Abstractive Document Summarization with Summary-length Prediction

Jingun Kwon[1], **Hidetaka Kamigaito**[1,2], **and Manabu Okumura**[1]
[1]Tokyo Institute of Technology
[2]Nara Institute of Science and Technology (NAIST)
kwon.j.ad@m.titech.ac.jp
kamigaito.h@is.naist.jp
oku@pi.titech.ac.jp

Abstract

Recently, we can obtain a practical abstractive document summarization model by fine-tuning a pre-trained language model (PLM). Since the pre-training for PLMs does not consider summarization-specific information such as the target summary length, there is a gap between the pre-training and fine-tuning for PLMs in summarization tasks. To fill the gap, we propose a method for enabling the model to understand the summarization-specific information by predicting the summary length in the encoder and generating a summary of the predicted length in the decoder in fine-tuning. Experimental results on the WikiHow, NYT, and CNN/DM datasets showed that our methods improve ROUGE scores from BART by generating summaries of appropriate lengths. Further, we observed about 3.0, 1,5, and 3.1 point improvements for ROUGE-1, -2, and -L, respectively, from GSum on the WikiHow dataset. Human evaluation results also showed that our methods improve the informativeness and conciseness of summaries.

1 Introduction

Current abstractive summarization models mostly utilize pre-trained language models (PLMs) (Liu and Lapata, 2019; Dou et al., 2021; Liu and Liu, 2021; Narayan et al., 2021; Liu et al., 2022a). Abstractive document summarization requires an encoder to determine the important parts in an input text and a decoder to output a non-redundant summary of the appropriate length relevant to the input. Thus, the characteristics required for an abstractive summarization model differ from those required as a language model, and are not usually considered in the pre-training for PLMs (Devlin et al., 2019; Zhang et al., 2019; Lewis et al., 2020). Hence, we need to fine-tune a PLM with a summarization dataset to treat it as an abstractive summarization model. Unlike training a randomly initialized model, this fine-tuning maintains and inherits the parameters learned as an original language model. Therefore, to learn an abstractive summarization model by fine-tuning a PLM, it is necessary to suppress its characteristics as a language model while enabling it to learn the unique properties of abstractive summarization.

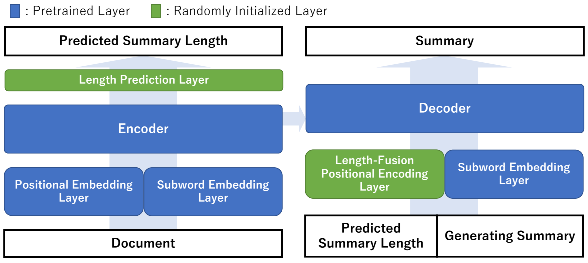

Figure 1: Overview of our methods. The length prediction layer predicts the summary length. The length-fusion positional encoding layer controls the decoder to generate a summary of the appropriate summary length.

For this purpose, we propose two regularization methods for fine-tuneing a PLM to learn abstractive summarization. Figure 1 shows an overview of our methods. The first method is a regularization method that uses the encoder's hidden states to predict the length of an output summary. When the length is not given for a summary to be generated, we believe it is difficult to determine what volume of important key contents to select from the original document. Thus, fixing the length for a summary can make it easier to select key contents for it. We think humans can also create more informative and concise summaries when a summary length is given. The system should also be better trained for selecting key contents in the original document for a summary in case when it can be provided with the length of the summary.

The second method provides the decoder with the length predicted by the first method and enables it to learn to output a summary of the length. In addition to regularizing the training of the decoder, this method reduces the search space by searching

Findings of the Association for Computational Linguistics: EACL 2023, pages 618–624
May 2-6, 2023 ©2023 Association for Computational Linguistics

only for summaries of the appropriate length during generation, and so it is expected to produce a concise and informative summary. Although there have been studies on adjusting the output length of summaries, they have focused on controlling the output length for a given desired length (Kikuchi et al., 2016; Liu et al., 2018; Takase and Okazaki, 2019; Makino et al., 2019; Saito et al., 2020; Yu et al., 2021).[1] We incorporate a target-length prediction task to the encoder side and then inject the predicted length to the decoder side to generate the final summary.

In an evaluation on the WikiHow, NYT, and CNN/DM datasets, our methods improve the ROUGE scores of BART with appropriate lengths of summaries. On the WikiHow dataset, the performance improvement reached about 3.0, 1,5, and 3.1 points for ROUGE-1, -2, and -L, respectively, from GSum. Human evaluation results also showed that our methods enable the fine-tuning for a PLM to generate informative and concise summaries.

Our contributions are as follows: (1) We propose a regularization method that uses the encoder's hidden states by predicting the length of a summary. (2) We propose a regularization method that reduces the search space by injecting the predicted length of a summary. (3) Both automatic and human evaluation results show that our novel model that combines (1) and (2) can generate a summary closer to its gold summary length by improving informativeness.

2 Our Methods

We apply our regularization methods to a transformer-based (Vaswani et al., 2017) PLM to generate a summary from a given document.

2.1 Predicting Summary Length

We impose summary-length prediction on the encoder during fine-tuning to make it easier for the encoder to determine how much important information the given document contains. The encoder converts a sequence of n tokens in a document $\mathbf{x} = \{x_1, x_2, ..., x_n\}$ into hidden states $\{h_1, h_2, ..., h_n\}$. Note that h_n is a hidden state of an end-of-document symbol x_n.

Then, we propose the length-prediction layer by using h_n and a 2-layer feed-forward neural network u to predict the summary length, which is the number of subwords in the summary, as follows:

$$\ell_{pred} = u(h_n). \tag{1}$$

After that, by using the root-mean-square error (RMSE), the regularization loss for the encoder $\mathcal{L}_{len}$ is calculated as follows:

$$\mathcal{L}_{len} = \sqrt{(\ell_{pred} - \ell_{gold})^2}, \tag{2}$$

where ℓ_{gold} is the gold length of the target summary.

2.2 Generating a Summary with the Predicted Length

We provide the decoder with the predicted summary length to generate a concise summary of the appropriate length relevant to the given document.

To encode the information of the predicted length into the decoder while keeping its pre-trained information, we insert our Length-Fusion Positional Encoding layer (LFPE), which is a transformer layer, before the decoder. Our LFPE consists of the length-ratio positional encoding (LRPE) (Takase and Okazaki, 2019) and a transformer layer. LRPE converts the position information of an output token y_t at time t to a continuous vector p_t with considering the predicted length ℓ_{pred} as follows:

$$p_t = \begin{cases} \sin(t/\ell_{pred}^{2i/dim}) & (i \equiv 0 \ (mod\ 2)) \\ \cos(t/\ell_{pred}^{2i/dim}) & (i \equiv 1 \ (mod\ 2)), \end{cases} \tag{3}$$

where dim is the dimension size of the embedding.

Then, the transformer layer converts $\{p_1, p_2, ..., p_t\}$ into $E_t = \{e_1, e_2, ..., e_t\}$ at a decoding time-step t. When adopting LFPE, we replace the original sinusoidal positional encoding of the pre-trained decoder with E_t. After that, the decoder calculates the output probability of y_t as $P(y_t|y_{t-1}, \cdots, y_1, \mathbf{x}, \ell_{pred})$.

Finally, the regularization loss for the decoder $\mathcal{L}_{gen}$ is calculated as follows:

$$\mathcal{L}_{gen} = -\sum_{t=1}^{m} \log P(y_t|y_{t-1}, \cdots, y_1, \mathbf{x}, \ell_{pred}), \tag{4}$$

where m is the number of tokens in the target summary. Note that we replace ℓ_{pred} with ℓ_{gold} in the decoder during training.

[1] Previous work assumes the desired length is given.

Dataset	Training	Valid	Test
WikiHow	168,126 (47.2)	6,000 (45.2)	6,000 (45.4)
NYT	44,382 (28.9)	5,523 (31.2)	6,495 (30.9)
CNN/DM	287,084 (20.5)	13,367 (25.1)	11,490 (22.0)

Table 1: Statistics of document summarization datasets. The value in parentheses indicates the variance of target summary lengths.

Model	R-1	R-2	R-L	VAR	AVG
	WikiHow				
PEGASUS$_{\mathrm{LARGE}}$*	43.06	19.71	41.35	-	
GSum*	41.74	17.73	40.09	-	
GSum	42.04	18.03	40.47	1.38	61.3
BART	$\underline{42.05}$	$\underline{18.06}$	$\underline{40.50}$	$\underline{1.34}$	57.5
BART w/ $\mathrm{R}_{\mathrm{enc}}$	44.68†	19.48†	43.31†	0.98†	51.5
BART w/ $\mathrm{R}_{\mathrm{enc+dec}}$	**45.02**†	**19.53**†	**43.56**†	**0.82**†	54.4
	NYT				
GSum	$\underline{57.63}$	$\underline{37.74}$	$\underline{41.99}$	1.62	151.8
BART	57.32	37.63	41.88	$\underline{1.55}$	149.3
BART w/ $\mathrm{R}_{\mathrm{enc}}$	57.50	37.67	41.92	1.43†	146.8
BART w/ $\mathrm{R}_{\mathrm{enc+dec}}$	**58.52**†	**38.65**†	**43.48**†	**0.89**†	129.9
	CNN/DM				
PEGASUS$_{\mathrm{LARGE}}$*	44.17	21.47	41.11	-	
GSum*	45.94	22.32	42.48	-	
GSum	**45.79**	**22.21**	**42.37**	$\underline{0.76}$	69.7
BART	44.48	21.41	41.19	0.78	70.7
BART w/ $\mathrm{R}_{\mathrm{enc}}$	44.59	21.40	41.07	0.59†	64.3
BART w/ $\mathrm{R}_{\mathrm{enc+dec}}$	44.65	21.60	41.25	**0.36**†	51.0

Table 2: Experimental results on `WikiHow`, `NYT`, and `CNN/DM`. † indicates the improvement is significant (p<0.05) compared with the best baseline score (underlined) on each dataset. ∗ indicates the reported score in the original paper. AVG indicates the average generated summary length.

2.3 Objective Function

To balance the encoder and decoder regularization, we sum the two losses through a hyperparameter λ for calculating the final loss as follows:

$$\mathcal{L} = \mathcal{L}_{gen} + \lambda \cdot \mathcal{L}_{len}. \qquad (5)$$

3 Experiments

3.1 Experimental Settings

Datasets: We used `WikiHow` (Koupaee and Wang, 2018) in the knowledge base domain and `NYT`[2] (Sandhaus, 2008) and `CNN/DM` (Hermann et al., 2015) in the news domain. Table 1 shows the dataset statistics.

Evaluation Metrics: We used F-scores of ROUGE-1 (R-1), -2 (R-2), and -L (R-L) in our experiments. To evaluate the quality of the predicted length and the length-controllability, we employed the length variance (VAR): $\mathrm{VAR} = 0.001 \times \frac{1}{\mathrm{n}} \sum_{\mathrm{i=1}}^{\mathrm{n}} |\mathrm{y}_{\mathrm{pred}} - \mathrm{y}_{\mathrm{gold}}|$, where y_{pred} is the length of the generated summary and y_{gold} is the length of the reference summary in word level, respectively.

Compared Methods: We used BART-large (Lewis et al., 2020) for constructing baselines and our models by following the previous work (Dou et al., 2021). The proposed models are as follows. **BART w/ $\mathbf{R}_{\mathbf{enc}}$** employs our method only for the encoder in §2.1. **BART w/ $\mathbf{R}_{\mathbf{enc+dec}}$** employs our methods both for the encoder and the decoder. The baseline models are as follows. **BART** and **PEGASUS** (Zhang et al., 2019) are the original pre-trained BART and PEGASUS. **GSum** (Dou et al., 2021) is a BART-based combination model that utilizes extracted sentences as a guidance signal to consider extractive aspects for a summary. For the guidance signal, it uses the MatchSum model (Zhong et al., 2020).

We followed the hyperparameters of **BART** and **GSUM** for training and testing the baselines and our models. We set λ to 0.1, 0.05, and 0.05 for `WikiHow`, `NYT`, and `CNN/DM`, respectively, on the basis of validation performances.[3]

3.2 Automatic Evaluation

The results are shown in Table 2. We can see that both of our models, BART w/ $\mathrm{R}_{\mathrm{enc}}$ and BART w/ $\mathrm{R}_{\mathrm{enc+dec}}$, showed significant improvement in ROUGE scores over BART on `WikiHow`. These scores were higher than the combination model of GSUM and PEGASUS (Zhang et al., 2019), which yields the current best results reported on `WikiHow`. We analyzed relations between lengths and ROUGE scores. When our BART w/ $\mathrm{R}_{\mathrm{enc+dec}}$ predicted summary lengths closer to gold summary lengths than BART, 95.4% of generated summaries from ours obtained higher R-1 scores than BART. In addition, VAR and AVG scores show that our models can generate summaries closer to the gold summary lengths and can actually reduce the search space in decoding steps. These results indicate that the proposed methods enable BART to generate highly abstractive summaries of appropriate lengths.

We can also confirm that the proposed methods improved summarization performance over BART on `NYT`[4] and `CNN/DM`. We can also see that

[2] Detailed pre-processing steps are described in Appendix A.

[3] Further details are described in Appendix B.

[4] There is no reported result for PEGASUS on NYT. For GSum, since the pre-processing could not be made identical, the reported and our scores were a bit different.

Model	WikiHow		CNN/DM	
	Info	Con	Info	Con
GSUM	-	-	3.97	4.02
BART	4.00	**4.22**	3.98	4.02
BART w/ $\mathrm{R}_{\mathrm{enc+dec}}$	**4.09**†	4.19	**4.05**†	**4.07**

Table 3: Human evaluation results. The notations are the same as in Table 2.

our model BART w/ $\mathrm{R}_{\mathrm{enc+dec}}$ showed significant improvement in ROUGE scores over GSUM on NYT. Although GSUM outperformed our BART w/ $\mathrm{R}_{\mathrm{enc+dec}}$ in ROUGE scores on CNN/DM, it could generate summaries closer to the gold summary lengths.

Thus, we tried to investigate what types of datasets our methods can work better on and found that the variance of reference summary lengths might be related to the performance of our models. Based on the observations from Tables 1 and 2, our BART w/ $\mathrm{R}_{\mathrm{enc+dec}}$ can largely improve performances on summarization datasets with a high variance of summary lengths, such as WikiHow and NYT.

3.3 Human Evaluation

For human evaluation, we sampled 100 documents each from WikiHow and CNN/DM. By using Amazon Mechanical Turk, we assigned 40 evaluators who obtained both US high school and US bachelor's degrees to each dataset for grading the results with scores from 1 to 5 (5 is the best) in terms of informativeness (Info) and conciseness (Con).

Table 3 shows the results. These results indicate that BART w/ $\mathrm{R}_{\mathrm{enc+dec}}$ generated more informative summaries than BART, that is consistent with the results from the automatic evaluation. In some cases, the generated summaries with BART are just short summaries on WikiHow due to a high variance of reference summary lengths, and so the Con score is slightly lower than the one for BART w/ $\mathrm{R}_{\mathrm{enc+dec}}$. However, BART w/ $\mathrm{R}_{\mathrm{enc+dec}}$ yields the best overall Info and Con scores, which shows our regularization methods are essential for fine-tuning a PLM to learn abstractive summarization models. We also evaluated GSUM together. BART attained a 0.01 better score for Info than GSUM even on CNN/DM since GSUM focuses on generating faithful summaries with injecting outputs from an extractive summarization model.

We investigated the tendency of the length of generated summaries. Figure 2 shows the relationship between gold and generated summary lengths for each model. We used WikiHow because it contains various target summary lengths. When we injected the gold summary length, the length of generated summaries from LFPE (Gold) was almost the same as the gold summaries. These results indicate that LFPE can precisely control various output lengths.[5] In addition, generated summary lengths from BART w/ $\mathrm{R}_{\mathrm{enc+dec}}$ show that the length-prediction layer can also predict various target summary lengths.

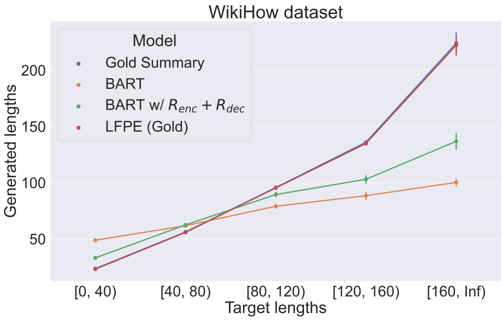

Figure 2: For x-axis, we divided the gold target lengths into 5 bins with 40 words interval. Y-axis indicates the length of generated summaries.

BART	use this method if you have a digital multimeter with a diode check function. set your multimeter to resistance mode. plug the leads into the correct ports. disconnect the diode from the circuit. touch the leads in the forward-bias direction. lower the resistance range if the result is 0. test the resistance in the reverse direction. test a new diode or a working diode.
BART w/ $\mathrm{R}_{\mathrm{enc+dec}}$	set your multimeter to resistance mode. plug the leads into the multimeter. disconnect the diode from the circuit. touch the leads in the forward-bias direction. test in the reverse direction. try a new diode.
Gold	use this method when necessary. set your multimeter to resistance mode. plug in the leads. disconnect the diode. measure the forward bias. measure the reverse bias. compare to a working diode.

Table 4: Example summaries generated from BART w/ $\mathrm{R}_{\mathrm{enc+dec}}$, BART, and gold summaries on WikiHow.

Table 4 shows example generated summaries with BART w/ $\mathrm{R}_{\mathrm{enc+dec}}$, BART, and gold summaries on WikiHow. The summary length prediction is essential for creating an informative and concise summary that is closer to the gold summary length.

4 Related Work

In summary length control, previous work mostly focuses on controlling models for generating summaries with a predefined length (Kikuchi et al.,

[5] Further details are described in Appendix C.

2016; Liu et al., 2018; Takase and Okazaki, 2019; Makino et al., 2019; Saito et al., 2020; Yu et al., 2021). Our work is novel because it enables a model dynamically predicts the appropriate summary length from the input text without relying on any predefined length.

From the viewpoint of regularization, we can see such a regularization term like L_{len} in recent works of summarization tasks. Kamigaito et al. (2018); Kamigaito and Okumura (2020) in sentence compression and Ishigaki et al. (2019) in extractive document summarization incorporate dependency tree information into the attention (Kamigaito et al., 2017). Hsu et al. (2018) integrate extractive and abstractive summarization. MatchSum (Zhong et al., 2020) considers the semantic similarity between a document and its extracted summary. BRIO (Liu et al., 2022a) takes multiple similar abstractive summaries into account by contrastive learning in sequence-to-sequence (Edunov et al., 2018). Different from these works, our approach focuses on summary lengths through L_{len} and can be incorporated into these works by adding L_{len} to their loss function.

5 Conclusion

To fine-tune a pre-trained language model for abstractive document summarization, we proposed a regularization method that uses the encoder's hidden states to predict the length of an output summary. We also proposed LFPE, that focuses on generating a summary with a given target length while keeping pre-trained information of the transformer-based model. We used LFPE to regularize the decoder during training to generate a summary with the predicted length.

Automatic evaluation results showed that the proposed methods enable BART to generate summaries of appropriate lengths while improving ROUGE scores. Human evaluation results also showed that the proposed methods enable BART to generate more informative and concise summaries.

6 Limitations

Although our models can largely improve performances on datasets with a high variance of summary lengths, the gain was small for datasets with a low variance of summary lengths. In the future, we will consider external resources to predict a summary length for the datasets with a low variance of target summary lengths. We plan to form document clusters based on each topic since different topics may have different reference lengths. We believe this may improve performances for the datasets with a low variance of summary lengths.

Acknowledgements

We would like to gratefully acknowledge the anonymous reviewers for their helpful comments and feedbacks. This work was supported by Google AI Focused Research Award.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. GSum: A general framework for guided neural abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4830–4842, Online. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Classical structured prediction losses for sequence to sequence 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 355–364, New Orleans, Louisiana. Association for Computational Linguistics.

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. MIT Press.

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, Melbourne, Australia. Association for Computational Linguistics.

Tatsuya Ishigaki, Hidetaka Kamigaito, Hiroya Takamura, and Manabu Okumura. 2019. Discourse-aware hierarchical attention network for extractive single-document summarization. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 497–506, Varna, Bulgaria. INCOMA Ltd.

Hidetaka Kamigaito, Katsuhiko Hayashi, Tsutomu Hirao, and Masaaki Nagata. 2018. Higher-order syntactic attention network for longer sentence compression. 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 1716–1726, New Orleans, Louisiana. Association for Computational Linguistics.

Hidetaka Kamigaito, Katsuhiko Hayashi, Tsutomu Hirao, Hiroya Takamura, Manabu Okumura, and Masaaki Nagata. 2017. Supervised attention for sequence-to-sequence constituency parsing. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 7–12, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Hidetaka Kamigaito and Manabu Okumura. 2020. Syntactically look-ahead attention network for sentence compression. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8050–8057.

Chris Kedzie, Kathleen McKeown, and Hal Daumé III. 2018. Content selection in deep learning models of summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1818–1828, Brussels, Belgium. Association for Computational Linguistics.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1328–1338, Austin, Texas. Association for Computational Linguistics.

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

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.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. 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 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Yixin Liu and Pengfei Liu. 2021. SimCLS: A simple framework for contrastive learning of abstractive 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 2: Short Papers)*, pages 1065–1072, Online. Association for Computational Linguistics.

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

Yizhu Liu, Qi Jia, and Kenny Zhu. 2022b. Length control in abstractive summarization by pretraining information selection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6885–6895, Dublin, Ireland. Association for Computational Linguistics.

Yizhu Liu, Zhiyi Luo, and Kenny Zhu. 2018. Controlling length in abstractive summarization using a convolutional neural network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4110–4119, Brussels, Belgium. Association for Computational Linguistics.

Takuya Makino, Tomoya Iwakura, Hiroya Takamura, and Manabu Okumura. 2019. Global optimization under length constraint for neural text summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1039–1048, Florence, Italy. Association for Computational Linguistics.

Shashi Narayan, Yao Zhao, Joshua Maynez, Gonçalo Simões, Vitaly Nikolaev, and Ryan McDonald. 2021. Planning with learned entity prompts for abstractive summarization. *Transactions of the Association for Computational Linguistics*, 9:1475–1492.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Itsumi Saito, Kyosuke Nishida, Kosuke Nishida, Atsushi Otsuka, Hisako Asano, Junji Tomita, Hiroyuki Shindo, and Yuji Matsumoto. 2020. Length-controllable abstractive summarization by guiding with summary prototype.

Evan Sandhaus. 2008. Ldc corpora. In *Linguistic Data Consortium*.

Sho Takase and Naoaki Okazaki. 2019. Positional encoding to control output sequence length. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*

(Long and Short Papers), pages 3999–4004, Minneapolis, Minnesota. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Zhongyi Yu, Zhenghao Wu, Hao Zheng, Zhe XuanYuan, Jefferson Fong, and Weifeng Su. 2021. LenAtten: An effective length controlling unit for text summarization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 363–370, Online. Association for Computational Linguistics.

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

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online. Association for Computational Linguistics.

A Statistics of the datasets

NYT dataset consists of articles from the New York Times and the associated summaries.[6] we followed the previous preprocessing step and splitting (Kedzie et al., 2018). There are two types of the reference summaries, which are archival abstracts and online teaser meants. From this collection, we take all articles that have a concatenated summary length of at least 100 words.

B Model details

We introduce the detailed information of the baseline and our models.

We used Fairseq[7] (Ott et al., 2019) for the model implementation. As the pretrained weight, we used `bart-large` in huggingface[8]. We used the original implementation for GSum[9]. We ran training for the models on two NVIDIA Tesla V100 with the multi-GPU setting. As described in the experimental settings, all hyperparameters were the same as for the large-scale BART in Lewis et al. (2020). Hyperparameter λ was set to 0.1, 0.05 and 0.05 for the WikiHow, CNN/DM, and NYT datasets, respectively, on the basis of validation performances.

Model	**R-1**	**R-2**	**R-L**	**VAR**
GOLC* (Makino et al., 2019)	38.27	16.22	34.99	5.13
PALUS* (Yu et al., 2021)	39.82	17.31	36.20	0.01
LPAS* (Saito et al., 2020)	43.23	20.46	40.00	-
PtLAAM* (Liu et al., 2022b)	44.17	20.63	40.97	-
BART	44.48	21.41	41.19	0.78
LRPE (Takase and Okazaki, 2019)	45.67	22.11	42.20	**0.03**
LFPE (Our)	**45.93**†	**22.30**	**42.44**†	**0.03**

Table 5: Experimental results on `CNN/DM` with using the gold summary length information. The notations are the same as in Table 2.

Δ	**Generated Summary**
+1	She and her husband are celebrating their 10th wedding anniversary.
0	She and her husband are celebrating their 10th anniversary.
-1	She and her husband are now married 10 years.

Table 6: Example summaries generated from BART with LFPE for different lengths on `CNN/DM`. $\mathbf{\Delta} = +1/-1$ indicates the injected length is larger/smaller than the gold summary.

C Length-controllability

We investigated the length-controllability of our LFPE in §2.2 by comparing it with the original BART and LRPE. We also compared these methods with the previously reported scores of GOLC, PALUS, LPAS, and PtLAAM. We used `CNN/DM` and gave the gold summary length to the models by following the previous work. The results in Table 5 show that LFPE outperformed other methods in terms of ROUGE scores and VAR. Thus, our LFPE can control the output summary length while keeping ROUGE scores and outperform the state-of-the-art length-controllable methods.

Next, we analyzed the effect of length-controllability in actually generated summaries in `CNN/DM`. Table 6 shows example generated summaries with injecting different lengths into LFPE. In this example, when there is no possibility of dropping a subword, our model paraphrases "10th" to "10" while maintaining the informativeness and grammaticality. From this observation, we can understand that our LFPE controls the summary length through subword-based paraphrasing, which is supported by the decoder's ability of abstraction.

[6] `https://catalog.ldc.upenn.edu/LDC2008T19`

[7] `https://github.com/pytorch/fairseq`

[8] `https://huggingface.co/facebook/bart-large`

[9] `https://github.com/neulab/guided_summarization`