Project Proposal: Efficient Long-Text Understanding with Short-Text Models

Bhabaranjan Panigrahi
bpanigr@gmu.edu

Kumar Shubham
kfnu@gmu.edu

Meghana Purijala
vpurijal@gmu.edu

1 Introduction

1.1 Task / Research Question Description

Transformer-based pretrained language models (LMs) have revolutionized the Natural Language Understanding landscape, but tend to not perform well in the both long text understanding benchmarks and computational efficiency. The main source of bottleneck comes from their quadratic complexity. Although, a variety of efficient transformers have been introduced, they use specialized and custom architectures. Most of these architectures have the limitation of scalability and do not exploit already present pre-trained LMs. Till date the pre-trained LMs have not been able to reached the same level of performance as compared to their short-range task benchmarks.

This research project involves using efficient techniques named as **SLED**: **SL**iding- **E**ncoder and **D**ecoder, for long sequence text such as stories, scientific articles, and long documents. The primary goal is to overcome the quadratic complexity issue associated with Transformer-based language models, making them more applicable to long texts, while utilising transfer learning from pretrained LMs. The research project aims to investigate if it's possible to leverage existing pre-trained language models for short texts to handle long texts effectively by using **SLED** and in-depth analysis that is of interest to the ML and NLP community.

1.2 Motivation and Limitations of existing work

In this research paper, SLED model has been compared with seven different variants of Transformer Language Models (BART-long, BART-base, Long T5-XL, Long T5-large, T5-base, UL2), pre-trained models that addressed to solve the quadratic complexity involved in the long-range text sequences, but they have expensive pre-training (fine tuning the model parameters) as compared to SLED. Having said that SLED has better performance in terms of efficiency in comprehending the SCROLLS (long documents) when compared with all the different language models.

The motivation behind this paper is to create **cost-effective** solution for **Natural Language Understanding** of long texts that can utilize pre-trained encoder-decoder models designed for short texts. The limitations of existing models involve costly fine-tuning of parameters (around 20B for UL2) while pre-taining whereas SLED obtains the same performance in very less parameters (around 400M parameters).

1.3 Proposed Approach

SLED follows the method of *sparse attention*, which is mainly used to handle quadratic complexity. In *sparse attention* each token attends to a fixed number of tokens that are around the main token. The main idea behind this is the sentences in long texts are generally related to the content surrounding them instead of the whole document. SLED relies on the local attention mechanism to work but unlike other methods it uses short-range encoders.

SLED depends on the assumption that the encoder can effectively contextualize smaller partitioned input tokens, and decoder can extract long range understanding. It uses a pretrained encoder-decoder model M as a backbone. The model receives a tokenized input of length n, and an extra prefix token of length m. Where m is significantly smaller than n. Generally, m represents a tokenized version of a question.

The following steps are being followed for the SLED

- The document is split in to C chunks of length c [e.g; c = 4]. For local contextualization, main tokens are surrounded by both left and right tokens. The middle tokens are called *effective tokens* and rest is called *context padding*.

- Each chunk is pre-appended by prefix tokens (usually the questions).
- Each partition is encoded independently using an encoder.
- The effective tokes are extracted from the encoder, and the representation is passed to the decoder to generate output.

1.4 Likely challenges and mitigations

There are no critical limitation, but a few edge cases need to be handled in order for SLED to work. The first and last chunk do not have bi-directional context padding. The last token index is intentionally chosen so that it should have a prefix token and post-fix token.

2 Related Work

The transformer based models has a core limitation, and that is the quadratic dependency on the long text sequences due to the full attention mechanism. To overcome this limitation, several researches have been done successfully in the past, such as followings:
LongT5: Efficient text-to-text transformer for long sequences (Guo et al., 2022). propose that either by increasing the input length or by increasing the model size, it improves the performance of the Transformers.
Big Bird: Transformer for Longer sequence (Zaheer et al., 2021) introduces a sparse attention mechanism has been proposed which reduces the quadratic dependency to linear. BigBird approximator is universal and Turing complete, therefore it preserves the properties of the full attention model.
Reformer: The efficient transformer (Kitaev et al., 2020) introduces two techniques for more memory-efficient and much faster on long sequences. First, by replacing dot-product attention by one that uses locality-sensitive hashing. Second, by using reversible residual layers, meaning, storing activations only one time in the training process instead of N times.
Other papers such as (Mehta et al., 2022), (Shaham et al., 2022) introduces different mechanisms to improve transformers on long text sequences.

Our work is different from all the above cited research papers and is closely related to (Izacard and Grave, 2021), it uses locality of information method, meaning, it contextualize input tokens with local context only, and the long range dependencies are handled by the decoder with limited number of tokens.

We have demonstrated it in two controlled experiments. Firstly, by fusing separate pieces of information in the decoder, and Secondly, by contextualizing each chunk with a prefix instead of using global tokens.

3 Experiments

3.1 Datasets

SLED is evaluated on **SCROLLS** (Shaham et al., 2022), which is used as a benchmark for evaluating long-text understanding. SCROLLS contains a total of seven datasets divided into three different language understanding tasks namely Summarization, Question answering (QA), and Natural language inference.

These datasets are publicly available and part of the SCROLLS benchmark. The paper uses the official evaluation metrics defined in SCROLLS, which are based on the metrics from the original datasets. We are able to access and use the same for preprocessing and train, Dev and Test splits for reproducing the results and comparing them against the evaluation results ran on the same datasets as mentioned in the paper.

Summarization: It contains the summary of reports from diverse domains.

GovReport (Huang et al., 2021) - A summarization task over reports from the Congressional Research Service. https://doi.org/10.18653/v1/2021.naacl-main.112

SummScreenFD (Chen et al., 2022) - A summarization dataset over TV scripts. https://doi.org/10.18653/v1/2022.acl-long.589

QMSum (Zhong et al., 2021) - Query-based summarization dataset over meeting transcripts from various domains. In this dataset we consider query as the Prefix. https://doi.org/10.18653/v1/2021.naacl-main.472

Question answering (QA) : In this type we consider question as the prefix.

Qasper - QA is a benchmark that contains questions about NLP papers. https://doi.org/10.18653/v1/2021.naacl-main.365

NarrativeQA - This is a QA-type dataset that contains questions about entire books and movie scripts. https://openreview.net/

QuALITY - It contains a multiple-choice QA dataset over books and articles. In this dataset, we consider the four answer options as a part of the question. https://doi.org/10.18653/v1/2022.naacl-main.391

Natural language inference: Short legal hypotheses are considered as prefixes.

ContractNLI : Legal documents as the premise. Models are tasked to predict whether the premise entails, contradicts or is neutral w.r.t. to the hypothesis. https://doi.org/10.18653/v1/2021.findings-emnlp.164

The SLED processes long sequences versus pretrained models such as BART and T5 which just processes the initial 1024 tokens. In SLED, we use the maximum sequence length as 16K tokens and chunk size of 256 to allow for a fair evaluation.

3.2 Implementation

We did not re-implemented the code from scratch. However, we used the author's implementation and compared our results with the claimed results of the paper. For all our experiment we used BART_BASE (Lewis et al., 2019) with SLED.

Following is the Github repository we used to run our code. The repository is **forked** from the main branch of the author's implementation. https://github.com/bhaba-ranjan/SLED-Copy (Author's implementaion: https://github.com/Mivg/SLED)

3.3 Results

We ran all of the experiments on SQuAD (Rajpurkar et al., 2016) and HotpotQA (Yang et al., 2018). We were successfully able to reproduce the results that is claimed by the paper.

Dataset	Claimed [F1]	Reproduced[F1]
Qspr	27.6	26.1
Nrtv	16.0	15.8
SQuAD	87.6	87.1
SQuAD(s)	87.2	86.6
HotpotQA	76.5	76.5

3.4 Discussion

We did a sensitivity analysis on the batch size, learning rate and token length. When we tested with a chunk size of 1 to find out if all the fusion can happen in the decoder. We observed that by removing local contextualized information in the encoder results in a significant drop in performance. This refers that model is dependent on the encoder to collect contextualized information.

We also tested with batch sizes(over multiple runs) and observed, higher batch size leads to drop in performance with the same number of training epochs. Also, the training loss drop is more smooth with higher batch size. However, increasing the learning rate helped mitigate the drop-in performance. It is because, small batch size introduced more stochasticity and the model ends up learning a bit more (not significant) contextualized representation.

3.5 Resources

We used ORC Cluster (Hopper) for all of our experiments. We reproduced our results on NVIDIA-A100 GPU machine with 40gigs of GPU memory. Also, we wrote a SLURM script to submit the job and get our results for comparision.

3.6 Error Analysis

The model suffers from two main problem that are Long Text Generation and Co-reference Resolution and Fact Retention. For the first problem, even if the encoder encodes the input efficiently in a liner time complexity the decoder still use quadratic attention. As SLED relies on the local information extremely distant and connected piece of information can still hinder the performance relying heavily on the decoder to understand global context hence reducing the NLU aspect of the model.

4 Conclusion

We can conclude that we have successfully replicated the results stated by the paper using datasets such as SQuAD and HotpotQA. Comparing the SLED model, a straightforward sliding method for long text sequences, against larger models and models that require specialised and costly pre-training, it performs competitively on the SCROLLS benchmark. This implies that SLED is a flexible and promising tool to handle challenges related to long-range text understanding.

Our inspection shows that the SLED is capable of processing and understanding lengthy texts with effectiveness. This research could improve language models suitability for usage in practical scenarios.

References

Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. 2022. Summscreen: A dataset for abstractive screenplay summarization.

Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2022. Longt5: Efficient text-to-text transformer for long sequences.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Harsh Mehta, Ankit Gupta, Ashok Cutkosky, and Behnam Neyshabur. 2022. Long range language modeling via gated state spaces.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, and Omer Levy. 2022. Scrolls: Standardized comparison over long language sequences.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2021. Big bird: Transformers for longer sequences.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. Qmsum: A new benchmark for query-based multi-domain meeting summarization.