LongT5: Efficient Text-To-Text Transformer for Long Sequences

Abstract

LongT5 explores the effects of scaling both the input length and model size at the same time to improve the performance of transformer-based neural models.

Specifically, we integrate TGlobal attention mechanism and adopt PEGASUS pre-training strategies into LongT5 architecture.

1. Introduction

2. T5

T5 is a transformer-based text-to-text pre-trained language model. T5 is popular for its unified framework, and its parameter scaling capability (60M - 11B) with model parallelism. However, T5's full attention mechanism leads to quadratic computational complexity.

Recently, several studies explored scaling up T5 at inference time to longer sequences, but how to scale up T5 in the input sequence length during training remains untapped.

3. LongT5

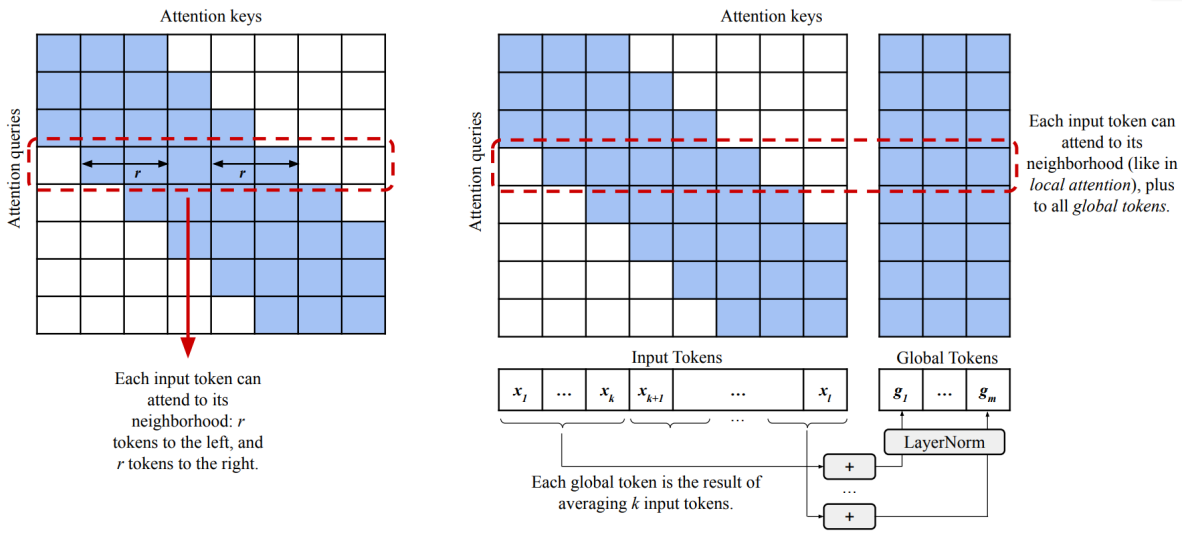
3.1. Architecture

We extend T5 encoder with (1) Local Attention and (2) Transient Global Attention. Both variations preserve several properties of T5: relative position representations, support for example packing, and compatibility with T5 checkpoints.

We use a standard T5 decoder since all of the tasks we considered require relatively short output sequence lengths.

3.1.1. Local Attention

Local Attention only allows each token to attend r tokens to the left and right of it ($r = 127$). It does not introduce any new parameters, easily accommodates the attention masking required for example packing and complexity is $O(l \times r)$.



a) LongT5 Local Attention **b) LongT5 Transient Global (TGlobal) Attention**

3.1.2. Transient Global Attention

We divide the input sequence into blocks of k tokens ($k = 16$), and for each block we compute a global token by summing (and then normalizing). We allow each input token to attend not only to nearby tokens like in Local Attention but also to every global token. Global tokens are dynamically constructed (and subsequently discarded) within each attention operation.

TGlobal Attention introduces a couple new parameters (1) Relative position biases representing the distance from an input token's block to the block of each global token it's attending to (2) Layer normalization parameters. Complexity is $O(l(r + l/k))$.

3.2. PEGASUS Principle Sentences Generation Pre-training

Predicting more informative tokens from the text could force the model to learn better semantics of the text. Motivated by that, we adopt the Gap Sentences Generation with Principle Ind-Uniq strategy.

We select top-m scored sentence based on ROUGE-F1 score using $s_i = rouge(x_i, D \backslash x_i, \forall i)$.

4. Experiments

4.1. Configurations

JAX, Flaxformer, 220M, 770M, 3B, SentencePiece, Adafactor, batch size 128, greedy decoding.

4.1.1. Pre-training

- 1M steps, 4096 input length, 910 output length
- Inverse square root
- C4 without dropout
- Masked sentence ratio is 0.2

4.1.2. Fine-tuning

- Learning rate is 0.1
- Experiment with 4096, 8192, 16382 input length, 512 output length

4.2. Evaluation on Summarization Tasks

4.2.1. Datasets

Dataset	**Example Count**			**Input Length**			
	Train	Validation	Test	Average	Median	Max	90th percentile
CNN / Daily Mail	287,113	13,368	11,490	982.39	894	5268	1659
arXiv	203,037	6,436	6,440	10,720.18	8,519	378,825	20,170
PubMed	119,924	6,633	6,658	4,747.97	3,883	452,915	8,883
BigPatent	1,207,222	67,068	67,072	6,537.32	5,236	294,004	11,328
MediaSum	443,596	10,000	10,000	2,302.02	1,748	125,974	4,128
Multi-News	44,972	5,622	5,622	2,593.81	1,902.5	683,544	4,853

4.2.2. Results

	arXiv		
Approach	R-1	R-2	R-L
DANCER PEGASUS	45.01	17.6	40.56
BigBird-PEGASUS (large)	46.63	19.02	41.77
HAT-BART	46.68	19.07	42.17
LED (large)	46.63	19.62	41.83
PRIMER	47.6	20.8	42.6
LongT5 (large - 16k input)	48.28	21.63	44.11
LongT5 (xl - 16k input)	**48.35**	**21.92**	**44.27**
	PubMed		
Approach	R-1	R-2	R-L
DANCER PEGASUS	46.34	19.97	42.42
BigBird-PEGASUS (large)	46.32	20.65	42.33
HAT-BART	48.36	21.43	37.00
LongT5 (large - 16k input)	49.98	24.69	46.46
LongT5 (xl - 16k input)	**50.23**	**24.76**	**46.67**