Positional Encoding to Control Output Sequence Length

Sho Takase **Naoaki Okazaki**
Tokyo Institute of Technology
{sho.takase@nlp.c, okazaki@c}.titech.ac.jp

Abstract

Neural encoder-decoder models have been successful in natural language generation tasks. However, real applications of abstractive summarization must consider additional constraint that a generated summary should not exceed a desired length. In this paper, we propose a simple but effective extension of a sinusoidal positional encoding (Vaswani et al., 2017) to enable neural encoder-decoder model to preserves the length constraint. Unlike in previous studies where that learn embeddings representing each length, the proposed method can generate a text of any length even if the target length is not present in training data. The experimental results show that the proposed method can not only control the generation length but also improve the ROUGE scores.

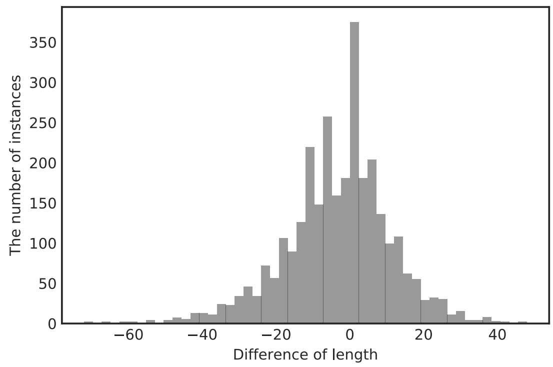

Figure 1: Difference in number of characters between correct headlines and outputs of a widely used LSTM encoder-decoder (Luong et al., 2015) which is trained on sentence-headline pairs created by Rush et al. (2015) from the annotated English Gigaword corpus. The difference was investigated for 3,000 sentence-headline pairs randomly sampled from the test splits.

1 Introduction

Neural encoder-decoder models have been successfully applied to various natural language generation tasks including machine translation (Sutskever et al., 2014), summarization (Rush et al., 2015), and caption generation (Vinyals et al., 2015). Still, it is necessary to control the output length for abstractive summarization, which generates a summary for a given text while satisfying a space constraint. In fact, Figure 1 shows a large variance in output sequences produced by a widely used encoder-decoder model (Luong et al., 2015), which has no mechanism for controlling the length of the output sequences.

Fan et al. (2018) trained embeddings that correspond to each output length to control the output sequence length. Since the embeddings for different lengths are independent, it is hard to generate a sequence of the length that is infrequent in training data. Thus, a method that can model any lengths continuously is required.

Kikuchi et al. (2016) proposed two learning based methods for an LSTM encoder-decoder: LenEmb and LenInit. LenEmb inputs an embedding representing the remaining length in each decoding step. Since this approach also prepares embeddings for each length independently, it suffers from the same problem as that in Fan et al. (2018).

On the other hand, LenInit can handle arbitrary lengths because it combines the scalar value of a desired length with a trainable embedding. LenInit initializes the LSTM cell of the decoder with the embedding depending on the scalar value of the desired length. Liu et al. (2018) incorporated such scalar values into the initial state of the decoder in a CNN encoder-decoder. These approaches deal with any length but it is reasonable to incorporate the distance to the desired terminal position into each decoding step such as in LenEmb.

In this study, we focused on Transformer (Vaswani et al., 2017), which recently achieved the state-of-the-art score on the machine

Proceedings of NAACL-HLT 2019, pages 3999–4004
Minneapolis, Minnesota, June 2 - June 7, 2019.

translation task. We extend the sinusoidal positional encoding, which represents a position of each token in Transformer (Vaswani et al., 2017), to represent a distance from a terminal position on the decoder side. In this way, the proposed method considers the remaining length explicitly at each decoding step. Moreover, the proposed method can handle any desired length regardless of its appearance in a training corpus because it uses the same continuous space for any length.

We conduct experiments on the headline generation task. The experimental results show that our proposed method is able to not only control the output length but also improve the ROUGE scores from the baselines. Our code and constructed test data are publicly available at: https://github.com/takase/control-length.

2 Positional Encoding

Transformer (Vaswani et al., 2017) uses a sinusoidal positional encoding to represent the position of an input. Transformer feeds the sum of the positional encoding and token embedding to the input layer of its encoder and decoder. Let pos be the position and d be the embedding size. Then, the i-th dimension of the sinusoidal positional encoding $PE_{(pos,i)}$ is as follows:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right), \quad (1)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right). \quad (2)$$

In short, each dimension of the positional encoding corresponds to a sinusoid whose period is $10000^{2i/d} \times 2\pi$. Since this function returns an identical value at the same position pos, the above positional encoding can be interpreted as representing the absolute position of each input token.

In this paper, we extend Equations (1) and (2) to depend on the given output length and the distance from the terminal position. We propose two extensions: length-difference positional encoding ($LDPE$) and length-ratio positional encoding ($LRPE$). Then we replace Equations (1) and (2) with (3) and (4) (or (5) and (6)) on the decoder side to control the output sequence length. We define $LDPE$ and $LRPE$ as follows:

$$LDPE_{(pos,len,2i)} = \sin\left(\frac{len - pos}{10000^{\frac{2i}{d}}}\right), \quad (3)$$

$$LDPE_{(pos,len,2i+1)} = \cos\left(\frac{len - pos}{10000^{\frac{2i}{d}}}\right), \quad (4)$$

$$LRPE_{(pos,len,2i)} = \sin\left(\frac{pos}{len^{\frac{2i}{d}}}\right), \quad (5)$$

$$LRPE_{(pos,len,2i+1)} = \cos\left(\frac{pos}{len^{\frac{2i}{d}}}\right), \quad (6)$$

where len presents the given length constraint. $LDPE$ returns an identical value at the position where the remaining length to the terminal position is the same. $LRPE$ returns a similar value at the positions where the ratio of the remaining length to the terminal position is similar. Let us consider the d-th dimension as the simplest example. Since we obtain $\sin(pos/len)$ (or $\cos(pos/len)$) at this dimension, the equations yield the same value when the remaining length ratio is the same, e.g., $pos = 5$, $len = 10$ and $pos = 10$, $len = 20$.

We add $LDPE$ (or $LRPE$) to the input layer of Transformer in the same manner as in Vaswani et al. (2017). In the training step, we assign the length of the correct output to len. In the test phase, we control the output length by assigning the desired length to len.

3 Experiments

3.1 Datasets

We conduct experiments on the headline generation task on Japanese and English datasets. The purpose of the experiments is to evaluate the ability of the proposed method to generate a summary of good quality within a specified length. We used JAMUL corpus as the Japanese test set (Hitomi et al., 2019). This test set contains three kinds of headlines for 1,181[1] news articles written by professional editors under the different upper bounds of headline lengths. The upper bounds are 10, 13, and 26 characters ($len = 10, 13, 26$). This test set is suitable for simulating the real process of news production because it is constructed by a Japanese media company.

In contrast, we have no English test sets that contain headlines of multiple lengths. Thus, we randomly extracted 3,000 sentence-headline

[1] We obtained this test set by applying the pre-processing script at https://github.com/asahi-research/Gingo to the original JAMUL corpus.

pairs that satisfy a length constraint from the test set constructed from annotated English Gigaword (Napoles et al., 2012) by pre-processing scripts of Rush et al. (2015)[2]. We set three configurations for the number of characters as the length constraint: 0 to 30 characters ($len = 30$), 30 to 50 characters ($len = 50$), and 50 to 75 characters ($len = 75$). Moreover, we also evaluate the proposed method on the DUC-2004 task 1 (Over et al., 2007) for comparison with published scores in previous studies.

Unfortunately, we have no large supervision data with multiple headlines of different lengths associated with each news article in both languages. Thus, we trained the proposed method on pairs with a one-to-one correspondences between the source articles and headlines. In the training step, we regarded the length of the target headline as the desired length len. For Japanese, we used the JNC corpus, which contains a pair of the lead three sentences of a news article and its headline (Hitomi et al., 2019). The training set contains about 1.6M pairs[3]. For English, we used sentence-headline pairs extracted from the annotated English Gigaword with the same pre-processing script used in the construction of the test set. The training set contains about 3.8M pairs.

In this paper, we used a character-level decoder to control the number of characters. On the encoder side, we used subword units to construct the vocabulary (Sennrich et al., 2016; Kudo, 2018). We set the hyper-parameter to fit the vocabulary size to about 8k for Japanese and 16k for English.

3.2 Baselines

We implemented two methods proposed by previous studies to control the output length and handle arbitrary lengths. We employed them and Transformer as baselines.

LenInit Kikuchi et al. (2016) proposed LenInit, which controls the output length by initializing the LSTM cell m of the decoder as follows:

$$m = len \times b, \tag{7}$$

where b is a trainable vector. We incorporated this method with a widely used LSTM encoder-decoder model (Luong et al., 2015)[4]. For a fair comparison, we set the same hyper-parameters as in Takase et al. (2018) because they indicated that the LSTM encoder-decoder model trained with the hyper-parameters achieved a similar performance to the state-of-the-art on the headline generation.

Length Control (LC) Liu et al. (2018) proposed a length control method that multiplies the desired length by input token embeddings. We trained the model with their hyper-parameters.

Transformer Our proposed method is based on Transformer (Vaswani et al., 2017)[5]. We trained Transformer with the equal hyper-parameters as in the base model in Vaswani et al. (2017).

3.3 Results

Table 1 shows the recall-oriented ROUGE-1 (R-1), 2 (R-2), and L (R-L) scores of each method on the Japanese test set[6]. This table indicates that Transformer with the proposed method (Transformer+$LDPE$ and Transformer+$LRPE$) outperformed the baselines for all given constraints ($len = 10, 13, 26$). Transformer+$LRPE$ performed slightly better than Transformer+$LDPE$. Moreover, we improved the performance by incorporating the standard sinusoidal positional encoding (+PE) on $len = 10$ and 26. The results imply that the absolute position also helps to generate better headlines while controlling the output length.

Table 2 shows the recall-oriented ROUGE scores on the English Gigaword test set. This table indicates that $LDPE$ and $LRPE$ significantly improved the performance on $len = 75$. Moreover, the absolute position (PE) also improved the performance in this test set. In particular, PE was very effective in the setting of very short headlines ($len = 30$). However, the proposed method slightly lowered ROUGE-2 scores from the bare Transformer on $len = 30, 50$. We infer that the bare Transformer can generate headlines whose lengths are close to 30 and 50 because the majority of the training set consists of headlines whose lengths are less than or equal to 50. However, most of the generated headlines breached the length constraints, as explained in Section 3.4.

To investigate whether the proposed method can generate good headlines for unseen lengths, we excluded headlines whose lengths are equal to the

[2]https://github.com/facebookarchive/NAMAS

[3]We obtained this training set by applying the pre-processing script at https://github.com/asahi-research/Gingo.

[4]We used an implementation at https://github.com/mlpnlp/mlpnlp-nmt.

[5]We used an implementation at https://github.com/pytorch/fairseq.

[6]To calculate ROUGE scores on the Japanese dataset, we used https://github.com/asahi-research/Gingo.

	$len = 10$			$len = 13$			$len = 26$		
Model	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Baselines									
LenInit	38.08	17.72	36.84	41.83	19.53	39.22	47.07	22.02	38.36
LC	35.88	15.73	34.80	40.28	18.86	38.16	42.62	19.38	35.61
Transformer	34.63	15.48	33.02	43.94	21.35	40.77	46.43	23.03	38.10
Proposed method									
Transformer+$LDPE$	42.84	21.07	41.31	46.51	22.83	43.76	50.89	24.18	40.82
+PE	42.85	20.67	41.47	46.72	22.70	43.75	**51.32**	**25.15**	**41.48**
Transformer+$LRPE$	42.70	21.62	41.35	**47.05**	**23.70**	**44.13**	50.68	24.70	41.23
+PE	**43.36**	**21.63**	**41.93**	46.39	23.09	43.49	51.21	25.03	41.43
Proposed method trained on the dataset without headlines consisting of target lengths									
Transformer+$LDPE$	41.91	20.01	40.69	45.88	22.61	43.16	50.90	24.37	40.48
+PE	42.33	20.46	40.88	44.78	22.33	42.27	50.87	24.54	40.89
Transformer+$LRPE$	41.91	20.10	40.52	46.01	22.87	43.47	50.33	24.37	41.00
+PE	42.59	20.76	41.16	46.52	23.65	43.81	50.73	24.64	41.01

Table 1: Recall-oriented ROUGE scores for each length on Japanese test set. This test set contains three kinds of headlines, i.e., $len = 10, 13, 26$, tied to a single article.

	$len = 30$			$len = 50$			$len = 75$		
Model	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Baselines									
LenInit	44.58	25.90	43.34	48.42	25.47	45.56	50.78	25.74	46.42
LC	45.17	26.73	44.09	46.56	24.55	44.10	48.67	24.83	44.98
Transformer	47.48	**29.77**	46.17	50.02	**28.04**	47.29	47.31	24.83	43.75
Proposed method									
Transformer+$LDPE$	47.26	26.98	45.77	50.21	26.13	47.15	53.99	27.78	49.24
+PE	48.13	27.18	46.43	50.29	25.97	47.17	53.65	27.65	49.06
Transformer+$LRPE$	48.79	28.77	47.17	50.09	26.08	46.91	53.91	27.82	49.15
+PE	**49.23**	29.26	**47.68**	50.41	26.37	47.39	**54.21**	**27.84**	**49.38**
Proposed method trained on the dataset without headlines consisting of the target lengths									
Transformer+$LDPE$	47.35	26.76	45.70	50.46	25.96	47.30	53.69	27.61	49.04
+PE	47.44	27.42	45.99	50.67	26.07	47.57	53.76	27.53	49.03
Transformer+$LRPE$	48.54	28.89	47.06	50.65	26.19	47.34	53.94	27.88	49.11
+PE	49.08	29.09	47.58	**50.78**	26.64	**47.60**	53.77	27.68	48.93

Table 2: Recall-oriented ROUGE scores for each length on test data extracted from annotated English Gigaword.

Model	R-1	R-2	R-L
Baselines			
LenInit	29.78	11.05	26.49
LC	28.68	10.79	25.72
Transformer	26.15	9.14	23.19
Proposed method			
Transformer+$LDPE$	30.95	10.53	26.79
+PE	31.00	10.78	27.02
+Re-ranking	31.65	11.25	27.46
Transformer+$LRPE$	30.74	10.83	26.69
+PE	31.10	11.05	27.25
+Re-ranking	**32.29**	**11.49**	**28.03**
Previous studies for controlling output length			
Kikuchi et al. (2016)	26.73	8.39	23.88
Fan et al. (2018)	30.00	10.27	26.43
Other previous studies			
Rush et al. (2015)	28.18	8.49	23.81
Suzuki and Nagata (2017)	32.28	10.54	27.80
Zhou et al. (2017)	29.21	9.56	25.51
Li et al. (2017)	31.79	10.75	27.48
Li et al. (2018)	29.33	10.24	25.24

Table 3: Recall-oriented ROUGE scores in DUC-2004.

desired length (len) from the training data. The lower parts of Table 1 and 2 show ROUGE scores of the proposed method trained on the modified training data. These parts show that the proposed method achieved comparable scores to ones trained on whole training dataset. These results indicate that the proposed method can generate high-quality headlines even if the length does not appear in the training data.

Table 3 shows the recall-oriented ROUGE scores on the DUC-2004 test set. Following the evaluation protocol (Over et al., 2007), we truncated characters over 75 bytes. The table indicates that $LDPE$ and $LRPE$ significantly improved the performance compared to the bare Transformer, and achieved better performance than the baselines except for R-2 of LenInit. This table also shows the scores reported in the previous studies. The proposed method outperformed the previous methods that control the output length and achieved the competitive score to the state-of-the-art scores.

Since the proposed method consists of a character-based decoder, it sometimes generated

	Variance					
	Japanese dataset			English Gigaword		
Model	$len = 10$	$len = 13$	$len = 26$	$len = 30$	$len = 50$	$len = 75$
Baselines						
LenInit	0.047	0.144	0.058	0.114	0.112	0.091
LC	0.021	0.028	0.040	0.445	0.521	0.871
Transformer	181.261	115.431	38.169	193.119	138.566	620.887
Proposed method						
Transformer+$LDPE$	**0.000**	**0.000**	**0.000**	**0.015**	0.012	0.013
+PE	0.003	0.001	0.001	0.016	**0.009**	**0.007**
Transformer+$LRPE$	0.121	0.210	0.047	0.082	0.071	0.187
+PE	0.119	0.144	0.058	0.142	0.110	0.173
Proposed method trained on the dataset without headlines consisting of the target lengths						
Transformer+$LDPE$	**0.000**	0.002	**0.000**	0.018	**0.009**	0.009
+PE	0.021	0.001	0.003	0.021	0.013	0.010
Transformer+$LRPE$	0.191	0.362	0.043	0.120	0.058	0.133
+PE	0.183	0.406	0.052	0.138	0.081	0.154

Table 4: Variances of generated headlines.

words unrelated to a source sentence. Thus, we applied a simple re-ranking to each n-best headlines generated by the proposed method ($n = 20$ in this experiment) based on the contained words. Our re-ranking strategy selects a headline that contains source-side words the most. Table 3 shows that Transformer+$LRPE$+PE with this re-ranking (+Re-ranking) achieved better scores than the state-of-the-art (Suzuki and Nagata, 2017).

3.4 Analysis of Output Length

Following Liu et al. (2018), we used the variance of the generated summary lengths against the desired lengths as an indicator of the preciseness of the output lengths. We calculated variance (var) for n generated summaries as follows[7]:

$$var = \frac{1}{n}\sum_{i=1}^{n} |l_i - len|^2, \qquad (8)$$

where len is the desired length and l_i is the length of the generated summary.

Table 4 shows the values of Equation (8) computed for each method and the desired lengths. This table indicates that $LDPE$ could control the length of headlines precisely. In particular, $LDPE$ could generate headlines with the identical length to the desired one in comparison with LenInit and LC. $LRPE$ also generated headlines with a precise length but its variance is larger than those of previous studies in very short lengths, i.e., $len = 10$ and 13 in Japanese. However, we consider $LRPE$ is enough for real applications because the averaged difference between its output and the desired length is small, e.g., 0.1 for $len = 10$.

[7]Liu et al. (2018) multiplies Equation (8) by 0.001.

The lower part of Table 4 shows the variances of the proposed method trained on the modified training data that does not contain headlines whose lengths are equal to the desired length, similar to the lower parts of Table 1 and 2. The variances for this part are comparable to the ones obtained when we trained the proposed method with whole training dataset. This fact indicates that the proposed method can generate an output that satisfies the constraint of the desired length even if the training data does not contain instances of such a length.

4 Conclusion

In this paper, we proposed length-dependent positional encodings, $LDPE$ and $LRPE$, that can control the output sequence length in Transformer. The experimental results demonstrate that the proposed method can generate a headline with the desired length even if the desired length is not present in the training data. Moreover, the proposed method significantly improved the quality of headlines on the Japanese headline generation task while preserving the given length constraint. For English, the proposed method also generated headlines with the desired length precisely and achieved the top ROUGE scores on the DUC-2004 test set.

Acknowledgments

The research results have been achieved by "Research and Development of Deep Learning Technology for Advanced Multilingual Speech Translation", the Commissioned Research of National Institute of Information and Communications Technology (NICT), Japan.

References

Angela Fan, David Grangier, and Michael Auli. 2018. Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation (WMT 2018)*, pages 45–54.

Yuta Hitomi, Yuya Taguchi, Hideaki Tamori, Ko Kikuta, Jiro Nishitoba, Naoaki Okazaki, Inui Kentaro, and Manabu Okumura. 2019. A large-scale multi-length headline corpus for improving length-constrained headline generation model evaluation. *CoRR*.

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 (EMNLP 2016)*, pages 1328–1338.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, pages 66–75.

Haoran Li, Junnan Zhu, Jiajun Zhang, and Chengqing Zong. 2018. Ensure the correctness of the summary: Incorporate entailment knowledge into abstractive sentence summarization. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*, pages 1430–1441.

Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. 2017. Deep recurrent generative decoder for abstractive text summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 2091–2100.

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 (EMNLP 2018)*, pages 4110–4119.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 1412–1421.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated Gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, AKBC-WEKEX '12, pages 95–100.

Paul Over, Hoa Dang, and Donna Harman. 2007. Duc in context. *Information Processing & Management*, 43(6):1506–1520.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 379–389.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 1715–1725.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, pages 3104–3112.

Jun Suzuki and Masaaki Nagata. 2017. Cutting-off redundant repeating generations for neural abstractive summarization. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*, pages 291–297.

Sho Takase, Jun Suzuki, and Masaaki Nagata. 2018. Direct output connection for a high-rank language model. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pages 4599–4609.

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 30 (NIPS 2017)*, pages 5998–6008.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015)*, pages 3156–3164.

Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. 2017. Selective encoding for abstractive sentence summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pages 1095–1104.