 14 models which are tested on the CNN/DM dataset in their corresponding original papers. These models are:

- **Pointer Generator** (See et al., 2017) proposed an encoder-decoder network variant in which the decoder may either replicate words from the source articles or generate words depending on the embeddings. In addition, it employs a mechanism that can track the summary material to avoid duplication.



(a) BART

(b) T5

(c) LongFormer

Figure D2: The average values reported by different metrics for different generated summary lengths with different models, when controlling the length of source article, reference and the summarization quality by shuffling the summaries. Note that scores have been minimum-shifted so that the lowest score will be 0. For clarity, the scores have been rescaled from 0-1 to 0-100.

- **NEUSUM** (Zhou et al., 2018) offered an end-to-end neural network system for document summarization extraction by simultaneously learning to score and pick phrases.

- **RNES** (Wu and Hu, 2018) proposed a neural coherence model to capture semantic and syntactic coherence patterns across sentences. This is a reinforcement learning model with the neural coherence model and ROUGE package output merged as the reward.

- **Abs-rl** (Chen and Bansal, 2018) presented a methodology that first chooses salient sentences and then abstractly rewrites them to generate a compact overall summary. Non-differentiable computations between these two phases are bridged hierarchically using sentence-level policy-based reinforcement learning based on ROUGE-L reward.

- **Bottom-Up** (Gehrmann et al., 2018) established a bottom-up strategy that inhibits abstractive summarizers' capacity to replicate terms from the source articles.

- **Improved-abs** (Kryściński et al., 2018) enhanced the encoder-decoder model by complementing the decoder with an external LSTM language model and by including a reinforcement learning object during training.

- **Unified-ext-abs** (Hsu et al., 2018) suggested utilizing the sentence-level probability output of an extractive model to adjust the word-level attention scores of an abstractive model. An inconsistency loss is created to promote consistency between these two levels of attention.

- **ROUGESal-Entail** (Pasunuru and Bansal, 2018) proposed a keyphrase-based salience reward in addition to the ROUGE metric and an entailment-based reward in reinforcement learning.

- **Multi-task with Ent and QG** (Guo et al., 2018) increased the performance of abstractive approaches by adding the additional tasks of question generation and entailment generation. The former teaches the summarization model how to explore question-worthy features, while the latter teaches it how to write a summary.

- **Closed book decoder** (Jiang and Bansal, 2018) built upon a Pointer Generator Network by adding an extra decoder without attention and pointer mechanisms to enhance the memorizing capabilities of the encoder in the original network.

- **GPT-2** (Ziegler et al., 2019) utilized human reward learning to natural language tasks throughout the reinforcement learning process.

- **T5** (Raffel et al., 2020) performed a thorough analysis of transfer learning approaches and applied their findings to a collection of text-to-text generation tasks, including summarization.

- **BART** (Lewis et al., 2020) introduced a denoising autoencoder for sequence-to-sequence model pretraining. This model is based on the standard Tranformer-based neural machine translation architecture and combines many denoising techniques.

- **Pegasus** (Zhang et al., 2020) provided a novel objective intended for summarization during the pretraining phase. This objective removes/masks significant sentences from the input material and generates them from the remaining sentences.

F Comparison of the Best Performance of the Models

Table F1 shows the best-performing versions of each model from Figure 5.

	Ori	RN	LR	BN
ROUGE-1	**0.36**	0.25	0.41	0.30
ROUGE-2	0.19	0.23	**0.45**	0.36
ROUGE-3	0.3	0.41	0.43	**0.47**
ROUGE-4	0.36	0.41	0.36	**0.43**
ROUGE-L	**0.12**	**0.12**	**0.12**	**0.12**
ROUGE-su*	0.32	**0.45**	0.38	0.36
ROUGE-w	0.03	0.08	0.16	**0.23**
ROUGE-we-3	0.05	0.12	0.16	**0.19**
BERTScore	**0.27**	0.12	0.25	**0.27**
MoverScore	**0.32**	**0.32**	**0.32**	**0.32**
BLANC	0.08	**0.30**	0.08	0.08
BLEU	0.27	0.25	0.27	**0.41**
chrF	0.34	0.25	0.43	**0.62**
METEOR	0.30	0.19	0.45	**0.61**

Table F1: Comparison of the best performance averaged across all 14 metrics, with adjustment scale $\alpha = 2^{-2}$ for the Bayesian Network (BN) method and $\alpha = 2^{-1}$ for Linear Regression (LR). The original unadjusted scores (Ori) and other methods do not depend on α.