Group 5 Proposal, Paper:Linearizing Transformer with Key-Value Memory

Yaqi Hu yaqihu@usc.edu
Gonglin Chen gonglinc@usc.edu
Libang Xia libangxi@usc.edu
Shaojin Yin zhaojiny@usc.edu
Luoyuan Zhang luoyuanz@usc.edu

1 Project Domain

Our project aims to implement and reproduce the results of the selected paper, testing the performance of MemSizer and the Vanilla Transformer using WMT 16 En-De, XSUM, & WikiText-103. And extend the model to new dataset OpenOrca, the area that paper not reached, question answering. The evaluation involves a direct comparison of the two models with reproduced results and an assessment of question-answering models powered by each. The objective is to determine which model yields superior performance.

The article presents *MemSizer*, a novel variant of the Vanilla Transformer, employing low-rank projection techniques akin to Linformer and Kernel-based Transformers. This variant excels in tasks such as machine translation, text summarization, and language modeling. A distinctive feature of their approach is the introduction of a unique key-value memory layer, designed to replace the multi-head attention layer of the Vanilla Transformer.

2 Related Work

2.1 Transformer

Transformer has become the de facto standard for various natural language processing (NLP) tasks. However, the transformer models are known for their computational expense, especially due to the quadratic scaling of the attention mechanism with sequence length.

2.2 Transformer Variants

To overcome this limitation, some Transformer variants use low-rank projections to reduce pairwise interactions and attention matrix size. However, these methods struggle with modeling variable-length sequences and autoregressive attention, limiting their usefulness in sequence generation tasks.

2.3 Recent Works

Recent approaches employ kernelization techniques to approximate softmax attention, enabling constant memory complexity for sequence generation tasks. While this boosts efficiency for long-form generation, the computational benefits diminish for shorter sentences.

2.4 MemSizer: An Efficient Transformer Variant

In this work, we propose an approach called MemSizer. MemSizer is an efficient Transformer variant that replaces the multi-head attention layer with a key-value memory layer. It dynamically packs source sequence information into memory values using simplified key design, enabling faster multi-head computations and emphasizing value modeling, thus enhancing efficiency and performance.

3 Datasets

For our project, we will first use some original datasets to reproduce the result, and we find some new datasets to evaluate the model. Moreover, we will extend further to see if MemSizer could be generalized to other tasks like conversations.
Note: Alternate plan: if time is not enough, we will only test one of the area from three.
Note: Due to space limitation, we only show the urls that we used in our project, for both reproduce and new work.

3.1 Paper Used Datasets

Baseline paper used six datasets in total for three different tasks. All of them are benchmark datasets, which are widely used in multiple paeprs. Three for Machine Translation. Two for Text Summarization, and one for language modeling.

3.1.1 Machine Translation dataset

1. WMT 16 En-De
https://huggingface.co/datasets/wmt16/

viewer/de-en/train
2. WMT 14 Fr-En
3. WMT 17 Zh-En

3.1.2 Text Summarization

1. CNN/Dailymail
2. XSUM
https://huggingface.co/datasets/xsum

3.1.3 Language Modeling

1. WikiText-103
https://huggingface.co/datasets/wikitext

3.2 Our Project Used Datasets

3.2.1 Reproduce Paper Result

First, as we decided to reproduce the paper result, we need to use the original datasets. Moreover, consider that we have limited computational resource, we choose the smallest datasets from original, to reduce the reproduce time.
Therefore, we choose three datasets from six to reproduce the result. The dataset we select are as below:

- WMT 16 En-De & XSUM & WikiText-103

3.2.2 Preprocessing

For both original and our new used datasets, we use the standard preprocessing method (so do as the paper author). It contains several steps: Tokenization, Lowercasing, Removing Special Characters and etc. Just like the homework 1 we did before.

3.2.3 New Datasets and Our Works

In order to explore MemSizer's ability in other task, we plan to use dataset OpenOrca (https://huggingface.co/datasets/Open-Orca/OpenOrca). The OpenOrca dataset is a conversational dataset, comprising three primary components: a system prompt, a question, and a response. Our strategic approach entails fine-tuning the pre-trained model that we initially developed for the purpose of replicating the results presented in the original paper.

4 Technical Challenge

4.1 Performance drop

The proposed MemSizer approach aims to close the performance gap between efficient transformer variants and the vanilla transformer. However, reproducing the results and achieving similar performance improvements may be challenging due to the unique characteristics of MemSizer. What's more, the performance is highly correlated with the hidden dimensionality of input source and the size of MemSizer, there is no guarantee that MemSizer always works better than vanilla transformer, but it definitely worth an investigation.

4.2 Variable-length sequences

Existing low-rank projection methods, such as Linformer, have difficulties in modeling variable-length sequences and autoregressive attention. Reproducing the MemSizer approach, which addresses these challenges, may require careful implementation and adaptation to different sequence generation tasks. Implementation of the new version of Q, K and V may lead to different bugs that cost a long time to fix.

4.3 Computation gain for short generation

While kernel-based approaches can achieve efficient computation for long-form generation, their computation gain diminishes when the generation is as short as a typical sentence length. Reproducing the MemSizer approach, which maintains computation efficiency even for short generation, may require understanding and implementing the underlying mechanisms.

4.4 Code availability and compatibility

The code for MemSizer is released on GitHub, but reproducing the paper's results may require ensuring compatibility with the specific code version and dependencies. Especially if we want to replace the original calculation of attention with our new attention structure, not only we need to change the QKV formula, we also need to modify the attention score formula and make sure it works as expected. Additionally, understanding and adapting the code to specific experimental setups may pose challenges such as modifying the model to adapt different downstream tasks.

5 Labor Division

Proposal:
1. Project Domain - Libang Xia
2. Related Work - Luoyuan Zhang
3. Datasets - Yaqi Hu, Gonglin Chen
4. Technical Challenge - Shaoyin Jin
Project Work:
Reproduce: Libang Xia, Gonglin Chen, Yaqi Hu
New Dataset: Shaoyin Jin, Luoyuan Zhang
Fine-tuning Work: All of us

Reference

1. Attention is all you need https://browse.arxiv.org/pdf/1706.03762.pdf
2. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension https://aclanthology.org/2020.acl-main.703.pdf
3. Linearizing Transformer with Key-Value Memory https://arxiv.org/abs/2203.12644
4. Generating Long Sequences with Sparse Transformers https://arxiv.org/abs/1904.10509