Towards a Holistic and Automated Evaluation Framework for Multi-Level Comprehension of LLMs in Book-Length Contexts

Yuho Lee[1,*], **Jiaqi Deng**[1,*], **Nicole Hee-Yeon Kim**[1], **Hyangsuk Min**[1], **Taewon Yun**[1], **Minjeong Ban**[1], **Kim Yul**[1], **Hwanjun Song**[1†]
[1]Korea Advanced Institute of Science and Technology
{yuholee, jiaqi.deng, songhwanjun}@kaist.ac.kr

Abstract

We introduce HAMLET, a holistic and automated framework for evaluating the long-context comprehension of large language models (LLMs). HAMLET structures key information of source texts into a three-level hierarchy at root-, branch-, and leaf-levels, and employs query-focused summarization to evaluate how well models faithfully recall the key information at each level. To validate the reliability of our fully automated pipeline, we conduct a systematic human study, demonstrating that our automatic evaluation achieves over 90% agreement with expert human judgments, while reducing the evaluation cost by up to 25×. HAMLET reveals that LLMs struggle with fine-grained comprehension, especially at the leaf level, and are sensitive to positional effects like the lost-in-the-middle. Analytical queries pose greater challenges than narrative ones, and consistent performance gaps emerge between open-source and proprietary models, as well as across model scales. Our code and dataset are publicly available at link.

1 Introduction

As LLMs are increasingly applied to long-form text understanding, recent advances now allow them to process inputs exceeding 100K tokens, enabling comprehension of book-length documents (Ding et al., 2024; Fu et al.; Jin et al., 2024). With the growing demand for accurate long-form processing, evaluating the comprehension capabilities of LLMs in book-length texts has become a critical challenge (Kryscinski et al., 2022; Zhang et al., 2024a). As with short texts, evaluating book-length comprehension needs evaluating faithfulness, coherence, and others, but poses challenges in holistic evaluation and the high cost of annotation, due to the complexity of extremely long inputs (Wu et al., 2023; Zhang et al., 2024b; Laban et al., 2024a).

Recent efforts have been made on the evaluation of the LLM's comprehension in a lengthy context, such as BOOOOKSCORE (Chang et al., 2024) and FABLES (Kim et al., 2024). However, existing works remain limited to the *coarse*-grained evaluation of the understanding of LLMs, *i.e.*, short-form generation tasks such as single-turn QA (Dong et al., 2024; An et al., 2024; Kwan et al., 2024; Wang et al., 2024), or whole-document summarization that only requires a shallow, surface-level understanding of the text (Kryscinski et al., 2022; Chang et al., 2024; Zhang et al., 2024a; Kim et al., 2024). That is, they overlook the LLM's ability to recall information across *varying* levels of detail, an aspect we refer to as *multi-level recall*[1]. This aspect is especially critical for book-length comprehension, which demands tracking both global themes and specific details and building logical relationships between different levels of information. Without it, LLMs risk generating responses that omit key information or lack coherence (Wan et al., 2024; Maharana et al., 2024).

In this paper, we propose a novel evaluation benchmark framework, HAMLET (Holistic and Automated Multi-Level Evaluation for Long Text) in Figure 1, a scalable and automated benchmark framework to evaluate the capabilities of LLMs on book-length contexts across varying levels of detail, in addition to faithfulness. To flexibly probe an LLM's comprehension, we introduce a *key-fact tree*, which is a hierarchical information structure derived from manageable chunks (*i.e.*, 4K-token segments) of long texts. Specifically, each tree captures multi-level content abstraction, structuring key-facts into a *root-branch-leaf* hierarchy of themes, supporting ideas, and fine-grained details (see Section 3.1.2). The key-fact tree enables the formulation of detail-aware queries to evaluate an

[*] Equal Contribution; the order is assigned randomly.
[†] Corresponding Author.

[1]Multi-level recall refers to an LLM's ability to retain and reproduce information at different levels of detail, from high-level summaries to fine-grained facts, in long-form contexts.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 24401–24425
November 4-9, 2025

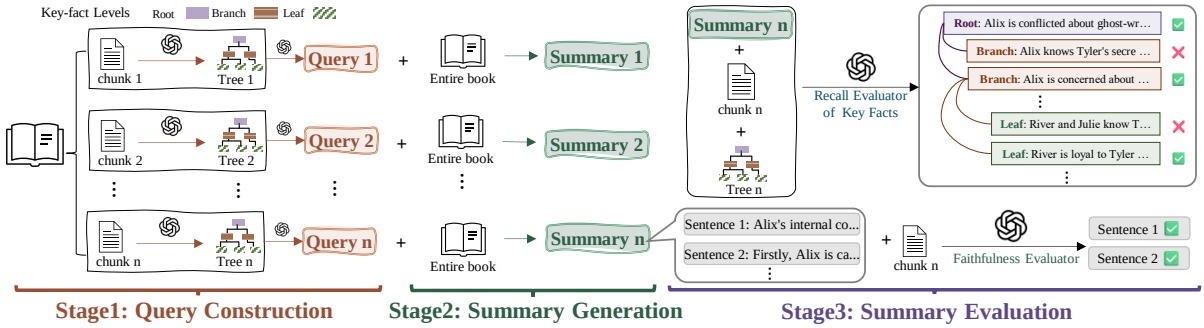


Figure 1: Overview of HAMLET: HAMLET constructs queries from a key-fact hierarchy (root, branch, leaf) per chunk, generates summaries using the full book, and evaluates recall and faithfulness using LLM-based evaluators.

LLM's ability to extract information across varying levels of abstraction, categorized into two perspectives: *analytical*, which focuses on deeper meaning and thematic interpretation, and *narrative*, which emphasizes story progression and key events.

Based on these queries, we adopt *query-focused* summarization (Xu and Lapata, 2020; Liu et al., 2024b) as the core task, where the LLM receives the entire book as input and generates a summary in response to each query. This setup aligns with the open-ended nature of long-form generation and facilitates fine-grained evaluation of LLMs' recall and factuality across different levels of abstraction. Particularly, by anchoring each query to a specific chunk at a distinct location in the book, HAMLET reveals positional challenges faced by LLMs, including insights into how the lost-in-the-middle effect (Liu et al., 2024a) manifests differently across levels of information abstraction.

To enable full automation of HAMLET, we conduct a systematic human study to assess the reliability of LLM-based evaluation across its three key components that typically require human verification (see Section 4): "key-fact tree construction," ensuring faithfulness, objectivity, and significance of the key-fact tree; "query validation," confirming that each query is a self-contained question grounded by the matched key-fact tree; and "summary evaluation," assessing summary quality via key-fact alignment and fact-verification to evaluate LLMs' recall and faithfulness. Our automated pipeline outperforms crowd workers, achieves over 90% agreement with expert annotations, and operates at up to 25× lower cost than the off-the-shelf evaluator FINESURE (Song et al., 2024), showing its reliability and efficiency.

This is notable, as assessing fine-grained comprehension across varying levels of abstraction in book-length contexts is challenging even for humans. Our automation enables scalable benchmarking and allows for effortless extension to new domains with minimal human intervention.

Our main contributions are: (1) we introduce the key-fact tree, a hierarchical abstraction of long texts that enables detail-aware evaluation of LLMs' comprehension; (2) we adopt query-focused summarization, a natural fit for open-ended evaluation, to probe LLMs' performance on both recall and faithfulness; (3) we present an automated evaluation pipeline, achieving over 90% agreement with expert annotations; and (4) we benchmark long-context comprehension across root-branch-leaf abstraction levels, comparing eight LLMs along three axes: open-source vs. proprietary; small vs. large; and non-reasoning vs. reasoning models.

2 Related Work

Long-form Comprehension Benchmarks. Existing LLM benchmarks for long-form comprehension have largely focused on assessing information retrieval through simple QA tasks (Wang et al., 2024; Karpinska et al., 2024; Hsieh et al., 2024), offering a limited view of an LLM's deeper understanding. Beyond this, several recent benchmarks have tuned to long-form generation tasks, generating a single, holistic summary of the entire book (Xu et al., 2023; Kwan et al., 2024; Zhang et al., 2024a; An et al., 2024; Kim et al., 2024; Laban et al., 2024a; Fan et al., 2025). For instance, BOOOOKSCORE (Chang et al., 2024) evaluates the coherence and readability of holistic summaries, while FABLES focuses on faithfulness by identifying unfaithful statements. However, both overlook the diverse abstraction levels in long texts, offering only shallow evaluations and failing to assess LLMs' recall, with an emphasis limited to coherence and factuality. In contrast, SUMMHAY (Laban et al., 2024b) adopts query-focused summarization, but its evaluation is limited to information retrieval performance, without assessing LLMs' recall or

Benchmark	Document Authenticity	Evaluation Dimensions	Evaluated Abstraction Level (Task)	Requires Human Annotation
BooookScore (Chang et al., 2024)	✓ Real	Coherence	High-level (Summary)	✗ Required
NoCha (Karpinska et al., 2024)	✓ Real	Faithfulness	Low-level (Claim Discrimination)	✗ Required
NovelQA (Wang et al., 2024)	✓ Real	Faithfulness	High-level (Simple QA)	✗ Required
FABLES (Kim et al., 2024)	✓ Real	Faithfulness	High-level (Summary)	✗ Required
∞Bench (Zhang et al., 2024a)	✓ Real	Faithfulness	High-level (Simple QA & Summary)	✓ Not Required
SummHay (Laban et al., 2024b)	✗ Synthetic	Faithfulness, Attribution	Low-level (Query-focused Summ.)	✗ Required
MedOdyssey (Fan et al., 2025)	✓ Real	Recall, Faithfulness	Low-level (Multi-choice QA)	✗ Required
HAMLET (Ours)	✓ **Real**	✓ **Multi-level Recall, Faithfulness**	✓ **High→Low (Query-focus Summ.)**	✓ **Not Required**

Table 1: Comparison of HAMLET with seven recent benchmark frameworks evaluating LLMs' comprehension of book-length contexts with respect to four key criteria: (Document Authenticity) use of real vs. synthetic documents; (Evaluation Dimension) targeted aspects of long-form understanding; (Abstraction Level) level of information abstraction assessed; and (Requires Human Annotation) reliance on human annotations.

faithfulness. Its fully synthetic data further limits the ability to assess LLMs' real capabilities.

Automated Evaluation. A critical challenge in human-based benchmarking is the high cost of human evaluation (Kim et al., 2024; Lee et al., 2024), making the benchmarking pipeline impractical for scalable and repeated use. To address this, many studies explore using LLMs as automated judges to reduce reliance on costly human annotators. Specifically for summary evaluation, traditional rule-based approaches (Lin, 2004; Zhang et al., 2019) and learning-based metrics (Zhong et al., 2022; Achiam et al., 2023) have shown low agreement with human judgments and are often confined to specific evaluation dimensions, such as faithfulness or completeness. In contrast, LLM-based metrics have demonstrated much stronger alignment with human assessments and can be easily extended to other dimensions through prompt tuning (Liu et al., 2023; Wang et al., 2023; Tang et al., 2024a; van Schaik and Pugh, 2024; Fu et al., 2024). Among these, FINESURE (Song et al., 2024) enables fine-grained, multi-dimensional evaluation with interpretable scores, such as factual error rates and missing content proportions.

Yet, automatic evaluation of book-length inputs remains challenging, as it requires tracking and understanding complex narratives over long contexts. Consequently, recent works, including FABLES (Kim et al., 2024) and NOVELQA (Wang et al., 2024), rely mostly on human annotators who have read the book; however, their memory-based judgments are limited to surface-level verification, making fine-grained evaluation infeasible.

3 HAMLET Framework

We first collect 16 novels,[2] which are recently published books with an average length of 101K tokens. These narrative-rich documents provide a challenging yet realistic testbed for evaluating long-form comprehension. Based on this, we construct our benchmark framework in three stages: (Stage 1) query construction, (Stage 2) summary generation, and (Stage 3) automated summary evaluation. In Section 4, we validate the reliability of automated components through expert human evaluation.

In contrast to recent benchmarks, HAMLET is the first automated framework to evaluate LLMs' long-context comprehension on multi-level recall and faithfulness, as summarized in Table 1.

3.1 Query Construction (Stage 1)

To evaluate the comprehension of long context for LLMs, we assess their response accuracy to queries grounded in different parts of a long document, focusing on recall and faithfulness. We construct such queries through three steps: text chunking, key-fact tree construction, and query formulation. These steps ensure that each query is anchored to specific locations within the long text and to different levels of information abstraction.

3.1.1 Text Chunking

Given the extreme length of book-scale inputs, generating targeted queries from the full text is impractical for both humans and LLMs. We therefore segment each book into sequential, non-overlapping 4K-token chunks, which serve as localized anchors for key-fact tree construction and query generation. A detailed justification and additional experiments on chunk size are provided in Appendix B.

3.1.2 Key-fact Tree Construction

Next, we extract key information from each chunk at varying levels of detail. Specifically, we organize this information into a hierarchical structure

[2]The selected books were mostly published after January 2025, ensuring they were not included in the training data of the evaluated LLMs (refer to the list of books in Table 8). The book list is easily configurable, and our automated framework generalizes seamlessly to new long-form inputs.

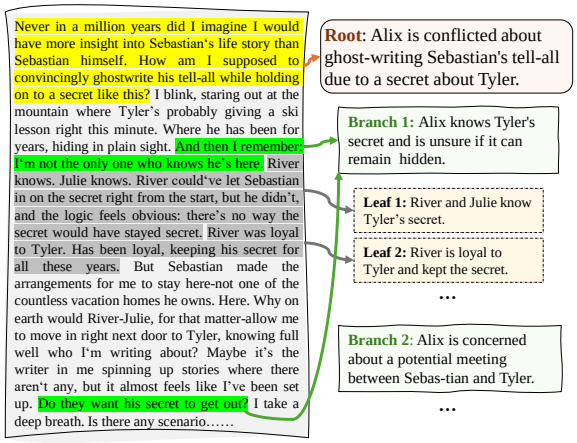


Figure 2: Example of an analytical key-fact tree extracted from a single chunk using GPT-4o.

called a *key-fact tree* (refer to Table 10 for a detailed definition of each level), which compresses all key information from a book chunk into three semantic levels, as exemplified in Figure 2. *Roots* capture the chunk's central theme; *branches* represent conceptual subtopics or supporting narrative arcs; and *leaves* enumerate specific facts or fine-grained evidence associated with each branch. This structure provides a principled way to assess LLM comprehension across abstraction levels.

Importantly, the key-fact tree is central to HAMLET: (i) it guides query generation by defining structure, detail level, and logical relations in Section 3.1.3, and (ii) serves as a reference for evaluating summaries through fine-grained, query-aligned recall and faithfulness analysis in Section 3.3.

Analytical vs. Narrative. To further structure the extracted key-fact trees, we classify them into two types: *analytical*, which emphasizes deeper meaning and logical reasoning, and *narrative*, which focuses on temporal progression and causal relationships. This distinction is essential for constructing key-fact trees, as it allows a structured representation of both reasoning and narrative elements, *i.e.*, two core aspects of long-form comprehension (Mar et al., 2021). See Table 10 for definitions and examples of the two key-fact types.

Generation and Verification. We generate the key-fact tree by prompting GPT-4o with carefully designed prompts, each tailored to either the analytical or narrative perspective, shown in Table 15 and Table 16. The output is returned in JSON format for ease of parsing and structured processing across the root–branch–leaf hierarchy.

All generated trees undergo an automated validation process, as raw key-fact trees may include hallucinations, subjective inferences, or trivial details due to reliance on a single model. Therefore, we configure GPT-4o as a specialized validator and implement a three-dimensional filter, covering: *faithfulness*, key-facts must be fully supported by the source chunk; *objectivity*, key-facts must be free from speculation or value judgments; and *significance*, key-facts must provide essential insight, not minor statements. See Tables 17–19 for the verification prompts. After the validation passes, `PASS`/`FAIL` judgments are merged. Any key-fact failing a single dimension is removed, and orphaned descendants are recursively pruned. This filtering ensures reliable key-fact trees for query formulation and summary evaluation.

We obtain 22,333 key-facts across 16 novels, spanning entire books and covering all three levels of detail. See Appendix A for statistics.

3.1.3 Query Formulation

To assess LLMs' long-context comprehension, we generate queries from previously extracted and verified key-fact trees. Each query is produced by GPT-4o, which takes a 4K-token text chunk and its validated key-fact tree as input and is prompted to generate a single, self-contained query. We use distinct prompts for the two types of key-fact trees, analytical or narrative, as shown in Tables 20 and 21. As key-fact trees span the entire book, queries are generated throughout the text, ensuring broad coverage. The formulated queries provide a robust benchmark for evaluating LLMs' comprehension of both analytical and narrative aspects in long documents.

As a result, we curate 814 queries (407 for each of the two query types) across 16 novels. The example of queries can be found in Table 10.

3.2 Summary Generation (Stage 2)

Based on the queries, we adopt *query-focused* summarization (Xu and Lapata, 2020; Liu et al., 2024b) as our task, where the LLM is given the full book (up to 114K tokens) as input and generates a summary in response to each query. This task evaluates the LLM's ability to extract all relevant information (recall) and generate factually accurate responses (faithfulness) to queries over long documents.

To perform this task, we evaluate eight high-performing LLMs, including GPT-4o (Base and Mini), Claude-3.5 (Sonnet and Haiku), Llama-3.1-Instruct (405B and 8B), and Qwen-2.5-32B (Instruct and R1-distill). They are assessed along three axes: proprietary vs. open-source, large vs. small, and general vs. reasoning-optimized. The model

specification is provided in Table 11.

We obtain 4,884 summaries from the entire 814 queries using six summarizers (excluding Qwen), and 632 summaries from a subset of 316 queries using two Qwen models, due to the high inference cost of reasoning models. All summaries are subject to automatic evaluation in our benchmark.

3.3 Summary Evaluation (Stage 3)

3.3.1 Evaluation Dimension

We benchmark LLMs' recall and faithfulness on extremely long texts through their query-based summaries. *Multi-level recall* measures how well the LLM retrieves relevant information across varying levels of detail, while *multi-level faithfulness* assesses the accuracy of that content without hallucination at each level. High scores in both indicate strong long-context comprehension, reflecting the LLM's ability to faithfully capture information from high-level concepts to fine-grained details.

Multi-level Recall. This metric quantifies how well an LLM-generated summary captures the key-facts needed to answer a query, evaluated at root, branch, and leaf levels of the reference key-fact tree. Let K_{level} denote the set of key-facts at a specific level (level $\in$ {root, branch, leaf}) from the reference tree. Then, for each level, recall is defined as:

$$\text{Recall}_{\text{level}} = |\{k | k \in S \wedge k \in K_{\text{level}}\}| / |K_{\text{level}}|, \quad (1)$$

where k is a content unit from the summary S.[3]

Multi-level Faithfulness. We first classify each content unit k in the summary S by whether it matches a key-fact in the reference tree. We then label each k as faithful or hallucinated across four levels (level $\in$ {root, branch, leaf, none}), where "none" indicates the content unit not aligned with any key-fact in the reference tree. Let S_{level} denote the set of content units in the summary assigned to a specific level. Then, faithfulness is defined as:

$$\text{Faithfulness}_{\text{level}} = |S^*_{\text{level}}| / |S_{\text{level}}|, \quad (2)$$

where $S^*_{\text{level}} \subseteq S_{\text{level}}$ is the subset consisting only of faithful content units at that level.

Note that by ignoring level distinctions and aggregating across all content units, we can easily compute an overall summary-level score.

[3]Following Song et al. (2024), we treat each summary sentence as a semantic content unit for simplicity.

3.3.2 Automatic Evaluation

Although methods like FINESURE (Song et al., 2024) have improved automated evaluation for summarization, evaluating models on inputs over 100K tokens remains an open challenge. Most long-document benchmarks still depend on human evaluation (Kim et al., 2024; Wang et al., 2024), yet even annotators who have read the book struggle to consistently assess fine-grained content.

HAMLET addresses this limitation by eliminating the need to reference the full document during evaluation. Instead, it anchors the evaluation of each query-based summary to a localized chunk and its associated key-fact tree, enabling fine-grained and scalable assessment without full-text access. This approach not only reduces the complexity and cost of human evaluation but also enhances the effectiveness of automated assessment by decomposing inputs of up to 114K tokens into manageable 4K-token chunks. To support this, we adapt FINESURE to our chunk-based pipeline for the two evaluation dimensions, recall and faithfulenss, using GPT-4o as the backbone. The detailed adaptation approach can be found in Appendix E.

We compare our method against human assessments by crowd workers and automated approaches that require full-book access, and find that HAMLET significantly outperforms them, achieving over 90% agreement with expert judgments while reducing cost by up to 25$\times$ (see Section 4).

4 Validation of HAMLET Pipeline

We evaluate the reliability of our three automated components, including (i) key-fact tree construction, (ii) query formulation, and (iii) summary evaluation, by comparing their outputs against expert human judgments. For this study, we recruit three graduate students with C2 English proficiency and NLP expertise as examiners. Expert disagreements are resolved through discussion, resulting in a consensus label. Note that the sample size used in this study ensures over 98% confidence level with ±5% margin of error. See Appendix F for details.

The three rigorous verification below demonstrate that our automated pipeline achieves high reliability close to expert-based evaluation.

Key-fact Tree Generation. Key-facts are finalized through our three-dimensional verification filter in Section 3.1.2. Thus, we assess the correctness of our filter, which assigns `Pass`/`Fail` judgments for faithfulness, objectivity, and significance to each

Judgement	Faithfulness	Objectivity	Significance	Mean
PASS	99.3%	99.1%	90.3%	96.2%
FAIL	100%	91.6%	97.5%	96.4%

Table 2: Accuracy (%) of PASS/FAIL judgments by the verification filter in key-fact tree construction.

key-fact. The three experts verify the correctness of each judgment on 1,185 key-facts randomly sampled from a total of 23,333. Table 2 reports the results of expert inspection of the automated filtering judgments. Overall, the filter exhibits the average of 96.2% and 96.4% accuracy for `PASS` and `FAIL` judgments, respectively, demonstrating the robustness of our verification process in cultivating high-fidelity key-fact trees.

Query Formulation. To evaluate the quality of the query generated in Section 3.1.3, we evaluate query quality along two dimensions: *naturalness*, referring to the query's fluency, grammaticality, and human-likeness; and *relevance*, indicating whether the answer can be derived from the corresponding 4K-token chunk. Thus, we ask the experts to evaluate whether each query is valid or not for the entire 814 queries. The results confirm that the queries are valid, with 100% for naturalness and 98.2% for relevance, respectively. This aligns with prior findings that modern AI models excel at naturalness and relevance (Liu et al., 2023). Therefore, our pipeline reliably produces valid, high-quality queries without manual intervention.

Summary Evaluation. In our framework, the key concern is the accuracy of automated summary evaluation compared to expert human assessments, which serves as a measure of its reliability. Hence, we first collect gold labels from three experts for two binary tasks: *key-fact alignment*, whether each key-fact appears in the summary, and *fact-checking*, whether each summary sentence is supported by the source document; these labels are used to compute the multi-level recall (in Eq. (1)) and faithfulness (in Eq. (2)) of LLM responses to the query, respectively. For each task, we sample 600 instances to annotate them with expert labels. Then, we compute binary accuracy (bACC) between the labels generated by the automated evaluation and the expert-annotated gold labels, as summarized in Table 3.

As an automated summary evaluation method, HAMLET differs from FINESURE (Song et al., 2024) in two key aspects: (i) it uses only a single chunk of text for faithfulness evaluation, and (ii) it employs a distinct key-fact extraction strategy (key-fact trees) to assess LLM recall on book-length

Eval. Method	Reference	Recall	Faithfulness
FineSurE (GPT-4o)	Full Text	N/A	52.9%
HAMLET (Crowd-sourced)	Chunk	86.8%	67.8%
HAMLET (GPT-4o-Mini)	Chunk	94.3%	64.3%
HAMLET (GPT-4o)	**Chunk**	**98.1%**	**91.6%**

Table 3: bACC (%) of automated evaluation methods for key-fact alignment (Recall) and fact-checking (Faithfulness) against expert-annotated gold labels. FINESURE does not support key-fact extraction for query-focused summarization, so the cell is marked 'N/A'.

context. Consequently, HAMLET achieves significantly higher bACC in evaluating LLMs' recall and faithfulness compared to FINESURE, based on expert labels. In particular, for faithfulness, HAMLET greatly improves bACC by anchoring the query to a 4K-token chunk, simplifying the task. This also reduces evaluation API cost from $10.50 to $0.53, achieving a 25× cost saving.

We also compare our automated pipeline with a variant that uses *crowd-sourced* workers instead of LLMs. We collect the label by the majority vote among three Amazon Mturk workers[4] for each instance (refer to Appendix F.2). The result in Table 3 show that a strong LLM like GPT-4o significantly outperforms crowd-sourced workers in summary evaluation, achieving over 90% accuracy. Even a weaker model, GPT-4o-Mini, surpasses the performance of crowd-sourced labels on average.

5 LLMs' Long-context Comprehension

To benchmark LLMs' long-context comprehension, we evaluate six LLMs, *i.e.*, two open-source and four proprietary, on book summarization. We further extend our benchmarking to how response length influences LLM recall and to examine the effectiveness of reasoning-optimized models.

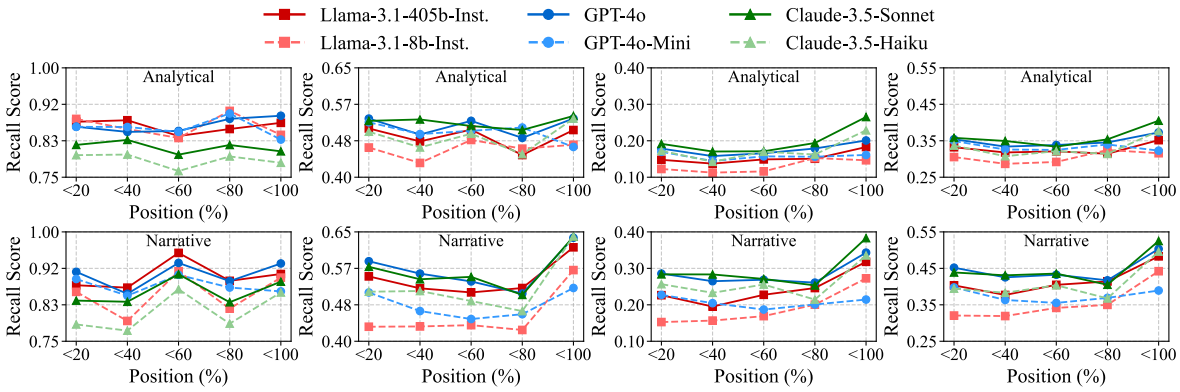
5.1 Main Experiment

5.1.1 Evaluating Multi-level Recall

Figure 3 shows LLMs' recall of key-facts across three abstraction levels, along with overall recall ("all"), computed over five input position splits based on the corresponding queries. Recall is measured by whether each key-fact is included in the generated summary, as mentioned in Eq. (1).

Overview. The results show *a consistent decline in recall from high-level gist (root-level in (a)) to fine-grained detail (leaf-level in (c))*. Considering all LLMs, recall drops from 0.764-0.902 at the root to

[4]The inter-annotator agreement (IAA) scores, measured using Gwet's AC1 (Gwet, 2008), are 0.53 for the key-fact alignment task and 0.47 for the fact-checking task.



(a) Root-level. (b) Branch-level. (c) Leaf-level. (d) All-level.

Figure 3: Recall scores of six LLMs across four key-fact levels, based on query chunk locations for lost-in-the-middle analysis. The first and second rows show recall for analytical and narrative contexts.

	Llama-3.1-405B		Llama-3.1-8B		GPT-4o		GPT-4o-Mini		Claude-Sonnet		Claude-Haiku		Total	
Level	Ana.	Nar.	Ana.	Nar.	Ana.	Nar.	Ana.	Nar.	Ana.	Nar.	Ana.	Nar.	Ana.	Nar.
Root-level Key-fact	0.036	0.079	0.062	0.114	0.037	0.071	0.060	0.048	0.034	0.064	0.038	0.095	0.045	0.079
Branch-level Key-fact	0.061	0.128	0.052	0.137	0.044	0.128	0.056	0.071	0.032	0.133	0.081	0.171	0.054	0.128
Leaf-level Key-fact	0.045	0.123	0.040	0.120	0.043	0.082	0.025	0.040	0.095	0.130	0.086	0.121	0.056	0.103
All-level Key-fact	0.037	0.106	0.040	0.123	0.039	0.086	0.026	0.042	0.071	0.120	0.069	0.128	0.047	0.101

Table 4: Recall Gap across varying abstraction levels under both analytical (Ana.) and narrative (Nar.) perspectives.

0.432-0.540 in the branch, and further to 0.113-0.265 at the leaf for analytical key-facts, while 0.774-0.952 to 0.426-0.640 and further to 0.153-0.383 for narrative ones. Thus, recent LLMs struggle more to generate responses reflecting detailed, leaf-level information; this limitation is more pronounced with analytical than narrative content.

Lost-in-the-Middle. LLMs show a pronounced drop in recall for queries linked to chunks appearing between 20%–80% of the way through the input, commonly referred to as the lost-in-the-middle effect (Liu et al., 2024a). *Compared to the root- or branch-levels (in (a) and (b)), the drop at the leaf-level (in (c)) is much sharper* with recall up to 0.10 lower than at the document's beginning or end. That is, detailed information is more vulnerable to the lost-in-the-middle effect.

Open-source vs. Proprietary LLMs. The results demonstrate that proprietary LLMs generally exhibit higher recall than open-source LLMs across all abstraction levels, as indicated by the red lines (Llama series) appearing primarily below the others (GPT and Claude series). However, it is noteworthy that the Llama series matches or exceeds proprietary models in root-level recall (in (a)), suggesting that *open-source LLMs are competitive in capturing abstract content but still lag behind in retrieving detailed, fine-grained information.*

Large vs. Small LLMs. We observe substantial and consistent gaps in recall scores between larger and smaller LLMs across all model series (shown by the gap between same-colored solid and dotted lines), highlighting that *larger LLMs are superior for long-context comprehension tasks*. Nevertheless, the two groups show negligible differences in vulnerability to the lost-in-the-middle effect, suggesting that *increasing parameter size alone does not mitigate this issue in recall.*

Positional Consistency. An important aspect of LLM recall is its positional consistency, indicating robustness to the lost-in-the-middle effect. Table 4 reports the recall gap, which is the difference between the highest and lowest recall across five input positions in Figure 3. Note that a smaller gap indicates greater positional consistency.

The GPT series exhibits the best positional consistency. Smaller LLMs tend to be slightly less consistent than their larger counterparts, though the differences are modest. By contrast, *the type of key-fact, analytical or narrative, has a greater impact on positional consistency than model type and size.* Specifically, LLMs exhibit higher recall consistency (*i.e.*, lower recall gaps) on analytical content, likely due to its more explicit and localized structure compared to narrative content.

5.1.2 Evaluating Multi-level Faithfulness

Table 5 shows the faithfulness scores of LLM generated summary sentences, labeled as four categories as defined in the Eq. (2). Overall, *hallucinations are frequent across all models*, with faithfulness

Sentence	Llama-3.1-405B		Llama-3.1-8B		GPT-4o		GPT-4o-Mini		Claude-Sonnet		Claude-Haiku		Total	
Category	Ana.	Nar.	Ana.	Nar.	Ana.	Nar.	Ana.	Nar.	Ana.	Nar.	Ana.	Nar.	Ana.	Nar.
Root	0.615	0.540	0.558	0.437	0.643	0.547	0.558	0.408	0.533	0.507	0.540	0.568	0.575	0.501
Branch	0.584	0.599	0.557	0.465	0.633	0.631	0.546	0.456	0.518	0.560	0.586	0.639	0.571	0.558
Leaf	0.616	0.648	0.571	0.515	0.655	0.664	0.547	0.482	0.539	0.596	0.611	0.674	0.590	0.597
None	0.497	0.243	0.404	0.151	0.507	0.378	0.464	0.281	0.481	0.317	0.382	0.272	0.456	0.274
All	0.602	0.594	0.560	0.468	0.642	0.614	0.551	0.446	0.529	0.553	0.578	0.627	0.577	0.550

Table 5: Faithfulness scores of summary sentences labeled under analytical and narrative perspectives.

Evaluation Dimension	Multi-level Recall				Multi-level Faithfulness				
Abstraction Level	Root-level	Branch-level	Leaf-level	All	Root-level	Branch-level	Leaf-level	None	All
Difference ({R1-Distil-Qwen} - {Qwen})	-0.049	-0.089	-0.062	-0.069	0.194	0.137	-0.001	0.253	0.128

Table 6: Recall and faithfulness score differences between Qwen2.5-32B-Instruct and DeepSeek-R1-Distil-Qwen-32B across varying abstraction levels. Negative values indicate the non-reasoning model performs better.

ranging from 0.151 to 0.674, much lower than in short-context tasks (Lee et al., 2024; Tang et al., 2024b). Moreover, *hallucinations are likely to occur independently of the level of abstraction*, as the scores show little variation across categories in {Root, Branch, Leaf}. However for the "None" category, faithfulness scores are notably lower, suggesting that *content unrelated to any key-facts is highly prone to hallucination especially in narrative contexts*, highlighting a unique challenge in LLMs' long-context comprehension.

From the model perspective (see the "All" row), *GPT-4o exhibits the highest faithfulness scores*, while Llama-3.1-405B surpasses the proprietary Claude. Interestingly, Claude shows little difference between its small and large versions, unlike other models where the gap is substantial. This suggests that *model size alone does not reliably predict factual alignment*; architecture and alignment objectives may play a greater role in faithfulness.

5.2 Additional Experiment

General vs. Reasoning Model. As reasoning models show significant performance gains on complex tasks, we examine whether similar benefits hold for long-context tasks. Table 6 shows recall and faithfulness differences between a reasoning model (R1-Distil-Qwen) and its non-reasoning version (Qwen), where the scores are aggregated for each abstraction levels. Contrary to our expectations, recall declines consistently, suggesting *reasoning-optimized training may hinder long-context comprehension* by prioritizing inference over extraction. But, this trade-off improves faithfulness, leading to more factually accurate responses.

Impact of Response Length on Recall. A potential factor affecting LLM recall is the length of the generated response, since a longer summary leads to greater recall. Hence, we compute the Pearson correlation between summary length and recall scores at each abstraction level, combining outputs from all LLMs. The results demonstrate that *response length has minimal impact on recall, with correlation coefficients close to zero across all levels.* Specifically, the correlation is 0.02, -0.02, -0.09, and -0.07 at root-, branch-, leaf-, and all-levels.

Model	Recall	Faithfulness	Mean
Llama-3.1-405B-Instruct	0.372	0.473	0.423
Llama-3.1-8B-Instruct	0.330	0.365	0.348
GPT-4o	0.398	0.528	0.463
GPT-4o-Mini	0.354	0.420	0.387
Claude-3.5-Sonnet	0.403	0.457	0.430
Claude-3.5-Haiku	0.370	0.478	0.424

Table 7: Summary-level recall and faithfulness of two open-source LLMs and four proprietary LLMs.

Summary-level Evaluation. While Section 5.1 presents fine-grained benchmarking via key-fact and sentence-level analyses, we also conduct a coarse-grained summary-level evaluation to assess how well LLMs generate query-focused summaries for long-context inputs. Table 7 compares the summary quality of six LLMs in terms of their summary-level recall and faithfulness scores. Proprietary and larger LLMs generate better query-focused summaries from long-context inputs. In detail, GPT-4o achieves the highest mean score of 0.463, while the Claude series performs comparably to, or slightly better than, the Llama series. A consistent trend across all three model series is that larger LLMs produce higher-quality summaries than their smaller counterparts.

6 Conclusion

We presented a holistic benchmark, automatically evaluating LLMs' long-context comprehension. It is built around a key-fact tree, a root-branch-leaf hierarchy enabling multi-level analysis from analytical and narrative aspects, revealing insights, *e.g.*, the impact of chunk position on LLM compre-

hension. In particular, its fully automated pipeline outperforms crowd-workers, achieving over 90% agreement with experts, and reduces cost by 25×.

Limitations

While HAMLET provides significant advancements in evaluating LLMs on book-length texts, several limitations present opportunities for future work. First, our benchmark currently focuses on literary novels, and expanding to additional domains such as academic texts, technical documentation, or non-fiction would broaden applicability and test comprehension across different writing styles. However, one practical reason we chose novels as our initial domain is the scarcity of publicly accessible, high-quality documents exceeding 100k tokens in length. Novels offer one of the few consistently high-quality and extensive textual resources freely accessible for research. This choice ensures that our benchmarking pipeline can robustly evaluate LLM comprehension on genuinely long-form content without compromising textual integrity or quality. Such extensive, quality-assured texts are significantly less common or unavailable in many other domains, especially when public accessibility is required for reproducibility and transparency in benchmarking research. Second, the current coverage is limited to English; extending HAMLET to other languages will test multilingual robustness and reveal challenges unique to different linguistic structures. Third, HAMLET focuses primarily on recall and faithfulness as evaluation dimensions due to strong alignment with human judgment. A promising direction for future work involves designing automated metrics for other nuanced dimensions, including bias, coherence and reasoning quality. Lastly, although HAMLET enables multi-level evaluation of book-length text comprehension across different information granularities, a more integrated approach to quantifying this capability in LLMs would be beneficial, such as investigating weighting schemes that account for hierarchical information structure.

Ethics Statement

Our research placed strong emphasis on transparent communication with all human annotators involved in the evaluation process. We ensured fair compensation practices, providing crowdsourced workers with payments that surpassed U.S. federal minimum wage standards, while our expert evaluators received professional-level compensation (over $30 hourly) plus additional incentives according to the quality of their work. We maintained strict data privacy protocols throughout the study, carefully anonymizing all personal identifiers in our dataset to protect annotator confidentiality.

Scientific Artifacts

Our benchmark utilizes 16 commercially published novels with appropriate copyright considerations. For summary generation, we used commercial APIs such as OpenAI and AWS Bedrock. Summary model details are in Table 11, providing comprehensive specifications of context window sizes, knowledge cutoffs, and model versions used.

Acknowledgements

This work was supported by the IITP grant funded by the Korea government(MSIT) (RS-2025-25410841, Beyond the Turing Test: Human-Level Game-Playing Agents with Generalization and Adaptation) and by the NRF grant funded by Ministry of Science and ICT (RS-2022-NR068758, Industry and Society Demand Oriented Open Human Resource Development). For GPU infrastructure, our work was supported by the IITP grant funded by MSIT (No. RS-2025-02653113, High-Performance Research AI Computing Infrastructure Support at the 2 PFLOPS Scale).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Chenxin An, Shansan Gong, Ming Zhong, Xingjian Zhao, Mukai Li, Jun Zhang, Lingpeng Kong, and Xipeng Qiu. 2024. L-eval: Instituting standardized evaluation for long context language models. In *ACL*.

Yapei Chang, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024. Booookscore: A systematic exploration of book-length summarization in the era of LLMs. In *ICLR*.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. 2024. LongRoPE: extending llm context window beyond 2 million tokens. In *ICML*.

Zican Dong, Tianyi Tang, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. 2024. BAMBOO: A comprehensive benchmark for evaluating long text modeling capacities of large language models. In *Proceedings of*

the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pages 2086–2099, Torino, Italia. ELRA and ICCL.

Yongqi Fan, Hongli Sun, Kui Xue, Xiaofan Zhang, Shaoting Zhang, and Tong Ruan. 2025. MedOdyssey: A medical domain benchmark for long context evaluation up to 200k tokens. In *NAACL*.

Jinlan Fu, See Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2024. GPTScore: Evaluate as you desire. In *NAACL*.

Qichen Fu, Minsik Cho, Thomas Merth, Sachin Mehta, Mohammad Rastegari, and Mahyar Najibi. LazyLLM: Dynamic token pruning for efficient long context llm inference. In *ICMLW*.

Kilem Li Gwet. 2008. Computing inter-rater reliability and its variance in the presence of high agreement. *British Journal of Mathematical and Statistical Psychology*, 61(1):29–48.

Linda He, WANG Jue, Maurice Weber, Shang Zhu, Ben Athiwaratkun, and Ce Zhang. 2025. Scaling instruction-tuned llms to million-token contexts via hierarchical synthetic data generation. In *ICLR*.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Dima Acharya, Shantanu Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. Ruler: What's the real context size of your long-context language models? In *COLM*.

Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Zirui Liu, Chia-Yuan Chang, Huiyuan Chen, and Xia Hu. 2024. LLM maybe longlm: Selfextend llm context window without tuning. In *ICML*.

Marzena Karpinska, Katherine Thai, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024. One thousand and one pairs: A "novel" challenge for long-context language models. In *EMNLP*.

Yekyung Kim, Yapei Chang, Marzena Karpinska, Aparna Garimella, Varuu Manjunatha, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024. Fables: Evaluating faithfulness and content selection in book-length summarization. In *COLM*.

Wojciech Kryscinski, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. 2022. BOOKSUM: A collection of datasets for long-form narrative summarization. In *EMNLP*.

Wai-Chung Kwan, Xingshan Zeng, Yufei Wang, Yusen Sun, Liangyou Li, Yuxin Jiang, Lifeng Shang, Qun Liu, and Kam-Fai Wong. 2024. M4LE: A multi-ability multi-range multi-task multi-domain long-context evaluation benchmark for large language models. In *ACL*.

Philippe Laban, Alexander Fabbri, Caiming Xiong, and Chien-Sheng Wu. 2024a. Summary of a haystack: A challenge to long-context LLMs and RAG systems. In *EMNLP*.

Philippe Laban, Alexander Richard Fabbri, Caiming Xiong, and Chien-Sheng Wu. 2024b. Summary of a Haystack: A challenge to long-context llms and rag systems. In *EMNLP*.

Yuho Lee, Taewon Yun, Jason Cai, Hang Su, and Hwanjun Song. 2024. UniSumEval: Towards unified, fine-grained, multi-dimensional summarization evaluation for LLMs. In *EMNLP*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024a. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 11:157–173.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *EMNLP*.

Yushan Liu, Zili Wang, and Ruifeng Yuan. 2024b. QuerySum: a multi-document query-focused summarization dataset augmented with similar query clusters. In *AAAI*.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of llm agents. In *ACL*.

Raymond A Mar, Jingyuan Li, Anh TP Nguyen, and Cindy P Ta. 2021. Memory and comprehension of narrative versus expository texts: A meta-analysis. *Psychonomic Bulletin & Review*, 28:732–749.

Hwanjun Song, Hang Su, Igor Shalyminov, Jason Cai, and Saab Mansour. 2024. FineSurE: Fine-grained summarization evaluation using llms. In *ACL*.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024a. Minicheck: Efficient fact-checking of llms on grounding documents. *arXiv preprint arXiv:2404.10774*.

Liyan Tang, Igor Shalyminov, Amy Wing-mei Wong, Jon Burnsky, Jake W Vincent, Yu'an Yang, Siffi Singh, Song Feng, Hwanjun Song, Hang Su, et al. 2024b. TofuEval: Evaluating hallucinations of llms on topic-focused dialogue summarization. In *NAACL*.

Tempest A van Schaik and Brittany Pugh. 2024. A field guide to automatic evaluation of llm-generated summaries. In *SIGIR*.

David Wan, Jesse Vig, Mohit Bansal, and Shafiq Joty. 2024. On positional bias of faithfulness for long-form summarization. *arXiv preprint arXiv:2410.23609*.

Cunxiang Wang, Ruoxi Ning, Boqi Pan, Tonghui Wu, Qipeng Guo, Cheng Deng, Guangsheng Bao, Qian Wang, and Yue Zhang. 2024. Novelqa: A benchmark for long-range novel question answering. *arXiv preprint arXiv:2403.12766*.

Jiaan Wang, Yunlong Liang, Fandong Meng, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023. Is chatgpt a good nlg evaluator? a preliminary study. *arXiv preprint arXiv:2303.04048*.

Yunshu Wu, Hayate Iso, Pouya Pezeshkpour, Nikita Bhutani, and Estevam Hruschka. 2023. Less is more for long document summary evaluation by llms. *arXiv preprint arXiv:2309.07382*.

Ruochen Xu, Song Wang, Yang Liu, Shuohang Wang, Yichong Xu, Dan Iter, Pengcheng He, Chenguang Zhu, and Michael Zeng. 2023. LMGQS: A large-scale dataset for query-focused summarization. In *EMNLP*.

Yumo Xu and Mirella Lapata. 2020. Coarse-to-fine query focused multi-document summarization. In *EMNLP*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. BERTScore: Evaluating text generation with bert. In *ICLR*.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2024a. ∞Bench: Extending long context evaluation beyond 100K tokens. In *ACL*.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, et al. 2024b. ∞ bench: Extending long context evaluation beyond 100k tokens. In *ACL*.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. In *EMNLP*.

Name	Author	Genre	Token Count	Publication Date	# Chunks*	Analytical				Narrative			
						Root	Branch	Leaf	Total	Root	Branch	Leaf	Total
Wyoming Burn	Jerry Fedora	Mystery	74,392	Jan 2, 2025	19	56	141	270	467	65	164	302	531
No Place Left to Hide	Megan Lally	Thriller	85,616	Jan 7, 2025	22	58	169	365	592	68	182	351	601
Lady's Steed	Eve Langlais	Romance, Fantasy	89,884	Dec 24, 2024	23	61	181	384	626	69	207	382	658
The Assassin's Guide to Babysitting	Natalie C. Parker	Young Adult	93,337	Jan 7, 2025	24	77	202	387	666	82	225	407	714
A Conventional Boy: A Laundry Files Novel	Charles Stross	Fantasy, Horror	96,274	Jan 7, 2025	24	56	181	379	616	74	210	394	678
Lies on the Serpent's Tongue	Kate Pearsall	Fantasy, Horror	96,311	Jan 7, 2025	24	73	192	377	642	84	209	377	670
All the Water in the World	Eiren Caffall	Science Fiction	98,082	Jan 7, 2025	25	65	195	413	673	97	251	402	750
Holmes is Missing	James Patterson	Mystery, Thriller	98,497	Jan 2, 2025	25	87	214	382	683	101	246	391	738
The Lodge	Kayla Olson	Romance	104,102	Jan 7, 2025	26	81	205	375	661	98	231	380	709
Switching Graves	Jen Stevens	Dark, Gothic	107,565	Jan 3, 2025	27	85	230	461	776	90	237	417	744
So Not My Type	Dana Hawkins	Romance	107,728	Dec 12, 2024	27	66	186	375	627	75	220	384	679
Close Your Eyes	Teresa Driscoll	Thriller	107,933	Jan 1, 2025	27	90	240	489	819	99	283	528	910
Kingdom of Faewood	Krista Street	Fantasy	108,780	Jan 3, 2025	28	92	225	447	764	92	220	424	736
Bitter Passage: An Allegheny Beckham Novel	Colin Mills	Mystery	111,603	Jun 22, 2024	28	72	220	456	748	95	241	460	796
The Three Lives of Cate Kay	Kate Fagan	Fiction, Mystery	113,253	Jan 7, 2025	29	105	248	459	812	104	255	424	783
Some Other Time	Angela Brown	Fiction	114,761	Jan 1, 2025	29	85	223	436	744	99	235	386	720
Average (Total)	-	-	100,507	-	25 (407)	76 (1209)	203 (3252)	403 (6445)	682 (10916)	87 (1382)	226 (3616)	401 (6409)	714 (11417)

Table 8: Statistics and details of the novels used in HAMLET. For each chunk, two key-fact trees and two queries are created corresponding to the two summarization perspectives (analytical and narrative).

A Dataset Statistics

Our benchmark consists of 16 novels, each divided into sequential chunks of approximately 4K tokens with preserved sentence boundaries. Table 8 summarizes the basic statistics of HAMLET. For text processing, we use OpenAI's `tiktoken`[2] tokenization library to compute token counts. Due to copyright restrictions, we release only the generated queries, key-fact trees, model-generated summaries, and evaluation labels, excluding the original book contents.

B Choice of Chunk Size

In this paper, we choose 4K as the chunk size. This size preserves coherence for hierarchical key-facts while remaining short enough for reliable positional evaluation of LLM recall, aligning with recent findings (He et al., 2025). To validate this choice, we conducted a comparative analysis across four chunk sizes (1K, 2K, 4K, and 8K) using an LLM-as-a-judge framework. The evaluation covered three key dimensions: *validity*, whether the extracted key-facts form a clear multi-level hierarchical structure; *coherence*, the structural integrity and logical flow of the chunk; and *cross-content*, the extent to which the chunk supports reasoning across different sections. Specifically, we first chunked five selected books using three different sizes (1K, 2K, 4K and 8K tokens), and then randomly sampled 100 chunks for each chunk size. Next, we employed two LLMs (GPT-4o and Claude 3.5 Sonnet) as judges to evaluate each chunk across three dimensions, using the standardized prompt shown below. The 1-5 likert-scale scores for each dimension were averaged over the two judges. The detailed prompt can be found in Table 24.

Chunk Size	Vadility	Coherence	Cross-content
1K-token	2.32	3.85	3.21
2K-token	3.50	4.15	4.20
4K-token	4.48	4.42	4.77
8K-token	4.76	4.40	4.84

Table 9: Scores of chunks across different chunk sizes.

Table 9 shows that scores across all three dimensions improve as the chunk size increases. While performance continues to rise up to 8K, the improvement from 4K to 8K is marginal, indicating that performance largely saturates at 4K. Also, the 4K-token chunk size achieves a high validity score of 4.48 (out of 5.0) for capturing hierarchical key-facts (i.e., root–branch–leaf structure) and a strong coherence score of 4.42 (out of 5.0), suggesting that chunks remain coherent and do not arbitrarily cut across scenes, dialogues, or paragraphs. These findings provide strong empirical support for 4K-token chunks as a meaningful unit of analysis. To sum up, 4K is a more practical choice, as it provides sufficient context to maintain high validity and coherence for hierarchical key-facts, while remaining short enough to enable reliable positional evaluation of LLM recall.

[2] https://github.com/openai/tiktoken

	Analytical	Narrative
Definition	Analytical perspective focuses on thematic elements, implications, character development, and symbolic patterns, examining how these literary components build toward deeper meaning and authorial intent.	Narrative perspective emphasizes chronological storytelling, key events, and plot developments, concentrating on how the storyline unfolds and progresses through the text.
Root Definition	A single concise sentence summarizing the overarching purpose, argument, or main analytical insight of the text.	A single concise sentence capturing the main idea or overarching sequence of events in the text.
Branch Definition	Key supporting ideas, arguments, or elements that develop the overarching purpose or insight, including significant stages, relationships, or turning points in the text.	Key supporting events or developments that progress the narrative logically, including major stages, actions or transitions.
Leaf Definition	Specific evidence, minor details, or examples that provide additional support or elaboration for each branch.	Specific details, minor events, or pieces of evidence that provide additional clarity or elaboration for each branch.
Query Example	How do the social rejection and personal challenges Holmes faces at the bar, his subsequent return home with Callie Brett's assistance, and the professional and personal challenges faced by Poe and Holmes, including the discovery of twins, collectively illustrate the overarching themes of vulnerability, trust, and commitment in the narrative?	Provide a detailed summary of the events involving Holmes at the bar, his interaction with Callie Brett, the subsequent health incident, and the developments with Poe and Grey, including their medical appointment and the team's commitment to a crime-writers' convention?

Table 10: Definitions of two summarization perspectives and their example queries.

Model Type	Model Name	Context Length	Knowledge Cutoff	Hugging Face Checkpoints & Official API Version
Proprietary	GPT-4o	128K	Oct, 2023	gpt-4o-2024-08-06
	GPT-4o-Mini	128K	Oct, 2023	gpt-4o-mini-2024-07-18
	Claude-3-5-Sonnet	200K	Apr, 2024	claude-3-5-sonnet-20240620
	Claude-3-5-Haiku	200K	Jul, 2024	claude-3-5-haiku-20241022
Open-Source	LLaMA-3.1-405B	128K	Dec, 2023	meta-llama/Llama-3.1-405B-Instruct
	LLaMA-3.1-8B	128K	Dec, 2023	meta-llama/Llama-3.1-8B-Instruct
	Qwen2.5-32B-Instruct	128K	–	Qwen/Qwen2.5-32B-Instruct
	DeepSeek-R1-Distill-Qwen-32B	128K	–	deepseek-ai/DeepSeek-R1-Distill-Qwen-32B

Table 11: Overview of model specifications including their types, context window sizes, and knowledge cutoff dates. Note that GPT-4o-Mini is also used as the baseline model for automated summary evaluation (Section 3).

C Key-fact Tree & Query Type Details

Table 10 shows the detailed definitions of each level in our key-fact hierarchy tree and the corresponding queries across analytical and narrative perspectives. Next, the automated key-fact tree validation process filters out about 4% of root-level key-facts, 13% of branch-level key-fact, and 30% of leaf-level key-facts, resulting in an overall pruning rate of 22%.

D Model Details

We compare various proprietary and open-source LLMs, highlighting their context lengths and knowledge cutoffs. The details of these models are presented in Table 11. For open-source LLMs, we use instruction-tuned versions. For proprietary models, we use official APIs. To ensure the reproducibility , we use the greedy decoding by setting the temperature parameter to 0. We spent a total of $800 on model inference. Due to the long input size, the cost remains significant even before factoring in annotation expenses.

Error Category	Error Type	Description
Extrinsic Error	Out-of-article Error	The summary introduces facts, opinions, or information not found in or reasonably inferrable from the text
Intrinsic Error	Entity Error	Incorrect reference to key subjects/objects (*e.g.*, wrong names, numbers, pronouns)
	Relation Error	Mistakes in semantic relationships (e.g., incorrect verbs, prepositions, adjectives)
	Sentence Error	Multiple errors causing an entire sentence to contradict the source text

Table 12: Faithfulness error categories.

E Evaluation Details

Our work builds on Song et al. (2024), adapting their automatic summarization evaluation framework to suit our task. The evaluation consists of two tasks: *Fact verification* and *Key-fact alignment*.

E.1 Fact Verification

To evaluate faithfulness of a summary, the summary is split into individual sentences, and the original text from which the query was generated is provided as context. LLM as an evaluator assigns a binary label to each sentence: faithful (1) or unfaithful (0). Additionally, following the taxonomy in Table 12, each unfaithful sentence is categorized, with an accompanying explanation. We use GPT-4o for the evaluation. Table 22 provides the fact verification prompt.

E.2 Key-fact Alignment

The key-fact alignment task is designed to compute recall and multi-level faithfulness scores. Table 23 provides the key-fact alignment prompt. The cor-

Perspective	Model	Position 0-20%				Position 20-40%				Position 40-60%				Position 60-80%				Position 80-100%				Average			
		Rt	Br	Lf	W	Rt	Br	Lf	W	Rt	Br	Lf	W	Rt	Br	Lf	W	Rt	Br	Lf	W	Rt	Br	Lf	W
Analytical	GTP-4o	0.87	0.53	0.18	0.35	0.85	0.50	0.16	0.33	0.86	0.53	0.17	0.34	0.88	0.49	0.18	0.35	0.89	0.53	0.20	0.37	0.87	0.52	0.18	0.35
	GPT-4o-Mini	0.87	0.53	0.17	0.35	0.87	0.50	0.14	0.33	0.85	0.51	0.16	0.33	0.90	0.51	0.16	0.34	0.84	0.47	0.16	0.32	0.86	0.50	0.16	0.33
	Claude-3.5-Sonnet	0.82	0.53	0.19	0.36	0.84	0.53	0.17	0.35	0.80	0.52	0.17	0.33	0.82	0.51	0.19	0.35	0.81	0.54	0.27	0.41	0.82	0.53	0.20	0.36
	Claude-3.5-Haiku	0.80	0.50	0.17	0.34	0.80	0.47	0.14	0.31	0.76	0.50	0.17	0.32	0.80	0.45	0.16	0.31	0.78	0.53	0.23	0.38	0.79	0.49	0.18	0.33
	LLaMA-3.1-405b-Inst.	0.88	0.51	0.15	0.33	0.88	0.48	0.14	0.32	0.84	0.51	0.15	0.32	0.86	0.45	0.15	0.31	0.87	0.51	0.18	0.35	0.87	0.49	0.15	0.33
	LLaMA-3.1-8b-Inst.	0.88	0.47	0.12	0.31	0.86	0.43	0.11	0.29	0.84	0.49	0.12	0.29	0.90	0.47	0.15	0.33	0.85	0.47	0.15	0.32	0.87	0.47	0.13	0.31
	Average	0.85	0.51	0.16	0.34	0.85	0.49	0.14	0.32	0.83	0.51	0.15	0.32	0.86	0.48	0.17	0.33	0.84	0.51	0.20	0.36	0.85	0.50	0.17	0.33
Narrative	GTP-4o	0.91	0.58	0.29	0.45	0.86	0.56	0.27	0.43	0.93	0.54	0.27	0.43	0.89	0.51	0.26	0.42	0.93	0.64	0.34	0.50	0.90	0.56	0.29	0.45
	GPT-4o-Mini	0.89	0.51	0.23	0.40	0.85	0.47	0.21	0.36	0.90	0.45	0.19	0.36	0.87	0.46	0.20	0.37	0.86	0.52	0.21	0.39	0.88	0.48	0.21	0.38
	Claude-3.5-Sonnet	0.84	0.57	0.28	0.44	0.84	0.54	0.28	0.43	0.90	0.55	0.27	0.44	0.84	0.51	0.25	0.41	0.89	0.64	0.38	0.53	0.86	0.56	0.30	0.45
	Claude-3.5-Haiku	0.79	0.51	0.26	0.39	0.77	0.52	0.23	0.38	0.87	0.49	0.26	0.40	0.79	0.47	0.21	0.37	0.86	0.64	0.34	0.50	0.82	0.53	0.26	0.41
	LLaMA-3.1-405b-Inst.	0.88	0.55	0.23	0.40	0.87	0.52	0.20	0.38	0.95	0.51	0.23	0.41	0.89	0.52	0.25	0.41	0.90	0.62	0.32	0.48	0.90	0.54	0.24	0.42
	LLaMA-3.1-8b-Inst.	0.86	0.43	0.15	0.32	0.80	0.43	0.16	0.32	0.91	0.44	0.17	0.34	0.83	0.43	0.20	0.35	0.90	0.56	0.27	0.44	0.86	0.46	0.19	0.36
	Average	0.86	0.53	0.24	0.40	0.83	0.51	0.22	0.38	0.91	0.50	0.23	0.40	0.85	0.48	0.23	0.39	0.89	0.60	0.31	0.47	0.87	0.52	0.25	0.41

Table 13: Model-wise breakdown of recall across key-fact levels, relative positions of information in input context, and summary perspectives. Rt: Root; Br: Branch; Lf: Leaf; W: Whole

responding key-fact tree is linearized into a key-fact list using depth-first traversal. Along with the original summary sentences, this list is provided to GPT-4o as the evaluator to determine whether each key-fact is included (1) or not included (0) in the summary. Additionally, the sentence number from the summary that contains each key-fact is recorded and aligned with the corresponding key-fact. This can be further used for multi-level recall and multi-level faithfulness scores calculation.

F Annotator Recruitment Details

F.1 Expert Annotators

We recruited three postgraduate students specializing in Natural Language Processing as expert annotators (with C2-level English proficiency) for the validation of the three critical components of our automated pipeline: key-fact tree generation, query generation, and automated evaluation (see Section 4). They were compensated at a rate of $30 per hour, with additional performance-based incentives to ensure high-quality contributions.

F.2 Crowd-sourced Annotators

As a baseline for our automated summarization with GPT-4o, we used summary evaluations from crowd-sourced annotators (see Section 4). For this purpose, we recruited three annotators for every Human Intelligence Task (HIT) from Amazon Mechanical Turk (MTurk) who met stringent qualification requirements.

Our recruitment criteria included successful completion of an English comprehension assessment that mirrored the actual annotation tasks of fact verification and key-fact alignment. Additionally, workers were required to maintain a minimum 90% lifetime approval rate and demonstrate experience with at least 500 previously accepted HITs. All annotators received compensation exceeding the U.S. federal minimum wage.

For quality control, we embedded 5–10% hidden attention-check questions with predetermined answers within each HIT. Any submissions failing these attention checks were rejected. This rigorous quality assurance procedure effectively filtered unreliable responses and ensured that all collected annotations were of high quality.

G Supplementary Result

We provide supplementary result that provides further insights into our primary findings.

Category	% of Total Faithfulness Errors
(A) Extrinsic Error	95%
(B) Intrinsic Error	5%
Subcategories of (B) Intrinsic Error	% of (B) Intrinsic Error
Relation Error	54%
Entity Error	45%
Sentence Error	1%

Table 14: Faithfulness error type distribution. The result is averaged over the six summarizers, excluding the two Qwen-2.5-32B models (Instruct and R1-distill).

G.1 Evaluation Performance

Table 13 shows the detailed model-wise breakdown of key-fact retention performance (*i.e.*, multi-level recall) across key-fact levels and relative positions

of information in input context, and summary perspectives

G.2 Faithfulness Error Distribution

Table 14 reveals that extrinsic errors constitute 95% of all faithfulness errors across the evaluated summarization systems, with intrinsic errors accounting for only 5%. Within the category of intrinsic errors, relation errors (54%) and entity errors (45%) account for nearly all cases, with sentence errors representing only 1%.

The overwhelming prevalence of extrinsic errors suggests that current LLMs have a fundamental tendency to generate plausible but unfounded content, rather than merely misrepresenting information that exists in the source. Notably, the near absence of sentence errors (only 1% of intrinsic errors) indicates that even when summarizing book-length content, models rarely produce statements that completely contradict the source material. Instead, when errors do occur within the bounds of input context, they typically manifest as more nuanced misrepresentations of specific entities or their relationships, rather than wholesale errors.

You will be given an excerpt of a longer text. Read the excerpt carefully and extract all the key analytical insights related to the structure, relationships, and significance of the content. Organize these insights into a hierarchical tree structure with three levels: Root, Branches, and Leaves.

Structure levels:

- Root: A single concise sentence summarizing the overarching purpose, argument, or main analytical insight of the text.
- Branches: Key supporting ideas, arguments, or elements that develop the overarching purpose or insight, including significant stages, relationships, or turning points in the text.
- Leaves: Specific evidence, minor details, or examples that provide additional support or elaboration for each branch.

Requirements:

1. Do not omit any significant information from the text.
2. Ensure clear relationships between roots, branches, and leaves.
3. All key-facts must be directly supported by the text.
4. Create as many roots, branches and leaves as needed to fully capture the text's key-facts.
5. All key-facts should NEVER be based on over-interpretation or logical leaps beyond the information provided in the text.
6. Focus on analyzing what is explicitly stated, supported, or implied within reasonable bounds, without adding subjective opinions or unsupported inferences.
7. NEVER use pronouns, such as he, she, it, that, or "the protagonist". ALWAYS USE PROPER NOUNS.
8. Make each key-fact as concise as possible, ensuring that each contain at most 2-3 entities.

Output format:
- Provide your answer in JSON format.
- The answer should ONLY be a dictionary with the valid JSON format as follows: <Tree>
- Include only the tree dictionary in the answer.

The excerpt:
{excerpt}

Table 15: Analytical key-fact tree generation prompt.

You will be given an excerpt of a longer text. Read the excerpt carefully and extract all the key-facts related to the sequence of events and key developments in a straightforward and chronological manner. Organize these key-facts into a hierarchical tree structure with three levels: Root, Branches, and Leaves.

Structure levels:

- Root: A single concise sentence capturing the main idea or overarching sequence of events in the text.
- Branches: Key supporting events or developments that progress the narrative logically, including major stages, actions or transitions.
- Leaves: Specific details, minor events, or pieces of evidence that provide additional clarity or elaboration for each branch.

Requirements:

1. Each key-fact is NOT a statement of theme or topic of the text, but a specific piece of information that can be directly extracted from the text.
2. Do not omit any significant information from the text.
3. Ensure clear relationships between roots, branches, and leaves.
4. All key-facts must be directly supported by the text.
5. Create as many roots, branches and leaves as needed to fully capture the text's key-facts.
6. Focus on how the story progresses from beginning to end, including any critical pivots or climaxes.
7. NEVER use pronouns, such as he, she, it, that, or "the protagonist". ALWAYS USE PROPER NOUNS.
8. Make each key-fact as concise as possible, ensuring that each contain at most 2-3 entities.

Output format:
- Provide your answer in JSON format.
- The answer should ONLY be a dictionary with the valid JSON format as follows: <Tree>
- Include only the tree dictionary in the answer.

The excerpt:
{excerpt}

Table 16: Narrative key-fact tree generation prompt.

You will receive:

- An excerpt from a novel (the source text).
- A hierarchical key-fact tree in JSON format, structured as follows: <Tree>

Your task:

- Evaluate the faithfulness of each fact in the key-fact tree by carefully comparing it to the provided novel excerpt.
- For every root key-fact, branch key-fact, and leaf key-fact, assign a binary label based on the following criteria:
  - 1 (Faithful): The fact is fully accurate and directly supported by the text.
  - 0 (Unfaithful): The fact is inaccurate, misleading, or not supported by the text.
- For each evaluation, provide a justification explaining why the fact is marked as Faithful or Unfaithful. Condense your reason to one or two sentences.

Important guidelines:

- Only accept facts that are explicitly stated. Do not accept information that is inferred, altered, or expanded beyond what the text directly says.
- Be precise and consistent. Evaluate each level—roots, branches, and leaves—independently for accuracy.
- Maintain input structure: The output must preserve the exact same hierarchical structure as the input key-fact tree. Each node of the output tree should represent the label (0 or 1) and its corresponding justification for that key-fact.

Output format:

- Provide your answer in JSON format.
- The answer should ONLY be a dictionary with the valid JSON format as follows: <Tree>
- Include only the tree dictionary in the answer.

The excerpt:

{excerpt}

The key-fact tree:

{key-fact tree}

Table 17: Faithfulness evaluation of key-fact tree prompt.

You will receive:

- An excerpt from a novel (the source text).
- A hierarchical key-fact tree in JSON format, structured as follows: <Tree>

Your task:

- Evaluate the subjectivity of each fact in the key-fact tree by comparing it to the provided excerpt.
- For every root key-fact, branch key-fact, and leaf key-fact, assign a binary label based on the following criteria:
  - 1 (Objective): The statement is purely fact-based and directly supported by the text, without subjective language or interpretation.
  - 0 (Subjective): The statement includes opinions, assumptions, interpretations, or evaluative/adjectival language.
- For each evaluation, provide a justification explaining why the fact is marked as objective or subjective. Condense your reason to one or two sentences.

Important guidelines:

- Accept only factual statements. Statements must reflect exactly what is stated in the text without additional interpretation.
- Reject subjective language. Statements that express opinions, emotions, or biases must be marked as 0.
- Evaluate each level independently. Assess the objectivity of roots, branches, and leaves separately.
- Maintain input structure: The output must preserve the exact same hierarchical structure as the input key-fact tree. Each node of the output should represent the label (0 or 1) and its corresponding justification for that key-fact.

Output format:

- Provide your answer in JSON format.
- The answer should ONLY be a dictionary with the valid JSON format as follows: <Tree>
- Include only the tree dictionary in the answer.

The excerpt:
{excerpt}

The key-fact tree:
{key-fact tree}

Table 18: Objectivity evaluation of key-fact tree prompt.

You will receive:

– An excerpt from a novel (the source text).

– A hierarchical key-fact tree in JSON format, structured as follows: <Tree>

Your task:

- Evaluate the significance of each fact in the key-fact tree based on the provided excerpt.
- For every root key-fact, branch key-fact, and leaf key-fact, assign a binary label based on the following criteria:
  – 1 (Significant): The fact is essential for understanding the text, such as driving the plot forward, developing characters, or revealing major conflicts.
  – 0 (Insignificant): The fact is trivial, background information, or does not meaningfully contribute to the story's progression or understanding.
- For each evaluation, provide a justification explaining why the fact is marked as significant or insignificant. Condense your reason to one or two sentences.

Important guidelines:

- Avoid trivial details: Facts that describe minor settings, insignificant actions, or irrelevant background information should be scored 0.
- Evaluate independently: Assess the significance of each root, branch, and leaf on its own merit.
- Maintain input structure: The output must preserve the exact same hierarchical structure as the input key-fact tree. Each node of the output should represent the label (0 or 1) and its corresponding justification for that key-fact.

Output format:

- Provide your answer in JSON format.
- The answer should ONLY be a dictionary with the valid JSON format as follows: <Tree>
- Include only the tree dictionary in the answer

The excerpt:

{excerpt}

The key-fact tree:

{key-fact tree}

Table 19: Significance evaluation of key-fact tree prompt.

Main objective:

Craft a single query that requests a summary of the analytical content represented by the key-fact tree. The query should address the overarching purpose or argument (Root), the supporting ideas or elements (Branches), and the specific evidence or examples (Leaves), guiding a coherent examination of how each component relates to the text's main insight.

Definition of a key-fact tree:

A key-fact tree is a hierarchical representation of the important information in a text, organized into three levels:

- Root: A single concise sentence summarizing the overarching purpose, argument, or main analytical insight of the text.
- Branches: Key supporting ideas, arguments, or elements that develop the overarching purpose or insight, including significant stages, relationships, or turning points in the text.
- Leaves: Specific evidence, details, or examples that provide additional support or elaboration for each branch.

You will receive:

– An excerpt from a text (the source text).

– A tree of key-facts in JSON format, with the structure: <Tree>

Requirements:

– The query should naturally lead to an answer that integrates the key-facts, showing how each piece of evidence or argument reinforces the main insight.

– Your query should be specific enough to address the contents in the key-fact tree.

– Make ONLY ONE query for the entire tree.

– Your query should be as concise as possible.

– Your query should NEVER mention anything about the key-fact tree.

The excerpt:
{excerpt}

The key-fact Tree:
{key-fact tree}

Table 20: Analytical query generation prompt.

Main objective:

Craft a single query that requests a summary of the narrative content represented by the keyfact tree. The query should address the overarching sequence of events (Root), the key supporting developments (Branches), and the specific details or events (Leaves), guiding a comprehensive and chronological explanation of how each component contributes to the overall narrative.

Definition of a key-fact tree:

A key-fact tree is a hierarchical representation of the important information in a text, organized into three levels:

- Root: A single concise sentence capturing the main idea or overarching sequence of events in the text
- Branches: Key supporting events or developments that progress the narrative logically, including major stages, actions or transitions
- Leaves: Specific details, events, or pieces of evidence that provide additional clarity or elaboration for each branch

You will receive:

– An excerpt from a text (the source text).

– A tree of key-facts in JSON format, with the structure: <Tree>

Requirements:

– The query should naturally lead to an answer that integrates the key-facts in a coherent, chronological narrative.

– Your query should be specific enough to address the contents in the key-fact tree.

– Make ONLY ONE query for the entire tree.

– Your query should be as concise as possible.

– Your query should NEVER mention anything about the key-fact tree.

The excerpt:
{excerpt}

The key-fact tree:
{key-fact tree}

Table 21: Narrative query generation prompt.

You will receive an excerpt of a novel and its corresponding summary, split into multiple sentences. Your task is to assess how faithfully each summary sentence represents the given excerpt's content. Faithfulness means the summary accurately reflects the information and meaning conveyed in the excerpt, without introducing unsupported claims or contradicting the source material.

When evaluating faithfulness, your decision should be one of these error categories:
-Out-of-article error: The summary introduces facts, opinions, or information not found in or reasonably inferrable from the text.
-Entity error: Incorrect reference to key subjects/objects (e.g., wrong names, numbers, pronouns).
-Relation error: Mistakes in semantic relationships (e.g., incorrect verbs, prepositions, adjectives).
-Sentence error: Multiple errors causing the entire sentence to contradict the text.
-No error: The summary statement aligns with the text's content.

Guidelines for evaluating abstractive summaries:
-Logical inference: If the summary makes reasonable conclusions based on information presented in the text, mark it as faithful (no error).
-Paraphrasing: Different word choices or sentence structures that preserve the original meaning are faithful.
-Generalization: Combining multiple specific details into a broader statement is faithful if accurate.
-Implicit information: Drawing on clearly implied information from context is faithful.

Instruction:
-Compare each summary sentence with the text.
-Provide a single, concise sentence explaining any factuality error, referencing specific elements from both texts.
-Classify the error category for each sentence.

Please provide your answer in JSON format as a list of dictionaries with keys "sentence", "reason", and "category" as follows:

```
[{
"sentence": "first sentence",
"reason": "your reason",
"category": "error category"
}, {
"sentence": "second sentence",
"reason": "your reason",
"category": "error category"
}]
```

Excerpt:
{excerpt}
Summary with {# sentences} sentences: {summary sentences}

Table 22: Summary evaluation prompt: fact-verification.

You will receive a summary and a set of key-facts for the same transcript. Your task is to assess if each key-fact is inferred from the summary.

Instruction:

- First, compare each key-fact with the summary.
- Second, check if the key-fact is inferred from the summary and then respond "Yes" or "No" for each key-fact. If "Yes", specify the line number(s) of the summary sentence(s) relevant to each key-fact.

Provide your answer in JSON format. The answer should be a list of dictionaries whose keys are "key-fact", "response", and "line number":

[{"key-fact": "first key-fact", "response": "Yes", "line number": [1]},
{"key-fact": "second key-fact", "response": "No", "line number": []},
{"key-fact": "third key-fact", "response": "Yes", "line number": [1, 2, 3]}]

Summary:
{summary}

{# key-facts} key-facts:
{key-fact list}

Table 23: Summary evaluation prompt: key-fact alignment.

You are an expert evaluator of novel chunks. Read the entire chunk, perform all reasoning silently, and output only the final JSON object described below—no other text.
DEFINITIONS
-Major fact (MF) = a distinct event, statement, or data point that introduces new information (e.g., "John confesses the theft," "The storm destroys the lighthouse"). Repeated or trivial details do not count.
-Internal connection (IC) = an explicit or implicit link between two MFs that shows foreshadowing, cause-effect, contrast, or resolution across sentences or paragraphs.
SCORING RUBRICS

1. HKF_validity — Validity of Hierarchical Key-Fact Tree Extraction Count the MFs and judge whether they can be arranged into a clear multi-level tree (root → branches → leaves).
   - Less than 25 MFs → Score 1-2
   - 25 - 35 MFs and at least a two-level hierarchy → Score 3-4
   - More than 35 MFs and a well-defined multi-level hierarchy → Score 5
2. Content_coherence - Evaluate the chunk using the following four signals:
   - A. Structural completeness (beginning → middle → end) : YES / NO
   - B. Abrupt transitions (N_abrupt) : 0, 1-2, ≥3
   - C. Unresolved references or unfinished plotlines : 0, 1, ≥2
   - D. Logical/temporal/spatial incoherences (N_incoherent) : 0, 1, ≥2

   Scoring
   - 1 point= A = NO and (N_abrupt ≥3 or N_unresolved ≥2 or N_incoherent ≥2)
   - 2 points= A = NO and at least two of B-D are in the "1-2 / 1" range
   - 3 points= A = YES but exactly one of B-D in the "1-2 / 1" range
   - 4 points= A = YES and at most one mild issue (B-D = 1-2 / 1); others 0
   - 5 points= A = YES and N_abrupt = N_unresolved = N_incoherent = 0
3. Cross_content_reasoning — Support for Cross-Content Reasoning Count the ICs that enable reasoning across different parts of the chunk.
   - Less than 8 ICs → Score 1-2
   - 8 - 16 ICs → Score 3-4
   - More than 16 ICs → Score 5

OUTPUT FORMAT:
{
"HKF_validity": <score from 1 to 5>,
"Content_coherence": <score from 1 to 5>,
"Cross_content_reasoning": <score from 1 to 5>,
"explanation": "Less than 25-word justification"
}
INPUT: {chunk}

Table 24: Chunk size validation prompt.