Incorporating Noisy Length Constraints into Transformer with Length-aware Positional Encodings

Yui Oka, Katsuki Chousa, Katsuhito Sudoh, and **Satoshi Nakamura**
Nara Institute of Science and Technology
{oka.yui.ov2, sudoh, s-nakamura}@is.naist.jp

Abstract

Neural Machine Translation often suffers from an under-translation problem due to its limited modeling of output sequence lengths. In this work, we propose a novel approach to training a Transformer model using length constraints based on length-aware positional encoding (PE). Since length constraints with exact target sentence lengths degrade translation performance, we add random noise within a certain window size to the length constraints in the PE during the training. In the inference step, we predict the output lengths using input sequences and a BERT-based length prediction model. Experimental results in an ASPEC English-to-Japanese translation showed the proposed method produced translations with lengths close to the reference ones and outperformed a vanilla Transformer by 3.22 points in BLEU on short sentences within ten subwords. The average translation results using our length prediction model were also better than another baseline method using input lengths for the length constraints. The proposed noise injection improved robustness for length prediction errors, especially within the window size.

1 Introduction

In autoregressive Neural Machine Translation (NMT), a decoder generates one token at a time, and each output token depends on the output tokens generated so far. The decoder's prediction of the end of the sentence determines the length of the output sentence. This prediction is sometimes made too early–before all of the input information is translated–causing a so-called under-translation.

Transformer has sinusoidal positional encoding to incorporate the token position information in the sequence into its encoder and decoder (Vaswani et al., 2017). There are some previous studies for controlling an output length in Transformer. Takase and Okazaki (2019) proposed two variants of length-aware positional encodings called length-ratio positional encoding (LRPE) and length-difference positional encoding (LDPE) to control the output length based on the given length constraints in automatic summarization. Lakew et al. (2019) applied LDPE and LRPE to NMT. They trained an NMT model using output length constraints based on LDPE and LRPE along with special tokens representing length ratio classes between input and output sentences, while they used the input sentence length at the inference time. However, the length of an input sentence is not a reliable estimator of the output length, because the actual output length varies with the content of the input.

Using length constraints in the decoder is a promising approach to the under-translation problem. We propose an NMT method based on LRPE and LDPE with a BERT-based output length prediction. The proposed method adds noise to the output length constraints in training to improve its robustness against the possible length variances in the translation. In our experiments with an English-to-Japanese dataset, the BERT-based output length prediction outperformed the use of the input length, and the proposed method, including noise injection into the training-time length constraints, improved the translation performance in BLEU for short sentences.

2 Positional encoding for control output length

The following is the sinusoidal positional encoding (PE) proposed by Vaswani et al. (2017):

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

where pos is the position in the sequence, $2i$ and $2i+1$ respectively represent even and odd dimensions in the PE vector, and d is the dimension of the embeddings. Length-ratio positional encoding (LRPE) considers the remaining length to the terminal position, and length-difference positional encoding (LDPE) considers the ratio of the remaining length to the final position as follows:

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

$$LDPE_{(pos,len,2i)} = sin\left(\frac{len-pos}{10000^{\frac{2i}{d}}}\right), LDPE_{(pos,len,2i+1)} = cos\left(\frac{len-pos}{10000^{\frac{2i}{d}}}\right), \quad (3)$$

where len is the given output sequence length. LRPE and LDPE are expected to generate sentences of any length even if sentences of an exact length are not included in the training data. Takase and Okazaki (2019) used character-based lengths for summarization constraints in the number of characters.

3 Proposed method

The Transformer-based model with LRPE and LDPE generates a sequence that almost matches the given length. This characteristic is not always appropriate in the problem of machine translation because some translation variants have different lengths. In this paper, we incorporate random noise to the length constraints used by LRPE and LDPE during training to improve the robustness for such length variants. We also propose using an output length prediction based on BERT. Noise injection is expected to improve the robustness against possible length prediction errors.

3.1 Random noise injection to LRPE/LDPE length constraints

The existing studies that used LRPE and LDPE used the exact output lengths as length constraints (len in Eqs. 2 and 3) in training. We introduce some random noise into the output lengths in the number of tokens. The noise is given as a random integer from a uniform distribution within a window, such as $[-2, 2]$. For example, we randomly chose an integer from $[-2, -1, 0, 1, 2]$. Although perhaps the same positional encoding vectors might appear in a different position when a negative value is applied as noise, we ignore such cases in this work for simplicity.

3.2 BERT-based output length prediction

We need length estimates when we use LRPE and LDPE in inferences. Instead of using the input lengths like Lakew et al. (2019), we propose using an output length prediction based on a pre-trained BERT model in the source language. We used the [CLS] vector in the last layer of the BERT encoder to predict the output length through an output layer as a regression problem.

4 Experiments

To investigate the performance of our proposed method, we conducted English-to-Japanese translation experiments between a vanilla Transformer and its variants with LRPE and LDPE, implemented using OpenNMT (Klein et al., 2017).

4.1 Setup

Datasets We used the Japanese-English portion of the ASPEC corpus (Nakazawa et al., 2016), which consists of 3 million parallel sentences for training, 1,790 sentences for development, 1,784 sentences for the devtest, and 1,812 sentences for the test. All the sentences were tokenized into subwords using a SentencePiece model (Kudo and Richardson, 2018) with a shared subword vocabulary of 16,000 entries, which were trained with 2M English and Japanese sentences that included the first set of the training sentences pairs (`train-1`). Throughout the experiments, we used subword-based lengths.

Hyperparameters Our hyperparameter settings came from OpenNMT-py FAQ[1] and are used commonly for all the compared methods described later in this section. We conducted five independent training runs with different random seeds and chose the best runs and training epochs in the devtest set to determine the models for the final evaluation.

Evaluation We used BLEU (Papineni et al., 2002) for our evaluation metric given by `multi-bleu.perl` and also investigated the length ratio (LR) of the input and output sentences ($LR = tgt\_len/ref\_len$). BLEU was calculated on translation results re-tokenized by MeCab (Kudo, 2005) after merging the subwords. We also calculated the variance of the length difference between the translation results and the references (VAR) to investigate the effects of the output length constraints, following Takase and Okazaki (2019). The variance of the length differences on a test set consisting of n sentences is given by:

$$VAR = \frac{1}{n}\sum_{i=1}^{n}|l_i - ref\_len_i|^2 \tag{4}$$

4.2 Compared methods

In addition to a vanilla Transformer (Vaswani et al., 2017), we compared three different windows for the length noise that was applied to LRPE and LDPE during training: the use of target lengths without noise, and the random noise within two different windows ($[-2, 2]$, $[-4, 4]$) from a uniform distribution over the integers in the window. We compared two inference-time length constraints: the proposed BERT-based length prediction ($BERT\_pred$) and using the input length (src_len). We also tested the reference lengths (ref_len) to investigate the upper-bound performance by the proposed method.

4.3 Results

Table 1 shows the results of the BLEU, length ratio, and variance by the compared methods.

BLEU The proposed method with the BERT-based length prediction ($BERT\_pred$) and an LDPE with a length noise window of $[-4, 4]$ resulted in a slightly better BLEU score (38.80) than the baseline Transformer (38.42), but the difference was not statistically significant by the bootstrap resampling test. On the other hand, the simple application of LDPE and LRPE resulted in a much lower BLEU score even with the correct reference lengths. This result suggests that the proposed training framework with some noise in the length constraints improved the robustness for length variances. The use of input length (src_len) instead of the output length prediction significantly decreased BLEU in most cases, although the random noise injection provided some improvements and even competitive performance, as shown in the bottom row. The oracle results with reference lengths (ref_len) were better than the baseline and the proposed method, but the BLEU differences were not significant.

Length ratio The length ratios by LDPE and LRPE with reference lengths clearly show that the length constraints induced more extended outputs than the vanilla Transformer, and the noise injection slightly shortened the outputs. Using the BERT-based output length prediction resulted in shorter outputs than the reference length due to the length prediction errors, as discussed below. On the other hand, the input lengths (src_len) induced 20% longer outputs than the references without noise injection. Such over-translation was reduced by the noise injection, although it remained longer in all the cases.

Variance The length error variances showed that LDPE and LRPE induced outputs in closer lengths to the references than the vanilla Transformer. A wider noise window increased the variances, as shown in the rightmost column (ref_len), although such differences became smaller when we used length prediction ($BERT\_pred$), possibly due to the length prediction errors. Using the input lengths resulted in much more significant variances due to relatively weak correlations between the input and output sentences.

[1] `https://opennmt.net/OpenNMT-py/FAQ.html#how-do-i-use-the-transformer-model`

Baseline		BLEU			LR			VAR		
Transformer		38.42			0.916			29.51		
Length-Constrained		$BERT\_pred$			src_len			ref_len		
		BLEU	LR	VAR	BLEU	LR	VAR	BLEU	LR	VAR
LDPE	(no noise)	35.87	0.938	19.91	29.70	1.226	71.83	37.19	1.000	0
LRPE	(no noise)	36.54	0.929	21.71	30.96	1.198	59.65	37.78	0.987	1.497
Proposed										
LDPE	$[-2, 2]$	**38.50**	0.927	20.11	32.81	1.162	55.85	38.72	0.972	2.411
	$[-4, 4]$	**38.80**	0.921	21.15	35.28	1.091	48.25	39.09	0.951	6.429
LRPE	$[-2, 2]$	**38.61**	0.911	23.87	36.34	1.075	35.18	39.28	0.943	7.502
	$[-4, 4]$	**38.44**	0.911	25.12	38.40	1.015	28.06	38.78	0.931	12.506

Table 1: BLEU, length ratio (LR), and variance (VAR) results with different models and inference-time length constraints: BLEU values in **bold** outperformed baseline.

	AVG Error	VAR	Corr	AVG Error	VAR	Corr
	src-ref			predicted-ref		
Train	7.51	99.43	0.86			
Dev	7.06	97.00	0.84			
Test	6.55	72.45	0.90	3.00	19.92	0.93

Table 2: Average error, variance and the Pearson correlation coefficient between the input and reference lengths, and between the predicted and reference lengths (in the number of tokens) in ASPEC dataset

Output length prediction We compared the differences of our length prediction ($BERT\_pred$) and the input lengths (src_len) with the corresponding reference length (ref_len) in the mean absolute error and error variances. The mean absolute error by the proposed method was 3.00, and the variance was 19.92, suggesting the BERT-based length prediction sometimes made serious length prediction errors, although it worked well in most cases. The proposed noise injection covered these relatively small errors tested in the experiments. On the other hand, the mean absolute error and variance by the substituted use of the input length were much more significant: 6.55 and 72.45, respectively. Such differences negatively affect translation results.

Table 2 shows the average and variance of the length prediction errors by the use of the input sentence length, together with the Pearson correlation coefficient. Moreover, Table 2 shows those results by the proposed BERT-based length prediction. They clearly show the use of the input length as a proxy for the output length constraints was not suitable in this experiment.

Analysis in length groups We scrutinized the results with different length groups to investigate the effects of the length noise injection, because the length constraints probably have more significant impact on shorter sentences and vice versa. Note that we excluded the longest length group that exceeded 80 tokens because it includes three sentences and serious length errors. In Table 3, the proposed method with LDPE with a noise window of $[-2, 2]$ significantly outperformed the vanilla Transformer by 3.22 points (50.81 vs. 47.59) in BLEU in the shortest length group with one to ten tokens. The other setups showed better BLEU results than the vanilla Transformer, although the differences were not statistically significant. Another clear finding is that the vanilla Transformer generated very short translation results for long sentences, as shown in the rightmost column; LDPE and LRPE created longer outputs. This finding is helpful for avoiding under-translation problems in NMT.

5 Conclusion

We proposed an NMT method using length-aware positional encodings with a training-time length noise injection and a BERT-based inference-time length prediction. The length noise injection improved the

Model		BLEU (LR)			
		Length range in number of tokens			
		$1 \sim 10$ (118 sentences)	$11 \sim 20$ (636 sentences)	$21 \sim 40$ (890 sentences)	$41 \sim 80$ (165 sentences)
Transformer (Baseline)		47.59 (1.004)	41.24 (0.951)	38.87 (0.920)	31.88 (0.862)
LDPE	(no noise)	40.30 (1.089)	38.10 (0.992)	36.43 (0.926)	30.12 (0.900)
LRPE	(no noise)	45.36 (1.018)	39.26 (0.975)	36.62 (0.923)	31.53 (0.889)
LDPE	$[-2, 2]$	***50.81 (0.997)**	**41.77 (0.966)**	**39.08 (0.971)**	**31.99 (0.976)**
	$[-4, 4]$	**48.05 (0.985)**	**42.38 (0.945)**	**39.46 (0.949)**	**32.54 (0.960)**
LRPE	$[-2, 2]$	**49.80 (0.980)**	**42.06 (0.940)**	**39.71 (0.950)**	**32.98 (0.924)**
	$[-4, 4]$	**49.19 (0.995)**	**42.50 (0.941)**	**38.90 (0.934)**	**32.12 (0.908)**

Table 3: Detailed results in different length ranges in number of tokens in reference sentences: BLEU values in **bold** outperformed baseline and * shows statistically significant difference from baseline Transformer.

robustness for translation length variations, including length prediction errors, especially for those within the noise window size. The experimental results show the effectiveness of the proposed method in short sentences. Our future work will pursue a more effective output length prediction to suppress over-translations because we also found some over-translations in the test set, possibly caused by the length constraints.

Acknowledgments

Part of this work was supported by JSPS KAKENHI Grant Number JP17H06101.

References

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada, July. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium, November. Association for Computational Linguistics.

Taku Kudo. 2005. Mecab : Yet another part-of-speech and morphological analyzer. *http://mecab.sourceforge.net/*.

Surafel Melaku Lakew, Mattia Di Gangi, and Marcello Federico. 2019. Controlling the Output Length of Neural Machine Translation. In *Proceedings of the 16th International Workshop on Spoken Language Translation (IWSLT 2019)*, October.

Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. 2016. ASPEC: Asian Scientific Paper Excerpt Corpus. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 2204–2208, Portorož, Slovenia, may. European Language Resources Association (ELRA).

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, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

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, June. 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 I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.