QUITO: Accelerating Long-Context Reasoning through Query-Guided Context Compression

Wenshan Wang[1], Yihang Wang[2], Yixing Fan*[1], Huaming Liao[1], and Jiafeng Guo[1]

1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
2. Beijing University of Posts and Telecommunications, Beijing, China

Abstract. In-context learning (ICL) capabilities are foundational to the success of large language models (LLMs). Recently, context compression has attracted growing interest since it can largely reduce reasoning complexities and computation costs of LLMs. In this paper, we introduce a novel Query-gUIded aTtention cOmpression (QUITO) method, which leverages attention of the question over the contexts to filter useless information. Specifically, we take a trigger token to calculate the attention distribution of the context in response to the question. Based on the distribution, we propose three different filtering methods to satisfy the budget constraints of the context length. We evaluate the QUITO using two widely-used datasets, namely, NaturalQuestions and ASQA. Experimental results demonstrate that QUITO significantly outperforms established baselines across various datasets and downstream LLMs, underscoring its effectiveness. Our code is available at https://github.com/Wenshansilvia/attention_compressor.

Keywords: Context Compression · In-context Learning · Large Language Model.

1 Introduction

In recent years, LLMs has demonstrated notable reasoning and generating capabilities, significantly enhancing the performance of natural language processing (NLP) tasks [4]. However, these models still exhibit limitations in acquiring real-time information and integrating external knowledge [8]. In-context learning (ICL) addresses these deficiencies by including examples and relevant contexts directly within the prompts[6]. This approach boost the performance of LLMs in downstream tasks without requiring additional training.

To better improve the reasoning ability of LLMs, researchers propose different ways to incorporate complex contexts in the input [4, 8]. For example, retrieval-augmented generation (RAG) employs an additional searcher to retrieve external relevant documents about the question as the context of inputs, which has attracted lots of attention for both the academia and industry [2, 3,

* Corresponding Author: fanyixing@ict.ac.cn

8]. In addition, Brown et al. [4] found that the number of examples has a great impact to the reasoning performance of LLMs, where more examples tend to bring better performances [17]. Moreover, the chain-of-thought (CoT) [24, 21] further improves the LLMs by involving the reasoning step of each example in the context. While these strategies have the potential to significantly improve the capabilities of LLMs, they also introduce challenges associated with the increased context length, such as higher inference complexity and costs.

To mitigate this issue, context compression in ICL is becoming a prominent solution. On one hand, reducing the length by removing noise from contexts can improve inference efficiency[11, 26]. On the other hand, it meets the input length restrictions of open-source LLMs[20, 27] while also reduces the costs associated with accessing proprietary LLMs. Several methods [11, 10] have been proposed to compress context by estimating the information entropy. This assessment is conducted by utilizing a small external LLM to evaluate the perplexity of individual tokens to identify those that contribute minimal information gain. Tokens that demonstrate low information are subsequently compressed or eliminated. However, neglecting the query during compression may result in the inadvertent deletion of key information.

For the above problem, recent methods such as LongLLMLingua [9] adopt a query-aware compression approach by calculating the perplexity of the context conditioned on the query. Despite this advancement, misalignment between compression model and generation model can lead to inconsistencies in determining which tokens are considered to have "low entropy gain". This discrepancy arises because models may differ in their interpretation and processing of the same information. Our work also scores tokens based on their relevance to the query. However, distinctively, we employ attention metrics rather than perplexity to assess the importance of tokens.

This paper introduces the Query-gUIded aTtention cOmpression (QUITO) method, which strategically selects the context to maintain supporting information by utilizing the attention mechanism. Intuitively, the attention mechanism offers a direct method for analyzing the interactions between the question and the context, moving beyond the sole reliance on models' probabilistic uncertainty. This technique facilitates a more precise identification of the information that is most crucial to the current task. More importantly, the attention-based filtering can be implemented with small LLMs, which improves the computation efficiency.

The main contributions of this study include:

1. This paper proposes a novel context compression method, named QUITO. It utilises self-attention mechanism of Transformers to score the importance of tokens, selecting context relevant to the current query.
2. In contrast to earlier methods that requires a compression model with 7 billion or 13 billion parameters, this method achieves superior results using a smaller LLM with only 0.5 billion parameters.

3. We conduct extensive experiments on two benchmark datasets, which demonstrate the effectiveness of the proposed QUITO. For example, it surpasses strong baselines with an increase in accuracy of up to 20.2.

2 Related Work

In this section, we briefly review two lines of related works, i.e., context compression task and attention mechanism.

2.1 Context Compression Task

To reduce the length of context, earlier efforts[13] opted to summarize and condense retrieved documents using models such as GPT. Other studies [1, 25, 23, 15] focused on distinguishing between useful and redundant information within documents, training a model to extract the most valuable sentences. For example, LeanContext [1] and FILCO [23] train the model to perform sentence-level extraction for the context. Fit-RAG [15] scores sub-paragraphs with sliding context windows. RECOMP [25] uses a generative model to rewrite extracted candidate sentences, thereby ensuring the coherence and naturalness of the summaries.

Approaches that generate summaries do not allow direct control over the compression ratio, resulting in a growing attention on token and word-level compression techniques in recent times. SelectiveContext [11] utilizes self-information within context for token selection. This approach considers perplexity (PPL) to be the representation of the uncertainty of an LLM regarding information carried by contexts. Based on [11], LLMLingua [10] introduces a two-stage, coarse to a fine, compression method. However, these methods fail to consider the relationship between the context and the query. LLMLingua [9] further addresses this gap by calculating context-specific perplexity conditioned on the query.

The aforementioned token-level compression methods utilize perplexity as the primary filtering criterion. However, discrepancies often arise between smaller compression models and larger generation models in their assessments of word perplexity, making it challenging to align their judgments on lexical importance.

2.2 Attention Mechanism

Attention is a significant breakthrough in deep learning, particularly shines in NLP tasks such as translation and summary generation[5]. The core concept behind Attention mechanisms involves assigning a specific weight to each input element, such as words or tokens, indicating their relevance to the task at hand. This allows models to focus selectively on more pertinent parts of the input data.

Self-attention, a particular category of the attention mechanism, measures the relationships between all input elements, assessing how each element influences and relates to the others[16]. Multi-head attention is a key component of the Transformers [22], which improves the model's capability in capturing diverse correlation patterns. Recent studies try to use the attention mechanisms

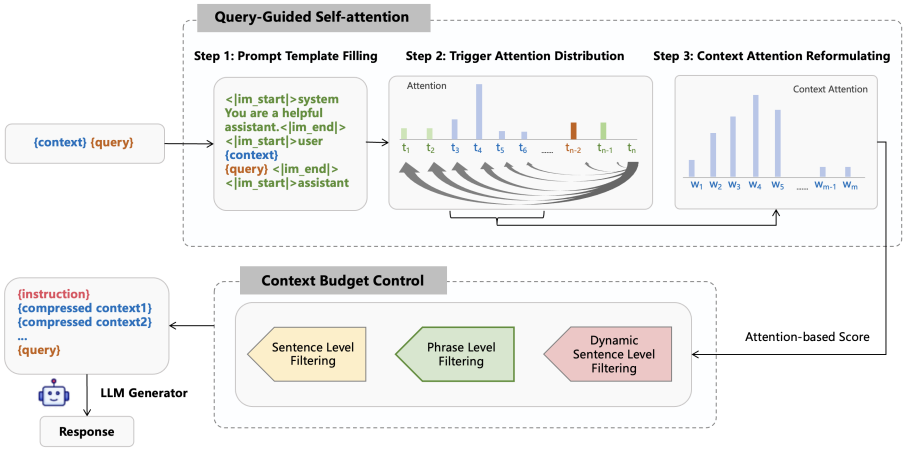

Fig. 1. The overall framework of QUITO

within LLMs to accomplish specific tasks. For instance, DRAGIN [19] use attention to evaluate the extent to which a given text segment significantly influences subsequent content. It employs the perplexity of tokens to determine whether to trigger re-retrieval and regeneration processes. In this paper, we also employ the multi-head attention mechanism to calculate the weights of tokens in context, thereby identifying useless content for answer generation.

3 Method

In this section, we introduce the QUITO method in detail. As illustrated in Figure 1, QUITO primarily consists of two main components, namely *the query-guided self-attention* component and *the context budget control* component. In what follows, we will firstly give a formal definition of the task, and then describe each component in detail.

3.1 Problem Formulation

Given an input $p = (s, C, q)$, where s is the instruction, q is the query, and $C = \{c_i\}_{i=1}^{n}$ is the context consisting n documents. Every document $c_i = \{w_{i,j}\}_{j=1}^{L_i}$ contains L_i word. The objective of context compression task can be formulated as:

$$\min_{\tilde{C}} dist(P(\tilde{y}|s, \tilde{C}, q), P(y|s, C, q)), \tag{1}$$

where $\tilde{y}$ represents the predicted response of the LLM, and y is the ground truth response. $dist(\cdot, \cdot)$ is a function that measures the distance of two distributions, such as KL divergence. $\tilde{C} = \{\tilde{c}_i\}_{i=1}^{n}$ is the compressed context, and $\tilde{c}_i = \{w_j | w_j \in c_i\}_{j=1}^{\tau L_i}, \tau \in [0, 1]$. $\tilde{c}_i$ is c_i being compressed with ratio at $1/\tau$, where smaller τ means higher compression ratio.

3.2 Query-Guided Self-Attention

The query-guided self-attention component aims to estimate the importance of tokens in context by calculating the trigger attention distribution. Firstly, we organize all input by *prompt template filling*. Then, we calculate the importance of all the input with *trigger attention distribution*. Finally, we obtain the lexical units importance within the context by *context attention reformulating*.

Prompt Template Filling It is crucial that the compression model fully understands the task at hand and accurately identifies the information most pertinent to the current query. A standard approach involves concatenating the context with the query and subsequently analyzing how tokens within the query attend to tokens in the context. However, in a Transformer decoder-only architecture, the visibility range of each token in the query varies. This variability suggests that tokens positioned later in the sequence more precisely reflect the model's comprehensive understanding of the task. Given the challenges associated with appropriately weighting tokens at different positions, we propose a novel method that utilize a conversational template and identify a specialized token that encapsulates the compression model's overall understanding of the task.

Trigger attention distribution We embed the context and query into a conversational template, concluding with a signal that prompts the model to initiate response generation. The terminal token within this sequence is designated as a trigger token, serving as an indicator of the model's assessment of information need after comprehensively understanding the task at hand. Subsequently, we employ a compression model equipped with a multi-head self-attention mechanism to process the completed template and compute the attention that the trigger token accords to the preceding text:

$$\{\alpha_i | \alpha_i = \frac{\exp(q_{L_{total}}^T k_i)}{\sum_{j=1}^{L_{total}} \exp(q_{L_{total}}^T k_j)}\}, \tag{2}$$

where q_i and k_i are query embedding and key embedding of the ith token, respectively. L_{total} is the total number of tokens in the completed template.

Context attention reformulating Once the attention allocated by the trigger token to all preceding tokens in the sequence has been determined, the subsequent step involves transforming this attention data into a quantified measure of significance for the lexical units within the context.

The array $\{\alpha_i\}$ signifies attention weights, with its length equating to the aggregate of the lengths of the conversational template, the context, and query. Within the scope of this task, it is imperative to concentrate on the attention distributed to the context segment. The attention should not be diluted by the segments pertaining to the template and the query. Consequently, we implement a normalization process, which is designed to ensure that the distribution of

attention across various tokens in the context remains unbiased, robust to the disparities in context and query lengths that may exist across different tasks. For the normalization we use softmax function:

$$\alpha'_i = \frac{\exp(\alpha_{i+doc_{start}})}{\sum_{j=doc_{start}}^{doc_{end}} \exp(\alpha_j)}, i \in [1, doc_{end} - doc_{start}], \tag{3}$$

where doc_{start} and doc_{end} represent the starting and ending positions, respectively, of the context segment.

We consider words to be the smallest semantic units within a document. In order to perform selection on semantic units, the next step involves transforming scores on token to scores attributed to each individual unit. In other words, we need to transform $\{\alpha'_i\}_{i=1}^{L_{doc}}$ to $\{\alpha''_i\}_{i=1}^{L}$, where L_{doc} is the length of token array $\{t_i\}_{i=1}^{L_{doc}}$ that belongs to context, and L is the length of word array $\{w_i\}_{i=1}^{L}$.

A word w_i may consist of one or more tokens. We can formulate a word as $w_i = \{t_j\}_{j=k+1}^{k+l}$, each of which has attention score:

$$\alpha''_i = \max_{k+1 \le j \le k+l} \alpha'_j, \tag{4}$$

where the length of the array $\{\alpha''_i\}_{i=1}^{L}$ is L.

3.3 Context Budget Control

In the previous section, we have derived a list of words, represented as $\{w_i\}_{i=1}^{L}$, and the corresponding array of attention weights, $\{\alpha''_i\}_{i=1}^{L}$. This section introduces the filtering methods that satisfy the requirement of the context budget control.

Phrase Level Filtering In the process of selecting based on attention scores, it is common to inadvertently overlook words adjacent to those with high attention, referred to as target words, which may also contain crucial knowledge for answering the query. To rectify this oversight and ensure these adjacent words are also considered, we apply a weighted adjustment, allowing them to receive a portion of the attention attributed to the target words. This is accomplished by implementing a Gaussian filter across the word attention array $\{\alpha''_i\}_{i=1}^{L}$.

$$G(x) = \frac{1}{2\pi\sigma^2} \exp(-\frac{x^2}{2\sigma^2}) \tag{5}$$

After the application of the Gaussian function $G(x)$ to $\{\alpha''_i\}_{i=1}^{L}$, the resulting Gaussian-modulated attention array is denoted as $\{\alpha'''_i\}_{i=1}^{L}$.

Subsequently, we identify the words from set $\{w_i\}_{i=1}^{L}$ that rank within the top τL based on their attention scores $\{\alpha'''_i\}_{i=1}^{L}$.

1. Perform a sort on $\{\alpha'''_i\}_{i=1}^{L}$ on descending order, which yields an ordered set of indices $\{j_1, j_2, \ldots, j_L\}$.

2. Select corresponding words from $\{w_i\}_{i=1}^{L}$ with index $\{j_1, j_2, \ldots, j_{\tau L}\}$,which yields $\{w'_i\}_{i=1}^{\tau L}$.
3. Reorganize the set of selected words $\{w'_i\}_{i=1}^{\tau L}$ to reflect their original sequential order within the context.

Although the selection process targets individual words, the application of Gaussian filtering often leads to the selection of contiguous words, thereby effectively forming phrases.

Sentence Level Filtering In addition to phrase-level filtering, sentence-level filtering is also implemented to preserve more comprehensive semantic information. Using the Natural Language Toolkit (NLTK) toolkit, we extract semantic units at the sentence level. Each sentence s_i, denoted as $s_i = \{w_j\}_{j=k+1}^{k+l}$, is assigned an attention score based on the maximum score of the tokens it contains. Subsequently, mirroring the phrase-level filtering process, we prioritize incorporating sentences with higher attention scores into the selection set, while ensuring that the aggregate word count remains below τL.

Dynamic Sentence Level Filtering Sentence-level filtering often leads to a compression ratio greater than the designated target $1/\tau$. To more effectively adhere to the predetermined compression rate and optimize budget utilization, we augment the results of sentence-level filtering with word-level filtering. Specifically, subsequent to sentence-level filtering, if the count of words is L', we are then able to select an additional $\tau L - L'$ words. These additional words are chosen via phrase-level filtering from the text that was not previously selected. The newly selected words are subsequently concatenated with the results from sentence-level filtering to form the final compressed output.

4 Experiments

4.1 Datasets and Evaluation Metrics

In this paper, we assess the efficacy of the proposed QUITO method across two distinct scenarios: open domain question answering and long-form question answering. Specifically, we employ the NaturalQuestions (NQ) and ASQA datasets as the testbed.

For NQ dataset, We employed a processed version as described in [14], where each query is paired with 20 documents, among which only one document contains the correct answer. In alignment with the procedures specified in [14], accuracy was used as the metric to determine whether the generated responses accurately included the correct answer. For the ASQA dataset, the answer to the question maybe multi-facet as there are many ambiguous questions. Each ambiguous question in the ASQA dataset has answers reflecting multiple interpretations of these ambiguities. We utilize the dataset version provided by [7], which includes 5 retrieved documents/snippets from Wikipedia for each query. In accordance with [18], our evaluation metrics included Exact Match (EM), a RoBERTa-based QA score (DisambigF1), and ROUGE [12].

Methods	NQ	ASQA		
	Accuracy	RougeL	EM	Disambig_F1
ratio=2x				
Selective-Context	53.2	-	-	-
LLMLingua	38.7	21.3	34.6	22.2
LongLLMLingua†	41.2	21.6	29.7	21.2
QUITO (Sentence Level)	49.9	23.5	**40.3**	23.6
QUITO (Dynamic Sentence Level)	58.3	**23.5**	40.0	**23.8**
QUITO (Phrase Level)	**58.9**	21.6	38.3	22.8
ratio=4x				
Selective-Context	38.2	-	-	-
LLMLingua	32.1	20.9	33.2	21.1
LongLLMLingua†	33.6	20.9	24.2	20.2
QUITO (Sentence Level)	52.1	22.1	30.1	20.2
QUITO (Dynamic Sentence Level)	**53.1**	**22.5**	**36.7**	**22.5**
QUITO (Phrase Level)	50.7	20.8	34.7	21.5
Original (without compression)	68.6	23.0	45.7	26.2

Table 1. Experimental results of various compression methods applied at different compression ratios on the NaturalQuestions and ASQA datasets.

4.2 Baselines and Implementation

Baselines We take three state-of-the-art compression approaches as baselines: For query-unaware methods, we select Selective-Context[11] and LLMLingua[10], which implements cross entropy scoring to remove redundant vocabulary. For query-aware method, we compare our approach with Longllmlingua[9]. LongLLMLingua implements a two-stage compression method. It first evaluates and reranks multiple retrieved contexts, followed by a token-level compression stage, allocating varying compression budgets to these contexts based on their initial scores. For fair comparison, we excluded the context reranking phrase of LongLLMLingua (marked as LongLLMLingua† in Table 1 and Figure 2), concentrating on the token-level compression.

Detailed Implementation For fair comparison, we follow LLMLingua [10] to use Longchat-13B-16k [1] as the generation model. To ensure the reproducibility of the results, we apply greedy decoding strategy throughout the inference process, with the temperature parameter set to zero. The compression model is implemented with Qwen2-0.5B-Instruct[2].

[1] https://huggingface.co/lmsys/longchat-13b-16k
[2] https://huggingface.co/Qwen/Qwen2-0.5B-Instruct

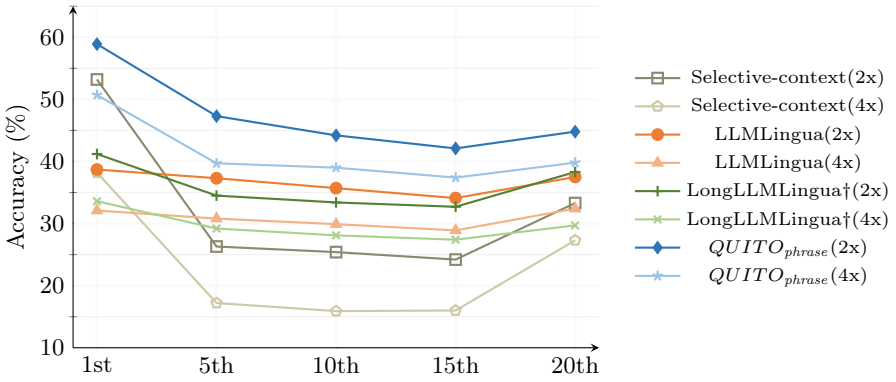

Fig. 2. Experimental comparison of different ground-truth context positions.

4.3 Main Results

Table 1 presents the comparative performance of our method, QUITO, against three baseline methods across various compression rates and datasets. Firstly, we can see that selective-context is a strong baseline compared with LLMLingua and LongLLMLingua† on both $2x$ and $4x$ compression rates. Secondly, QUITO obtains significantly better performances than all baselines, e.g., the improvement of QUITO with phrase level filtering against selective-context, LLMLingua, and LongLLMLingua†(i.e., $2x$ compression ratio) on NQ is 5.7, 20.2, and 17.7, respectively. Finally, we find that QUITO with different filtering method all achieve better performances on both datasets. However, there is no consistent advantages of each filtering method when compared on different datasets. This maybe that the context length on NQ and ASQA differs significantly, i.e., the average length of context on NQ and ASQA is about 2904 and 721 tokens, respectively. All the results demonstrate the effectiveness of QUITO in compressing contexts for the LLMs.

4.4 Analysis on different position of the ground truth context

We analyse the performance of the QUITO compression method across different ground truth context positions within the NQ dataset. This dataset comprises 20 context document fragments per query, of which only one contains the answer and is designated as the ground truth document. We assessed the impact of this document's positioning at the 1st, 5th, 10th, 15th, and 20th ranks on the efficacy of various compression strategies.

The results presented in Figure 2 indicate that all context compression methods struggle with the 'lost in the middle' phenomenon, as described by [14]. Performance is optimal when the ground truth context is positioned at the beginning; however, it deteriorates significantly when the ground truth context is placed in the middle. Among the evaluated methods, LLMLingua[10] exhibits

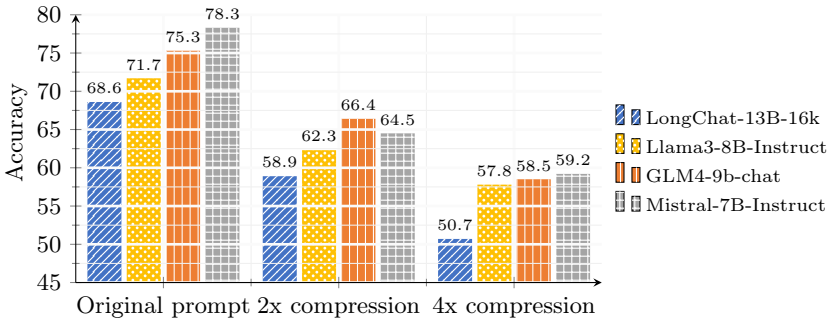

Fig. 3. Experimental results of different generation models on NQ dataset.

the most resilience to the 'lost in the middle' phenomenon. This robustness may be attributed to its strategy of allocating higher compression ratios to contexts containing a greater density of information. Overall, the QUITO method consistently surpasses the two baseline methods across a variety of ground truth context positions and compression rates. On average, QUITO improves upon the performance of Selective Context[11] by +19.6 and LLMLingua[10] by +13.6.

4.5 Analysis on different generation models

To better understand the generation ability of different LLMs, we evaluate the performance of 4 widely-used models, including Longchat-13B-16k [3], Llama3-8b-Instruct [4], GLM4-9b-chat [5], and Mistral-7b-instruct [6]. These models were tested with contexts compressed at a rate of 2 on the NQ dataset. The generated responses from these compressed contexts were then compared with those derived from uncompressed contexts.

As depicted in Figure 3, the Mistral-7B-Instruct model significantly outperforms the other three generation models despite having fewer parameters. This superior performance may be attributed to the incorporation of Grouped-Query Attention (GQA) and Sliding Window Attention (SWA) during its training phase, which enhances its capability to process long sequence inputs. While the context is compressed at 2x ratio, we find that the GLM4-9b-chat model show the smallest performance decline, with a decrease of 8.9, and the Mistral-7B-Instruct has the greatest decline. When the compression ratio is 4x, we can see that all generation models obtain a relative close performance except for LongChat-13B-16k. These maybe that the LongChat-13B-16k is released earlier than other three models, and the latter are trained more deeply.

[3] https://huggingface.co/lmsys/longchat-13b-16k
[4] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct
[5] https://huggingface.co/THUDM/glm-4-9b-chat
[6] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

5 Conclusion

This paper introduces the QUITO method, a novel attention-based importance estimation for long context compression in LLMs. The QUITO method employs a trigger token that comprehensively considers the query to assess the importance of each lexical unit within the context, thereby filtering out units with low relevance scores. Evaluations conducted on the NQ and ASQA datasets demonstrate that our method outperforms state-of-the-art compression methods such as Selective Context, LLMLingua, and LongLLMLingua, confirming its superior ability to preserve essential information needed by LLMs to respond to queries effectively. For future work, we would like to study the combination of the context compression and re-ranking module, since the re-ranking stage in RAG also targets on selecting useful information for final answer generation.

Acknowledgments. This work was funded by the National Natural Science Foundation of China (NSFC) under Grants No. 62372431, the Strategic Priority Research Program of the CAS under Grants No. XDB0680102, the National Key Research and Development Program of China under Grants No. 2023YFA1011602, the Youth Innovation Promotion Association CAS under Grants No. 2021100, the Lenovo-CAS Joint Lab Youth Scientist Project, and the project under Grants No. JCKY2022130C039.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Arefeen, M.A., Debnath, B., Chakradhar, S.: Leancontext: Cost-efficient domain-specific question answering using llms (2023), https://arxiv.org/abs/2309.00841
2. Asai, A., Wu, Z., Wang, Y., Sil, A., Hajishirzi, H.: Self-rag: Learning to retrieve, generate, and critique through self-reflection (2023), https://arxiv.org/abs/2310.11511
3. Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., van den Driessche, G., Lespiau, J.B., Damoc, B., Clark, A., de Las Casas, D., Guy, A., Menick, J., Ring, R., Hennigan, T., Huang, S., Maggiore, L., Jones, C., Cassirer, A., Brock, A., Paganini, M., Irving, G., Vinyals, O., Osindero, S., Simonyan, K., Rae, J.W., Elsen, E., Sifre, L.: Improving language models by retrieving from trillions of tokens (2022), https://arxiv.org/abs/2112.04426
4. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. arXiv preprint arXiv:2005.14165 (2020)
5. Chaudhari, S., Mithal, V., Polatkan, G., Ramanath, R.: An attentive survey of attention models (2021), https://arxiv.org/abs/1904.02874
6. Dong, Q., Li, L., Dai, D., Zheng, C., Ma, J., Li, R., Xia, H., Xu, J., Wu, Z., Chang, B., Sun, X., Li, L., Sui, Z.: A survey on in-context learning (2024), https://arxiv.org/abs/2301.00234

7. Gao, T., Yen, H., Yu, J., Chen, D.: Enabling large language models to generate text with citations. In: Bouamor, H., Pino, J., Bali, K. (eds.) Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. pp. 6465–6488. Association for Computational Linguistics, Singapore (Dec 2023). https://doi.org/10.18653/v1/2023.emnlp-main.398, https://aclanthology.org/2023.emnlp-main.398
8. Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Guo, Q., Wang, M., Wang, H.: Retrieval-augmented generation for large language models: A survey (2024)
9. Jiang, H., Wu, Q., , Luo, X., Li, D., Lin, C.Y., Yang, Y., Qiu, L.: Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. ArXiv preprint **abs/2310.06839** (2023), https://arxiv.org/abs/2310.06839
10. Jiang, H., Wu, Q., Lin, C.Y., Yang, Y., Qiu, L.: LLMLingua: Compressing prompts for accelerated inference of large language models. In: Bouamor, H., Pino, J., Bali, K. (eds.) Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. pp. 13358–13376. Association for Computational Linguistics, Singapore (Dec 2023). https://doi.org/10.18653/v1/2023.emnlp-main.825, https://aclanthology.org/2023.emnlp-main.825
11. Li, Y., Dong, B., Lin, C., Guerin, F.: Compressing context to enhance inference efficiency of large language models (2023)
12. Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: Text Summarization Branches Out. pp. 74–81. Association for Computational Linguistics, Barcelona, Spain (Jul 2004), https://aclanthology.org/W04-1013
13. Liu, J., Jin, J., Wang, Z., Cheng, J., Dou, Z., Wen, J.R.: Reta-llm: A retrieval-augmented large language model toolkit (2023), https://arxiv.org/abs/2306.05212
14. Liu, N.F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., Liang, P.: Lost in the middle: How language models use long contexts (2023), arXiv:2307.03172
15. Mao, Y., Dong, X., Xu, W., Gao, Y., Wei, B., Zhang, Y.: Fit-rag: Black-box rag with factual information and token reduction. ACM Trans. Inf. Syst. (jul 2024). https://doi.org/10.1145/3676957, https://doi.org/10.1145/3676957, just Accepted
16. de Santana Correia, A., Colombini, E.L.: Attention, please! a survey of neural attention models in deep learning (2021), https://arxiv.org/abs/2103.16775
17. Song, Y., Wang, T., Mondal, S.K., Sahoo, J.P.: A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities (2022), https://arxiv.org/abs/2205.06743
18. Stelmakh, I., Luan, Y., Dhingra, B., Chang, M.W.: ASQA: Factoid questions meet long-form answers. In: Goldberg, Y., Kozareva, Z., Zhang, Y. (eds.) Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. pp. 8273–8288. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Dec 2022). https://doi.org/10.18653/v1/2022.emnlp-main.566, https://aclanthology.org/2022.emnlp-main.566
19. Su, W., Tang, Y., Ai, Q., Wu, Z., Liu, Y.: Dragin: Dynamic retrieval augmented generation based on the real-time information needs of large language models. ArXiv **abs/2403.10081** (2024), https://api.semanticscholar.org/CorpusID:268509926
20. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G.: Llama: Open and efficient foundation language models (2023), https://arxiv.org/abs/2302.13971

21. Trivedi, H., Balasubramanian, N., Khot, T., Sabharwal, A.: Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions (2023), https://arxiv.org/abs/2212.10509
22. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS). pp. 6000–6010 (2017)
23. Wang, Z., Araki, J., Jiang, Z., Parvez, M.R., Neubig, G.: Learning to filter context for retrieval-augmented generation (2023), https://arxiv.org/abs/2311.08377
24. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., Zhou, D.: Chain-of-thought prompting elicits reasoning in large language models (2023), https://arxiv.org/abs/2201.11903
25. Xu, F., Shi, W., Choi, E.: Recomp: Improving retrieval-augmented lms with compression and selective augmentation. ArXiv **abs/2310.04408** (2023), https://api.semanticscholar.org/CorpusID:263830734
26. Xu, Z., Liu, Z., Chen, B., Tang, Y., Wang, J., Zhou, K., Hu, X., Shrivastava, A.: Compress, then prompt: Improving accuracy-efficiency trade-off of llm inference with transferable prompt (2023), https://arxiv.org/abs/2305.11186
27. Yang, A., Xiao, B., Wang, B., Zhang, B., Bian, C., Yin, C., Lv, C., Pan, D., Wang, D., Yan, D., Yang, F., Deng, F., Wang, F., Liu, F., Ai, G., Dong, G., Zhao, H., Xu, H., Sun, H., Zhang, H., Liu, H., Ji, J., Xie, J., Dai, J., Fang, K., Su, L., Song, L., Liu, L., Ru, L., Ma, L., Wang, M., Liu, M., Lin, M., Nie, N., Guo, P., Sun, R., Zhang, T., Li, T., Li, T., Cheng, W., Chen, W., Zeng, X., Wang, X., Chen, X., Men, X., Yu, X., Pan, X., Shen, Y., Wang, Y., Li, Y., Jiang, Y., Gao, Y., Zhang, Y., Zhou, Z., Wu, Z.: Baichuan 2: Open large-scale language models (2023), https://arxiv.org/abs/2309.10305