Sequential-NIAH: A Needle-In-A-Haystack Benchmark for Extracting Sequential *Needles* from Long Contexts

Yifei Yu
Tencent YouTu Lab
Shanghai, China
felixyfyu@tencent.com

Qian-Wen Zhang
Tencent YouTu Lab
Beijing, China
cowenzhang@tencent.com

Lingfeng Qiao
Tencent YouTu Lab
Shanghai, China
leafqiao@tencent.com

Di Yin
Tencent YouTu Lab
Shanghai, China
endymecyyin@tencent.com

Fang Li
Tencent YouTu Lab
Beijing, China
frankfangli@tencent.com

Jie Wang
Tencent YouTu Lab
Beijing, China
jieeewwwang@tencent.com

Zengxi Chen
Tencent
Shanghai, China
hackerchen@tencent.com

Suncong Zheng
Tencent
Beijing, China
congzheng@tencent.com

Xiaolong Liang
Tencent
Shanghai, China
mikexlliang@tencent.com

Xing Sun
Tencent YouTu Lab
Shanghai, China
winfredsun@tencent.com

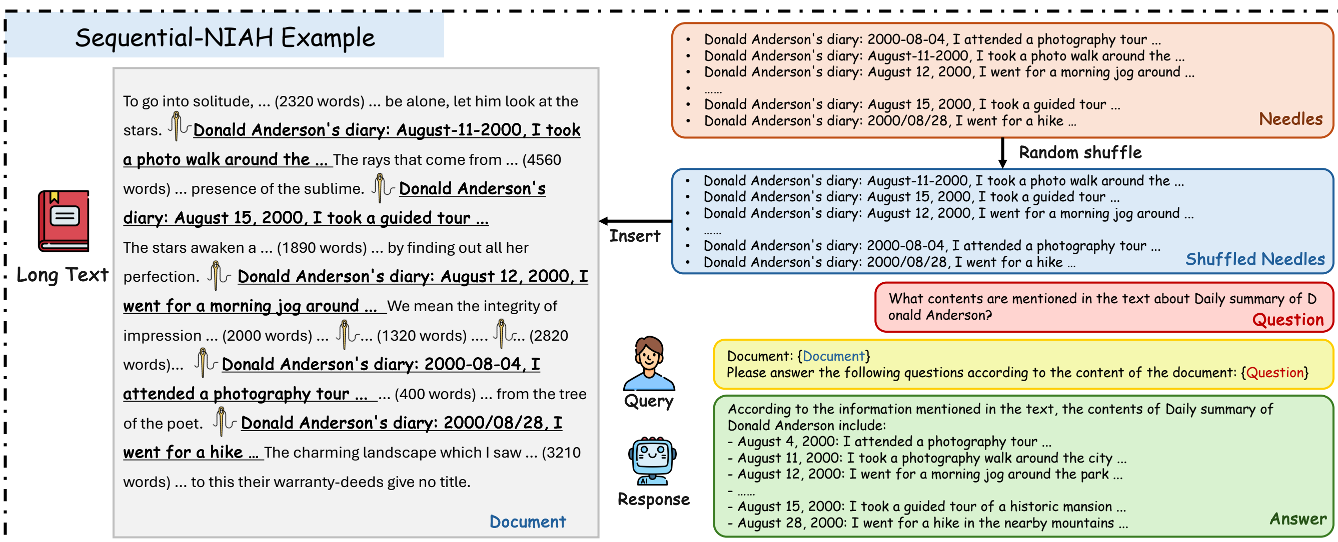


Figure 1: Sequential-NIAH example of a long text with shuffled needles.

Abstract

Evaluating the ability of large language models (LLMs) to handle extended contexts is critical, particularly for retrieving information relevant to specific queries embedded within lengthy inputs. We introduce *Sequential-NIAH*, a benchmark specifically designed to evaluate the capability of LLMs to extract sequential information items (known as *needles*) from long contexts. The benchmark comprises three types of needle generation pipelines: synthetic, real, and open-domain QA. It includes contexts ranging from 8K to 128K tokens in length, with a dataset of 14,000 samples (2,000 reserved for testing). To facilitate evaluation on this benchmark, we trained a synthetic data-driven evaluation model capable of evaluating answer correctness based on chronological or logical order, achieving an accuracy of 99.49% on synthetic test data. We conducted experiments on six well-known LLMs, revealing that even the best-performing model achieved a maximum accuracy of only 63.15%. Further analysis highlights the growing challenges

posed by increasing context lengths and the number of needles, underscoring substantial room for improvement. Additionally, noise robustness experiments validate the reliability of the benchmark, making Sequential-NIAH an important reference for advancing research on long text extraction capabilities of LLMs.

Keywords

Sequential Information Retrieval, Needle-In-A-Haystack Benchmark, Long Context Question Answering, Evaluation Model

1 Introduction

Enhancing the ability of large language models to understand long contexts has been a common pursuit among researchers in the field of NLP. In recent years, numerous LLMs, including Gemini-1.5 [12], GPT-4 [1], Claude-3.5 [3], Qwen-2.5 [36], GLM-4 [41], Kimi [28], and DeepSeek-V2 [23], have attempted to extend the context length of models to hundreds of thousands or even millions of tokens, without weakening the understanding and reasoning capabilities of the model. To evaluate the performance of these models on some challenging tasks, several long context-based benchmarks have been exposed for long context understanding, such as ∞Bench [43], L-Eval [2], LongBench [4], LongEval [19], LooGLE [20] and ZeroSCROLLS [33].

However, *Needle-in-a-Haystack* (NIAH) [13], as a special and challenging long context comprehensive task, is rarely addressed in these benchmarks. NIAH refers to the retrieval of one or more target pieces of information (*needles*) from a long text (*haystack*). In this setting, long text typically consists of irrelevant content with respect to the target information. RULER [15] and Counting-Stars [34] attempt to construct the NIAH benchmark in a synthetic way. But the needles they defined consist of character-level objects devoid of practical meaning, such as password, number, key-value pair, and special symbols, which leads to a significant gap with real-world application scenarios. NeedleBench [21] provides a more realistic NIAH benchmark designed to retrieve semantically meaningful *needles* within long contexts. Additionally, NeedleBench incorporates logical relationships between multiple *needles* to construct more complex question-answer (QA) pairs, thereby evaluating the model's comprehensive understanding of retrieved key information.

Nevertheless, NeedleBench does not address the understanding of chronological or logical sequences. Specifically, if the *needles* scattered throughout a long text have a chronological or logical order, it is desirable for the model not only to retrieve these items but also to understand the sequential relationships among them and list them in correct order. For example, in the case of diary-type information with specific timestamps, the task would require the model to retrieve and list all diary entries in chronological order. Similarly, for cooking instructions with a specific logical sequence, the task would require the model to retrieve and provide the detailed cooking process for a particular dish. These tasks will evaluate the capability of LLMs to not only extract relevant information items from a long text but also comprehend the sequential relationships among these items correctly to produce accurate response.

To supplement existing long context information retrieval evaluation methods, We introduce the Sequential-NIAH benchmark, which shuffles *needles* with chronological or logical order and inserts them into long contexts of varying lengths, as shown in Figure 1. Considering the completeness of the benchmark, we propose three different types of *needles* generation pipelines, including synthetic *needles*, real *needles*, and open-domain QA *needles*. Synthetic *needles* are generated by fake entities, timestamps, and events. Real *needles* are generated from the Temporal Knowledge Graph (TKG), which can be used to build sequential chronological items based on the relationship of two entities overtime. Integrated Crisis Early Warning System2 (ICEWS [11, 17]) and Finance Event Graph (FEG [25]) are used in this pipeline. Open-domain QA *needles* are generated from a private open-domain QA resource. The first two types are mainly aimed at retrieving chronological sequence items, while the last type is mainly aimed at retrieving logical sequence items.

To facilitate the evaluation of various LLMs on this benchmark, we employed synthetic answer pairs to train an evaluation model. A pair of answers consists of one designated as the ground truth and the other as the answer to be evaluated, which can either be correct or contain various types of errors. Furthermore, we designed different evaluation prompt templates for chronologically ordered QA and logically ordered QA. The validation results indicate that the evaluation model achieved an accuracy of 99.49% on the synthetic test set, which is sufficient to be used for reliable automatic evaluation of the performance of LLMs on this benchmark. Using the evaluation model, we evaluated the accuracy of several well-known LLMs on our benchmark. The experimental results indicate that this task is highly challenging, while even the best-performing model achieved a maximum accuracy of only 63.15%. Moreover, the increasing of context lengths and the number of needles will further enhance the challenge of the task. We also verified the reliability of the benchmark through noise robustness experiments, and explored the result consistency of LLMs' responses under noise disturbance.

Our major contributions are as follows:

- We proposed three different types of *needles* generation pipelines: Synthetic *needles*, Real *needles*, and Open-domain QA *needles*, which are used to build the Sequential-NIAH benchmark, providing diverse sequential QA pairs.
- We developed an evaluation model trained on synthetic data to facilitate the evaluation of LLMs' performance on this benchmark. An accuracy of 99.49% in the test set indicates that the model's evaluation of responses to this task is highly reliable and efficient.
- We evaluated the capabilities of six well-known LLMs in long-text information retrieval using the test set of our benchmark. We found that all current LLMs have significant room for improvement in this task, struggling with the complexity of sequential information retrieval within long contexts.

2 Related work

2.1 Long Context Language Models

Many techniques have been used to improve the context length that LLMs can handle. For instance, certain novel position embedding methods, such as ALiBi [32], Position Interpolation [6], RoPE [35] and its variants [24, 31, 38]. And some research aims to reduce context length by memory replay back-propagation [37], recurrent memory augmentation [5], and activation beacon [42]. In addition, there are several methods to extend the context length by modifying the model architecture, such as Mamba [14], FLASHBUTTERFLY [7], and RWKV [30]. Specifically, current mainstream LLMs have significantly enhanced their ability to process long contexts. For example, Gemini 1.5 supports a context length of 1 million tokens, and Kimi supports a context length of 2 million words. Currently, the majority of LLMs support a context length of at least 128k tokens, such as GPT-4o, Claude-3.5, Qwen-2.5, and LLaMA-3.3.

2.2 NIAH Benchmarks and Tasks

NIAH essentially represents a category of long-text information retrieval tasks, primarily assessing the capability of LLMs to extract detailed information relevant to questions from vast amounts of text. The RULER [15] and Counting Starts [34] benchmarks are designed with retrieval tasks at the word or character level, where the problems involved are relatively clear and singular. For instance, these tasks may include the retrieval of numbers or words, the counting of star symbols, etc. NeedleBench [21] takes this one step further by designing more complex information on logical reasoning, such as descriptions of relationships between entities or kinship relationships, and inserting it into a long context. This is done to verify the model's ability to retrieve and understand information from extensive contexts, which approximates the practical application of the NIAH task in real-world scenarios more closely.

2.3 Evaluation Model

The evaluation of natural language generation (NLG) is a vital but challenging problem in artificial intelligence [9]. Its primary methods include the following four types: LLM derived metrics [8, 16], prompt-based LLMs [10, 26, 27, 29], fine-tuning LLMs [18, 39, 45], and collaborative human-LLM evaluation [22, 44]. In practical applications, simply designing prompts often fails to achieve optimal performance in evaluations. Fine-tuning on open-source models can enhance the performance of the evaluation model effectively. If manpower is sufficient, the combination of human effort with LLMs can further improve the reliability of the evaluation. To facilitate a reliable automatic evaluation of Sequential-NIAH tasks, we trained an evaluation model based on Qwen2.5-Instruct-32B, which provides a convenient and reliable evaluation solution for this task.

3 The Sequential-NIAH Benchmark

The goal of Sequential-NIAH is to retrieve sequential needles from long contexts. Therefore, both the needles and the long contexts need to be prepared in advance. We will first introduce the sequential needle synthesis pipeline for the QA source, followed by the introduction of long context source, and finally provide detailed information about the entire synthetic dataset. See the link[1] for details of the dataset.

3.1 QA source

We propose three synthesis pipelines for sequential *needles*, as shown in Figure 2, which are used to build the question with sequential answer items, including synthetic needles (synthetic events in chronological order), real needles (real events in chronological order), and open-domain QA needles (answer items in logical order). All are ultimately presented in the form of a question with multiple answer items (*needles*) inserted into a long text that meets specific length requirements. Table 1 provides detailed information of the number and proportion of QA pairs constructed by different pipelines, collectively referred to as **QA source**. As can be seen, the QA source is mainly composed of synthetic needles, supplemented by real needles, with open-domain QA needles serving as a complement for logical order items. Meanwhile, we adjusted the proportion of Chinese and English QA pairs to maintain each around 50%.

[1]https://anonymous.4open.science/r/Sequential-NIAH-Benchmark-88B7

3.1.1 Synthetic needles. Synthetic needles refers to the synthesis of question-answer pairs using specific generation templates by combining subjects, event times, and event content. The question is usually posed about events that occur within a certain time period for a predefined fake subject, and the answer items are the synthesized events related to the fake subject listed in chronological order. This method can generate an unlimited number of qualified chronological question-answer pairs. We ensure the complexity of the task by designing various question templates and needle templates. It is important to note that the number of needles corresponding to a question can be set manually, and we randomly select the number of needles from 3 to 15.

3.1.2 Real needles. Real needles refers to the extraction of data on how relationships between two different entities change over time from open source TKG datasets (ICEWS and FEG). These datasets is usually organized to represent the evolving relationships between two entities over time, which is then further rewritten into chronological question-answer pairs. Unlike synthetic needles, the needles synthesized by this pipeline is composed of real subjects and events. And the question is usually posed about the changes in relationships between two specific entities over a period of time, and the answer items are the real events related to the two entities listed in chronological order. The amount of data that can be generated by this pipeline is limited by the size of the Graph, and the number of needles corresponding to a question depends on the number of relationships between two entities that change over time in the Graph and cannot be manually set.

Graph data only define structured information (entities and relations with time stamps); for NIAH tasks, needles must be at least a complete sentence describing an event or content. Therefore, we use GPT-4o to generate questions and needles for a specific set of relationships and events. After generating raw needles, to enhance the authenticity of the task, we further rewrite the raw needles to ensure diversity in their expression while constructing a total of 5,014 complete QA pairs.

3.1.3 Open-domain QA needles. In addition to items in chronological order, there are also cases where answers follow a precise logical order. To incorporate these into the benchmark, we filtered out question-answer pairs that meet the requirements from a private open-domain question-answering database, where the answer items all have a definite logical order. To enhance the efficiency of QA screening, we designed specific discriminative templates to filter with the help of GPT-4o, resulting in a total of 2,986 QA pairs

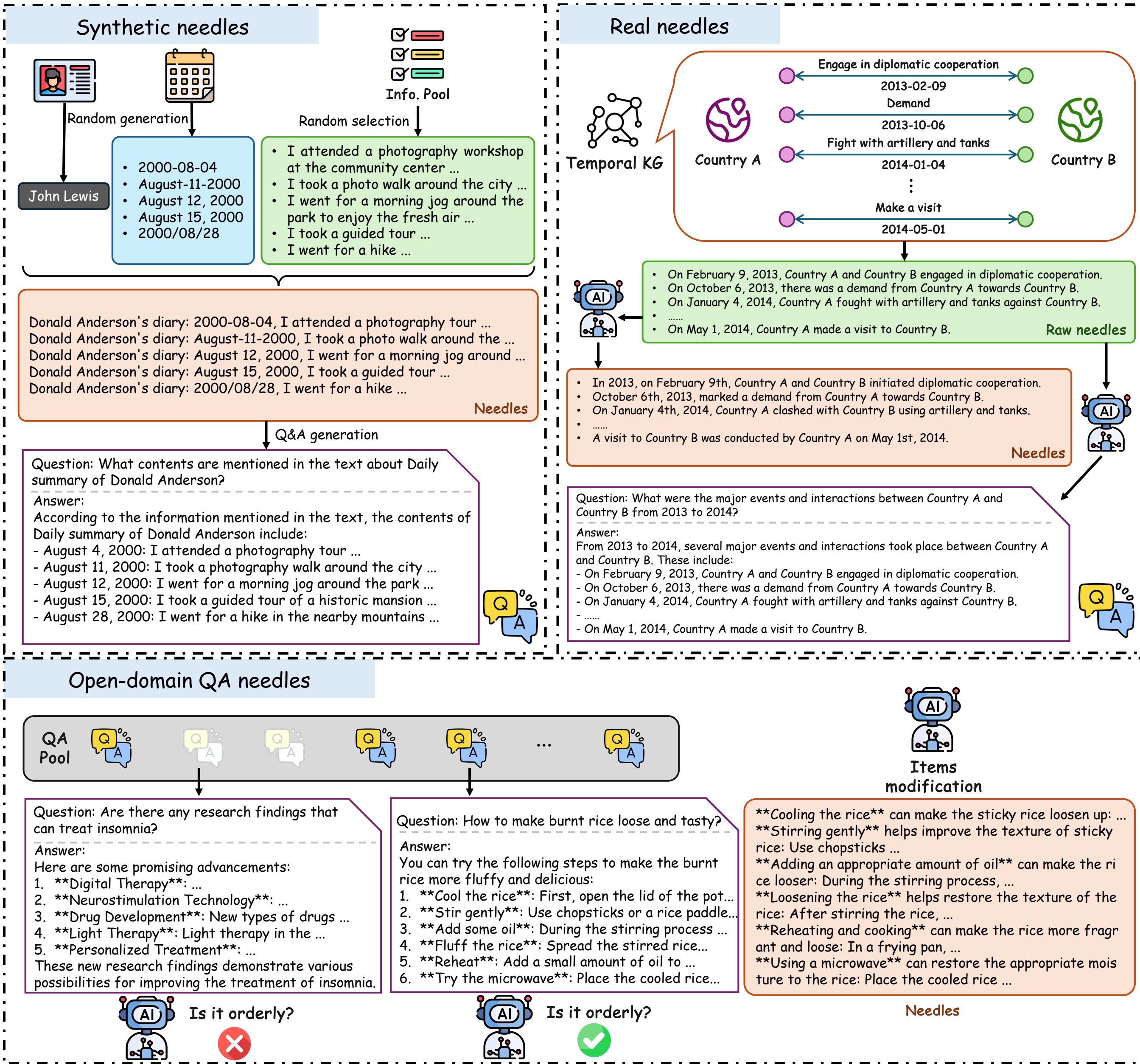


Figure 2: Three pipelines for sequential needles synthesis. Synthetic *needles* are generated by fake entities, time, and events (left). Real *needles* are generated from the TKG (right). Open-domain QA *needles* are generated from a private open-domain QA resource (bottom).

whose answer items strictly adhere to a logical order. Considering that directly inserting the answer items into the long context might seem abrupt (making it difficult to establish a direct connection between the question and the needles), we also use GPT-4o to rewrite the answer items to generate more naturally phrased needles. This ensures that when needles are inserted into the long context, they can still make connections to the question, maintaining the rationality of the task.

3.2 Long Context Source

To enhance the authenticity of the task, we use LongData-Corpus [40], a real long text corpus, to construct the **long context source**. The corpus contains more than 100k pieces of Chinese and English long texts with lengths exceeding 8k characters, with the longest text exceeding 256k characters. The text content covers a wide range of materials such as academic papers, novels, legal documents, news, patents, government work reports, etc. This provides ample long

QA source					Long context source				
Pipeline	English	Chinese	Total	Proportion	Token range	English	Chinese	Total	Proportion
QA (synthetic)	3,000	3,000	6,000	42.86%	8k-16k	1,000	750	1,750	12.50%
QA (real)	3,003	2,011	5,014	35.81%	16k-32k	2,000	1,750	3,750	26.79%
QA (Open-domain)	1,497	1,489	2,986	21.33%	32k-64k	2,000	1,750	3,750	26.79%
Total	7,500	6,500	14,000		64k-128k	2,500	2,250	4,750	33.93%
Proportion	46.43%	53.57%			Total	7,500	6,500	14,000	

Table 1: Information of QA source from three sequential needles synthetic pipelines (left) and long context source extracted from LongData-Corpus (right).

Algorithm 1: Sequential-NIAH Constructing

Data: long text C, question Q, answer A, needles $N = [n_1, n_2, \ldots, n_k]$.
Result: Input C_{in} and output C_{out}.
$[C_1, C_2, \ldots, C_{k+1}] \leftarrow \text{Segment}(C, k+1)$;
$N \leftarrow \text{Shuffle}(N)$;
Long text with needles: $\hat{C} \leftarrow C_1$;
for $i \leftarrow 1$ **to** k **do**
  $\hat{C} \leftarrow \hat{C} + n_i + C_{i+1}$;
end
$C_{in} \leftarrow \text{Prompt}(\hat{C}, Q)$;
$C_{out} \leftarrow A$;
return C_{in} and C_{out};

context data for the construction of the benchmark. When preparing the long context source, to keep the language and quantity of long texts consistent with the QA source, we randomly sample long texts within different token length ranges for each language (Chinese and English), as shown in Table 1. This forms a long context source that covers a wide enough range of topics, has a reasonable distribution of article lengths, and can match each QA pair in the QA source one-to-one.

3.3 Sequential-NIAH Sample Constructing

For a given long text and a QA pair, a specific Sequential-NIAH sample is constructed by randomly shuffling the order of needles in the answer of QA and inserting them into random positions within the long text. Subsequently, the sample is formatted into input (inserted long text and question) and output (reference answer) forms using a designed prompt template, as shown in Figure 1. The detailed procedure is described in Algorithm 1. At first, a long raw text C and a QA pair are provided, where answer A is the ground truth (GT) answer organized and refined by needles. The C is segmented into $k+1$ subtexts at random punctuation marks (the period (.) in English and the full stop (。) in Chinese), where k is the number of needles, i.e., the number of answer items. And then, k shuffled needles are insert into the k positions among $k+1$ subtexts to obtain a long text with needles $\hat{C}$. Finally, combine the document content $\hat{C}$ and the question Q into the input text using a pre-designed prompt template, and use the original answer A to the question as the output to obtain a Sequential-NIAH sample. Ultimately, the dataset is partitioned into three subsets: training set, development set, and test set. Each subset contains data from diverse languages, varying text lengths, and distinct QA synthesis pipelines, embodying a Sequential-NIAH benchmark, as shown in Table 2.

	English	Chinese	Total
Train	5,400	4,600	10,000
Development	1,015	985	2,000
Test	1085	915	2,000
Total	7,500	6,500	14,000

Table 2: Dataset information of Sequential-NIAH Benchmark

4 Evaluation Model

Due to the complexity of benchmark evaluation, we hope to automate the evaluation of this task by training an evaluation model f_θ. For each question Q_i, the ground truth answer A_i and an corresponding answer B_i to be evaluated are provided to constitute an input sample $X_i = T_{eval}(Q_i, A_i, B_i)$, where $T_{eval}(\cdot)$ is a specially designed prompt template for answer evaluation. For each X_i, the label $Y_i = T_{res}(y_i, R_i)$ is constructed by the result ($y_i \in$ {'wrong', 'correct'}) and the reason R_i, where $T_{res}(\cdot)$ is the prompt template for result analysis. To train the evaluation model, our objective is to learn the conditional probability distribution between X_i and Y_i, i.e., $P(Y_i|X_i;\theta)$. And the loss function can be defined as:

$$\mathcal{L}(\theta) = \arg\max_{\theta} \sum_{i=1}^{N} \log P(Y_i|X_i;\theta) \quad (1)$$

	English	Chinese	Total
Train	3,000	3,000	6,000
Test	984	967	1,960
Total	3,984	3,967	7,960

Table 3: Dataset information for evaluation model training and test.

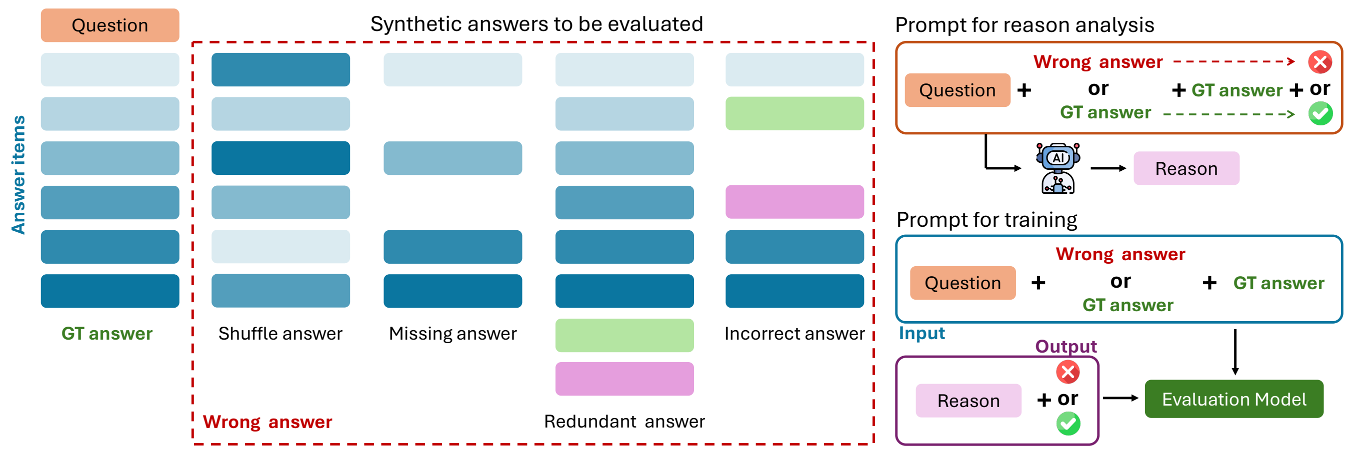


Figure 3: Synthetic wrong answers for generating various training data, reason analysis by GPT-4o with prompt, and input and output pair for evaluation model training.

To obtain the training data, four types of potential wrong answers are synthesized by changing GT answer items, including shuffled answer items (GT answer items with shuffled order), missing answer items (GT answer items with random missing), redundant answer items (GT answer items and random redundant items), and incorrect answer items (random missing items and random redundant items coexist). For the last three groups of answers, we uniformly consider them as wrong answers. For the first group of answer with only shuffled items, if the question does not require the answer items to be output in specific order, it will be treated as correct answers; otherwise, it will be treated as an wrong answers. Moreover, for chronological and logical sequential needles in answer items, we have designed two distinct evaluation prompt templates, respectively, to prevent the evaluation model from conflating on evaluating the two types of sequential answer items.

Figure 3 provides an example of how to generate training data for evaluation model training. At first, four kinds of wrong answer are generated from the ground truth (GT) answer by randomly shuffling, removing, adding, and modifying the answer items. For each question Q_i we can get the answer pair by combining the GT answer A_i and the answer B_i to be evaluated. Then, the question Q_i, the answer pair A_i and B_i, and the known result y_i are organized into the reason analysis prompt template to get the reason analyzed by an LLM (GPT-4o). Finally, the obtained reason R_i and the known result y_i can be combined as the output Y_i, and the question Q_i, the answer pair A_i and B_i will be organized as the input X_i for evaluation model training.

As shown in Table 3, a total of 6,000 samples are constructed to train the evaluation model, and 1,960 samples are used to evaluate its performance. The data used to train and test the evaluation model are randomly sampled from the QA source. We used Qwen2.5-Instruct-32B[36] as the foundation for our evaluation model and performed full-parameter SFT training on it. We utilized the AdamW optimizer, setting the learning rate to 8×10^{-6} with 4 epoch. We set the warm-up ratio to 0.1 and the weight decay to 0.1.

QA types	Answer groups	No.	Claude-3.5 (%)	GPT-4o (%)	Ours (%)
Chronological order	GT	241	85.06	90.87	**95.85**
	Missing	235	93.62	97.45	**100**
	Redundant	265	77.74	85.28	**100**
	Incorrect	229	95.20	97.38	**100**
Logical order	GT	229	86.03	99.13	**100**
	Missing	249	93.17	**100**	**100**
	Redundant	266	85.34	99.25	**100**
	Incorrect	246	82.11	**100**	**100**
	Total/Avg.	1960	87.09	96.07	**99.49**

Table 4: Evaluation model performance on synthetic test data with various QA types and answer groups.

5 Experiments & Results

5.1 Evaluation Model Performance

To demonstrate the need to train the evaluation model, we compared the performance of Claude-3.5, GPT-4o, and the evaluation model on 1,960 test samples using the same prompt templates (T_{eval} and T_{res}). The experimental results are shown in Table 4, and the results are divided into two groups (QA with needles of chronological and logical order) for analysis.

According to analysis, the evaluation model we trained achieved a total accuracy rate of **99.49%** in two groups of test data, which is 3.42% higher than GPT-4o (96.07%) and 12.4% higher than Claude-3.5 (87.09%). In 7 groups of answer evaluation, our model achieved an accuracy rate of 100%. We have checked that the only 0.5% misjudgment of our evaluation model came from the model's slight confusion about whether the question requires listing answer items in chronological order (only occurring in answer generated by shuffle GT answer items).

5.2 Benchmark Results of Current LLMs

To evaluate the performance of different models on this benchmark, we conducted inference on 2,000 test samples using four closed-source models, including **Claude-3.5** (Claude-3.5-sonnet-20241022), **GPT-4o** (GPT-4o-20240806), **GPT-4o-mini**, **Gemini-1.5** (Gemini-1.5-pro), and two open-source models, including **Qwen-2.5** (Qwen-2.5-72B-Instruct), and **LLaMA-3.3** (LLaMA-3.3-70B-Instruct). We have conducted a validity check on the results of all models on the test set to prevent situations such as token lengths exceeding the model's acceptable range or refusal to answer due to sensitive words, ensuring that all six models can provide valid answers for evaluation across all data in the test set.

Overview of results: Figure 4 illustrates the overall performance of different LLMs on this benchmark. It can be seen that Gemini-1.5 exhibits the best performance, achieving an accuracy of 63.15%. Qwen-2.5 follows closely behind with an accuracy of 51.10%, while LLaMA-3.3 and Claude-3.5 demonstrate comparable levels of performance. In contrast, GPT-4o-mini and GPT-4o perform poorly on this task, with GPT-4o being the less performance of the two.

Results depend on lengths: Different text lengths may affect the difficulty of this task. Figure a shows the performance of various LLMs in different text length groups. It is evident that the accuracy of most LLMs decreases as the text length increases. However, Gemini-1.5 and Qwen-2.5 maintain better and more stable accuracy. The accuracy rates of LLaMA-3.3 and GPT-4o decrease significantly with increasing text length, where GPT-4o exhibits a more pronounced trend. Claude-3.5 and GPT-4o-mini demonstrate relatively stable performance across different text lengths, although not as outstanding. Overall, Gemini-1.5 still exhibits superior performance.

Results depend on the number of needles: The number of needles inserted in each long text varies, and the more needles will significantly increase the difficulty of the task. Figure b shows the performance of various LLMs in different number of needles groups. It is evident that the accuracy of all LLMs decreases as the number of needles increases. Similarly, Gemini-1.5 maintain better and more stable accuracy. The performance of Qwen2.5 has also shown the most significant degradation, indicating that the increase in the number of needles seriously affected the model's performance on this task. The accuracy of Claude-3.5 and LLaMA-3.3 decrease similarly with increasing number of needles.

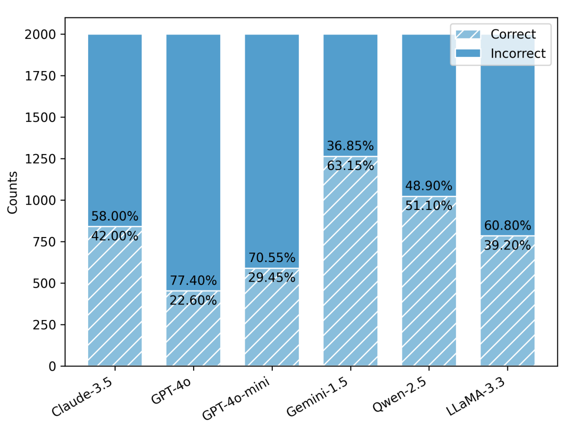


Figure 4: Results comparison of LLMs on all test data of Sequential-NIAH benchmark.

Results depend on QA synthesis pipelines: Figure c presents the performance of various LLMs under different QA synthesis pipelines. It can be observed that most LLMs perform better on test data composed of open-domain QA needles. This improvement may be attributed to the fact that the questions in this group include some general queries, allowing the models to provide answers that are relatively close to the GT answers based on their inherent capabilities, rather than having to retrieve relevant information from long texts. However, the models do not perform well on test data from synthetic needles and real needles, both of which require chronological order, indicating that retrieving and listing information in chronological order from long texts is more challenging. Among them, Gemini-1.5 performs more consistently across both types, while Qwen-2.5 and Claude-3.5 excel in open-domain QA needles. Overall, the synthetic data presents the most challenging task difficulty.

Results depend on languages: Figure d depicts the performance of multiple LLMs in different language groups. It can be seen that the performance of the same model on test samples in different languages is generally consistent, indicating that language is not a key factor affecting the difficulty of the task.

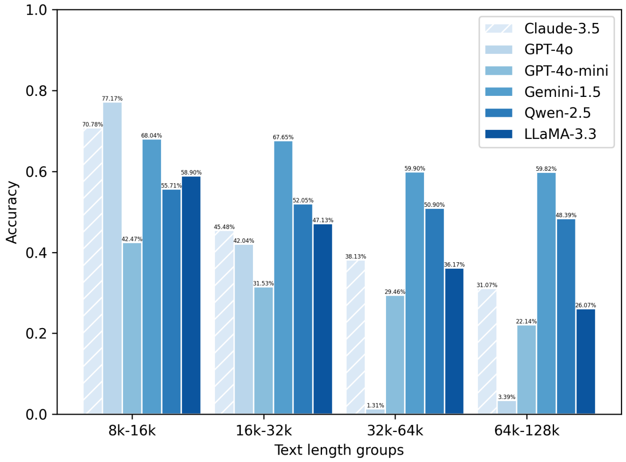
5.3 Noise Analysis for Benchmark Results

In our investigation of this benchmark's characteristics, we meticulously selected 200 samples from its test set for conducting a noise analysis. Noise analysis in this context involves introducing perturbations to the position or order of *needles* inserted into a long context. This process is aimed at evaluating the stability of various Large Language Models (LLMs) when confronted with such changes. Specifically, we introduced three distinct types of noise to each sample:

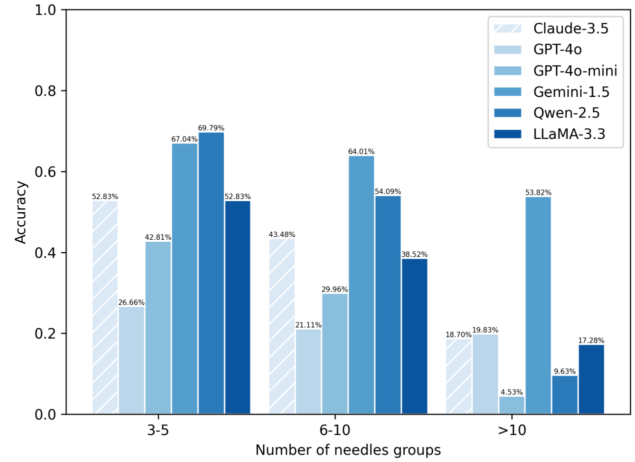
- **Tiny movement (TM)**: Each needle within the long text undergoes a slight positional shift, either forward or backward, by no more than two sentence positions. The order of the needles within the long text remains unchanged
- **Significant movement (SM)**: Multiple needles within the long text are repositioned significantly, with their original sequence maintained. This simulates larger displacements,

Model	Metrics	Ref. (%)	TM	SM	RO	All
Gemini-1.5	Acc. (%)	62.50	**65.00**	**63.13**	**64.12**	**64.40**
	Cons. (%)	-	56.50	52.00	47.50	30.50
Qwen-2.5	Acc. (%)	51.50	48.50	48.25	48.00	47.60
	Cons. (%)	-	**67.50**	**65.50**	**64.50**	**46.50**
LLaMA-3.3	Acc. (%)	38.00	38.00	36.50	42.00	39.00
	Cons. (%)	-	63.50	55.50	55.50	35.50

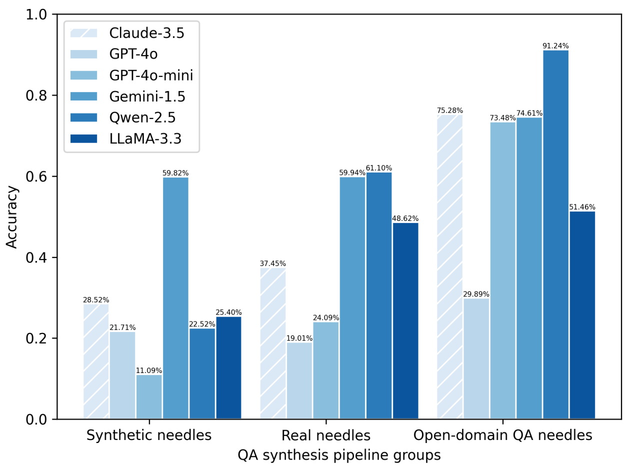
Table 5: Average accuracy and result consistency of LLMs with different position and order noise assigned on needles.



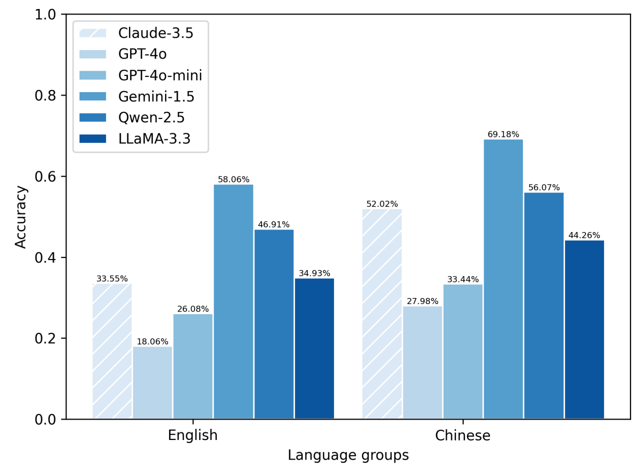
(a) Results across under length groups.



(b) Results across the number of needles groups.



(c) Results across QA synthesis pipeline groups.



(d) Results across language groups.

Figure 5: Benchmark results of well-known LLMs on test data.

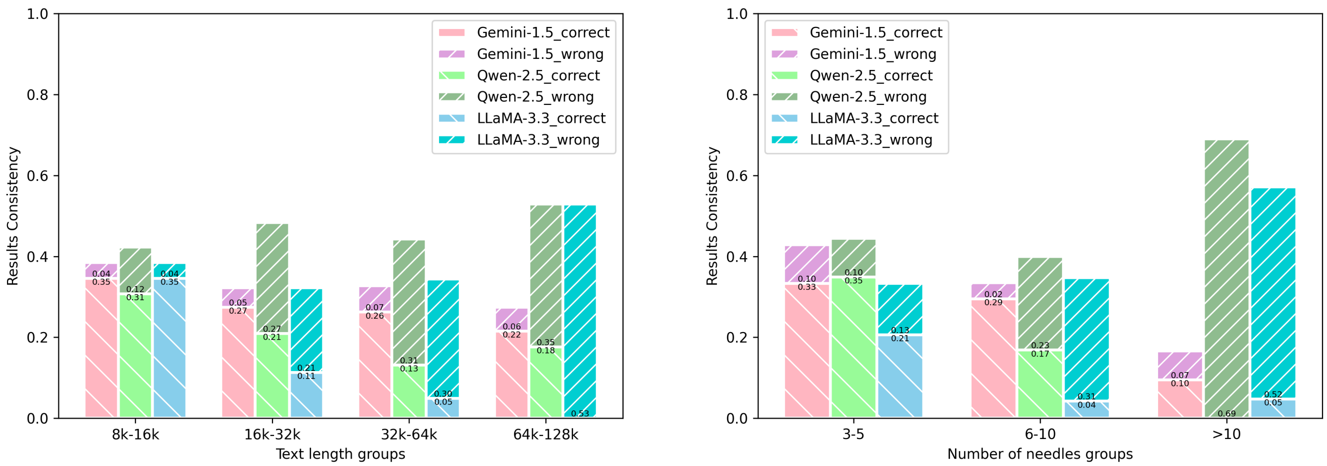
enabling analysis of the model's robustness to more pronounced positional alterations.

- **Reorder (RO)**: The positions of multiple needles remain static, but their sequence of appearance within the text is shuffled.

For each type of noise, we generated three variations of the original data, culminating in a total of 1800 noisy data samples used for inference and experimentation. In this section, three specific models, Gemini-1.5, Qwen-2.5, and LLaMA-3.3, were subjected to the noise analysis experiments to discern their resilience and performance under these controlled perturbations.

Two metrics were employed for the noise analysis experiment: average accuracy (**Acc.**) and result consistency (**Cons.**). **Acc.** represents the mean accuracy across the original 200 test samples and additional test samples with introduced noise. For the TM noise group as an example, average accuracy is computed as the mean accuracy from four sets: the original set and three noise-altered sets. **Cons.** assesses the stability of the model's responses by comparing the correctness (correct or incorrect) of answers from the original 200 test samples with those from the noise-altered sets. The consistency value, expressed as a percentage, indicates the proportion of samples where the correctness of the answers remains unchanged from the original set. A higher consistency percentage signifies greater model stability, indicating that the model's response is less affected by the introduced noise.

Accuracy analysis: As depicted in Table 5, Gemini-1.5 maintains the highest accuracy among the models tested. In the table, the reference accuracy (**Ref.**) represents the original accuracy achieved by the 200 samples drawn from the benchmark test set. Notably, there are slight variations in the accuracies of the noise groups compared to the reference accuracy, suggesting that noise has influenced the accuracy of certain samples. Nevertheless, the overall impact remains relatively minor. For Gemini-1.5, the average accuracy across noise groups shows an improvement over the reference



(a) Results across length groups.

(b) Results across the number of needles groups.

Figure 6: Noise analysis results of LLMs on noise test data across different groups.

accuracy. In contrast, Qwen-2.5's average accuracy generally decreases compared to the reference accuracy. For LLaMA-3.3, the average accuracy of the noise groups fluctuates, with both increases and decreases relative to the reference accuracy. These observations indicate that the benchmark test set is reliable for evaluating the models' capabilities in this task, as the experimental results are not significantly affected by the perturbations introduced.

Result Consistent analysis: Table 5 also illustrates the consistency of results (answer correctness) produced by various LLMs when subjected to noise. Qwen-2.5 demonstrates the highest result consistency, reaching up to 67.50% within individual noise groups. Although the introduction of each noise group generally diminishes result consistency, Qwen-2.5's consistency drops to 46.50% when all nine noise groups are considered—a figure that remains notably higher than those of Gemini-1.5 and LLaMA-3.3. The discrepancy between Gemini-1.5's higher accuracy but lower result consistency may be attributed to its tendency to produce correct answers with minor adjustments to the needles in long texts. However, this also suggests that Gemini-1.5's responses to the same Sequential-NIAH sample are less stable, with its information retrieval more affected by text content variations. On the other hand, Qwen-2.5's higher result consistency but lower accuracy could imply that, for most incorrectly answered samples, altering the needles does not enable Qwen-2.5 to provide correct answers, thereby maintaining a consistent level of incorrect responses. This highlights Qwen-2.5's limitations in this task, as it struggles to enhance its accuracy despite adjustments to the needles. Additionally, we observed that SM noise tends to more significantly disrupt model result consistency compared to TM noise, indicating that substantial shifts in needle positions are more likely to alter the correctness of model outputs. Similarly, RO noise more readily affects result consistency than TM noise, suggesting that merely changing the order of needles in long texts can lead to variations in answer correctness.

Result Consistent analysis depend on groups: Figure 6 presents the result consistency metric of the LLMs' responses across all noise test groups, organized by different text lengths and numbers of needles. The data shows that variations in result consistency across different text lengths are minimal, suggesting that the complexity of test samples constructed by this benchmark is largely uniform across various text lengths. However, as text length increases, there is a rise in the proportion of consistently incorrect answers. This trend indicates that the task becomes more challenging with longer texts, making it more difficult to enhance model accuracy by adjusting the needles. Similarly, when the number of needles is 10 or fewer, the variation in result consistency remains small. However, when the number of needles surpasses 10, there is a marked increase in result consistency. This rise is primarily due to the higher task difficulty associated with a larger number of needles, leading to a corresponding increase in consistently incorrect results, which aligns with expectations.

In summary, the test set provided by this benchmark demonstrates robust stability in evaluating model performance, as both minor and significant changes in the data have minimal impact on the accuracy metrics of LLMs. Additionally, the noise test set effectively assesses the consistency of output results from different LLMs for the same sample, thereby reflecting the stability of LLMs in this task.

6 Conclusion

We present the Sequential-NIAH benchmark, a comprehensive dataset designed to evaluate large language models (LLMs) in the context of sequential information extraction from long texts, encompassing up to 128K tokens. This benchmark includes synthetic, real, and open-domain question-answering pipelines, with a dataset distribution of 10,000 training samples, 2,000 development samples, and 2,000 test samples. Additionally, we have released an evaluation model to facilitate efficient and precise assessment metrics for our benchmarks.

In our experiments, popular LLMs such as Claude, GPT-4.0, Gemini, LLaMA, and Qwen exhibited suboptimal performance, underscoring the complexity of the benchmark and highlighting the necessity for advancements in current models. The noise analysis further validates the stability of this benchmark, signifying a meaningful contribution to the NLP community.

7 Limitations and Ethics

Model evaluations may be biased by the dataset's domain, and unoptimized API parameters could affect performance and fairness. Addressing these is crucial for accurate assessments.

The dataset is for academic and research use only; commercial or misuse is prohibited to maintain integrity and ethical standards.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2024. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2024). arXiv:2303.08774 [cs.CL] https://arxiv.org/abs/2303.08774

[2] Chenxin An, Shansan Gong, Ming Zhong, Xingjian Zhao, Mukai Li, Jun Zhang, Lingpeng Kong, and Xipeng Qiu. 2023. L-Eval: Instituting Standardized Evaluation for Long Context Language Models. *arXiv preprint arXiv:2307.11088* (2023). arXiv:2307.11088 [cs.CL] https://arxiv.org/abs/2307.11088

[3] Anthropic. 2024. Introducing the next generation of claude 3.5. https://www.anthropic.com/news/claude-3-5-sonnet.

[4] Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding. *arXiv preprint arXiv:2308.14508* (2024). arXiv:2308.14508 [cs.CL] https://arxiv.org/abs/2308.14508

[5] Aydar Bulatov, Yuri Kuratov, Yermek Kapushev, and Mikhail S. Burtsev. 2024. Scaling Transformer to 1M tokens and beyond with RMT. arXiv:2304.11062 [cs.CL] https://arxiv.org/abs/2304.11062

[6] Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023. Extending Context Window of Large Language Models via Positional Interpolation. doi:10.48550/arXiv.2306.15595 arXiv:2306.15595 [cs]

[7] Daniel Y Fu, Elliot L Epstein, Eric Nguyen, Armin W Thomas, Michael Zhang, Tri Dao, Atri Rudra, and Christopher Ré. 2023. Simple hardware-efficient long convolutions for sequence modeling. In *International Conference on Machine Learning*. PMLR, 10373–10391.

[8] Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. GPTScore: Evaluate as You Desire. arXiv:2302.04166 [cs.CL] https://arxiv.org/abs/2302.04166

[9] Mingqi Gao, Xinyu Hu, Jie Ruan, Xiao Pu, and Xiaojun Wan. 2024. LLM-based NLG Evaluation: Current Status and Challenges. arXiv:2402.01383 [cs.CL] https://arxiv.org/abs/2402.01383

[10] Mingqi Gao, Jie Ruan, Renliang Sun, Xunjian Yin, Shiping Yang, and Xiaojun Wan. 2023. Human-like Summarization Evaluation with ChatGPT. arXiv:2304.02554 [cs.CL] https://arxiv.org/abs/2304.02554

[11] Alberto García-Durán, Sebastijan Dumančić, and Mathias Niepert. 2018. Learning sequence encoders for temporal knowledge graph completion. *arXiv preprint arXiv:1809.03202* (2018). arXiv:1809.03202 [cs.AI] https://arxiv.org/abs/1809.03202

[12] Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530* (2024). arXiv:2403.05530 [cs.CL] https://arxiv.org/abs/2403.05530

[13] gkamradt. 2023. Needle In A Haystack - Pressure Testing LLMs. https://github.com/gkamradt/LLMTest_NeedleInAHaystack

[14] Albert Gu and Tri Dao. 2024. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. arXiv:2312.00752 [cs.LG] https://arxiv.org/abs/2312.00752

[15] Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. RULER: What's the Real Context Size of Your Long-Context Language Models? *arXiv preprint arXiv:2404.06654* (2024). arXiv:2404.06654 [cs.CL] https://arxiv.org/abs/2404.06654

[16] Qi Jia, Siyu Ren, Yizhu Liu, and Kenny Q. Zhu. 2023. Zero-shot Faithfulness Evaluation for Text Summarization with Foundation Language Model. arXiv:2310.11648 [cs.CL] https://arxiv.org/abs/2310.11648

[17] Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. 2020. Recurrent Event Network: Autoregressive Structure Inference over Temporal Knowledge Graphs. *arXiv preprint arXiv:1904.05530* (2020). arXiv:1904.05530 [cs.LG] https://arxiv.org/abs/1904.05530

[18] Pei Ke, Bosi Wen, Zhuoer Feng, Xiao Liu, Xuanyu Lei, Jiale Cheng, Shengyuan Wang, Aohan Zeng, Yuxiao Dong, Hongning Wang, Jie Tang, and Minlie Huang. 2024. CritiqueLLM: Towards an Informative Critique Generation Model for Evaluation of Large Language Model Generation. arXiv:2311.18702 [cs.CL] https://arxiv.org/abs/2311.18702

[19] Kalpesh Krishna, Erin Bransom, Bailey Kuehl, Mohit Iyyer, Pradeep Dasigi, Arman Cohan, and Kyle Lo. 2023. LongEval: Guidelines for Human Evaluation of Faithfulness in Long-form Summarization. *arXiv preprint arXiv:2301.13298* (2023). arXiv:2301.13298 [cs.CL] https://arxiv.org/abs/2301.13298

[20] Jiaqi Li, Mengmeng Wang, Zilong Zheng, and Muhan Zhang. 2024. LooGLE: Can Long-Context Language Models Understand Long Contexts? arXiv:2311.04939 [cs.CL] https://arxiv.org/abs/2311.04939

[21] Mo Li, Songyang Zhang, Yunxin Liu, and Kai Chen. 2024. NeedleBench: Can LLMs Do Retrieval and Reasoning in 1 Million Context Window? *arXiv preprint arXiv:2407.11963* (2024). arXiv:2407.11963 [cs.CL] https://arxiv.org/abs/2407.11963

[22] Qintong Li, Leyang Cui, Lingpeng Kong, and Wei Bi. 2023. Collaborative Evaluation: Exploring the Synergy of Large Language Models and Humans for Open-ended Generation Evaluation. arXiv:2310.19740 [cs.CL] https://arxiv.org/abs/2310.19740

[23] Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, et al. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434* (2024). arXiv:2405.04434 [cs.CL] https://arxiv.org/abs/2405.04434

[24] Jiaheng Liu, Zhiqi Bai, Yuanxing Zhang, Chenchen Zhang, Yu Zhang, Ge Zhang, Jiakai Wang, Haoran Que, Yukang Chen, Wenbo Su, Tiezheng Ge, Jie Fu, Wenhu Chen, and Bo Zheng. 2024. E^2LLM: Efficient and Extreme Length Extension of Large Language Models. arXiv:2401.06951 [cs.CL] https://arxiv.org/abs/2401.06951

[25] liuhuanyong. 2022. Finance Event Graph. https://github.com/liuhuanyong/FinanceEventGraph

[26] Adian Liusie, Potsawee Manakul, and Mark J. F. Gales. 2024. LLM Comparative Assessment: Zero-shot NLG Evaluation through Pairwise Comparisons using Large Language Models. arXiv:2307.07889 [cs.CL] https://arxiv.org/abs/2307.07889

[27] Z Luo, Q Xie, and S Ananiadou. 2023. Chatgpt as a Factual Inconsistency Evaluator for Abstractive Text Summarization (2023). *arXiv preprint arXiv:2303.15621* (2023).

[28] Moonshot. [n. d.]. Kimi. https://www.moonshot.cn.

[29] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 27730–27744. https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf

[30] Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, Kranthi Kiran GV, Xuzheng He, Haowen Hou, Jiaju Lin, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartlomiej Koptyra, Hayden Lau, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Bolun Wang, Johan S. Wind, Stanislaw Wozniak, Ruichong Zhang, Zhenyuan Zhang, Qihang Zhao, Peng Zhou, Qinghua Zhou, Jian Zhu, and Rui-Jie Zhu. 2023. RWKV: Reinventing RNNs for the Transformer Era. arXiv:2305.13048 [cs.CL] https://arxiv.org/abs/2305.13048

[31] Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2023. YaRN: Efficient Context Window Extension of Large Language Models. arXiv:2309.00071 [cs.CL] https://arxiv.org/abs/2309.00071

[32] Ofir Press, Noah Smith, and Mike Lewis. 2022. Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation. In *International Conference on Learning Representations*. https://openreview.net/forum?id=R8sQPpGCv0

[33] Uri Shaham, Maor Ivgi, Avia Efrat, Jonathan Berant, and Omer Levy. 2023. ZeroSCROLLS: A Zero-Shot Benchmark for Long Text Understanding. *arXiv preprint arXiv:2305.14196* (2023). arXiv:2305.14196 [cs.CL] https://arxiv.org/abs/2305.14196

[34] Mingyang Song, Mao Zheng, and Xuan Luo. 2024. Counting-Stars: A Multi-evidence, Position-aware, and Scalable Benchmark for Evaluating Long-Context Large Language Models. *arXiv preprint arXiv:2403.11802* (2024). arXiv:2403.11802 [cs.CL] https://arxiv.org/abs/2403.11802

[35] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. RoFormer: Enhanced Transformer with Rotary Position Embedding. *Neurocomputing* 568 (Feb. 2024), 127063. doi:10.1016/j.neucom.2023.127063

[36] Qwen Team. 2024. Qwen2.5: A Party of Foundation Models. https://qwenlm.github.io/blog/qwen2.5/

[37] Qingyang Wu, Zhenzhong Lan, Kun Qian, Jing Gu, Alborz Geramifard, and Zhou Yu. 2022. Memformer: A Memory-Augmented Transformer for Sequence

Modeling. arXiv:2010.06891 [cs.CL] https://arxiv.org/abs/2010.06891

[38] Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, Madian Khabsa, Han Fang, Yashar Mehdad, Sharan Narang, Kshitiz Malik, Angela Fan, Shruti Bhosale, Sergey Edunov, Mike Lewis, Sinong Wang, and Hao Ma. 2023. Effective Long-Context Scaling of Foundation Models. arXiv:2309.16039 [cs.CL] https://arxiv.org/abs/2309.16039

[39] Wenda Xu, Danqing Wang, Liangming Pan, Zhenqiao Song, Markus Freitag, William Yang Wang, and Lei Li. 2023. INSTRUCTSCORE: Explainable Text Generation Evaluation with Finegrained Feedback. arXiv:2305.14282 [cs.CL] https://arxiv.org/abs/2305.14282

[40] yuyijiong. 2023. LongData-Corpus. https://huggingface.co/datasets/yuyijiong/LongData-Corpus. Accessed: 2023-12-20.

[41] Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, et al. 2024. ChatGLM: A Family of Large Language Models from GLM-130B to GLM-4 All Tools. *arXiv preprint arXiv:2406.12793* (2024). arXiv:2406.12793 [cs.CL] https://arxiv.org/abs/2406.12793

[42] Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. 2024. Long Context Compression with Activation Beacon. doi:10.48550/arXiv.2401.03462 arXiv:2401.03462 [cs]

[43] Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Khai Hao, Xu Han, Zhen Leng Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2024. ∞Bench: Extending Long Context Evaluation Beyond 100K Tokens. *arXiv preprint arXiv:2402.13718* (2024). arXiv:2402.13718 [cs.CL] https://arxiv.org/abs/2402.13718

[44] Yangjun Zhang, Pengjie Ren, and Maarten de Rijke. 2021. A Human-machine Collaborative Framework for Evaluating Malevolence in Dialogues. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, Online, 5612–5623. doi:10.18653/v1/2021.acl-long.436

[45] Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2023. JudgeLM: Fine-tuned Large Language Models are Scalable Judges. arXiv:2310.17631 [cs.CL] https://arxiv.org/abs/2310.17631