Proceedings of the 8th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/2026.35523

A Review of Long-Context Processing Ability in Large Language Models: Technical Progress and Challenges

Junhao Wu

Guangzhou Foreign Language School, Guangzhou, China
shofairlyplayuld@outlookl.com

Abstract. In recent years, large language models (LLMs) have achieved significant results in natural language processing. They are applied to various tasks, including text generation, question answering, automatic summarization, code generation, and complex reasoning. With the increasingly complex real scenarios, the length of input text that models need to deal with also grows. Thus, the long-context processing ability of language models has gradually become an important factor in evaluating the practicability of LLMs. This paper gives an introduction to the long-context processing ability of large language models. It first introduces the background of large language models and the basic concept of long-context processing. It then summarizes the main technical methods of long-context modeling, such as improving positional encoding, training stage expansion, inference-stage optimization, and architecture-level innovation. Third, the paper also discusses the use of long-context ability in long-document question answering, long-text summarization, multi-document integration, code understanding and long-context evaluation tasks. Then, summarize the current main challenges and prospects of research work. This paper argues that the ability of long context should not only come from increasing the context window, but also from the ability of the model to locate, integrate and reason about important information in long text.

Keywords: large language models, long-context processing, context window, Transformer, long-document understanding

1. Introduction

1.1. Research background

Recently, large language models have achieved great success in natural language processing and have been widely used in text generation, translation, automatic summarization, question answering, code generation, complex reasoning and so on. Large language models are usually larger in parameter scale, richer in training data, and stronger in the ability of language understanding and generation compared with previous natural language processing models. In 2017, Vaswani et al. introduced the Transformer structure, which replaced traditional recurrent neural network and convolution neural network with a self-attention mechanism. It improved the computing parallelism and modeling long distance dependency in sequence modeling and provided an important basis for later large language models [1]. Later, Devlin et al. proposed the BERT (Bidirectional Encoder

DOI: 10.54254/2755-2721/2026.35523

Representations from Transformers), which improved the natural language understanding tasks through bidirectional Transformer pre-training [2]. Brown et al. proposed the GPT-3 (Generative Pre-training 3), showcasing the strong few-shot learning ability of large-scale autoregressive language models [3].

These studies inspired later work on language models and helped them evolve from language modeling tools into more general NLP systems.

As LLMs become more powerful, their context windows are also increasing. Early LLMs were also good at language generation, but they could read at a time only a limited amount of text. GPT-3 is a good example. Its context window was 2048 tokens [3]. This is long enough for many short prompts but limiting when the task is a long document, a long summary, a long conversation, etc. In recent years, as model architectures, training schemes and computing resources improve, many representative frontier LLMs have begun to support longer context windows. It enables models to handle longer input text and also makes models more practically useful for document analysis, code understanding, multi-turn conversations, knowledge retrieval, etc.

1.2. Definition and scope of long-context processing

For language models, context means the range of input information that a model can receive and use when understanding or generating text. A context window indicates the maximum number of tokens a model can use in an input. In NLP, context matters because a word or sentence usually depends on what comes before and after it. If the context is too short, the model might overlook earlier parts of a conversation or evidence in a long document. It is also because the context length can influence the above tasks such as multi-turn dialogue, long-document understanding, information extraction, and complex question answering. So, the ability of a model to effectively use context is an important point in assessing language understanding and task completion ability of the model.

However, long-context processing is not the same as simply trying to increase the input length. Long-sequence modeling mainly concerns whether a model can process longer text sequences. Long-context reasoning further requires the model to find the important content in the long text, to integrate the semantic information from different paragraphs and to complete reliable reasoning according to the context. In other words, even if a model has a large context window, it may not be able to use long-context information well and stably. To solve the fixed-length context issue, Dai et al. proposed a transformer named transformer-XL, which uses segment-level recurrence and relative positional encoding to reduce the limitations of traditional Transformer models in modeling long-distance dependencies [4], indicating that long-context processing involves not only input length increase, but also model ability to maintain, retrieve, and reason over long-distance information.

1.3. Research significance and main challenges

Long-context processing ability is important for the practical use of a large language model. In the actual use case, the user often needs models to process long texts (for example, academic papers, contracts, business, enterprise articles, code archives, conversation transcripts, knowledge reference documents) and so on. These tasks usually require the model to capture important information over a long range of texts and ensure consistency in logic and accuracy when the model outputs answers. So long-context ability has become a crucial factor affecting the capability of large language models to solve practical problems and real tasks.

However, improving the long-context processing ability also has many new technical challenges: first, for the standard Transformer model, its self-attention mechanisms require high computational

DOI: 10.54254/2755-2721/2026.35523

and memory overhead when processing long sequences. The training and inference cost increases sharply when the input length increases. To tackle this challenge, Beltagy et al. proposed Longformer, in which local window attention and global attention are used to reduce the computational complexity in processing long-document input [5]. Zaheer et al. proposed BigBird, in which sparse attention is used to enhance Transformer model ability to process long sequences [6]. Second, in a long-text input, a model will experience issues such as attention diffusion, key information loss, or inferior utilization of middle position information. Liu et al. showed that even if language models can receive long contexts, they cannot always use contextual information stably.

In particular, the performance of a model might drop when key information is in the middle of the input text [7]. In addition to directly increasing the context window, retrieval-augmented generation provides another way to process long context. Lewis et al. proposed Retrieval-Augmented Generation, which combines a pre-trained generation model with external document retrieval. Such a model can use external information during generation, and thus the model can achieve better performance for many knowledge-intensive natural language processing tasks [8]. This approach is not equivalent to the model receiving all the long text at a time. Instead, it retrieves relevant information first and uses this information to support generation, so it has an important value for long-document QA, knowledge retrieval and domain question answering.

1.4. Contributions and structure of this paper

Considering the above background, this paper introduces the basic concepts and background of long-context processing of large language models. First, it reviews the development of context window expansion in large language models and give brief introductions of long-context processing. It then summarizes the main routes of long-context modeling, such as improving the attention mechanism, enhancing the position encoding, the sparse attention mechanism, the retrieval augmented generation and memory mechanism. Next, it describes typical application scenarios of the long-context ability in natural language processing tasks. Finally, it summarizes the current research challenge and prospects for the further development.

2. Core technical methods for long-context modeling

Improving the ability of long context processing is not only a matter of increasing the number of tokens that the model can receive. This also involves positional modeling, training scheme, inference algorithm and model architecture. For large language models, long-context ability mainly involves two aspects. The first is whether the model can formally input a longer sequence. The second is whether the model can effectively locate, maintain and use important information in long text during reasoning. Accordingly, current research on long-context modeling focuses mainly on improved model positional encoding, long-context training scheme extension, long-context inference scheme optimization and long-context modeling architecture level innovation.

2.1. Positional encoding improvement

Positional encoding is another factor affecting the long context capability of large language models. Because the self-attention mechanism of Transformer itself has no inherent information about sequence order, the model needs positional encoding to distinguish the order of different tokens and relative distance. For short-text tasks, simple positional encoding can usually satisfy their needs. However, when the context length becomes as large as tens of thousands or even hundreds of

DOI: 10.54254/2755-2721/2026.35523

thousands of tokens, the model might have encountered some positions outside its training length. This will cause the poor length extrapolation, abnormal attention distribution, and unstable long-distance dependency modeling. So, improving the positional encoding has become an important approach for long context modelling.

ALiBi is an early positional modelling method to enhance length extrapolation. Press et al. developed ALiBi, and it does not use the common strategy of adding position vectors to token embedding. Instead, ALiBi adds a distance-related linear bias to attention scores. It thus enables a model trained on short sequences to generalize to long inputs [9]. This method is simple and decreases model dependency on fixed length positional embedding during inference for long sequences.

RoPE and various improvement methods of RoPE are some typical positional encoding approaches in current large language models. RoPE adopts rotary positional encoding to embed the positional information into attention calculation, and can well express the relative positional information, but models using RoPE still exhibit lower generalization ability when the input length is beyond training context length. To overcome this problem, the researchers have developed several RoPE extension methods. For instance, Position Interpolation reduces the position indices of long inputs into the range of original training length of the model, so that the model can extend its context window with only a small-scale fine-tuning [10]. YaRN also adopts scaling strategies based on RoPE. YaRN extends the context window of the model with lower cost for training and reduce the data resources and training resource required for long-context fine-tuning [11].

Similarly, PoSE and CLEX also study long-context ability from the point of view of positional modeling. PoSE performs positional skip-wise training, to simulate longer positions under a short training window, to reduce the cost of full long-sequence training [12]. CLEX models positional encoding scaling as a continuous length extrapolation problem and, tries to enhance the stability of the generalization of the model on the different length ranges [13]. Generally, the goal of improving positional encoding is to help models maintain relatively stable position awareness and long-distance dependency modeling ability for inputs longer than the original training length.

2.2. Training-stage extension

Besides improving positional encoding, the data and length strategy during training is also important to enhance long-context ability. Although a model architecture may be designed to support longer context length, if it does not see sufficient long-text samples during training, it may still not learn how to find and use important information in long texts. Long-context ability is affected by model architecture as well as data organisation and training length design.

Progressive training is also a common idea. In progressive training, the model is first trained on shorter sequences, and then progressively adjusted to longer input sequence lengths. For example, the sequence length can be extended from 4K tokens to 32K tokens, then to 128K tokens or even longer. This can reduce the workload of directly training on super-long sequences, and meanwhile can make the model slowly adjust to longer positional distributions and semantic dependencies. However, for closed-source commercial models, the exact training length schedule is usually not completely open. For this reason, in this paper, the progressive training is discussed as a common training idea rather than a firm description of the training method of any specific closed-source model.

Position Interpolation is another approach to expand the context window during training. Chen et al. proposed Position Interpolation to linearly interpolate the position index and map positions outside of the original context window to positions inside the model's context range trained during

DOI: 10.54254/2755-2721/2026.35523

training [10]. Such a method to expand the context window does not affect the model's structure greatly. A relatively small fine-tuning effort can enable a model originally trained with short context windows to adapt to longer context inputs.

Other training approaches for improving long-context ability are mixed-length training. Usually, both the short-text samples and the long-text samples remain in the training data for mixed-length training. In this way, the model retains the ability for short-text tasks, while gradually adapting to the understanding ability of long documents, long-text question answering, multi-turn dialogue tasks and so on. Compared with training with only long text data, mixed-length training reduces to some extent the dependence of the model on one length distribution, and helps the model maintain a more stable effect under different input lengths.

2.3. Inference-stage optimization

Long-context modeling has cost problems not only in training, but also in inference. For autoregressive large language models, the model needs to store the Key and Value states of historical tokens during the generation process. This is called the KV cache. The longer the context length is, the larger the memory cost of the KV cache is. This directly limits the usage of models in long conversations, long-document question answering, and real-time generation. Hence, the optimization during the inference stage focuses on saving computation and memory cost without significantly reducing model performance.

StreamingLLM is an inference-stage optimization strategy. Xiao et al. found that when doing long-text streaming generation, some initial tokens become attention sinks. "Attention sink" means that the model still pays attention to these initial positions too much even though those positions may not be the most important positions with semantic information. According to this phenomenon, StreamingLLM keeps the initial sink tokens and then combines them with a sliding-window strategy, so that the model can have stable generation under limited cache in long conversations or streaming text generation [14].

LM-Infinite and LongLLaMA also aim to improve models' generalization capability in extremely long contexts. LM-Infinite tries to improve generalization of existing language models in dealing with inputs that are beyond the original length in training, without updating model parameters, and brings down the memory cost under some settings [15]. LongLLaMA is designed based on the idea of Focused Transformer. It uses training techniques to enhance the ability of model's attention focus in long contexts to help the model to deal with long-distance information better [16].

H2O reduces the computation of long context inference by pruning the KV cache. Zhang et al. proposed a novel Heavy-Hitter Oracle, arguing that only a few tokens contribute the most to generation in attention computation. Thus, the model can dynamically keep important tokens and recent tokens, while removing less useful cache content [17]. It alleviates the KV cache memory cost and can increase the efficiency of long-text generation to some degree. This method has large usefulness for multi-turn dialogue and long-content generation.

2.4. Architecture-level innovation

Except for the encoding of positions, training strategies and acceleration of inference, the innovation of architecture level is also another important direction for solving the problem of long context. Self-attention mechanism in the standard Transformer has high computational complexity with the increase in the sequence length. So researchers started to consider more possible ways such as linear attention, state space models, hybrid architecture to reduce the cost of long-sequence modelling.

DOI: 10.54254/2755-2721/2026.35523

Mamba is a state space model architecture structure that has been paid much attention to recently. Gu and Dao proposed Mamba, which realizes the selective state space mechanism. It can selectively hold or forget information according to the input content, and achieve linear scaling for sequence length dimension [18]. Compared with the standard Transformer, Mamba has advantages of efficiency for the modelling of long sequence. So it can be considered as an important direction for exploring long context in modeling beyond Transformer.

RetNet and RWKV also have similar ideas. RetNet has a retention mechanism and tries to achieve parallel training, low-cost inference, and long-sequence model in parallel [19]. RWKV combines RNN and Transformer characteristics. RWKV has a good parallel training ability and the inference efficiency close to recurrent model, at the same time [20]. These methods strive for the seeking balance between the strong representation ability of Transformer and the efficient long-sequence processing ability of the RNN-like model.

Hybrid architecture even attempts to merge the strengths of different model structure. For instance, Jamba composes of Transformer, Mamba and mixture-of-experts mechanisms. Jamba expects to retain strong language modeling ability and decrease the long-context processing cost [21]. Zamba also adopts a hybrid structure of SSM and Transformer, and aims to get a balance of the performance on inference speed, memory occupation and model accuracy [22]. Hybrid architectures suggest that researchers are testing a long-sequence model avenue beyond scaling up the standard Transformer. It also indicates that the fusion of different architectures may help in addressing the long-context processing efficiency.

Another system-level method for the expansion of long context is Ring Attention. Liu et al. proposed Ring Attention which applies block computing and communicating on several devices. Ring Attention divides extremely long sequences into several devices to expand the context length that the model could process [23]. Different from the method directly modifying the model structure, Ring Attention pays more attention to the parallel design of the training system and the inference system. Ring Attention is suitable for long text or multimodal long sequences of extremely long length in large model hardware environments.

In summary, the core methods for long-context modeling can be divided into four categories. The first category is positional encoding improvement, which mainly addresses stable generalization when models face positions beyond their training length. The second category is training-stage extension, which focuses on how models learn long-distance dependencies through long-text data and length strategies. The third category is inference-stage optimization, which mainly reduces memory and computation costs during long-context generation. The fourth category is architecture-level innovation, which explores new long-sequence modeling methods beyond or together with Transformer. Collectively, these technical methods have helped extend LLMs from short-text processing to long-document understanding, long-dialogue reasoning, and complex information integration.

3. Application scenarios and evaluation tasks of long-context ability

The previous section reviewed the main technical methods for long-context modeling, including positional encoding improvement, training-stage extension, inference-stage optimization, and architecture-level innovation. The techniques of long-context modelling are not developed only to increase the number of tokens a model can get. More profoundly, they help models understand, locate, combine and use long-text information better for actual tasks. So, the scenarios for long-context ability can be discussed through long-document question answering, long-text

DOI: 10.54254/2755-2721/2026.35523

summarization, long-document information integration, code understanding, long-context reasoning evaluation etc.

3.1. Long-document question answering and academic paper reading

Long-document question answering is one of the most straightforward application scenarios of long-context ability. In traditional short-text question answering, a model only needs to answer questions based on a short passage. In long-document question answering, the model needs to find relevant evidence in a full paper, report, contract or document, and then produce an answer after integrating information in a long-text range. Such tasks not only need a long model input, but also require information searching, relevant evidence finding, cross-paragraph information understanding etc.

Academic paper reading is the typical application of long-document question answering. Research papers usually contain sections such as the abstract, introduction, method, experiment, result and discussion. User questions may involve the experimental setting, model method, data source, explain conclusion and so on. The Qasper dataset is designed for question answering over academic papers [24]. Its questions are created by experts that read the paper titles and abstracts, and the answers need to be found in the full text of the papers. It shows that academic reading aid, paper question answering and research information retrieval are typical scenarios for which long-context models can show their potential value.

Finally, long-context benchmarks such as LongBench and LooGLE also have long-document question answering tasks. LongBench tests models' long-text understanding capability through single-document question answering and multi-document question answering tasks [25]. LooGLE uses long documents and questions taken from the real world and with different dependency distances, and tests whether models are capable of capturing long-distance dependence [26]. Such evaluations show that long-document question answering does not only test the input length but also test if the model can reliably use the important information in the long text.

3.2. Long-text summarization and document compression

Long-text summarization is another useful application of long-context ability. If the length of a document grows, summarization is not just shortening the text anymore. A model needs to find important information in a long article, understand the overall structure of the article and produce a summary that is coherent and comprehensive. For example, government reports, papers, enterprise documents, laws and regulations, long literary works may have long articles, chapters and complex themes. A model should be capable of remembering the entire document over long contexts.

GovReport is a dataset for long-document summarization. GovReport mostly consists of long reports from USA government agencies and the summaries of these reports. Unlike a general news summary, government reports tend to be longer and more complex in structure. Their summaries also need to highlight important information from various parts of the whole document [27]. BookSum is for long narrative summarization, such as novels, plays and stories, covering paragraph-level, chapter-level and book-level summarization tasks [28]. These datasets demonstrate that long-text summarization needs not only long input, but also understanding of timelines, characters, causes and effects, and structural aspects of discourse.

From a technical aspect, the positional encoding extension, sparse attention, the inference-stage cache optimization, and architecture-level innovation mentioned in the second section can also provide technical support for long-text summarization. Positional encoding extension can help models understand the order of information in long documents. Sparse attention, linear attention can

DOI: 10.54254/2755-2721/2026.35523

help to reduce the cost for long-document processing. Inference-stage optimization can also help to alleviate the memory pressure when the model generates long summaries.

3.3. Multi-document information integration and knowledge retrieval

However, in real applications, the information users need is not concentrated in one document, but is scattered in multiple documents. For instance, enterprise knowledge base QA system needs product manual, email, meeting records, technical documents at the same time; legal consulting needs laws, contracts and judicial cases documents, etc.; academic studies may also require comparison of several papers. Thus, multi-document information integration is an important application field of long-context ability.

These difficulties of multi-document tasks come from the fact that the model not only has to understand each document, but it also needs to connect and correlate information across different documents, and decide which information is really pertinent to the question. Lengthening the context window enables the model to get more materials at once. But if the model does not understand how to filter and integrate the information, it still might lose key information, mix source information, and yield inaccurate conclusions. Thus, retrieval-augmented generation is valuable for these situations. RAG retrieves document passages relevant to the question, and then sends these passages to the generation model, which reduces the interference of long and irrelevant texts to some extent [8].

The multi-document question answering tasks in LongBench can be used to examine whether the model is able to find and integrate information in multiple documents [25]. LV-Eval also uses single-hop and multi-hop question answering tasks and confusing fact insertion to test the model's anti-interference ability and its multi-step information integration ability in long context [29]. So, multi-document information integration is not only an application of long-context technology, but also an important task type for testing whether the model's long-context ability is reliable.

3.4. Code understanding and repository-level completion

Code understanding is another significant application of long-context capability in software engineering. Unlike general natural language texts, code has strict syntax and interfile dependency. In a real software project, an implementation of a function can depend on classes, variables, interfaces, configuration files of other files. Consequently, if a model needs to accomplish some tasks such as repository-level code completion, code question answering, bug location in a codebase, it should use a wider code context.

Traditional code generation evaluations evaluate either a single file or short function. However, real development work involves cross-file dependencies. RepoBench focuses on this problem and proposes a repository-level code completion benchmark. It covers code retrieval, code completion and retrieval plus completion pipeline tasks. It is used to evaluate whether a model can find relevant code snippets from another file and combine them with the current file context for prediction [30]. This tells us the importance of long context ability to code models directly.

From the technical routes presented in the second section, inference stage optimization and architecture innovation are particularly important for code scenarios. In general, code repositories are long, and the straightforward way is to feed all of code into the model, which brings high cost of computation and memory. Hence, the retrieval-augmented generation, sliding window, KV cache pruning, more efficient architectures of sequence models can all improve the practical usability of code understanding and code generation.

DOI: 10.54254/2755-2721/2026.35523

3.5. Long-context reasoning and anti-interference evaluation

Aside from specific applications, long-context ability also needs to be checked by specialized long-context evaluation benchmarks. A model that has a big context size does not mean that it well understands long texts. Existing studies have shown that model understanding is prone to attention dispersion, information loss and positional bias in long text. In particular, when key information is located in the middle of input text, the model's accuracy may be lower [7]. Thus, the long-context evaluation needs to check if a model can stably locate key information in very long inputs and do dependable reasoning.

BABILong is a representative benchmark for the ultra-long-context reasoning task. BABILong inserts the facts for reasoning inside very long natural texts. The model has to locate useful facts from a large amount of irrelevant information and finish many kinds of tasks, like fact-chain reasoning, induction, deduction, counting, set computing, and others [31]. This kind of evaluation actually evaluates whether the model can reason over long texts and not perform simple information retrieval.

LV-Eval has different length levels (16K, 32K, 64K, 128K and 256K words) to evaluate model performance for different length contexts [29]. It also conducts confusing fact addition, keyword swap and answer keyword recall to reduce knowledge leakage and evaluation bias. LooGLE also pays attention to real long documents and understanding of long-distance dependencies. It also uses relatively new documents and manually designed questions to see if the model can understand long-distance dependency under enough larger context window [26]. These evaluations show that long context research has gradually been transferred from "how many tokens the model can input?" to "whether the model can effectively use these tokens?" So, when assessing the long-context ability of large language models, researchers should not only pay attention to context window size, but also information location, cross-paragraph inference, anti-interference ability and factual consistency.

3.6. Summary

Overall, long-context ability has been gradually applied to tasks like long-document QA, long-text summarization, multi-document integration, code comprehension, and complex reasoning evaluation. Positional encoding improvement, training-stage extension, inference-stage optimization and architecture-level innovation discussed in the second section offer technical support for these applications. The application situations and evaluation scenarios discussed in this section also show the practice value and research significance of long-context ability. In the future, research for long-context models not only should continue in a context window extension and practice, but also in strengthening the ability of the model for information locating, information integration across documents, reasoning over long-distance dependencies.

4. Current challenges

Although large language models do have some progress in the area of long-context processing, they still have many challenges in current research and in practical applications. Such challenges are not limited to whether a model can get longer inputs, but also whether it can precisely understand, properly employ and reliably produce information based on long context. Thus, the main problem of research on long context has gradually transformed from "How long is the context window?" to "Can the model effectively employ long context?"

DOI: 10.54254/2755-2721/2026.35523

4.1. Computational complexity and inference cost

Computational complexity is one of the first challenges for long-context modeling. Self-attention in standard Transformer requires computing the relationship between all the tokens in a sequence. Therefore, as the input length gets bigger, the computation cost and the memory cost get enormous. Longformer and BigBird use local attention, global attention and sparse attention to reduce the cost of long-sequence computation [5, 6]. But the cost efficiency and representation ability need to be balanced in those methods. Similar situations appear in inference. As the length of context becomes larger, the memory cost of the KV cache will continue to grow and give deployment pressure on long conversation and long-document generation. H2O avoids the problem by dynamically pruning the KV cache [17]. But how to compress the cache but still keep the key information is one interesting research question.

4.2. Insufficient use of long context

Long-context ability is not equivalent to having a larger context window. Many models can take long inputs, but they may not stably use all of the information in the context. Liu et al. discussed that language models have positional bias when taking long contexts. When critical information occurs in the middle of input text, model performance might degrade [7]. This indicates that models may pay more attention to the start and end of long text and may disregard the information in the middle. For tasks like long-document question answering, legal contract analysis, academic paper reading, etc., if a model is not able to stably locate information, it may lead to incomplete and incorrect answers even if the context window size is sufficiently large. Therefore, improving long-context ability requires not only expansion of the length of input, but also strengthening of information location, information memory and cross-paragraph reasoning.

4.3. Difficulties in information retrieval and multi-document integration

Real applications show that users also want models to refer to multiple documents or more sources of information. Multi-document question answering, enterprise knowledge base question answering and domain-specific question answering all require models to filter information from large volumes of text, and establish connections between different sources. Multi-document integration is more influenced by the irrelevant information, repeated information and inconsistent information. Retrieval augmented generation can reduce the burden of processing all text at once by using external retrieval [8]. But good retrieval results directly affect the final answer. If in retrieval process some key information is missed or the retrieved passages look alike but are in fact unrelated, the model will still get wrong answers. Therefore, long-context processing not only needs strong generation capabilities, but also reliable information retrieval and evidence integration capabilities.

4.4. Hallucination and reliability problems

Long-context scenarios may further increase hallucination risks and reliability problems for large language models. When the input text is long, the sources of information are abundant, and the task is complex, the model may be more prone to confusing different kinds of information, even producing conclusions that do not appear in the original text. In long-document summarization, a model may miss some information or add information which does not exist in the source. In multi-document question answering, a model may even combine information across different documents

DOI: 10.54254/2755-2721/2026.35523

incorrectly. In the professional domain, wrong answers may be more dangerous. Hence, long-context models should not only seek longer model inputs, but also strive for better factual consistency, source traceability, and answer verifiability. The future long-context models need to learn to incorporate citation mechanism, evidence location and fact-checking methods to enhance the reliability of practical applications.

4.5. Evaluation Standards are still not unified

Evaluation of long-context ability is in initial development. Benchmarks, like LongBench, LooGLE, LV-Eval, and BABILong, measure long-context understanding from different angles [25, 26, 29, 31], but their purposes are not exactly same. LongBench pays more attention to multi-task ability. LooGLE pays attention to long-distance dependency understanding in real long documents. LV-Eval pays more attention to the stability and anti-interference ability under different length levels. BABILong pays attention to the reasoning in facts of ultra-long context. This indicates that long-context ability is a complex concept and cannot be measured only by one index. Future evaluation systems need to consider context length, information positions, task types, reasoning depth, anti-interference ability, and factual consistency at the same time, in order that they can reflect the actual ability of models more completely.

Currently, long-context processing of large language models still faces problems such as high computation cost, unsteady information use, hard to integrate multi-documents, hallucination risk and incomplete evaluation system. All of these problems indicate that research on long-context cannot stop at enlarging the context window. Long-context research should further pay attention to how models understand, select, organize and verify information in long text.

5. Conclusion and future directions

5.1. Conclusion

This paper reviewed the long-context processing ability of large language models. The first section of this paper included the background of large language models and the importance of long-context ability. In the development of Transformer structure and pre-training language models, large language models show strong ability in natural language processing task text understanding and generation [1-3]. However, early models had relatively short contexts, which made them unable to meet the application needs for long-document question answering, long-text summarization, multi-turn long conversations and code comprehension. So, the long-context processing ability has become an important research direction of large language models gradually.

In terms of technical methods, this paper summarized the major ways of long-context modeling, such as positional encoding improvement, training stage extension, inference stage improvement, architecture-level innovation. Positional encoding improvement mainly addresses the problem of position generalization beyond the training length, such as ALiBi, Position interpolation, YaRN [9-11]. Training stage extension uses long-text data, long-text training, position interpolation to make the model more able to adapt to long-distance relationship. Inference stage improvement applies methods such as StreamingLLM, H2O to reduce the memory cost and computational cost of long-context inference, and so on [14, 17]. Architecture-level innovation goes to explore new long-sequence modeling methods, such as Mamba, RetNet, RWKV and novel architectures [18-22].

As to application cases, long-context ability has gradually been applied to long-document question answering, long-text summarization, multi-document information integration, code

DOI: 10.54254/2755-2721/2026.35523

understanding, long-context reasoning evaluation, etc. There are related benchmarks such as LongBench, LooGLE, LV-Eval, and BABILong which investigate model ability on long-text understanding, long-text information location, anti-interference, complex reasoning and other aspects from different angles [25, 26, 29, 31], etc., long-context ability is becoming an important angle to measure the practical value of language models.

Overall, in developing long-context processing ability, large language models have in some sense moved from simple short-text generation tools towards complex information processing systems. Models are no longer only employed for simpler tasks such as simply question answering or simple text generation. Models are starting to show long-document processing capability, being able to utilize multiple sources of information, support more complex tasks, and so on. Of course, for current long-context models, some issues still exist, such as the high computing cost, unstable use of information, hallucination risk, incomplete evaluation systems etc. Therefore, long-context ability will still be an important focus for research in the development of large language models.

5.2. Future directions

For the future, long-context processing ability of large language models can continue to advance in several directions: First, there are still many important studies to do on more efficient long-sequence modeling methods. Even after sparse attention, linear attention and state space models have made some progress on reducing the complexity of computational models, the long-context model is still relatively high training and inference cost for real application scenarios. Future research will need to explore more efficient attention, cache management methods and parallel training strategies in depth, so that models can process longer texts with less cost.

Second, the capabilities of models to use long-context information effectively still need to be improved. More future research should not only attend to the size of context windows, but also focus on whether models can properly find key information, understand relationships between information, and perform reliable reasoning. Especially when the key information appears in the middle of a long text, or in different segments of input, models need a better ability to locate the information and perform across-paragraph reasoning.

Third, retrieval-augmented generation and long-context models can also become an important direction in future research. For extremely long documents or large knowledge bases, simply enlarging context windows may not always be the solution. A more practical method could be retrieval to select relevant content first, followed by a long-context model for understanding and generation. Future systems can combine RAG, a long context window, evidence citation mechanisms to improve answer effectiveness and traceability.

Fourth, the reliability and safety of the long-context models require more attention. For many professional areas such as law, medicine, finance, scientific research, models should not only be able to process long text, but also be accurate and come from reliable sources. Therefore, the future research should further improve the factual consistency check, hallucination reduction, evidence tracking and human review mechanisms, so that long-context models can be used more safely in safety-critical scenarios.

Finally, evaluation systems of long-context ability also need further improvement. Existing benchmarks on long-context ability also investigated long-context ability from different points of view, such as multi-task evaluation, ultra-long reasoning, length-level evaluation and real long documents. These benchmarks still cannot meet the complex needs of real applications. Future evaluations should pay more attention to real scenarios, multi-document integration, long-distance

DOI: 10.54254/2755-2721/2026.35523

causal reasoning, and verifiability of model answers, etc. This will help researchers estimate the long-context ability of large language models more comprehensively.

References

[1] Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*, 5998–6008.

[2] Devlin, J., Chang, M. W., Lee, K., et al. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186.

[3] Brown, T. B., Mann, B., Ryder, N., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems, 33*, 1877–1901.

[4] Dai, Z., Yang, Z., Yang, Y., et al. (2019). Transformer-XL: Attentive language models beyond a fixed-length context. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2978–2988.

[5] Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

[6] Zaheer, M., Guruganesh, G., Dubey, K. A., et al. (2020). Big Bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems, 33*, 17283–17297.

[7] Liu, N. F., Lin, K., Hewitt, J., et al. (2024). Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics, 12*, 157–173.

[8] Lewis, P., Perez, E., Piktus, A., et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems, 33*, 9459–9474.

[9] Press, O., Smith, N. A., & Lewis, M. (2022). Train short, test long: Attention with linear biases enables input length extrapolation. *International Conference on Learning Representations*.

[10] Chen, S., Wong, S., Chen, L., et al. (2023). Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*.

[11] Peng, B., Quesnelle, J., Fan, H., et al. (2024). YaRN: Efficient context window extension of large language models. *International Conference on Learning Representations*.

[12] Zhu, D., Yang, N., Wang, L., et al. (2024). PoSE: Efficient context window extension of LLMs via positional skip-wise training. *International Conference on Learning Representations*.

[13] Chen, G., Li, X., Meng, Z., et al. (2024). CLEX: Continuous length extrapolation for large language models. *International Conference on Learning Representations*.

[14] Xiao, G., Tian, Y., Chen, B., et al. (2024). Efficient streaming language models with attention sinks. *International Conference on Learning Representations*.

[15] Han, C., Wang, Q., Peng, H., et al. (2024). LM-Infinite: Zero-shot extreme length generalization for large language models. *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 3991-4008.

[16] Tworkowski, S., Staniszewski, K., Pacek, M., et al. (2023). Focused transformer: Contrastive training for context scaling. *arXiv preprint arXiv:2307.03170*.

[17] Zhang, Z., Sheng, Y., Zhou, T., et al. (2023). H2O: Heavy-Hitter Oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems, 36*, 34661-34710.

[18] Gu, A., & Dao, T. (2023). Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

[19] Sun, Y., Dong, L., Huang, S., et al. (2023). Retentive network: A successor to Transformer for large language models. *arXiv preprint arXiv:2307.08621*.

[20] Peng, B., Alcaide, E., Anthony, Q., et al. (2023). RWKV: Reinventing RNNs for the Transformer era. *Findings of the Association for Computational Linguistics: EMNLP 2023*, 14048–14077.

[21] Lieber, O., Lenz, B., Bata, H., et al. (2024). Jamba: A hybrid Transformer-Mamba language model. *arXiv preprint arXiv:2403.19887*.

[22] Glorioso, P., Anthony, Q., Tokpanov, Y., et al. (2024). Zamba: A compact 7B SSM hybrid model. *arXiv preprint arXiv:2405.16712*.

[23] Liu, H., Zaharia, M., & Abbeel, P. (2024). RingAttention with blockwise Transformers for near-infinite context. *International Conference on Learning Representations*.

[24] Dasigi, P., Lo, K., Beltagy, I., et al. (2021). A dataset of information-seeking questions and answers anchored in research papers. *Proceedings of the 2021 Conference of the North American Chapter of the Association for*

DOI: 10.54254/2755-2721/2026.35523

Computational Linguistics: Human Language Technologies, 4599–4610.

[25] Bai, Y., Lv, X., Zhang, J., et al. (2024). LongBench: A bilingual, multitask benchmark for long context understanding. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 3119-3137.

[26] Li, J., Wang, M., Zheng, Z., et al. (2024). LooGLE: Can long-context language models understand long contexts? *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 16304-16333.

[27] Huang, L., Cao, S., Parulian, N., et al. (2021). Efficient attentions for long document summarization. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1419-1436.

[28] Kryscinski, W., Rajani, N., Agarwal, D., et al. (2022). BOOKSUM: A collection of datasets for long-form narrative summarization. *Findings of the Association for Computational Linguistics: EMNLP 2022*, 6536-6558.

[29] Yuan, T., Ning, X., Zhou, D., et al. (2024). LV-Eval: A balanced long-context benchmark with 5 length levels up to 256K. *arXiv preprint arXiv:2402.05136*.

[30] Liu, T., Xu, C., and McAuley, J. (2024). RepoBench: Benchmarking repository-level code auto-completion systems. *International Conference on Learning Representations*.

[31] Kuratov, Y., Bulatov, A., Anokhin, P., et al. (2024). BABILong: Testing the limits of LLMs with long context reasoning-in-a-haystack. *arXiv preprint arXiv:2406.10149*.