CAN PERPLEXITY REFLECT LARGE LANGUAGE MODEL'S ABILITY IN LONG TEXT UNDERSTANDING?

Yutong Hu, Quzhe Huang, Mingxu Tao, Chen Zhang, Yansong Feng [*]
Wangxuan Institute of Computer Technology, Peking University
{huyutong,huangquzhe,thomastao,zhangch,fengyansong}@pku.edu.cn

ABSTRACT

Recent studies have shown that Large Language Models (LLMs) have the potential to process extremely long text. Many works only evaluate LLMs' long-text processing ability on the language modeling task, with perplexity (PPL) as the evaluation metric. However, in our study, we find that there is no correlation between PPL and LLMs' long-text understanding ability. Besides, PPL may only reflect the model's ability to model local information instead of catching long-range dependency. Therefore, only using PPL to prove the model could process long text is inappropriate. The local focus feature of PPL could also explain some existing phenomena, such as the great extrapolation ability of the position method ALiBi. When evaluating a model's ability in long text, we might pay more attention to PPL's limitation and avoid overly relying on it.

1 INTRODUCTION

Recently, many researchers (Chen et al., 2023b;a; Xiong et al., 2023; Ding et al., 2023; Chen et al., 2023c) have proposed various approaches to scale up the context window of LLMs to more than 100k. Since there is not a comprehensive benchmark tailored for the evaluation of such extremely long text understanding, such as question answering (QA) over 100K tokens, researchers use perplexity (PPL), an evaluation metric for language modeling [1], to demonstrate the model's ability to process long text (Chen et al., 2023c; Ding et al., 2023; Liu et al., 2023; Peng et al., 2023).

However, only given LLMs are excellent in language modeling, can it indicate LLMs' ability to understand long text? We conduct experiments on three long context window LLM variants to figure out this. We use several available benchmarks of downstream tasks, such as QA and summerization, to evaluate their long-text understanding ability. Surprisingly, the models' performance on language modeling is inconsistent with their performance on most downstream tasks, implying the PPL can not be a good indicator of the model's long-text understanding ability.

We speculate that the phenomenon above may be because PPL is a reflection of the model's ability to model **local** information. We use LLaMA2, which only has a short context window of 4,096 and cannot handle long context, to prove our speculation. The experiment results show that, LLaMA2 delivers comparable PPL with the long context window LLMs. The feature of PPL in reflecting local information modeling ability can also explain why methods such as ALiBi (Press et al., 2022), which makes the model mainly focus on local information, could enable models to extrapolate to longer inference sequences while keeping the PPL at a low level.

2 LOWER PPL ≠ UNDERSTANDING LONG TEXT BETTER

In this section, we compare the models' performance in language modeling and several downstream tasks, and figure out whether a lower PPL in language modeling indicates a better long-text understanding ability. We choose three model variants whose context windows are longer than 100K

---

[*]Corresponding author
[1]The definition and calculation method of PPL is shown in Appendix A.1. A lower PPL shows a higher accuracy of a model in long-text language modeling

| Task | LM | QMSUM | NaQA | NaQA (32K-64K) | NaQA (64K-128K) | Retrieval |
Metric	PPL ↓	Rouge-L ↑	F1 ↑	F1 ↑	F1 ↑	Acc. ↑
YARN	**1.878**	19.79	16.84	26.75	17.49	0.065
Yi	2.069	20.48	14.49	23.00	21.26	0.38
LongLoRA	2.002	**21.53**	**18.38**	**29.23**	**23.19**	**0.485**
LLaMA	1.935	-	-	-	-	-

Table 1: Performance on downstream tasks. LM is language modeling and NaQA is NarrativeQA. 32K-64K (64K-128K) indicates the input length is between 32K and 64K (64K and 128K).

tokens for experiments: 1) YARN-7B-128K (Peng et al., 2023), 2) Yi-6B-200K [2], 3) LongLoRA-7B-100K (Chen et al., 2023c).

We first calculate the PPL of each model with a 76k input length[3] on 50 books randomly sampled from the test set of PG-19 (Rae et al., 2019). While for downstream tasks, we use two public benchmarks, QMSUM (Zhong et al., 2021) and NarrativeQA (Kočiský et al., 2017), to evaluate the models' performance in long question answering and long document summarization. In addition, following Dacheng Li* & Zhang (2023), we use a finer-grained line retrieval test to evaluate models' retrieval ability. The experiment details are shown in Appendix A.1 and Appendix A.2.

The experiment results are shown in Table 1. Column **LM** reports models' PPL on the long text. YARN delivers the lowest PPL of 1.878, compared to 2.069 of Yi and 2.002 of LongLoRA, which indicates YARN's better ability in long text language modeling. However, YARN does not deliver the best performance in downstream tasks. Instead, LongLoRA outperforms other models on all downstream tasks. The inconsistency between the models' performance in language modeling and downstream tasks demonstrates that, although PPL is a fair evaluation for language modeling, it can not be a good indicator of long-text understanding ability.

3   PPL MIGHT REFLECT LOCAL LANGUAGE MODELING ABILITY

To explain the phenomenon in Sec 2, we speculate PPL may be related to how well LLMs handle the local information, rather than understanding the whole long text. Specifically, for long text language modeling, LLMs may only use local tokens to predict the next token, finally achieving a low PPL.

To prove the speculation, we calculate the PPL of LLaMA2-7B on the same test corpus as the experiment in Sec 2. Each token is predicted based on the previous 4k tokens, ensuring LLaMA2 can only use the local information in language modeling. We show experiment details in Appendix A.1.

The experiment results in Table 1 show that LLaMA2 delivers a PPL of 1.935, lower than Yi and LongLoRA with an input length of 76K, demonstrating that a model incapable of understanding long text can also deliver a low PPL. In other words, the good performance of a model in long-text language modeling can not indicate the model's ability in long-text understanding. Therefore, PPL may only reflect LLMs' ability to handle the local information, rather than understanding long text.

As LLMs can only use local information to deliver low PPL on long text, it can explain the great extrapolation ability of position embedding ALiBi (Press et al., 2022). Since the position representation method of ALiBi penalizes attention scores between distant query-key pairs, LLMs still focus on the local information even with longer input, which is why the PPL can remain at a similar level.

4   CONCLUSION

PPL can be an effective evaluation metric for long-text language modeling ability, but not for long-text understanding. A model without the ability to understand long text can also effectively use local information to model a long text. Considering PPL can not be a good indicator for long text understanding ability, except for using PPL to evaluate a model, we call for more diversified evaluation metrics for long text processing ability from multiple aspects.

---

[2]https://github.com/01-ai/Yi

[3]76K is the longest input length our machine can process. We get Out Of Memory Error with longer inputs

URM Statement

The authors acknowledge that at least one key author of this work meets the URM criteria of ICLR 2024 Tiny Papers Track.

References

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. Longbench: A bilingual, multitask benchmark for long context understanding, 2023.

Guanzheng Chen, Xin Li, Zaiqiao Meng, Shangsong Liang, and Lidong Bing. Clex: Continuous length extrapolation for large language models, 2023a.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation, 2023b.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. Longlora: Efficient fine-tuning of long-context large language models, 2023c.

Anze Xie Ying Sheng Lianmin Zheng Joseph E. Gonzalez Ion Stoica Xuezhe Ma Dacheng Li*, Rulin Shao* and Hao Zhang. How long can open-source llms truly promise on context length?, June 2023. URL https://lmsys.org/blog/2023-06-29-longchat.

Jiayu Ding, Shuming Ma, Li Dong, Xingxing Zhang, Shaohan Huang, Wenhui Wang, Nanning Zheng, and Furu Wei. Longnet: Scaling transformers to 1,000,000,000 tokens, 2023.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The narrativeqa reading comprehension challenge, 2017.

Xiaoran Liu, Hang Yan, Shuo Zhang, Chenxin An, Xipeng Qiu, and Dahua Lin. Scaling laws of rope-based extrapolation, 2023.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models, 2023.

Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation, 2022.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling, 2019.

Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, Madian Khabsa, Han Fang, Yashar Mehdad, Sharan Narang, Kshitiz Malik, Angela Fan, Shruti Bhosale, Sergey Edunov, Mike Lewis, Sinong Wang, and Hao Ma. Effective long-context scaling of foundation models, 2023.

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

A   APPENDIX

A.1   PPL CALCULATION METHOD

For the method we use for LongLoRA, Yi and YARN in Sec 2, we split the given text into N chunks with the size of 76k per chunk. Taking each chunk as input, the LLM makes predictions on every token based on all of its previous tokens in the chunk. Specifically, the probability of token x_i in the chunk is calculated as:

$$P_1(x_i) = p(x_i|x_0, ..., x_{i-1}), i \in [0, 76000) \tag{1}$$

After obtaining the prediction results of all chunks, i.e., obtaining the probability of every token in the given text, we calculate the

$$PPL = average(-log(P_1(x_i))), i \in [0, L) \tag{2}$$

where L is the length of the given text. The calculation method is the same as Chen et al. (2023c); Peng et al. (2023).

In the method we use for LLaMA2 in Sec 3, every token is predicted based on a fixed number of previous tokens. This is different from the former method, which makes predictions on the i_{th} token based on all previous tokens in the chunk. Specifically, if the index of the token in the given text is equal to or larger than the context size of 4k, the probability of the token is calculated based on its corresponding previous 4k tokens. Otherwise, its probability is calculated based on all of its previous tokens. The equations are:

$$P_2(x_i) = p(x_i|x_{i-4000}, ..., x_{i-1}), i \in [4000, L) \tag{3}$$

$$P_2(x_i) = p(x_i|x_0, ..., x_{i-1}), i \in [0, 4000) \tag{4}$$

After obtaining the probability of every token in the given text, we calculate the PPL:

$$PPL = average(-log(P_2(x_i))), i \in [0, L) \tag{5}$$

Such a calculation method can help models deliver a lower PPL compared to the former calculation method under the same test corpus. Therefore, we adopt it for LLaMA2, which only has a limited context window of 4,096, to show that a model can be good at long text language modeling even though it cannot understand long text.

A.2   DETAILS FOR DOWNSTREAM TASKS

We adopt QMSUM and NarrativeQA from LongBench (Bai et al., 2023), a comprehensive benchmark for long-text understanding, to evaluate the models' ability in long-text understanding. The average length of test cases in QMSUM and NarrativeQA are 10,614 and 18,409 respectively. Following Bai et al. (2023), we use the Rouge-L score as the evaluation metric for QMSUM and the F1 score for NarrativeQA.

For the finer-grained line retrieval test, following Dacheng Li* & Zhang (2023), we construct a synthetic dataset, which consists of 3,200 lines of key-value pairs, and ask the model to retrieve the value of a certain key. The key-value pair is shown in the format: "A person named ⟨key⟩ has a cat. His cat's name is ⟨value⟩". The average length of each test case is 76K. We report the average retrieval accuracy of 200 runs.

The detailed prompt is:

Below is a record of lines I want you to remember. Each line begins with a person's name and its cat's name. For each line index, memorize the person's name and the cat's name. At the end of the record, I will ask you to retrieve the cat's name of a certain person. Now the record start:

A person named ⟨key_1⟩ has a cat. His cat's name is ⟨value_1⟩....A person named ⟨key_3200⟩ has a cat. His cat's name is ⟨value3_200⟩.

Now the record is over. I remember the cat's name of person ⟨example_key_1⟩ is ⟨example_value_1⟩.

I remember the cat's name of person ⟨example_key_2⟩ is ⟨example_value_2⟩.

I remember the cat's name of person ⟨example_key_3⟩ is ⟨example_value_3⟩.

I remember the cat's name of person ⟨test_key⟩ is