Abstractive Summarization using Longformer-PEGASUS

Akila Jeeson-Daniel[1]* **Curtis Lin**[2]* **Elaine Lau**[3]*
Mila - Quebec Artificial Intelligence Institute[1]
McGill University, Montreal, Canada[2,3]
akila.jeeson.daniel@umontreal.ca[1]
{curtis.lin[2], tsoi.lau[3]}@mail.mcgill.ca

Abstract

PEGASUS has achieved state-of-art results on 12 diverse downstream datasets in summarization, but has only tested input lengths up to 1,024 tokens. This work proposes a technique for extending beyond 1,024 tokens: replacing self-attention in PEGASUS with Longformer's attention mechanism for improvements in processing documents of thousands of tokens or longer. Modifications on PEGASUS's positional embeddings, dropout probabilities, and attention pattern are required to create our model, Long-PEGASUS. We evaluated Long-PEGASUS on the PubMed dataset. Experiments demonstrate that we do not have sufficient evidence to prove that it has outperformed the original PEGASUS on long-document summarization due to computational constraints.

1 Introduction

Abstractive text-summarization is the task of generating a summary while retaining the essential information of a given document. Such generated summaries may compose new phrases or sentences and should be similar to how human summarize documents.

PEGASUS (Zhang et al., 2019) has achieved state-of-the-art results on 12 diverse downstream datasets. This success is partly due to applying both gap-sentence generation and masked language modeling simultaneously as a pre-training objective. Having achieved competitive results of input lengths up to 1,024 tokens, it hypothesizes that it would remain competitive on longer documents through the generalization of sinusoidal positional encoding. We believe that its memory and computing power requirements make it infeasible to process long sequences on standard hardware.

Besides PEGASUS, recent work on abstractive summarization has shown encouraging results on relatively short documents (Nallapati et al., 2016; See et al., 2017; Paulus et al., 2017; Li et al., 2017), these approaches struggle with summarizing longer sequences. Their inability to capture long-term dependencies across the entire sequence is due to the restriction of packing all of the information into a single vector(Shao et al., 2017; Cohan et al., 2018).

To address this issue, we propose replacing the self-attention in PEGASUS with Longformer's attention mechanism, a self-attention operation that scales linearly with the sequence length. (Beltagy et al., 2020a) We hypothesize that, with sufficient training, our model, Long-PEGASUS, will outperform vanilla PEGASUS and other models on the task of long document summarization. The evaluation metrics is ROUGE (Lin, 2004) score and the dataset used is PubMed (Cohan et al., 2018).

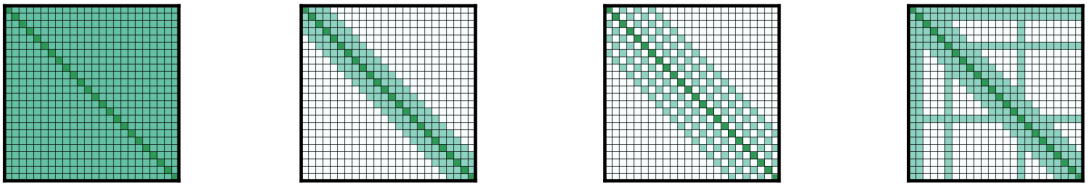
Our motivation for this work is to verify the possibility of switching out the self-attention of transformer models with Longformer's attention mechanism as proposed by the authors of Longformer. By doing so, we will help to provide the NLP community an easier integration of their model with Longformer's attention mechanism.

2 Related Work

Left-to-Right Language Models Models like Transformer-XL (Dai et al., 2019), Adaptive Span (Sukhbaatar et al., 2019), Compressive (Rae et al., 2019) are some prior work on adapting Transformers for long-documents that uses left-to-right (ltr) approach to process long documents. However, these models are unsuitable for bidirectional context for transfer learning summarization. (Beltagy et al., 2020a)

Discourse-Aware Attention Model Cohan et al.

*These authors contributed equally



(a) Full n^2 attention (b) Sliding window attention (c) Dilated sliding window (d) Global+sliding window

Figure 1: Comparing the full self-attention pattern and the configuration of attention patterns in our Longformer. (Beltagy et al., 2020a)

(2018) includes a hierarchical encoder, which captures scientific documents' discourse structure, and a discourse-aware decoder that generates the summaries. The decoder uses the different discourse information to modify the word-level attention function. Doing so allows their model to have a better context vector and have higher accuracy in representing important information.

Reinforced Model Paulus et al. (2017) The model introduces a neural network model with an intra-attention in which a decoder has an attention mechanism attend to previously generated words to generate output for long-document abstractive summarization.

Compared with other pre-existing models, our work's strength is that we are exploring the concept of extending pre-trained models to be useful for longer document summarization. Instead of training from scratch, extending PEGASUS with Longformer will reserve the prior knowledge from the pre-training. Long-PEGASUS will significantly reduce the computational cost when fine-tuning on a task-specific summary.

3 Methodology

3.1 Longformer

Longformer (Beltagy et al., 2020a) introduces an attention mechanism that acts as a drop-in replacement for the standard transformer self-attention by combining local windowed attention with a task motivated global attention. This modified transformer architecture allows the self-attention operation to scale linearly with the sequence length, making it versatile to process long documents.

Sliding Window First, the sliding window within the attention pattern is used to capture the local context's importance (Figure 1b). Fixed-size window attention is surrounding each token, and multiple stacked of such layers results in a large receptive field, allowing the top layer to have access to all input locations and build representations that incorporate information across the entire input.

Dilated Window and Global Attention Supposedly, a dilated sliding window is used to increase further the receptive field, similar to a dilated CNN where the window has gaps of size dilation d (Figure 1c). However, dilated sliding window is not used in our implementation as it is not implemented in the Huggingface module (Wolf et al., 2020). It was noted that the implementation of dilated windows needs special computation optimization modules, which are currently unavailable (Beltagy et al., 2020a). It is also noted that the implementation of global attention is also added on a few pre-selected input locations as windowed and dilated attention are not flexible enough to learn task-specific representations (Figure 1d).

3.2 PEGASUS

PEGASUS is trained on a vast corpus of web-crawler documents. In PEGASUS, important sentences from documents are selected and masked. Then, these gap-sentences are concatenated into a pseudo-summary with the corresponding position of each selected gap sentence being replaced by a mask token [MASK1] to inform the model.(Zhang et al., 2019) Important sentences are chosen by following a similar algorithm similar to ROUGE.

Due to similarity in model architecture of BART and PEGASUS, in Huggingface module, Pegasus implementation inherits from Bart class. Therefore, the implementation of Long-BART and Long-PEGASUS is almost identical except for a few changes, such as a different positional embedding - Learned Embeddings for Bart and static Sinusoidal embedding for Pegasus.

```
PegasusForConditionalGeneration(
  (model): BartModel(
    (shared): Embedding(96103, 1024, padding_idx=0)
    (encoder): BartEncoder(
      (embed_tokens): Embedding(96103, 1024, padding_idx=0)
      (embed_positions): SinusoidalPositionalEmbedding(4096, 1024)
      (layers): ModuleList(
        (0): EncoderLayer(
          (self_attn): LongformerSelfAttentionForBart(
            (longformer_self_attn): LongformerSelfAttention(
              (query): Linear(in_features=1024, out_features=1024,
bias=True)
```

Figure 2: Long-PEGASUS Congifuration

3.3 Implementation

Dataset The PubMed dataset (Cohan et al., 2018) composes of biomedical and life science literature, and is used for training the Long-PEGASUS model. The entire dataset consists of 130,397 articles with 3,031 and 198 word tokens on average for articles and abstracts respectively. We first preprocessed the articles by removing tokens such as <S> and </S> from the beginning and the end of sentences. As the articles are tokenized into sentences, we have to concatenate the sentences to form a string consisting of the entire article and a string for the entire abstract.

Long-PEGASUS Our Long-PEGASUS model is implemented by replacing the self-attention in PEGASUS with the Longformer attention module by following closely to the RoBERTa (Liu et al., 2019) fine-tuning example (Beltagy et al., 2020c) in the Longformer paper and fine-tuning example on Long-BART (Suraj, 2020). To convert PEGASUS to Long-PEGASUS, here are the following changes:

Dropout The dropout ratio for attention probabilities is expected in the Longformer self attention. Therefore, attention_probs_dropout_prob in Long-PEGASUS is set to attention_dropout in PEGASUS.

Attention Pattern We replaced the PEGASUS self-attention with the Longformer self-attention in the self-attention part of the encoder. We left the cross attention and the decoder self-attention with the traditional (n^2) attention module as changing cross attention to the local attention module did not seem sensible. To keep computation small in the decoder and the cross attention, we left the decoder size as the default values in Pegasus. Following closely to Longformer, sliding window attention with a window size of 512 is used for Long-PEGASUS.

Positional Embeddings Extending the position embedding from 512 positions of PEGASUS is needed as Longformer sets the max position as 4,096, as shown in Figure 2. Then, to leverage PEGASUS's pre-trained weights, we initialize the new position embeddings by copying the 512 position embeddings from PEGASUS multiple times as it is noted that analysis of BERT's (Devlin et al., 2018) attention heads show strong learned bias to attending to local context (Beltagy et al., 2020a), including the previous or next token. By using the copied initialization, it will preserve the local structure everywhere except for the partition boundaries. Making these changes will allow Longformer's pre-training to converge with a small number of gradient updates rapidly.

Training We used train/validation/test ratio of 60/20/20 for training Long-PEGASUS. Long-PEGASUS is trained without freezing any layers, as the extended embeddings were initialized through copying the existing positional embeddings. Ideally, we would like to train Long-PEGASUS with ample training samples to demonstrate the hypothesized improved results on longer documents. We attempted standard training and Longformer's (Beltagy et al., 2020a) staged training procedures where attention window size and sequence length were increased across multiple training stages. However, issues arises regarding RAM overflow. We could not train it beyond ten samples per epoch with batch size two as both the RAM and time limit of Google Colaboratory prevented any further training.

Evaluation Evaluation is conducted on the evaluation set in full sequence length measured by ROUGE (Lin, 2004) scores. (Kryściński et al., 2019). The comparison is conducted through comparing predicted abstracts and actual abstracts.

Model	0 examples R_1 / R_2 / R_3	5 examples R_1 / R_2 / R_3	10 examples R_1 / R_2 / R_3
BART (pre-trained)	44.04/14.99/29.16	44.04/14.99/29.16	44.04/14.99/29.16
PEGASUS (pre-trained)	58.31/21.97/46.26	58.31/21.97/46.26	58.31/21.97/46.26
Long-BART	44.81/9.58/31.30	NA - RAM Overflow	NA - RAM Overflow
Long-PEGASUS	53.34/13.76/48.75	NA - RAM Overflow	NA - RAM Overflow

Table 1: ROUGE-F1, ROUGE2-F1 and ROUGEL-F1 scores of low resource summarization on PubMed.

Epoch	Train Loss	Val Loss
1	9.454	9.069
2	9.269	9.004
3	9.432	8.942
4	9.190	8.885
5	9.152	8.762

Table 2: PEGASUS fine-tuning with Batch Size 4 and Number of Samples 10, Learning Rate 2e-6

Epoch	Train Loss	Val Loss
1	15.514	14.893
2	13.307	11.669
3	12.079	10.868
4	8.431	10.427
5	7.268	10.264

Table 3: BART fine-tuning with Batch Size 4 and Number of Samples 10, Learning Rate 2e-6

4 Results

ROUGE Comparison Table 1 shows that that PEGASUS has the highest score across ROUGE scores at 0 examples of training, but Long-PEGASUS and Long-BART remain competitive compared to vanilla PEGASUS and BART.

RAM Overflow Due to computational constraints of Google Colab Pro, training of Long-BART and Long-PEGASUS with sequences of length 4,096 was not possible as the RAM limit was reached. To overcome this issue, modifications were made to lower the batch size, number of epochs, and training samples with no success in the end. We attempted to reduce the max length of a sequence to 2,048, but an error occurred when fine-tuning Long-BART and Long-PEGASUS, caused by an invalid lookup of the embedding weight matrix. For that reason, we decided to run our training function on vanilla BART and PEGASUS only to test out the fine-tuning process.

Training Results Table 2 and Table 3 show the fine-tuning loss value for PEGASUS and BART respectively. Despite the limited training samples, a trend has been observed: as epoch increases, training loss, and validation loss gradually decrease.

5 Discussion and Conclusion

From the results in Table 1, we concluded that there is not enough evidence to support our initial claim that Long-PEGASUS will outperform vanilla PEGASUS and other models on the long document summarization task. Potential reasons and struggles in providing strong evidence to support our hypothesis are as follows:

Model Variations The results of the Longformer-infused models are expected as their additional embeddings are copied, which adds noise to sequences that may have previously worked well on their vanilla counterparts. To improve the results, sufficient training is required to optimize the new weights. PEGASUS remains at the top as no modification has been made. BART placed second overall. Meanwhile, it is shown that Long-PEGASUS and Long-BART have a similar ROUGE score, with Long-PEGASUS scoring slightly higher. We believe this outcome is due to the differences in performance between their base models, PEGASUS and BART. This is further proven as PEGASUS outperforms BART in several short summaries in the PEGASUS paper (Zhang et al., 2019).

ROUGE ROUGE offers a cheap evaluation method for summarization models. However, it has been widely discussed on whether it is qualified to be the standard evaluation metric for summarization tasks, with strong evidence against it (Kryściński et al., 2019). It is also found that low-ROUGE summaries often were high-quality (Zhang et al., 2019). These findings brought us to question and discuss whether there will ever be an evaluation metric for summarization tasks that evaluate summaries as humans do. To achieve such a feat, we believe that the metric must achieve human analogous understanding to determine relevance, consistency, fluency

and coherence in generated summaries (Kryściński et al., 2019).

Version Difference The Original Longformer model was in version 2.11.0 of HuggingFace which did not have PEGASUS implemented. Therefore, a lot of effort was put into changing it to the current version. About midway through the project, AllenAi's Longformer team released their version of LongformerEncoderDecoder (Beltagy et al., 2020b) which was used for debugging and changing to version 3.1.0. Adding a global attention mask to the Bart class of the Huggingface module needed some effort to consistently track it through the model into the Longformer Attention class.

Accessibility With the discovery of transformers (Vaswani et al., 2017), state-of-the-art benchmarks in natural language processing are being surpassed left and right. These new found discoveries have attracted increasingly more students to deep dive and experiment with the model themselves. Though models like BERT (Devlin et al., 2018) have allowed students to achieve impressive results with minimal data needed for fine-tuning, the fact remains that transformer models are computationally expensive. The ability to pre-train, tweak a transformer's inner workings to test their hypotheses still lies in the hands of people with sufficient compute power. For this project, progress was hindered multiple times due to computational constraints. We look forward to a future where significant improvements have been made to the accessibility of transformers, such that anyone can have a shot at discovering something new.

Future Work For future work, we want to construct a detailed tutorial on replacing BERT-style models with Longformer's attention mechanism. We would also like to obtain sufficient compute to test Long-PEGASUS on datasets with longer documents such as books. We believe that abstractive summarization of books can produce meaningful insight that readers may not have expected due to the generative nature of abstractive summarization.

6 Statement of contributions

Three of us designed the project together. Akila's primary focus is on implementing the Long-PEGASUS and Long-BART models and writing the paper. Curtis and Elaine's primary focus is on training and evaluating the two models, preprocessing data, and writing the paper.

References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020a. Longformer: The Long-Document Transformer. *arXiv e-prints*, page arXiv:2004.05150.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020b. Longformer: The long-document transformer. https://github.com/allenai/longformer.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020c. Longformer: The long-document transformer - convert to long model notebook. https://colab.research.google.com/github/allenai/longformer/blob/master/scripts/convert_model_to_long.ipynb.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents. *arXiv e-prints*, page arXiv:1804.05685.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. *arXiv e-prints*, page arXiv:1901.02860.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv e-prints*, page arXiv:1810.04805.

Wojciech Kryściński, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural Text Summarization: A Critical Evaluation. *arXiv e-prints*, page arXiv:1908.08960.

Piji Li, Wai Lam, Lidong Bing, Weiwei Guo, and Hang Li. 2017. Cascaded attention based unsupervised information distillation for compressive summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2081–2090, Copenhagen, Denmark. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. 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. *arXiv e-prints*, page arXiv:1907.11692.

Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond. *arXiv e-prints*, page arXiv:1602.06023.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A Deep Reinforced Model for Abstractive Summarization. *arXiv e-prints*, page arXiv:1705.04304.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, and Timothy P. Lillicrap. 2019. Compressive Transformers for Long-Range Sequence Modelling. *arXiv e-prints*, page arXiv:1911.05507.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. *arXiv e-prints*, page arXiv:1704.04368.

Yuanlong Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil. 2017. Generating high-quality and informative conversation responses with sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2210–2219, Copenhagen, Denmark. Association for Computational Linguistics.

Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin. 2019. Adaptive Attention Span in Transformers. *arXiv e-prints*, page arXiv:1905.07799.

Patil Suraj. 2020. Longbart. `https://github.com/patil-suraj/longbart.git`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *arXiv e-prints*, page arXiv:1706.03762.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. *arXiv e-prints*, page arXiv:1912.08777.