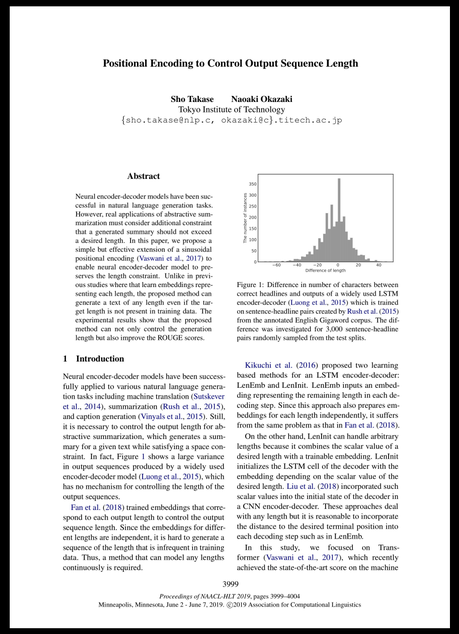
# Text Summarization Models with Length Control in ACL 2019 & NAACL 2019

Zhenghao WU, Jack
Zhongyi YU, Jerry
Hao ZHENG, Lebron

BNU-HKBU **United International College**

**Artificial Intelligence** SIG
Special Interest Group in FinTech, NLP, CV/CG and More



**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.

**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.




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.

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

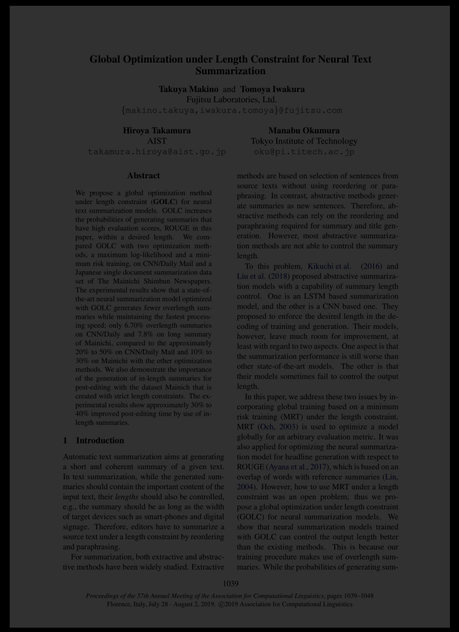


*Proceedings of NAACL-HLT 2019*, pages 3999–4004
Minneapolis, Minnesota, June 2 - June 7, 2019. 

*NAACL-HLT 2019*

# Positional Encoding to Control Output Sequence Length

Sho Takase & Naoaki Okazaki
Tokyo Institute of Technology



**Global Optimization under Length Constraint for Neural Text Summarization**

**Takuya Makino** and **Tomoya Iwakura**
Fujitsu Laboratories, Ltd.
{makino.takuya,iwakura.tomoya}@fujitsu.com

**Hiroya Takamura**
AIST
takamura.hiroya@aist.go.jp

**Manabu Okumura**
Tokyo Institute of Technology
oku@pi.titech.ac.jp

**Abstract**

We propose a global optimization method under length constraint (**GOLC**) for neural text summarization models. GOLC increases the probabilities of generating summaries that have high evaluation scores, ROUGE in this paper, within a desired length. We compared GOLC with two optimization methods, a maximum log-likelihood and a minimum risk training, on CNN/Daily Mail and a Japanese single document summarization data set of The Mainichi Shimbun Newspapers. The experimental results show that a state-of-the-art neural summarization model optimized with GOLC generates fewer overlength summaries while maintaining the fastest processing speed; only 6.70% overlength summaries on CNN/Daily and 7.8% on long summary of Mainichi, compared to the approximately 20% to 50% on CNN/Daily Mail and 10% to 30% on Mainichi with the other optimization methods. We also demonstrate the importance of the generation of in-length summaries for post-editing with the dataset Mainich that is created with strict length constraints. The experimental results show approximately 30% to 40% improved post-editing time by use of in-length summaries.

**1 Introduction**

Automatic text summarization aims at generating a short and coherent summary of a given text. In text summarization, while the generated summaries should contain the important content of the input text, their *lengths* should also be controlled, e.g., the summary should be as long as the width of target devices such as smart-phones and digital signage. Therefore, editors have to summarize a source text under a length constraint by reordering and paraphrasing.

For summarization, both extractive and abstractive methods have been widely studied. Extractive methods are based on selection of sentences from source texts without using reordering or paraphrasing. In contrast, abstractive methods generate summaries as new sentences. Therefore, abstractive methods can rely on the reordering and paraphrasing required for summary and title generation. However, most abstractive summarization methods are not able to control the summary length.

To this problem, Kikuchi et al. (2016) and Liu et al. (2018) proposed abstractive summarization models with a capability of summary length control. One is an LSTM based summarization model, and the other is a CNN based one. They proposed to enforce the desired length in the decoding of training and generation. Their models, however, leave much room for improvement, at least with regard to two aspects. One aspect is that the summarization performance is still worse than other state-of-the-art models. The other is that their models sometimes fail to control the output length.

In this paper, we address these two issues by incorporating global training based on a minimum risk training (MRT) under the length constraint. MRT (Och, 2003) is used to optimize a model globally for an arbitrary evaluation metric. It was also applied for optimizing the neural summarization model for headline generation with respect to ROUGE (Ayana et al., 2017), which is based on an overlap of words with reference summaries (Lin, 2004). However, how to use MRT under a length constraint was an open problem; thus we propose a global optimization under length constraint (GOLC) for neural summarization models. We show that neural summarization models trained with GOLC can control the output length better than the existing methods. This is because our training procedure makes use of overlength summaries. While the probabilities of generating sum-



*Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1039–1048
Florence, Italy, July 28 - August 2, 2019. 

*ACL 2019*

**Global Optimization under Length Constraint for Neural Text Summarization**

Maybe Next Time …

People from
Fujitsu Laboratories, AIST, Tokyo Institute of Technology

# 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.

## 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.

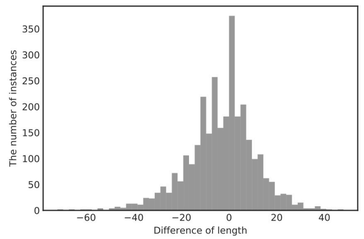



Figure 1: Difference in number of characters between correct headlines and outputs of a widely used LSTM encoder-decoder (Luong et al., 2015) which is trained on sentence-headline pairs created by Rush et al. (2015) from the annotated English Gigaword corpus. The difference was investigated for 3,000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

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. 

| | |
|---|---|
| **Type** | Abstractive |
| **Model** | Transformer |
| **Corpora** | • **JAMUL**(JP-Test), **annotated English Gigaword**(EN_Extracted-Test-3K)<br>• **DUC-2004**(Evaluation)<br>• **JNC**(JP-Train-1.6M), **annotated English Gigaword**(EN_Extracted-Train-3.8M) |
| **Evaluation** | ROUGE(1,2,L), Variances of generated headlines. |

# Paper #1- What is Transformer?

arXiv:1706.03762v5 [cs.CL] 6 Dec 2017

**Attention Is All You Need**


**Ashish Vaswani**[*]
Google Brain
avaswani@google.com

**Noam Shazeer**[*]
Google Brain
noam@google.com

**Niki Parmar**[*]
Google Research
nikip@google.com

**Jakob Uszkoreit**[*]
Google Research
usz@google.com

**Llion Jones**[*]
Google Research
llion@google.com

**Aidan N. Gomez**[*] [†]
University of Toronto
aidan@cs.toronto.edu

**Łukasz Kaiser**[*]
Google Brain
lukaszkaiser@google.com

**Illia Polosukhin**[*] [‡]
illia.polosukhin@gmail.com



**Abstract**

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.


**1 Introduction**

Recurrent neural networks, long short-term memory [13] and gated recurrent [7] neural networks in particular, have been firmly established as state of the art approaches in sequence modeling and

[*]Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

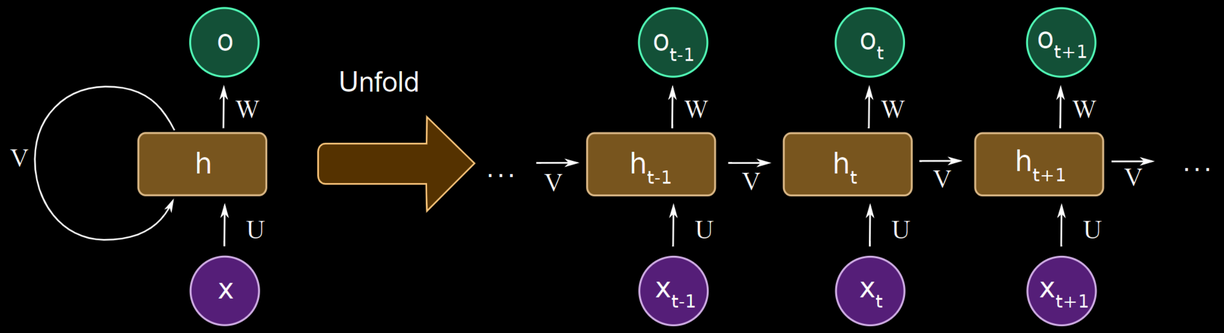
[†]Work performed while at Google Brain.

[‡]Work performed while at Google Research.


31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

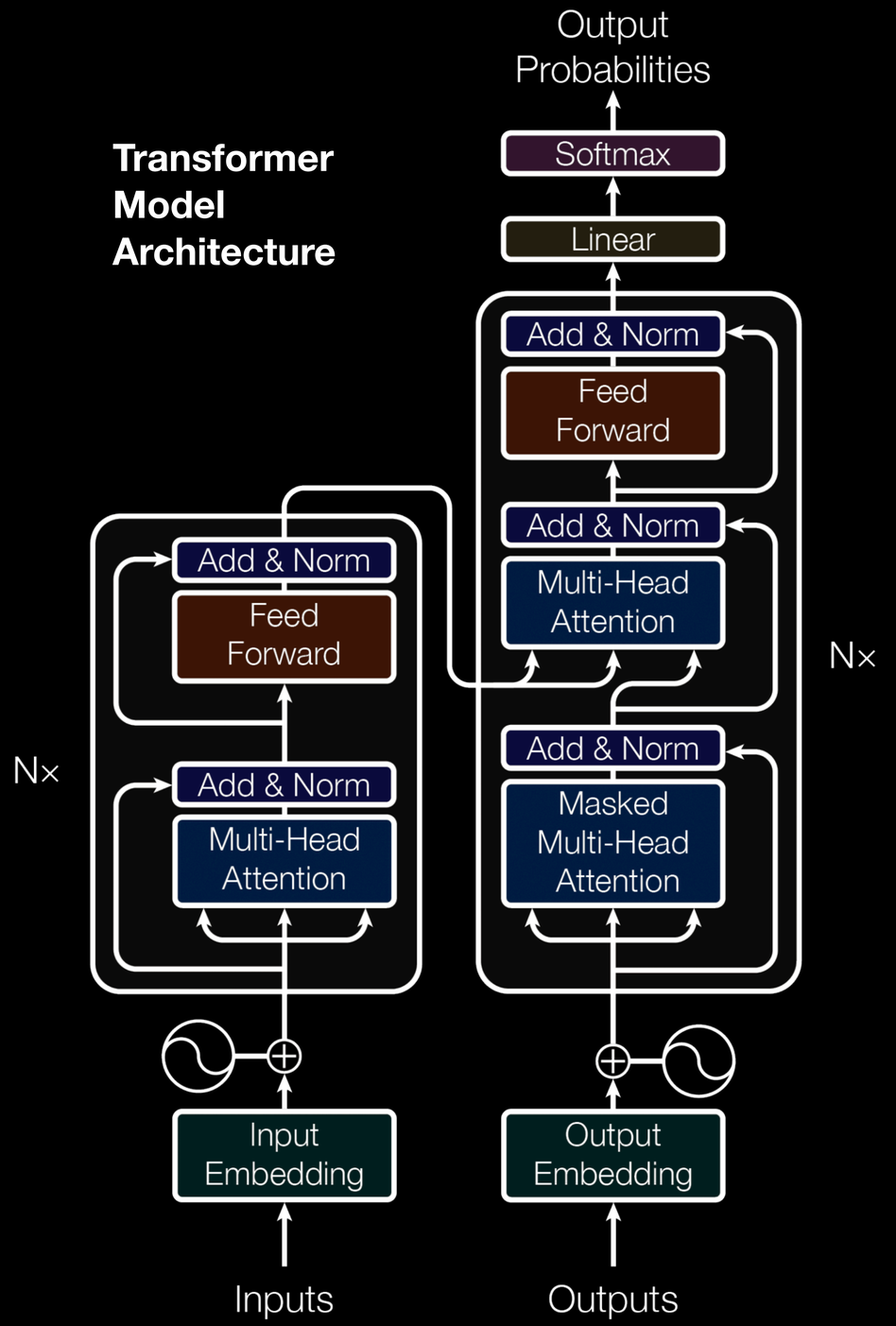

- An new type of neural network architecture
- Proposed in paper ***Attention Is All You Need***(2017)
- Currently considered as a good practice to build Language Model

# Paper #1, Positional Encoding to Control Output Sequence Length



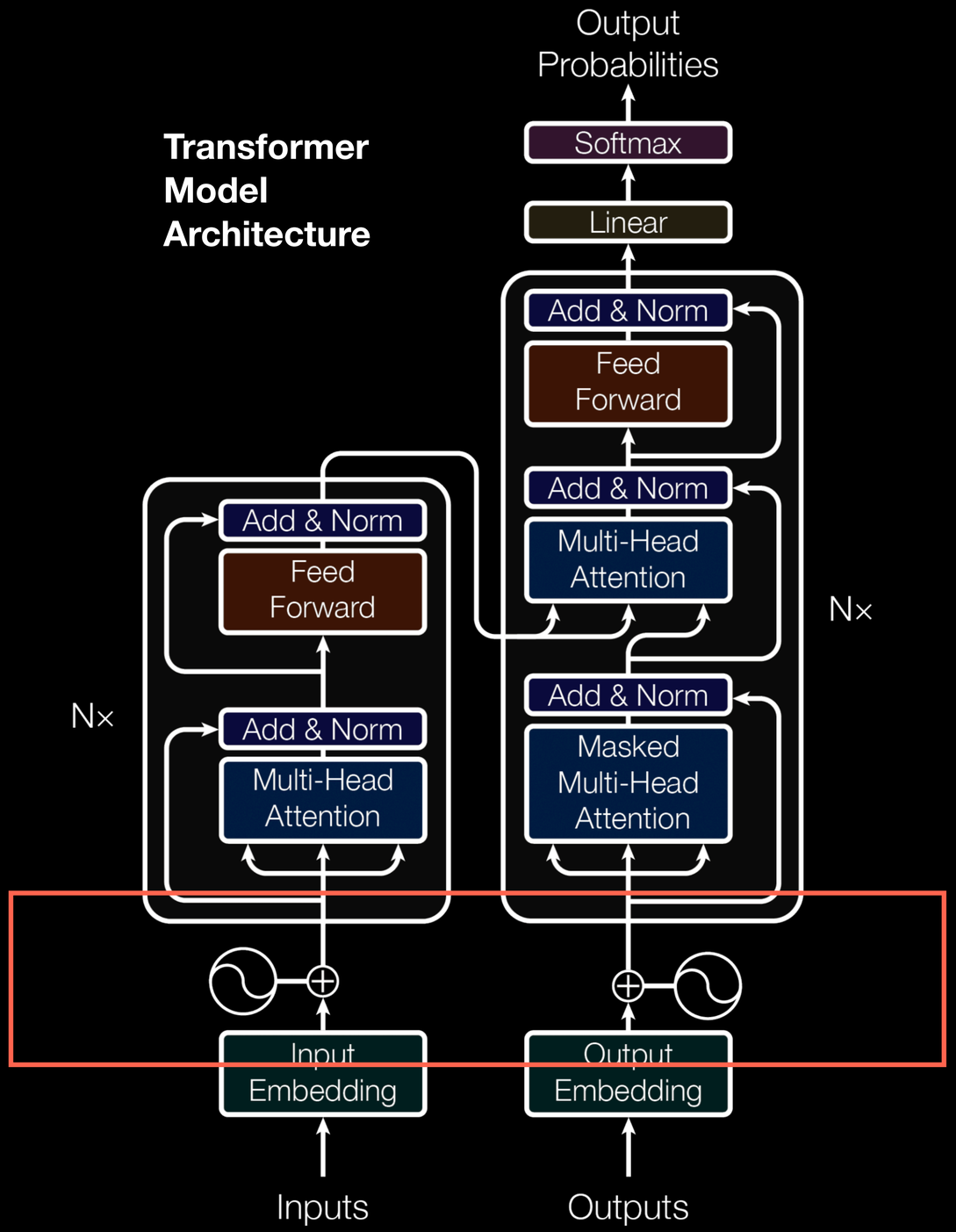

**Vanilla (Not LSTM or GRU) RNN Structure**
Born with implicit position information

**Transformer Model Architecture**



# Paper #1, Positional Encoding to Control Output Sequence Length

- Transformer doesn't contain any recurrence or convolution, positional encoding is added **provide extra information about the relative position of the words** in the sentence.



# Paper #1, Positional Encoding to Control Output Sequence Length

## Sinusoidal positional encoding

**Authors hypothesized it would work!**

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

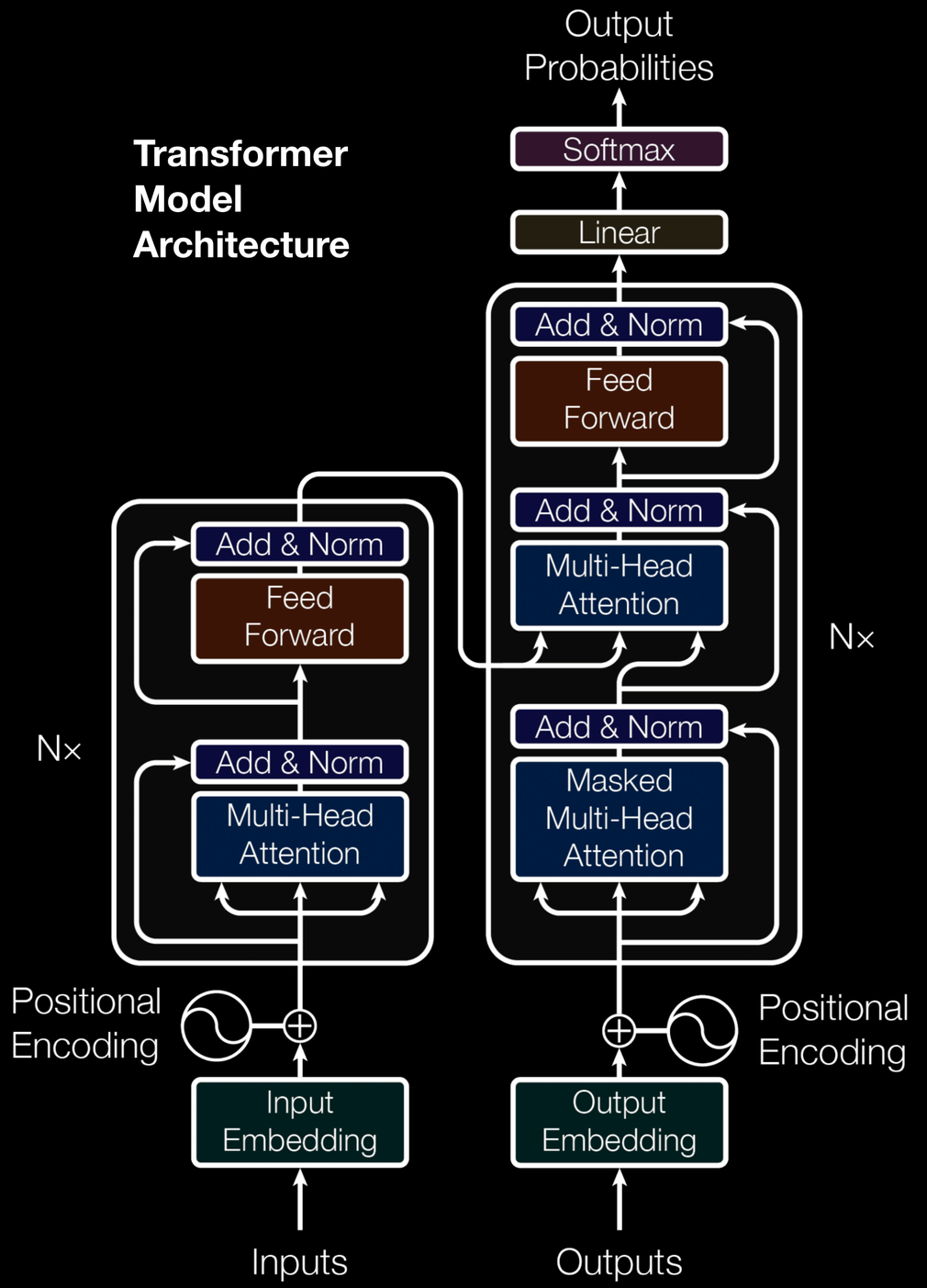
$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

$pos$ **Position index for that word in the sentence**

$d$ **Word vector dimensions**

$i$ **Row index in positional embedding vector**

**Transformer Model Architecture**



# Paper #1, Positional Encoding to Control Output Sequence Length

For each word that feed into Transformer

## Word Embedding + Positional Encoding

# Paper #1, Positional Encoding to Control Output Sequence Length

“… enough information about how …”

**For this word, it is the fourth word in this sentence.**

# Paper #1, Positional Encoding to Control Output Sequence Length

“… enough information about how …”

**For this word, it is the fourth word in this sentence.**

Word Vector for this word: (Using GloVe or Word2Vec)

Information = [ 0.50451 , 0.68607 , -0.59517 , …, -0.51042 ] (512D Vector)

# Paper #1, Positional Encoding to Control Output Sequence Length

“… enough information about how …”

**For this word, it is the fourth word in this sentence.**

Positional Encoding for this word: (Using Sinusoidal Positional Encoding)

$$PE_{(4,2i)} = \sin\left(\frac{4}{10000^{\frac{2i}{512}}}\right)$$

$$PE_{(4,2i+1)} = \cos\left(\frac{4}{10000^{\frac{2i}{512}}}\right) \quad (i \in [0,511])$$

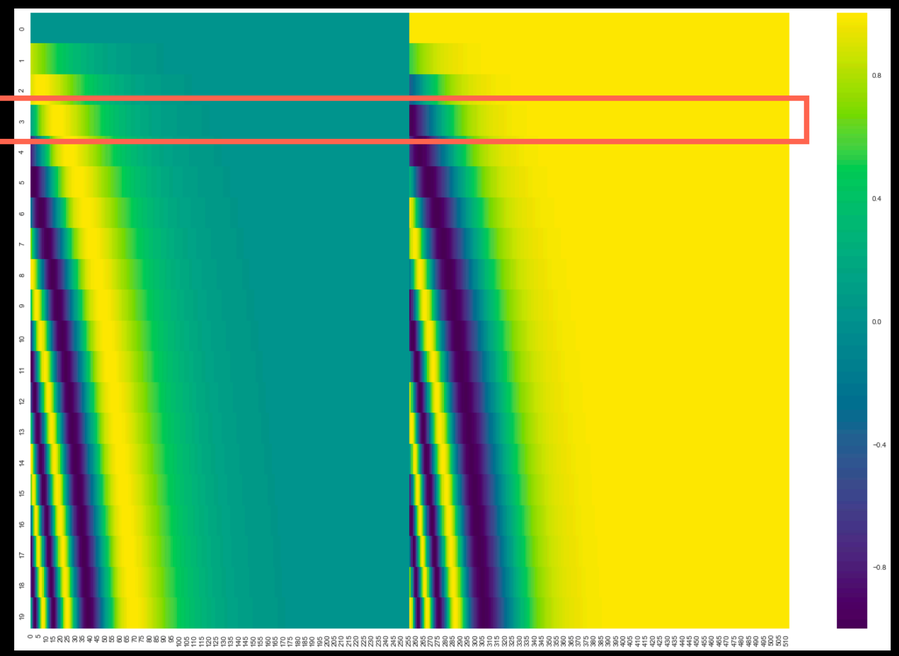
# Paper #1, Positional Encoding to Control Output Sequence Length

## “… enough information about how …”

**For this word, it is the fourth word in this sentence.**

**Positional Encoding for the fourth word in a sentence**

PE(Information) = [ 0.4 , 0.6 , 0.8, …, 0.9 ] (512D Vector)




**Sinusoidal positional encoding visualization**
**Color represent value**

# Paper #1, Positional Encoding to Control Output Sequence Length

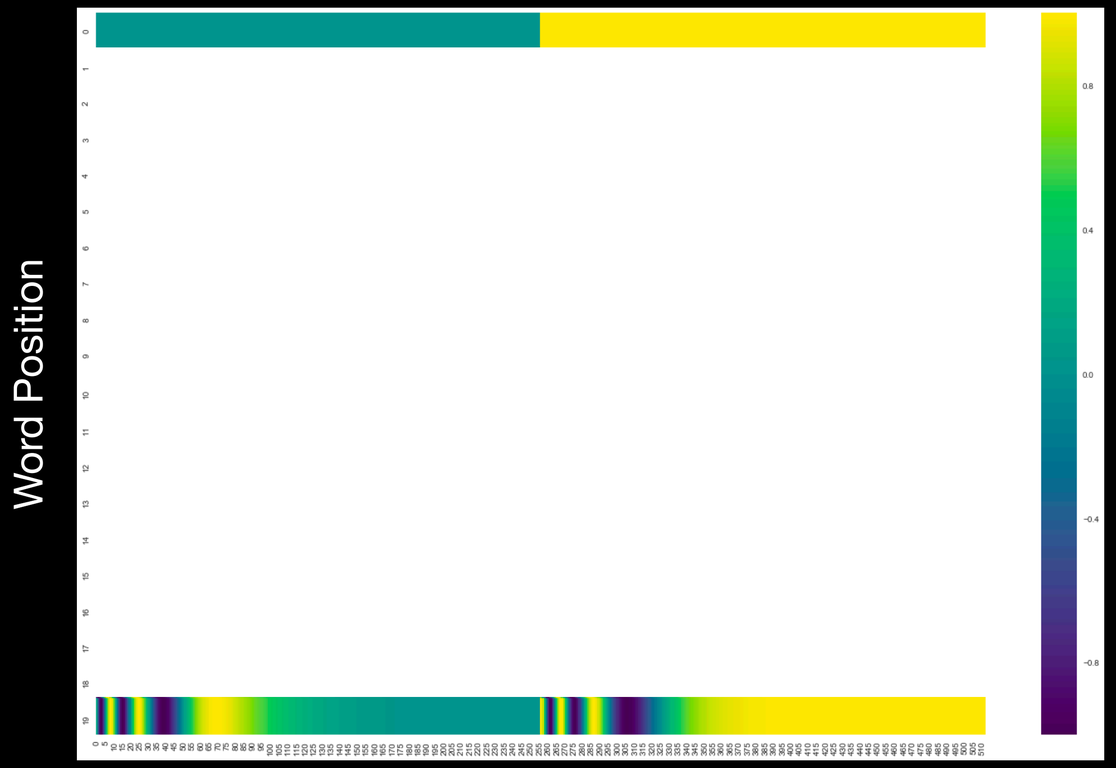
For each word that feed into Transformer

## Word Embedding + Positional Encoding

# Paper #1, Positional Encoding to Control Output Sequence Length

Sinusoidal positional encoding

- The Positional Encoding is only related to the word position in a sentence. (regardless of the word per se.)
- Can be considered as: adding a pattern to the sentence.
- We hope transformer can learn this pattern, and figure out its relationship with word position.




**Sinusoidal positional encoding visualization**
**Color represent value**

# Paper #1, Positional Encoding to Control Output Sequence Length

## Sinusoidal positional encoding

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

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

## Proposed positional encoding

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

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

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

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

**LDPE: Length-difference positional encoding**
**LRPE: Length-ratio positional encoding**

# Paper #1 - Experiments
# Test set

- For Japanese
  - **JAMUL** corpus: contain 3 kinds of headline (len=10, 13, 26) for 1,181 news articles
- For English
  - No corpus that contain headlines of multiple lengths.
  - Alternative Solution: Randomly extracted 3,000 sentence-headline pairs that satisfy a length constraint (len: 0-30, 30-50, 50-75) from **Gigaword***

*** Annotated English Gigaword (Napoles et al., 2012)**
*** Len: Characters for Japanese, Words for English**

# Paper #1 - Experiments
# Evaluate set

- On **DUC-2004** Task for comparison.

# Paper #1 - Experiments
# Training set

- No available supervision data for both language
  - For Japanese
    - **JNC** corpus (1.6M pairs): Lead 3 sentences of a article & headline
  - For English
    - **Annotated English Gigaword** (3.8M pairs): News article & headline

# Paper #1 - Result

LenInit: LSTM

LC: CNN

PE: Positional Encoding

| | $len = 10$ | | | $len = 13$ | | | $len = 26$ | | |
|---|---|---|---|---|---|---|---|---|---|
| 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 |

**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.**

# Paper #1 - Result

| | $len = 30$ | | | $len = 50$ | | | $len = 75$ | | |
|---|---|---|---|---|---|---|---|---|---|
| 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 |

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

# Paper #1 - Result

| 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 |

**Recall-oriented ROUGE scores in DUC-2004.**

$$var = \frac{1}{n}\sum_{i=1}^{n} |l_i - len|^2$$

| | 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 |

**Variances of generated headlines (Lower is better)**

# Paper #1 - Summary

- The model control the length of the generated headline
  - They modify the Positional Encoding to include length constraint.
- Better summarization quality (SoTA at that time)

# That's all. Thank you.

BNU-HKBU **United International College**

**Artificial Intelligence** SIG

Special Interest Group in FinTech, NLP, CV/CG and More