🧘 *Taking a Deep Breath:* Enhancing Language Modeling of Large Language Models with Sentinel Tokens

Weiyao Luo[1,2,3], **Suncong Zheng**[3], **Heming Xia**[4], **Weikang Wang**[3],
Yan Lei[3,5], **Tianyu Liu**[6], **Shuang Chen**[3] and **Zhifang Sui**[1*]

[1] State Key Laboratory of Multimedia Information Processing, Peking University
[2] School of Software & Microelectronics, Peking University   [3]Tencent MLPD
[4]Department of Computing, The Hong Kong Polytechnic University
[5]Institute of Computing Technology, Chinese Academy of Sciences   [6]Alibaba Group
luoweiyao993@gmail.com

Abstract

Large language models (LLMs) have shown promising efficacy across various tasks, becoming powerful tools in numerous aspects of human life. However, Transformer-based LLMs suffer a performance degradation when modeling long-term contexts due to they discard some information to reduce computational overhead. In this work, we propose a simple yet effective method to enable LLMs to *take a deep breath*, encouraging them to summarize information contained within discrete text chunks. Specifically, we segment the text into multiple chunks and insert special token <SR> at the end of each chunk. We then modify the attention mask to integrate the chunk's information into the corresponding <SR> token. This facilitates LLMs to interpret information not only from historical individual tokens but also from the <SR> token, aggregating the chunk's semantic information. Experiments on language modeling and out-of-domain downstream tasks validate the superiority of our approach. [1]

1 Introduction

In recent years, Transformer-based large language models (LLMs) have become a focal point of research, leading to the emergence of numerous powerful models such as ChatGPT (OpenAI, 2022), GPT-4 (OpenAI, 2023), LLaMA (Touvron et al., 2023a,b) and Mistral (Jiang et al., 2023). However, in mainstream decoder-only models, subsequent tokens can only attend to preceding historical individual tokens without acquiring information from aggregated local contexts, thereby limiting the language modeling capability of LLMs.

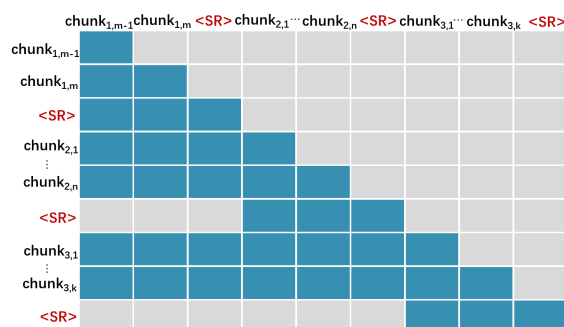To tackle this problem, existing studies have explored various approaches to compress contexts,

Figure 1: The modified attention mask is illustrated in the figure, where cell (r, c) signifies whether token r can attend to token c. $chunk_{2,1}$ represents the first token of the second chunk, with similar patterns for other chunks.

such as sentinel tokens (Ren et al., 2023), memory slots (Ge et al., 2023), and summary vectors (Zhang et al., 2024; Chevalier et al., 2023). However, these methods may lose useful context information during the compression process, leading to performance degradation. Besides, some of these compression and accumulation strategies still exhibit quadratic computing complexity of self-attention (Zhang et al., 2024).

In this work, we introduce a simple yet effective method that allows LLMs to *take a deep breath*, enabling them to gather information not only from preceding historical individual tokens but also from special tokens that encapsulate the holistic information of chunks. Specifically, we propose a strategy to insert new tokens denoted as <SR> (**S**entinel **R**ight) at the end of each chunk. During training, the <SR> corresponding to each chunk is capable of attending to the entire content of the chunk. That is to say, after processing each chunk, the LLMs are prompted to summarize key information within this chunk. Consequently, subsequent tokens can acquire information not only from the preceding individual tokens as in the original approach, but also from the sentinel token <SR> which aggregates the holistic information of the chunk. As a result,

---

[*]Corresponding author.
[1]The code for conducting the experiments is released at https://github.com/WeiyaoLuo/Taking_a_Deep_Breath.

when generating the next token, tokens can harness both local information (individual tokens within a chunk) and relatively holistic information (the <SR> that represents the entire chunk's context). By prompting the LLMs to take a deep breath, we enable them to obtain richer semantic information during decoding, thereby enhancing its language modeling capability.

We conducted experiments on the Wikitext-2 language modeling benchmark using models ranging from 1.3B to 13B in size, employing diverse positional encoding strategies (Devlin et al., 2018; Su et al., 2024). The experimental results demonstrate that the introduction of sentinel tokens enhances the language modeling capabilities of LLMs. Additionally, we further demonstrated the effectiveness of our method on out-of-domain downstream tasks.

2 Related Work

Attention Mask   A number of architectural modifications have been proposed to constrain and sparsify the attention window (Dai et al., 2019; Child et al., 2019; Luo et al., 2024). Ainslie et al. (2023) introduced conditional computation. Beltagy et al. (2020) and Zaheer et al. (2020) introduced some sparse attention mechanisms to reduce computational complexity. However, most of these architectures require expensive training from scratch. Our approach modifies the attention mask while requiring only a small amount of fine-tuning.

Context Distillation   Various strategies (Askell et al., 2021; Snell et al., 2022) have been proposed for prompt compression and context distillation. Ren et al. (2023) adopted a random partitioning approach to compress random contiguous tokens into a single token, resulting in a significant performance degradation. The AutoCompressors (Chevalier et al., 2023) compress context into summary vectors, exhibiting quadratic complexity. Mu et al. (2023) compressed instructions into short prefixes, a method similar to the memory slots introduced by Ge et al. (2023), which may lead to the loss of some useful information. Our method prompts LLMs to take a deep breath, yielding richer semantic information.

3 Approach

In this section, we will present our approach which integrating the comprehensive information of each chunk into the sentinel <SR>. This is achieved by strategically placing sentinel token <SR> at the right flank of a chunk and modifying the attention mask rules accordingly.

3.1 Adding Sentinel Tokens

Assuming there is a text segment that has been divided into multiple chunks, to enable subsequent tokens to extract information not only from individual tokens in the preceding text but also from the collective information from the aggregated semantic content of a chunk, we introduce the special sentinel token: <SR>, which represents sentinel right. The sentinel token is inserted at the end of a chunk to mark its boundary, and it is also added to the model's vocabulary.

Specifically, to absorb the information of a chunk into the <SR> sentinel, we implement a modified causal attention mask, as illustrated in Figure 1. The strategy for ordinary tokens (excluding <SR>) is the same as that of a standard causal attention mask, where they can attend to all preceding tokens. For <SR>, which is the crux of our method, to enable it to encapsulate the semantic information of the corresponding chunk, <SR> can attend to the ordinary tokens within the chunk. Through this modification of the mask strategy, in conjunction with fine-tuning, we promote the condensation of semantic information from a chunk of tokens into the sentinel token at the end, allowing <SR> to become an aggregator capable of selecting and retrieving information from the corresponding chunk.

3.2 Adapting Model Inputs for Sentinel Integration

For the general next token prediction task, the computation of the loss relies on labels corresponding to each token. With the introduction of sentinel tokens, these labels must be adjusted accordingly. Specifically, if the current token is a sentinel token, identified by <SR>, the label for the subsequent position should be uniquely designated to ensure that the location corresponding to the current sentinel token is excluded from the loss calculation.

Conversely, if the next token is a sentinel, its label should be set to that of the subsequent non-sentinel token. Moreover, the position ids of sentinel tokens should be congruent with the position id of the last non-sentinel token preceding the current position.

Table 1: Perplexity (the lower, the better) of six LLMs on the WikiText-2 language modeling benchmark.

Method	OPT-1.3B	OPT-2.7B	RedPajama-3B	Mistral-7B	Llama-2-7b	Llama-2-13b
Origin	14.044	12.416	10.557	6.408	6.235	5.709
Sentinel	**12.664**	**11.313**	**9.387**	**6.176**	**6.01**	**5.512**

4 Experiments

4.1 Models and Data

To validate the effectiveness of our method, we selected models of various sizes and position encoding methods, including OPT-1.3B, OPT-2.7B (Zhang et al., 2022), RedPajama-3B (Computer, 2023), Mistral-7B (Jiang et al., 2023), Llama-2-7B, and Llama-2-13B (Touvron et al., 2023b).

We choose the Wikitext-2 dataset (Merity et al., 2016), which is composed of Wikipedia articles and widely used for evaluating language modeling. By fine-tuning on Wikitext-2, we report perplexity (PPL) on the test set as an evaluation metric. It is noteworthy that sentinel tokens are not included in the computation of PPL, as discussed in Section 3.2.

4.2 Experimental Setup

We treat each individual sentence as a chunk, more details in 4.4. For training, the LoRA (Hu et al., 2021) technique was adopted for fine-tuning during training, with all parameters of the LLM frozen except for the embeddings of the special sentinel tokens and the LoRA matrices. In our approach, the LoRA module is applied to all attention layers, typically comprising the q_proj, k_proj, v_proj, and o_proj parts. The rank of the LoRA is set to 16. We used the AdamW (Loshchilov and Hutter, 2018) with a learning rate of 5e-5. The batch size is set to 12. The entire experiment, including modifications to the attention mask, was based on the Huggingface transformers library (Wolf et al., 2020).

4.3 Results

The experimental results are shown in Table 1, where **Origin** represents the results of the original model after standard fine-tuning without any special modifications, and **Sentinel** denotes the results fine-tuning the model after adding the special sentinel token <SR>. Notably, the methods in Section 2 center on context compression, trading off information for reduced computation and, consequently, lower performance. Given their predictable inferiority to **Origin**, it is superfluous to include these
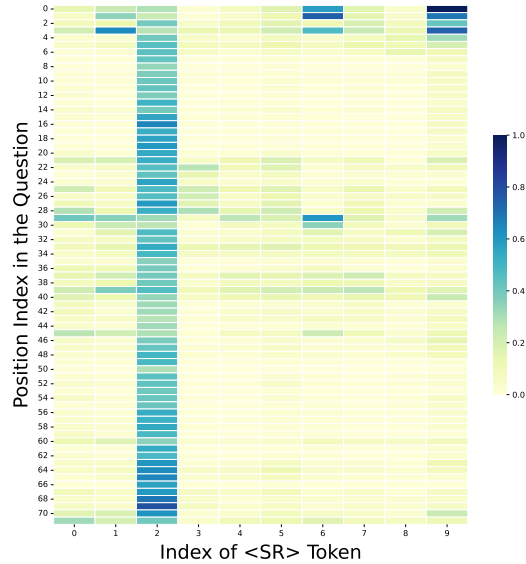

Figure 2: An example of DocumentQA illustrating the attention between each position in the question and sentinel tokens, where correct <SR> index is 2. More detailed explanation provided in Section 4.5.3.

results in Table 1. Our work stands as a pioneering exploration of this method in language modeling.

It can be observed that our method achieved notable performance enhancements across all models, as evidenced by a reduction in perplexity. In the OPT series models and the RedPajama-3B model, perplexity significantly decreased by approximately 10% compared to the vanilla method. In the Llama series and Mistral-7B, perplexity also decreased by about 3.5%.

4.4 Analysis

These results suggest that our approach effectively assists LLMs with the ability in acquiring information from diverse perspectives during next token prediction. By prompting LLMs to take a deep breath, they not only obtain information from each individual token, similar to general LLMs, but also capture more holistic information. This is achieved through the use of the special sentinel token <SR>, which represents the entire chunk of information. By incorporating, the model can take a deep breath to access richer semantic information during decoding, thereby enhancing its language modeling

Table 2: Perplexity (the lower, the better) when a chunk contains different numbers of sentences.

#Sentences	OPT-1.3B	OPT-2.7B	RedPajama-3B	Mistral-7B	Llama-2-7b	Llama-2-13b
1	**12.664**	**11.313**	**9.387**	**6.176**	**6.01**	**5.512**
2	13.637	12.113	10.026	6.438	6.278	5.756
3	14.078	12.509	10.309	6.550	6.412	5.852
4	14.181	12.605	10.385	6.591	6.404	5.863

capability.

Breath Length Analysis To explore the optimal interval for taking a breath to achieve the best results, we conducted experiments with chunks containing 1 to 4 sentences, as shown in Table 2. The findings indicate that the best performance is achieved when the chunk contains only one sentence, and a decrease in performance is observed across all models as the number of sentences increases. Therefore, in the main results presented in Table 1, we adopted the strategy of allowing the model to take a deep breath after every sentence to achieve the best effect.

4.5 Generalization of Model Performance on Out-of-Domain

4.5.1 Data

We select 1,105 samples of the MLQA dataset (Lewis et al., 2019) for DocumenQA task evaluation. Additionally, we choose 1,120 samples of MultiNews (Fabbri et al., 2019) to assess the model's performance on summarization. More details can be found in Appendix A.

4.5.2 Models and Metrics

We utilize OPT-1.3B and OPT-2.7B (Zhang et al., 2022), which were previously fine-tuned on the Wikitext-2 dataset, for out-of-domain DocumentQA and summarization tasks, respectively. The evaluation follows LongBench (Bai et al., 2023) methodologies. For DocumentQA, the model generates an output based on the given context and question, and the F1 score is calculated by comparing the predictions with ground-truth references. For summarization, the model generates summaries from a simple prompt and the provided article, with Rouge-L scores (Lin, 2004) computed between the generated and reference summaries.

4.5.3 Out-of-domain Results and Analysis

Table 3 presents the experimental results, showing that compared to the original models, adding sentinels significantly improved the performance

Task	OPT-1.3B		OPT-2.7B	
	Origin	Sentinel	Origin	Sentinel
DocumentQA	3.97	**19.89**	17.41	**21.78**
Summarization	2.77	**7.69**	7.58	**11.02**

Table 3: Experimental results for OPT-1.3B and OPT-2.7B on two out-of-domain tasks. We show the F1-Score for the DocumentQA task and Rouge-L for the Summarization task, respectively.

on both out-of-domain tasks, particularly for OPT-1.3B on the DocumentQA task, where it nearly quadrupled the F1 score. Additionally, the larger OPT-2.7B with more parameters outperformed the smaller OPT-1.3B, which is consistent with the scaling law (Kaplan et al., 2020). These findings further validate the effectiveness and robustness of our approach across various out-of-domain tasks.

Figure 2 illustrates the attention distribution of the question sequence towards the special token <SR> in a DocumentQA task case. In this example, the document contains 10 chunks, each followed by an <SR> token. Therefore, the horizontal axis displays the indices of these <SR> tokens, ranging from 0 to 9. In this DocumentQA example, the question segment has a length of 72, as indicated on the vertical axis. This means that the vertical axis represents the index of each position in the question, and each position corresponds to the attention scores for the 10 <SR> tokens in the document.

The index corresponding to the chunk that contains the true answer to the question is 2. The figure reveals that nearly every position within the question segment exhibits a heightened attention value for the token <SR> at index 2. This suggests that the token <SR> encapsulates the semantic information of its corresponding chunk, and this information can be accurately captured during decoding, thereby improving the model's performance.

5 Conclusions

In this work, we introduce a novel approach that prompts LLMs to take a deep breath after encountering each chunk. This strategy enables LLMs

not only to extract information from individual tokens but also to be capable of selecting and retrieving information from aggregators(denoted as `<SR>`) corresponding to each chunk. Experiments on language modeling and out-of-domain downstream tasks demonstrate the effectiveness of our method.

Limitations

In this work, we only prompt Large Language Models (LLMs) to take a deep breath on OPT models with up to 2.7 billion parameters, RedPajama model with 3 billion parameters, the Mistral-7B, and Llama2 models with up to 13 billion parameters. Future endeavors should be directed towards establishing the efficacy of this approach on models of an even greater magnitude. Furthermore, the exploration of a broader range of potential chunk division strategies presents a valuable avenue for further research.

Acknowledgements

References

Joshua Ainslie, Tao Lei, Michiel de Jong, Santiago Ontañón, Siddhartha Brahma, Yury Zemlyanskiy, David Uthus, Mandy Guo, James Lee-Thorp, Yi Tay, et al. 2023. Colt5: Faster long-range transformers with conditional computation. *arXiv preprint arXiv:2303.09752*.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. 2023. Adapting language models to compress contexts. *arXiv preprint arXiv:2305.14788*.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Together Computer. 2023. Redpajama: an open dataset for training large language models.

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 preprint 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 preprint arXiv:1810.04805*.

Alexander R. Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R. Radev. 2019. Multi-news: a large-scale multi-document summarization dataset and abstractive hierarchical model.

Tao Ge, Jing Hu, Xun Wang, Si-Qing Chen, and Furu Wei. 2023. In-context autoencoder for context compression in a large language model. *arXiv preprint arXiv:2307.06945*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2019. Mlqa: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in adam.

Weiyao Luo, Junfeng Ran, Zailong Tian, Sujian Li, and Zhifang Sui. 2024. FaGANet: An evidence-based fact-checking model with integrated encoder leveraging contextual information. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7082–7088, Torino, Italia. ELRA and ICCL.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. 2023. Learning to compress prompts with gist tokens. *arXiv preprint arXiv:2304.08467*.

OpenAI. 2022. Chatgpt.

OpenAI. 2023. Gpt-4 technical report.

Siyu Ren, Qi Jia, and Kenny Zhu. 2023. Context compression for auto-regressive transformers with sentinel tokens. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12860–12867.

Charlie Snell, Dan Klein, and Ruiqi Zhong. 2022. Learning by distilling context. *arXiv preprint arXiv:2209.15189*.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 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.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. 2024. Soaring from 4k to 400k: Extending llm's context with activation beacon. *arXiv preprint arXiv:2401.03462*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Appendix

A   More Data Details

For the document question answering (DocumentQA) task, we selected the MLQA (Lewis et al., 2019) dataset for evaluation. MLQA is a multilingual QA task dataset, from which we choose the English portion as the test set for our task.

DocumentQA refers to a task where a model is provided with a **Document**, which contains several sentences, along with a corresponding **Question**. The expectation is that the model can process the **Document** and output the correct answer to the question. Evaluation is conducted by comparing the model-generated results with the ground-truth labels to judge the quality of the model's output.

For the summarization task, we chose the Multi-News (Fabbri et al., 2019) dataset and filtered it based on example length, ultimately selecting 1120 data points for the evaluation of the summarization task. The original MultiNews dataset includes over 5k examples, but we find that the length of some examples might exceed the context length that our evaluation model can handle, so we ultimately selected 1120 examples for testing.