Controlling Summarization Length Through EOS Token Weighting

Anonymous ACL submission

Abstract

Controlling the length of generated text can be crucial in various text generation tasks including summarization. Existing methods often require complex model alterations, limiting compatibility with pre-trained models. We address these limitations by developing a simple approach for controlling the length of automatic text summaries by increasing the importance of correctly predicting the EOS token in the cross entropy loss computation. The proposed methodology is agnostic to architecture and decoding algorithm and orthogonal to other inference-time techniques for controlling generation length, allowing for powerful hybrid combinations. We test it with encoder-decoder and modern GPT-style LLMs. We show that our method can consistently control generation length without affecting the quality of the summary.

1 Introduction

Text summarization is the task of condensing essential information from longer texts in a shorter summary. Extractive text summarization methods create summaries by taking the most representative sentences from the original text, whereas abstractive text summarization focuses on generating completely new texts (Witbrock and Mittal, 1999). This task finds applications in various domains such as news (Hermann et al., 2015b), scientific papers (Luhn, 1958), conversations (Gliwa et al., 2019), and review (Hu and Liu, 2004) summarization.

Summarization tasks tend to be accompanied by various constraints, often dictated by an application or product requirements. Example of these constraints are capping the maximum length of the generated text, using specific keywords in the summary, following a specific format or style (Fan et al., 2018).

Furthermore, despite the rise of large language models like ChatGPT or GPT-4 (OpenAI, 2023), we speculate (and confirm in Section 5) that simpler models can offer comparable summarization quality at a lower cost, making research in this field still relevant.

In this work, we focus on controlling length in abstractive text summarization. This problem is motivated by the necessity to meet interface requirements, such as element sizes in mobile applications. In this context, summaries need to be of a desired length to fit into the page to optimize user experience.

To address this problem, we introduce a novel method of controlling summary length which involves weighting the end-of-sentence (EOS) token more than other tokens at training time. Intuitively, this allows the model to focus on correctly predicting when to stop the generation, thus inducing it to respect the summary length distribution in its training data. We conduct experiments on two model families and multiple decoding strategies to show that our method not only is able to control generation length without compromising the quality of the summary, but also is transferable among architectures and complementary to other inference-time length controlling techniques.

2 Previous work

Methods for controlling the length of generated text can be categorized into two groups: learning-based and decoding-based approaches. While learning-based methods entail alterations to the training architecture or loss function, decoding-based methods operate during the inference phase.

Decoding-based techniques often involve preventing the model from producing the EOS token by assigning it a probability of negative infinity and truncating the text once the desired token count is achieved (Rush et al., 2015), or by incorporating a length penalty into the beam-search decoding algorithm (Murray and Chiang, 2018).

On the other hand, learning-based methods adapt the attention mechanism to be more sensitive to length (Yu et al., 2021; Liu et al., 2022) or train specialized embeddings that factor in the desired length of the generated text (Kikuchi et al., 2016; Fan et al., 2017; Liu et al., 2018; Takase and Okazaki, 2019). In addition, Makino et al. (2019) devised a modification of the objective function that boosts the effectiveness of embedding-based methods, thus showing that modifications of the training architecture and of the objective function are complementary to each other. Many of these techniques, however, entail intricate implementation steps and necessitate training new models from scratch, making them less feasible for integration with pre-trained models.

Notable exceptions to this constraint are the work of Miculicich et al. (2023), who fine-tuned a pre-trained model with reversed positional encodings and showed competitive results both in terms of summary quality and length, and that of Chan et al. (2021) and Jie et al. (2023) who used a Markov decision process and reinforcement learning respectively, to control the generation length.

In line with this research trajectory, our method can be applied to train a new model from scratch as well as to fine-tune pre-trained models. We refrain from altering the underlying architecture, and instead adopt a straightforward modification of the objective function which enhances our ability to govern generation length without compromising quality.

3 Methodology

The intuition behind our method lies in the special importance of the EOS token during training. We note that the cross-entropy loss calculated on that particular token is the only loss component directly teaching the model to respect the summary length distribution in its training data. During the computation of the loss, the signal from that particular token gets diluted by the averaging operation among all other generated tokens, which depending on the dataset can range in number from a few dozens to a few hundreds.

We therefore hypothesize that simply boosting the weight of that loss component will help the model follow the training length distribution more closely, without significantly affecting overall performance. To be precise, our work aims at enforcing an upper bound for the generation length, which is why we are only interested in disproportionately penalising false negatives when predicting EOS token (the loss component when the ground truth **is** EOS). The exact weight to be applied is a hyperparameter on which we run an ablation study.

In formal terms, we start from the original form of the cross entropy loss calculated over the sequence:

$$L_1 = -\frac{1}{N}\sum_{n=1}^{N} \log \frac{e^{x_n^{y_n}}}{\sum_{v=1}^{|V|} e^{x_n^v}} \quad (1)$$

where V is the vocabulary, N is the sequence length, y_n the ground truth token at time-step $n \in (1, N)$ and x_n^v is the logit for token $v \in V$ at time-step n. We then add a weighting term to derive:

$$L_2 = -\frac{1}{N}\sum_{n=1}^{N} w_{y_n} \log \frac{e^{x_n^{y_n}}}{\sum_{v=1}^{|V|} e^{x_n^v}} \quad (2)$$

where

$$w_{y_n} = \begin{cases} W, & \text{if } y_n = [EOS] \\ 1, & \text{otherwise} \end{cases}$$

The weight of the EOS token W is a hyperparameter that controls the balance between semantics and length: when $W = 1$, L_2 goes back to treating EOS just as another token ($L_1 = L_2$); as $W \to \infty$ the loss assigns higher importance to not missing the EOS token, thus making its predicted sequences increasingly short (potentially at the expense of quality).

4 Experiments

Because our method requires training datasets with summaries that respect the desired length constraint, we create subsets of **CNN/Daily Mail** (Hermann et al. (2015a) and See et al. (2017)) and **Wikihow** (Koupaee and Wang (2018)). Specifically we set a hard constraint on summary length of 235 characters for CNN/Daily Mail and 125 characters for Wikihow. We pick different thresholds to explore whether our method's performance depends on it. From each filtered dataset, we sample 10,000, 500, and 500 instances for training, validation, and test, respectively. We call this the *Fixed Length approach*. We apply it by fine-tuning pre-trained models and showing that the resulting models are consistently better able to generate summaries within the character count limitation.

A natural extension of our method, which circumvents the need for manually curating datasets with a specific summary length, would be prepending the instruction 'Summarize with up to {K} characters the following text:' to each sample in the dataset, where K is the exact number of characters in the reference summary. This would induce the model to "learn to count" the number of characters at inference time, thus being able to generate summaries of any desired length. We then fine-tune the LLM with the same procedure explained in Section 3 and call this the *Dynamic Length approach*. We test this hypothesis on a subset of CNN/Daily Mail with 100k samples filtered to have summaries shorter than 800 characters. For every sample, we prepend the prompt above and, for simplicity, round num_chars up to the closest number in the range from 75 to 800 with a stride of 50.

4.1 Base Models and Hyperparameters

For the Fixed Length approach, we fine-tune the pre-trained T5-base (Raffel et al., 2020) and Llama-2 7B (Touvron et al., 2023) models. We evaluate EOS token weight with the values of 1 (baseline) and 10. We compare two decoding strategies: greedy decoding, and beam search with 5 beams and length penalty values of -1, 0 and 1.

As baselines, we use gpt-3.5-turbo and gpt-4 by OpenAI[1] on the CNN/Daily Mail dataset with default generation parameters and the following prompt template: "{source_text}. Summarize with up to 235 characters."[2].

For the Dynamic Length approach, we only fine-tune the pre-trained T5-base and evaluate against gpt-3.5-turbo and gpt-4.

4.2 Metrics

As metrics, we report (a) **ROUGE-2** (Lin, 2004): a relevance score for text generation tasks which relies on the intersection of bi-grams between the reference and prediction; (b) **BERTScore** (Zhang et al., 2019): a semantic similarity score calculated using contextual embeddings from a pre-trained BERT model, in our case RoBERTa-large (Liu et al., 2019); (c) **Percent of too long summaries**: the percent of generated summaries that exceed the number of character limitation. This is our primary metric.

[1]https://openai.com/

[2]We also tried prepending the instruction but obtained slightly worse results

5 Results

Our main results are shown in Tables 1 and 2. We observe that our method always controls length better than the baseline, across architectures and decoding strategies. Furthermore, for Llama-2 7B, our method also improves summary quality across all settings, both in terms of Rouge-2 and BertScore. For T5-base our results on summary quality are inconclusive: sometimes the metrics slightly improve and other times slightly degrade. Note that deviations in quality metrics are small in comparison to the improvements achieved in terms of summary length.

We observe that the positive effects of our method are consistent across decoding strategies and, in particular, are present even when beam search with length penalty[3] is used, proving that our method is indeed orthogonal to inference-time length control techniques. Moreover, it's worth noting that the baseline models, namely gpt-3.5-turbo and gpt-4, failed to adhere to the specified length constraints provided via prompts. Both models demonstrated inferior performance compared to our fine-tuned Llama-2 7B across all the metrics we monitored, and T5-base across all metrics except BertScore.

	Rouge-2		BertScore		% of too long	
	w=1	w=10	w=1	w=10	w=1	w=10
T5-base						
Greedy	14.7	**15.2**	23.7	**24.7**	27.0	**10.0**
Beam_{-1}	**15.1**	14.9	23.8	**24.1**	15.6	**8.8**
Beam_{0}	**15.0**	14.0	23.4	**24.0**	20.8	**13.0**
Beam_{+1}	**14.7**	14.5	21.6	**21.8**	63.4	**60.8**
Llama-2 7B						
Greedy	16.0	**16.0**	**32.1**	32.0	15.0	**6.6**
Beam_{-1}	15.3	**15.6**	28.4	**28.5**	3.8	**1.0**
Beam_{0}	15.2	**15.6**	28.4	**28.5**	4.4	**1.8**
Beam_{+1}	15.2	**15.7**	28.2	**28.5**	6.0	**2.6**
OpenAI						
gpt-3.5	12.8		26.2		51.6	
gpt-4	13.0		26.1		48.2	

Table 1: Results for modified CNN/Daily Mail, Fixed Length approach (235 characters). The subscripts in Beam denote the value of the length penalty parameter.

Table 3 shows the effectiveness of our methodology for the Dynamic Length setting. We observe better adherence to the length constraint than the baseline, and better or equal performance in terms of summary quality. Note that the results in Tables 1 and 3 cannot be meaningfully compared as

[3]The *lp* parameter is actually a length reward as implemented in HuggingFace, i.e. positive values penalise short, rather than long generations

	Rouge-2		BertScore		% of too long	
	w=1	w=10	w=1	w=10	w=1	w=10
T5-base						
Greedy	**18.0**	17.6	48.9	48.9	6.6	**3.4**
Beam$_{-1}$	**18.2**	17.5	**49.4**	47.8	3.2	**1.4**
Beam$_{0}$	**18.4**	17.6	**49.3**	49.1	4.0	**2.2**
Beam$_{+1}$	17.0	**17.1**	43.5	**47.1**	22.2	**9.6**
Llama-2 7B						
Greedy	22.1	**23.1**	53.4	**54.0**	3.2	3.2
Beam$_{-1}$	22.4	**23.2**	53.6	**54.1**	2.8	**1.6**
Beam$_{0}$	22.6	**23.3**	53.8	**54.1**	2.8	**1.6**
Beam$_{+1}$	22.6	**23.2**	53.5	**53.8**	3.8	**2.2**

Table 2: Results for modified Wikihow, Fixed Length approach (125 characters).

the length constraints for the Dynamic length setting are more granular and therefore significantly harder to respect compared to the single constraint in the Fixed Length setting.

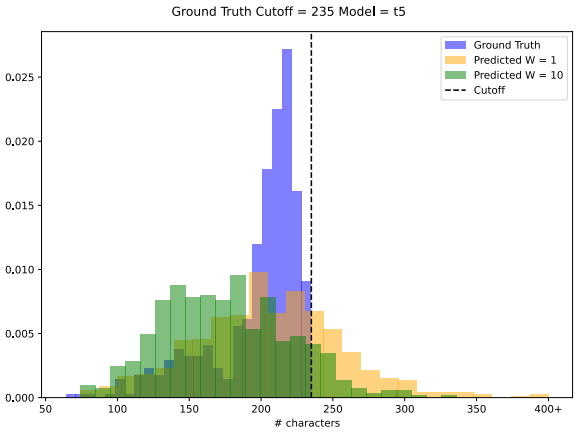
	Rouge-2		BertScore		% of too long	
	w=1	w=10	w=1	w=10	w=1	w=10
T5-base						
Greedy	15.9	**16.0**	25.5	**25.7**	35.6	**22.4**
Beam$_{-1}$	16.0	**16.4**	25.4	25.4	22.8	**7.0**
Beam$_{0}$	16.1	16.1	**25.0**	24.8	30.6	**15.0**
Beam$_{+1}$	15.8	15.8	23.8	**24.0**	63.6	**46.8**
OpenAI						
gpt-3.5	12.7		26.8		42.6	
gpt-4	12.5		26.8		33.2	

Table 3: Results for CNN/Daily Mail, Dynamic Length approach with `num_chars in range(start=75, stop=800, step=50)`.

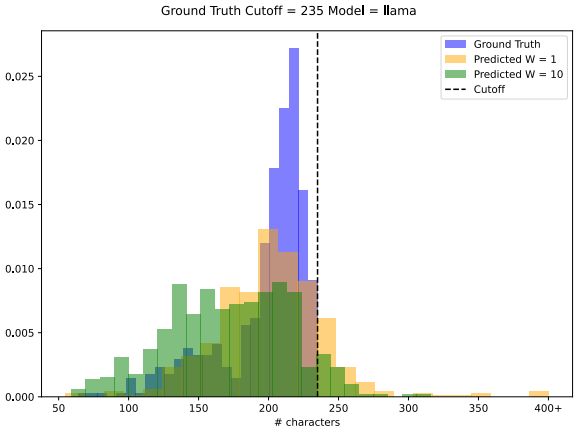
w	Rouge-2	BertScore	% of long
T5-base			
1	14.7	23.7	27.0
2	14.9	24.3	19.4
5	**15.4**	**25.0**	11.6
10	15.2	24.7	**10.0**
50	14.8	24.2	10.4
1e7	11.9	22.1	0
Llama-2 7B			
1	16.0	32.1	15.0
2	15.4	31.9	10.8
5	**16.5**	**32.5**	8.4
10	16.0	32.0	**6.6**
50	16.0	32.1	7.8

Table 4: Results for modified CNN/Daily Mail, fixed length (235 characters), different EOS weights and greedy decoding.

Table 4 shows how metrics differ across several W settings. As expected, higher values result in better length control by shifting the distribution of generated length to the left as shown in Figure 1a and Figure 1b. However we note there are diminishing returns after a certain value of W which in our



(a) T5-base



(b) Llama-2 7B

Figure 1: Length distributions of predicted test summaries with different EOS weights. For visualisation purposes we truncated the lengthier baseline ($W = 1$) distributions: the actual maximum lengths were 639 and 852 characters for T5 and LLAMA-2 respectively.

setting lies somewhere between 10 and 50. Finally, in Appendix A we demonstrate the effectiveness of the Fixed Length approach across datasets of different sizes and with different cutoffs.

6 Conclusions

This paper presents a novel, easy to implement method for limiting the length of generations without significant quality degradation. We provided evidence that our method works across different model architectures (encoder-decoder and decoder only), different datasets and in combination with other inference time length control methods.

7 Limitations

While our method shows promising results, certain limitations need to be acknowledged. Firstly, our evaluation was conducted on a limited set of

models, which may not fully represent the diversity of available architectures and configurations. Future work should include ablation studies across a broader range of models to ascertain the generalizability and robustness of our approach.

Secondly, the fine-tune process required for pre-trained language models incurs significant computational costs, potentially limiting the scalability and accessibility of our method compared to approaches that solely rely on inference-time operations.

8 Ethics Statement

This research adheres to the principles outlined in the ACL Ethics Policy. We comply with the licenses of all used datasets. Despite we did not use sensitive data to train our models, we encourage all summarization application developers using our method to honor the ethical code for conducting linguistic and cognitive research. (Mao et al., 2022) found that pre-trained language models might suffer from certain biases. Since our methodology suggests to fine-tune such models, we cannot guarantee that the resultant models produce unbiased outputs.

References

Hou Pong Chan, Lu Wang, and Irwin King. 2021. Controllable summarization with constrained Markov decision process. *Transactions of the Association for Computational Linguistics*, 9:1213–1232.

Angela Fan, David Grangier, and Michael Auli. 2017. Controllable abstractive summarization. *arXiv preprint arXiv:1711.05217*.

Angela Fan, David Grangier, and Michael Auli. 2018. Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54, Melbourne, Australia. Association for Computational Linguistics.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015a. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.

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

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, page 168–177, New York, NY, USA. Association for Computing Machinery.

Renlong Jie, Xiaojun Meng, Lifeng Shang, Xin Jiang, and Qun Liu. 2023. Prompt-based length controlled generation with reinforcement learning. *arXiv preprint arXiv:2308.12030*.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. *arXiv preprint arXiv:1609.09552*.

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

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

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. *CoRR*, abs/1907.11692.

Yizhu Liu, Qi Jia, and Kenny Zhu. 2022. 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.

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.

H. P. Luhn. 1958. The automatic creation of literature abstracts.

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.

Rui Mao, Qian Liu, Kai He, Wei Li, and Erik Cambria. 2022. The biases of pre-trained language models: An empirical study on prompt-based sentiment analysis and emotion detection. *IEEE Transactions on Affective Computing*.

Lesly Miculicich, Yujia Xie, Song Wang, and Pengcheng He. 2023. Summarization with precise length control. *arXiv preprint arXiv:2305.05171*.

Kenton Murray and David Chiang. 2018. Correcting length bias in neural machine translation. *arXiv preprint arXiv:1808.10006*.

OpenAI. 2023. Gpt-4 technical report.

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

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.

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, Vancouver, Canada. Association for Computational Linguistics.

Sho Takase and Naoaki Okazaki. 2019. Positional encoding to control output sequence length. *arXiv preprint arXiv:1904.07418*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Michael J. Witbrock and Vibhu O. Mittal. 1999. Ultra-summarization (poster abstract): a statistical approach to generating highly condensed non-extractive summaries. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, page 315–316, New York, NY, USA. Association for Computing Machinery.

Zhongyi Yu, Zhenghao Wu, Hao Zheng, Zhe XuanYuan, Jefferson Fong, and Weifeng Su. 2021. Lenatten: An effective length controlling unit for text summarization. *arXiv preprint arXiv:2106.00316*.

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

A Ablations

A.1 Dataset Size

To demonstrate that the effectiveness of our method is independent from the size of the fine-tuning dataset, we test it on subsamples of CNN/Daily Mail of different sizes: 2k, 10k, 50k samples. Table 5 shows that our method yields a consistent improvement in length control with minimal to no performance degradation across all dataset sizes. We note, however, that as the dataset size increases the difference between our method and the baseline reduces, thus indicating that increasing dataset size positively contributes to better length control and is complementary to our method.

	Rouge-2		BertScore		% of too long	
	w=1	w=10	w=1	w=10	w=1	w=10
T5-base						
2k samples	**14.7**	14.3	23.5	**24.3**	38.6	**18.8**
10k samples	14.7	**15.2**	23.7	**24.7**	27.0	**10.0**
50k samples	15.5	**15.7**	25.4	**25.6**	19.4	**12.2**
Llama-2 7B						
2k samples	14.4	**15.7**	28.3	**31.8**	45.0	**10.0**
10k samples	16.0	16.0	**32.1**	32.0	15.0	**6.6**
50k samples	**16.7**	16.5	**35.7**	33.3	13.2	**3.0**

Table 5: Results for modified CNN/Daily Mail with 2k, 10k, 50k samples and reference summary length of at most 235 characters.

A.2 Reference summary length

We also test our method on subsets of CNN/Daily Mail with 10k samples and different maximum lengths for the reference summary, namely 175, 235, and 500 characters. We do this in order to demonstrate that the method is portable across datasets with any summary length. Table 6 shows that our method yields a consistent improvement in length control with minimal to no performance degradation across all dataset sizes also in this case.

	Rouge-2		BertScore		% of too long	
	w=1	w=10	w=1	w=10	w=1	w=10
T5-base						
175 chars	13.2	**13.5**	24.8	**25.1**	17.6	**10.4**
235 chars	14.7	**15.2**	23.7	**24.7**	27.0	**10.0**
500 chars	**16.3**	16.2	24.4	**24.6**	8.6	**0.0**
Llama-2 7B						
175 chars	15.6	**16.1**	32.8	**33.1**	12.2	**3.2**
235 chars	16.0	16.0	**32.1**	32.0	15.0	**6.6**
500 chars	**16.6**	16.1	**31.7**	31.3	7.8	**1.4**

Table 6: Results for modified CNN/Daily Mail with 10k samples and reference summary lengths of at most 175, 235, and 500 characters.