LiteraryQA: Towards Effective Evaluation of Long-document Narrative QA

Anonymous ACL submission

Abstract

Question Answering (QA) on narrative text poses a unique challenge for current systems, requiring a deep understanding of long, complex documents. However, the reliability of NarrativeQA, the most widely used benchmark in this domain, is hindered by noisy documents and flawed QA pairs. In this work, we introduce LiteraryQA, a high-quality subset of NarrativeQA focused on literary works. Using a human- and LLM-validated pipeline, we identify and correct low-quality QA samples while removing extraneous text from source documents. We then carry out a meta-evaluation of automatic metrics to reveal that all *n*-gram-based metrics have a low system-level correlation to human judgment, while LLM-as-a-Judge evaluations, even with small open-weights models, can strongly agree with the ranking identified by humans. Finally, we benchmark a set of long-context LLMs on LiteraryQA. We release our code and data at https://omitted.link.

1 Introduction

Question Answering (QA) has long been a core task in Natural Language Processing, supported by a large number of datasets that differ between themselves across several dimensions (Rogers et al., 2023): question type and objective (information-seeking or probing); answer format (extractive, multiple-choice or free-form); given context (quantity and modality of information). These datasets have enjoyed widespread adoption by the community, making up an important part of the evaluations of current models (Anthropic, 2024b; Yang et al., 2024; DeepSeek-AI, 2025). A particular QA setting is the one that focuses on whole books and narrative corpora. Books, and in general narrative text, express intricate sequences of events that unfold across a very long text, as recounted by characters or an external narrator (Piper et al., 2021). The need for the model to understand the underlying plot and link information from different parts of the (long) story makes it a challenging setting even for the latest Large Language Models (Pang et al., 2022).

Question: What happened to the cargo when it was near the coast of Western Australia?
Reference Answers: It spontaneously combusted. // It spontaneously combusted
Prediction: The cargo caught fire through spontaneous combustion and burned.
ROUGE-L: 0.0 **METEOR:** 0.55 **F1:** 0.0

Question: What is the name of the sun that the Centaurian Empire is built around?
Reference Answers: Centarus // Centarus
Prediction: Proxima Centaurus
ROUGE-L: 0.0 **METEOR:** 0.0 **F1:** 0.0

Question: What is the name of the leader of the reavers who attack the Argos when they first arrive in Kush?
Reference Answers: Belit // Belit
Prediction: Bêlit.
ROUGE-L: 0.0 **METEOR:** 0.0 **F1:** 0.0

Table 1: Illustrative example of the failure modes that automatic metrics incur when evaluating predictions on the original NarrativeQA.

In this context, NarrativeQA (Kočiský et al., 2018) is arguably the most established benchmark for the evaluation of long-context models' capabilities on English narrative text. It is included in many long-context benchmarks, notably ∞Bench(Zhang et al., 2024), L-Eval (An et al., 2024), LongBench (Bai et al., 2024a,b), HELMET (Yen et al., 2025). NarrativeQA was constructed by tasking crowd-sourced annotators to create a set of questions and answers from a *summary* of a book or a screenplay for which the source text is available. This annotation method prevents the model

from answering questions through shallow pattern-matching (McCoy et al., 2019), requiring it instead to synthesize the answer from the whole book. NarrativeQA is also very different from other free-form QA datasets, as a majority of its questions require understanding and differentiating narrative events and their relations (Mou et al., 2021).

This fundamental difference is reflected in model performance: systems on NarrativeQA achieve low scores compared to results on other free-form QA datasets. It is unclear if the reason for the low performance stems from the inherent difficulty of summary-derived questions on narrative text or the metrics used to evaluate predictions compared to the reference answers. Many automatic metrics have been used to evaluate performance on NarrativeQA: Kočiský et al. (2018) measured BLEU-1, BLEU-4 (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005) and ROUGE-L (Lin, 2004), while later evaluations adopted token-level F1 from extractive QA tasks (An et al., 2024) or tasked a model to evaluate if an answer is correct or not, a paradigm referred to as LLM-as-a-judge (Chen et al., 2019; Wang et al., 2023). Except for the latter, these metrics are based on the exact match of the words appearing in the reference answers and the prediction of a system and are susceptible to issues as presented in Table 1. As we show in our work, NarrativeQA also contains plenty of *noise*: there are instances of misaligned summaries and source texts, questions and answers that are grammatically and semantically incorrect with respect to the reference summary, and incorrect or malformed reference answers (a particular issue combined with the metrics used).

In order to address these issues, we propose LiteraryQA, a human- and LLM-validated subset of NarrativeQA focused only on literary works. Following recent literature that found LLMs to be capable annotators (Gilardi et al., 2023), we employ Claude 3.5 Haiku (Anthropic, 2024a) in a multi-step pipeline to first identify and subsequently correct questions and answers that are not acceptable according to a set of guidelines. We also carry out an extensive meta-analysis on which automatic metric to use, according to its agreement with human judgments, considering common n-gram-based metrics and LLM-as-a-judge solutions. We then benchmark current long-context LLMs, both open- and closed-weights, on both NarrativeQA and LiteraryQA, demonstrating the challenge it poses to current state-of-the-art systems.

2 Related Work

2.1 Narrative-Based QA

NarrativeQA (Kočiský et al., 2018) was an early effort to scale QA to entire books and movie scripts, with an average length of around 60,000 tokens. Despite its scale and free-form format, answers tend to be short and often paraphrased from summaries, leading to the inconsistencies pointed out in the introduction. Recent benchmarks have advanced long-context QA over narrative texts. QuALITY (Pang et al., 2022) presents multiple-choice questions over medium-length fiction text, averaging 5,159 tokens, which cannot be considered long-context in modern scenarios. NarrativeXL (Moskvichev and Mai, 2023) scales to 700k multiple-choice questions across 1,500 novels, but its reliance on structured questions limits its semantic depth. Contemporarily to our work, NovelQA (Wang et al., 2025) offers full-book contexts, free-form answers, and annotated supporting evidence, though it is restricted to 60 publicly available books.

2.2 Long-Document, Non-Narrative Tasks

Beyond narrative text, several benchmarks target long-document reasoning across diverse domains. The SCROLLS benchmark suite (Shaham et al., 2022) aggregates tasks such as summarization and QA over government reports (GovReport (Huang et al., 2021)), TV transcripts (SummScreenFD (Chen et al., 2022)), and meeting notes (QMSum (Zhong et al., 2021)). While valuable for studying long-context understanding, it does not investigate the specific narrative setting that we are interested in. Other QA-specific datasets include Qasper (Dasigi et al., 2021), which requires fine-grained fact retrieval from research papers, and ContractNLI (Koreeda and Manning, 2021), which frames contract understanding as a document-level entailment task. To probe deep retrieval and reasoning, RULER (Hsieh et al., 2024) introduces synthetic tasks over extremely long sequences, such as variable tracking and information chaining. While useful for stress-testing model capacities, synthetic tasks may not reflect the complexity of real-world documents.

Our work contributes a natural, generative QA benchmark over long-form narrative texts, designed to balance document-level scope, high-quality supervision, and flexible answer generation.

Step	# Docs	# QA samples
NarrativeQA	355	10557
– movies	−178	−5207
– plays	−20	−573
– other	−11	−234
– mismatched	−8	−320
Filtered	138	4223
– duplicates	/	−125
– double change	/	−308
– duplicates	/	−5
LiteraryQA	138	3785

Table 2: Breakdown of the changes in the test set due to our Data Refinement pipeline.

3 LiteraryQA

We hypothesize that the challenging aspect of NarrativeQA can be ascribed in part to inconsistencies in text quality and formatting, which include HTML artifacts and unrelated content, and to problematic QA samples containing wrong and misspelled reference answers or unanswerable questions. To mitigate this, we develop and validate a human-curated data refinement pipeline that, applied to NarrativeQA, creates an improved high-quality dataset, LiteraryQA.

3.1 Data Refinement Pipeline

Our pipeline is composed of two main phases: document-level and QA-level. In the following sections, we detail our filtering approach designed to produce a more balanced and narrative-representative dataset. Table 2 contains a breakdown of these steps and their impact on the dataset size.

3.1.1 Document-level phase

Our preliminary qualitative inspection of NarrativeQA reveals potential concerns regarding the pairing between book texts and their corresponding summaries, raising the need for a systematic alignment check. This is a fundamental issue since documents with misaligned summaries will have unanswerable questions, as the QA samples cannot be answered from an unrelated source text.

Moreover, NarrativeQA contains different document types, spanning novels, movie screenplays, poetry collections, theatrical plays, fairytales, and other types of text that do not strictly fit the conventional narrative text definition (Piper et al., 2021). This heterogeneity introduces substantial variance in the dataset in terms of format and style and distracts from the challenge of understanding a narrative plot. Thus we limit our focus on the book categories to develop a structurally and stylistically homogeneous high-quality narrative dataset.

Document filtering Our aim in this step is to filter out non-narrative text. As a first step, we manually annotate all documents in the book category[1] of the test set to identify and exclude mismatched documents, theatrical plays, and non-narrative texts. Our annotation process reveals that of the 177 books in the test set, there are 8 mismatched samples (4.5%), 20 theatrical plays (11.3%), and 11 non-narrative documents (6.2%), for a total of 39 documents (22%) that we subsequently removed from the dataset.

Text cleaning When examining the filtered documents, we discover that many documents contain text unrelated to the book which inflates document length and could confuse models. Such text includes HTML and Markdown strings, Project Gutenberg headers and footers, and legal license sections. To address this issue, we downloaded the original HTML versions of all documents through the URL included in the dataset[2]. Then, we isolated the narrative content of each book through an algorithm that heuristically extracts text within certain HTML tags. This algorithm was iteratively tested on the manually cleaned test set. Throughout this process, we prioritized recall over precision, ensuring that all narrative elements were preserved even at the cost of including occasional non-narrative content. We also fixed several encoding errors within the summaries, particularly regarding incorrect diacritical marks (e.g., `Ăvariste` instead of `Évariste`). On average, our cleaning procedure produces texts with 3k tokens less than the original texts in NarrativeQA (Figure 2 in Appendix A).

3.1.2 QA-level steps

Our second refinement phase focuses on individual question-answer pairs. Upon manual inspection, we found questions duplicated within the same book and issues with the grammatical and (especially) the semantic correctness of question-answer pairs. Given the high number of 4223 QA samples in the filtered NarrativeQA test split, we employed

[1] Movie documents are clearly categorized in NarrativeQA, making it possible to filter them easily.

[2] We used Gutenberg's mirrors as some of the original URL are no longer available.

an LLM to identify and correct QA issues, validating its outputs on a set of 10 documents spanning different genres (full list in Table 12, Appendix A). Full prompts can be seen in Appendix A (Tables 13 and 14).

Question Deduplication We implemented a simple ROUGE filtering mechanism to identify and remove duplicate questions. Questions with a ROUGE-L similarity score exceeding a given threshold were classified as duplicates. This process identified 125 (1.2%) duplicate questions, which were subsequently removed after manual validation[3].

Questions refinement After deduplication, we wanted to assess whether the questions were acceptable from both a grammatical and semantic point of view. We defined *malformed questions* as questions containing lexical errors, such as misspelled character names or "fat-fingers" typographical mistakes, as well as grammatical issues. *Ill-posed questions*, on the other hand, were defined as those containing false assumptions, misrepresenting facts presented in the summary, or being fundamentally unanswerable based on the available information. We tasked an LLM with identifying and correcting malformed and ill-posed questions. Since the original questions of NarrativeQA were generated from the summaries, we provided the summary as a reference to the LLM (Table 13).

Answers refinement We evaluated the reference answers following a similar approach. We applied identical criteria used for malformed questions to identify *malformed answers*, and we defined *invalid answers* as answers failing to be i) factually accurate, ii) complete in addressing all aspects of the question, or iii) directly relevant to the information requested. As in the previous step, we used an LLM to identify and correct these issues. We prompted it with the document summary, the question that had passed our previous refinement step (either because originally correct or subsequently corrected) and the reference answer to evaluate[4] (Table 14).

3.2 Evaluation of pipeline steps

When designing our pipeline evaluation strategy, we prioritized precision over recall to ensure that only only high-quality samples contribute to the final dataset. Each step in the pipeline processes the output from the previous step, creating a cascading refinement process.

In addition to the manual annotations for the document filtering steps, we also developed an automatic approach employing an LLM to classify the training and validation sets' samples. For each document, we prompted a Llama 3.1 8B Instruct model (Grattafiori et al., 2024), either with the Wikipedia page of the document or with the starting paragraphs of the text (Tables 15 to 17). We validated this approach on the test set, resulting in good performances (Table 11). We think this automated approach to be a good starting point for solving some issues present in the training and validation sets, which however we leave for future work due to their prohibitive scale for manual refinement.

For the QA-level steps, which pose a greater evaluation challenge compared to Document-level ones, we employed Claude 3.5 Haiku (Anthropic, 2024a). We evaluated the quality of the QA-level steps of our pipeline examining and annotating the outputs of the LLM. Except for the question deduplication step, for which we examined all identified duplicates, we performed our evaluation on a selected subset of 10 documents from the test set, comprising a total of 298 QA samples.

First, we annotate the original questions and reference answers using the same criteria established in our methodology. We computed the inter-annotator agreement using Cohen's Kappa coefficient, resulting in an average value of $\kappa = 0.83$, indicating excellent agreement (Table 3).

Then, to assess the quality of the LLM corrections, we evaluate the samples modified by the LLM. This qualitative analysis revealed that many false positives (instances classified as acceptable by human annotators but rejected and subsequently

Acceptability	κ	A1 %	A2 %
Questions	0.8384	91.61	92.62
Answer 1	0.8820	86.91	86.91
Answer 2	0.7836	78.52	77.85
Average	0.8346	85.68	85.79

Table 3: Inter-annotator agreement on the classification of 298 QA samples (10 documents) in the original NarrativeQA test set, before refinement. Values in columns A1 and A2 are the percentage of accepted modifications according to the annotators.

[3]We repeated this step at the end of the pipeline to remove new duplicates resulting from the LLM corrections.

[4]We evaluated each reference answers independently.

# Corrections	Acc. A1	Acc. A2	κ
1 (Questions)	0.96	0.95	0.88
1 (Answers)	0.96	0.96	1.00
2 (both)	0.65	0.65	1.00

Table 4: Analysis on the QA samples modified by Claude 3.5 Haiku of the annotated subset. We report the accuracy of the corrections computed on the annotators' judgments and the inter-annotator agreement with Cohen's Kappa.

Length in tokens	μ	σ
NarrativeQA questions	8.60	$\pm$3.30
LiteraryQA questions	8.62	$\pm$3.24
Only modified questions	9.76	$\pm$3.43
NarrativeQA answers	4.22	$\pm$3.63
LiteraryQA answers	4.33	$\pm$4.07
Only modified answers	6.86	$\pm$6.21

Table 5: Average length of questions and answers before (NarrativeQA) and after (LiteraryQA) our Data Refinement pipeline. We also report the length of only the modified samples.

corrected by the LLM) involved only minor modifications. These corrections typically produced paraphrases that preserved the essential meaning of the original samples.

We also observed that samples with corrections to either the question or the answer resulted in predominantly valid instances. However, samples with corrections to both the question and answer often contained compounding errors that resulted in invalid pairs. Based on this finding, we excluded all QA samples with double corrections from our dataset. We present the quantitative results of this last analysis on the modified samples in Table 4.

Table 5 presents the length distribution of question-answer pairs before and after processing. While modified answers show greater length variability, the mean length remains consistent across both versions.

4 Metrics Analysis

Our analysis aims to establish the quality of common *n*-gram-based metrics by measuring their system-level correlation with human annotations on LiteraryQA, as done by previous work on meta-evaluating QA systems (Chen et al., 2019). Following recent work that proved LLMs to be capable annotators (Gilardi et al., 2023), we also include LLM-as-a-judge as a metric to compare and contrast their agreement with *n*-gram-based measures. The LLM-as-a-judge paradigm involves querying an LLM with a question, two reference answers, a context, and a candidate answer to obtain a score generated by the LLM according to a rubric provided through system instructions. By measuring the system-level correlation between a metric and human judgment, we measure how closely the ranking that results from that metric aligns with the target human ranking.

Model	Size	Context
Qwen2.5-7B (Yang et al., 2025)	7B	1M
Qwen2.5-14B (Yang et al., 2025)	14B	1M
Llama3.1-8B (Grattafiori et al., 2024)	8B	128K
NExtLong (Gao et al., 2025)	8B	512K
GLM-4 (GLM et al., 2024)	9B	1M
Claude 3.5 Haiku (Anthropic, 2024a)	?	200K
Gemini 2.0 Flash Lite (Google, 2024)	?	1M

Table 6: Model tested in our experiments. All models are instruction finetuned. Text is truncated (preserving sentences) if longer than the model context window. The top 5 models are open-weights, the bottom 2 are only accessible via API. We focused on models with a context length larger or equal to 128k tokens.

We also compare the differences in system-level correlation of *n*-gram-based metrics and LLM-as-a-judge approaches on LiteraryQA against NarrativeQA: if a metric on the former correlates better with human judgment than the same metric on the latter, it would indicate that a large amount of noise from the dataset has been captured and corrected by our pipeline. We include metrics that have been used in literature to evaluate answers on NarrativeQA, namely: ROUGE-L (Lin, 2004), METEOR (Banerjee and Lavie, 2005), token-level F1 (F1) and exact-match (EM) taken from extractive QA (Yang et al., 2018). Regarding the LLM-as-a-judge setup, we use state-of-the-art LLMs accessed through APIs (GPT 4.1[5] and Claude 3.7 Sonnet[6]) and Prometheus 2 7B (Kim et al., 2024), an evaluator LM finetuned to provide direct assessments of candidate answer quality according to a user-defined rubric. Given our benchmark requires models to take as input the whole book for each question, we were restricted by our computational resources[7] to this choice of LLMs.

[5] `gpt-4.1-2025-04-14`

[6] `claude-3-7-sonnet-20250219`

[7] A single node equipped with 4 NVIDIA A100 GPUs

4.1 Experimental Setup

Human judgments In order to carry out our analysis, we must first collect human judgments of the quality of a response provided by an automatic system. We randomly sample $N = 100$ QA pairs from LiteraryQA's test split and input the question to each of the $M = 7$ systems in Table 6, obtaining their prediction. We repeat the same process for NarrativeQA, resulting in 1400 (QA, prediction) pairs collected across the two datasets. We then annotate the quality of each of the 1400 predictions according to human preference: given a question and two reference answers together with the automatic answer produced by a system, each human annotator is required to score the automatic answer in a range between 1 and 5, following the guidelines presented in Appendix Table 18. We also task each annotator with evaluating each automatic answer produced by a system with the added context of the summary of the book, considering it as a separate, summary-based setting.

Two of the authors of this paper annotated 1400 predictions each, with an estimated annotation time of 10 hours across multiple sessions. The inter-annotator agreement between them, measured through Kendall's τ correlation, amounts to 0.7876 for LiteraryQA and 0.8098 for NarrativeQA.

Correlation measurement We measure the system-level correlation r_{SYS} of each metric to evaluate if it can establish a preference ranking over a set of automatic systems that aligns with the one that emerges from human preferences. Following the notation established in recent literature (Deutch et al., 2024), the system-level correlation between a metric $\mathcal{X}$ and human judgment $\mathcal{Z}$ is calculated on the predictions provided by a set of M systems on N documents. Specifically,

$$r_{\text{SYS}} = \text{Corr}\left(\left\{\left(\frac{1}{N}\sum_{j}^{N} x_i^j, \frac{1}{N}\sum_{j}^{N} z_i^j\right)\right\}_{i=1}^{M}\right)$$

where x_i^j and z_i^j are the scores assigned by the metric $\mathcal{X}$ or human judgment $\mathcal{Z}$, respectively, to the output of the i-th system on the j-th sample, and Corr can be any measure of correlation, in our case Kendall's τ.[8]

[8]We chose Kendall's τ as we want to measure the correlation in ranking power of a metric compared to human scores. We computed it through its implementation in `scipy`.

Metric	LiteraryQA	NarrativeQA
EM	0.1099	-0.1014
F1	0.0872	-0.0277
ROUGE-L	0.1244	-0.0535
METEOR	0.1795	0.2690

Table 7: System-level correlation to human judgment according to Kendall's τ on LiteraryQA and NarrativeQA datasets.

Setting		Judge		
		Sonnet 3.7	GPT4.1	Prometheus2
Ref.	LQA	0.5213	0.6174	0.5246
	NQA	0.4807	0.5760	0.1900
Sum.	LQA	0.9750	0.6803	0.7802
	NQA	0.9078	0.7152	0.5268

Table 8: System-level correlation according to Kendall's τ between LLM-as-a-judge metrics and human judgments on the LiteraryQA (LQA) and NarrativeQA (NQA) datasets. 'Ref.': reference-based setting, 'Sum.': summary-based setting.

***N*-gram-based metrics** We calculate the correlation to human judgment of EM, F1, ROUGE-L and METEOR on two sets of $N \cdot M = 700$ samples from LiteraryQA and NarrativeQA, i.e. our human-annotated judgments.

LLM-as-a-judge We evaluate three LLMs to see how closely they correlate with human judgment, specifically GPT 4.1, Claude 3.7 Sonnet, and an open-weights option, Prometheus 2 (Kim et al., 2024). All models are initialized with a system prompt that describes the annotation required and provides an evaluation rubric (the prompt follows the same rubric defined in Appendix Table 18). Contrary to *n*-gram-based metrics, LLMs used as judges can also incorporate extra context when assigning the score of a predicted answer. We make use of this characteristic and devise two settings in which we measure system-level LLM-as-a-judge correlation: i) reference-based, where the LLM is given only the question, the reference answers, and the candidate answer; and ii) summary-based, where we also provide the model with the summary of the book, allowing it to disregard the reference answers when scoring a prediction if it can support it through the summary. We can carry out the latter setting as our human annotators also scored candidate answers with the added context of the summary of the book.

Model	Context	R-L	METEOR	EM	F1	Prom.
Llama3.1-8B	128k	0.3904	0.3669	0.1663	0.3785	2.981
Claude 3.5 Haiku	200k	0.2534	0.2988	0.0425	0.2818	**3.296**
NExtLong-8B	512K	**0.4155**	0.3617	**0.2015**	**0.4057**	2.836
Qwen2.5-7B	1M	0.3123	0.3311	0.0529	0.3033	2.843
GLM-4-9B	1M	0.3372	**0.3849**	0.0924	0.3319	3.149
Qwen2.5-14B	1M	0.3300	0.3632	0.0679	0.3216	3.123
Gemini 2.0 Flash Lite	1M	0.2299	0.2825	0.0158	0.2574	2.860

Table 9: Performance of seven models on LiteraryQA using six automatic n-gram-based metrics and Prometheus 2 as a Judge (ordered by context size). Best scores are in bold.

4.2 Results

***N*-gram-based metrics** Table 7 represents the system-level correlation according to Kendall's τ between four automatic metrics and human judgment. The results show poor correlations in general, especially in NarrativeQA: except METEOR, all other metrics are *negatively* correlated with our collected human judgments. This reflects the fragility that these metrics demonstrate concerning noise in the reference answers. Regarding METEOR, we hypothesize that its stemming and synonym-resolution features mitigate much of the noise that can be encountered in the original NarrativeQA.

On LiteraryQA, instead, *n*-gram metrics have a slightly positive correlation with human judgment, which indicates that it contains questions and reference answers of higher quality compared to NarrativeQA. This demonstrates the effectiveness of the pipeline we showcase in Section 3. Still, these correlations are very small and should not be trusted to reflect the real quality of a predicted answer. Especially, one should avoid assessing system performance on NarrativeQA using Exact Match, token-level F1, or ROUGE-L, and should place limited trust in METEOR.

LLM-as-a-judge In Table 8 we report the system-level correlation between LLM-as-a-judge systems and human judgment, both in the reference- and summary-based settings, on LiteraryQA and NarrativeQA.

When given only the question and reference answers as context to score the predicted answer (reference-based), each LLM achieves a moderately good correlation with human judgments on LiteraryQA, around 0.52 for Sonnet and Prometheus 2 and 0.61 for GPT 4.1. This indicates that our pipeline has succeeded in improving the quality of QA pairs. Notably, in this setting, there is a consistent gap across every LLM when comparing the correlation obtained on LiteraryQA and NarrativeQA, in favor of the former. This is especially visible in Prometheus 2: being only a 7B model could penalize it in handling noisy reference answers from NarrativeQA, but it reaches a correlation to human judgment comparable to Sonnet.

When considering the summary-based setting, all system-level correlations increase drastically, arriving at a maximum value of 0.975 for Sonnet. It is clear that letting the judge LLM consider the summary frees it from the restrictions of the reference answers, as the question could accept multiple valid answers within the context of the whole book represented by the summary.

We conclude that LLM-as-a-judge has a higher correlation than any *n*-gram-based metric on LiteraryQA, and advocate for its use in future work.

5 LLM benchmarking

In this section, we report on the performance of the selected models (Table 6) on the test set of LiteraryQA.

We evaluated the models in three distinct settings to isolate different aspects of model performance. In the open-book setting, models have access to the complete narrative text, testing their ability to locate and integrate relevant information across extensive narratives. We report the performance according to all metrics in the open-book setting in Table 9. According to the *n*-gram based metrics, the best performing model is NExtLong-8B, except for the METEOR metric that has GLM-4-9B as the winner. Notably, all open-weight LLMs surpass the two closed models; however, as we described in the previous section, a lower score in *n*-gram metrics does not necessarily imply a wrong output, but merely that the generated answer was *different* from the references. In fact, according

Dataset	# Docs	Claude 3.5 Haiku					
		R-1	R-2	R-L	METEOR	EM	F1
NarrativeQA (original)	177	0.2208	0.0771	0.2079	0.2743	0.0117	0.2380
NarrativeQA (filtered)	138	0.2305	0.0824	0.2174	0.2855	0.0122	0.2480
LiteraryQA	138	0.2655	0.1037	0.2534	0.2989	0.0425	0.2818
Dataset	# Docs	Gemini 2.0 Flash Lite					
		R-1	R-2	R-L	METEOR	EM	F1
NarrativeQA (original)	177	0.2307	0.0805	0.2201	0.2635	0.0237	0.2522
NarrativeQA (filtered)	138	0.2402	0.0850	0.2294	0.2745	0.0254	0.2612
LiteraryQA	138	0.2399	0.0863	0.2300	0.2827	0.0158	0.2575

Table 10: Performance increase of closed models across NarrativeQA, NarrativeQA filtered, and LiteraryQA.

to Prometheus 2 judgments on the predictions, the best performing model is a closed one, Claude 3.5 Haiku; however, the other closed model, Gemini 2.0 Flash Lite, is not among the top scoring.

In addition to the evaluation of the performance of the models on LiteraryQA, we establish comparative baselines on both the complete book section of NarrativeQA and the filtered subset containing only the 138 documents included in LiteraryQA. The results in Table 10 show that the predictions of closed-source models become progressively more similar to the reference answers following the steps of our pipeline, as measured by n-gram-based metrics. This hints at a reduction in noise in LiteraryQA compared to NarrativeQA, considering the widely recognized quality of both Claude 3.5 Haiku and Gemini 2.0 Flash Lite.

We also test the models in other two settings. In the closed-book setting, models are only given the title of the literary work, without any additional context, requiring them to rely entirely on their pretraining knowledge. This results in a lower overall performance, due to the absence of grounding context. Instead, in the summary setting, models are provided with the summary of the story instead of the full text. This represents the easiest setting, as summaries are brief (usually less than 500 words) and many reference answers appear almost verbatim within them. This difference in performance, according to Prometheus 2, is presented in Figure 1.

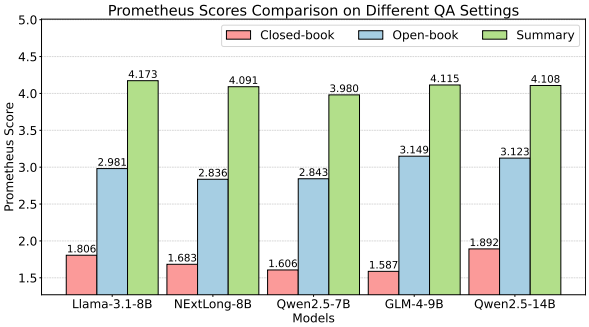

Figure 1: Prometheus-as-a-judge scores of the models across different settings.

6 Conclusions

In this work, we introduced LiteraryQA, an improved subset of NarrativeQA focused on literary works. Our extensive benchmarking demonstrates that the improved quality of LiteraryQA enables a more reliable and fair evaluation: tested models achieve higher scores in all metrics, and these metrics better reflect human judgments. We then carry out a meta-evaluation of automatic metrics, through which we identify METEOR as the most reliable among n-gram approaches, though LLM-as-a-judge systems demonstrated a significantly higher correlation with human judgments. However, despite these improvements, overall performance remains below those observed in other QA settings, indicating that LiteraryQA (and in general the "free-form" narrative QA setting) continues to represent a challenging benchmark for reading comprehension tasks.

Limitations

While LiteraryQA improves the quality and reliability of NarrativeQA, several limitations remain.

First, the refinement process partly relies on an LLM to support human validation, which introduces potential biases. Although human oversight mitigates this to some extent, the final dataset may still reflect these biases and subjective interpretations of question validity and answer correctness.

Second, our subset focuses exclusively on literary works, excluding other narrative forms such as movie scripts and theatrical plays. While this

design choice supports our goal of creating a reliable and homogeneous benchmark, the resulting dataset should not be taken as representative of the *full* narrative landscape.

Third, we did not include any retrieval-augmented generation (RAG) approaches in our evaluations, as our focus was on assessing the ability of the models to comprehend and reason over the entire narrative texts. Although RAG methods could potentially enhance performance by retrieving relevant context, they introduce additional complexity and issues that are orthogonal to our goal of evaluating narrative understanding. Retrieving small fragments can disrupt the narrative flow, which is critical for tasks where coherence and temporal structure are essential. Exploring RAG in this setting remains an interesting direction for future work.

Finally, LLM-as-a-judge evaluations, despite showing stronger alignment with human assessments, are i) costly to run at scale and ii) lack transparency, posing challenges for reproducibility and standardization. While small fine-tuned models like Prometheus have proven helpful even in this out-of-domain setting, we believe that models specifically fine-tuned for narrative evaluation could offer more accurate and cost-effective alternatives, especially if supported by structured knowledge or grounded in an external knowledge base, enabling more consistent and context-aware judgments.

References

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 *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14388–14411, Bangkok, Thailand. Association for Computational Linguistics.

Anthropic. 2024a. Claude 3.5 haiku. Accessed: May 16, 2025.

Anthropic. 2024b. Claude 3.7 sonnet. Accessed: May 16, 2025.

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. 2024a. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137, Bangkok, Thailand. Association for Computational Linguistics.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024b. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *arXiv preprint arXiv:2412.15204*.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Evaluating question answering evaluation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 119–124, Hong Kong, China. Association for Computational Linguistics.

Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. 2022. SummScreen: A dataset for abstractive screenplay summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8602–8615, Dublin, Ireland. Association for Computational Linguistics.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online. Association for Computational Linguistics.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Gilad Deutch, Nadav Magar, Tomer Natan, and Guy Dar. 2024. In-context learning and gradient descent revisited. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1017–1028, Mexico City, Mexico. Association for Computational Linguistics.

Chaochen Gao, Xing Wu, Zijia Lin, Debing Zhang, and Songlin Hu. 2025. Nextlong: Toward effective long-context training without long documents. *Preprint*, arXiv:2501.12766.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences of the United States of America*, 120.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, and 37 others. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

Google. 2024. Gemini 2.0 flash. Accessed: May 16, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. 2024. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436, Online. Association for Computational Linguistics.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353, Miami, Florida, USA. Association for Computational Linguistics.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Yuta Koreeda and Christopher Manning. 2021. ContractNLI: A dataset for document-level natural language inference for contracts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1907–1919, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Arsenii Moskvichev and Ky-Vinh Mai. 2023. NarrativeXL: a large-scale dataset for long-term memory models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15058–15072, Singapore. Association for Computational Linguistics.

Xiangyang Mou, Chenghao Yang, Mo Yu, Bingsheng Yao, Xiaoxiao Guo, Saloni Potdar, and Hui Su. 2021. Narrative question answering with cutting-edge open-domain QA techniques: A comprehensive study. *Transactions of the Association for Computational Linguistics*, 9:1032–1046.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel Bowman. 2022. QuALITY: Question answering with long input texts, yes! In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5336–5358, Seattle, United States. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Andrew Piper, Richard Jean So, and David Bamman. 2021. Narrative theory for computational narrative understanding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 298–311, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Anna Rogers, Matt Gardner, and Isabelle Augenstein. 2023. Qa dataset explosion: A taxonomy of nlp resources for question answering and reading comprehension. *ACM Comput. Surv.*, 55(10).

Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, and Omer Levy. 2022. SCROLLS: Standardized CompaRison over long language sequences. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 12007–12021, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Cunxiang Wang, Sirui Cheng, Qipeng Guo, Yuanhao Yue, Bowen Ding, Zhikun Xu, Yidong Wang, Xiangkun Hu, Zheng Zhang, and Yue Zhang. 2023. Evaluating open-QA evaluation. In *Proceedings of*

the 37th International Conference on Neural Information Processing Systems, NIPS '23, pages 77013–77042, Red Hook, NY, USA. Curran Associates Inc.

Cunxiang Wang, Ruoxi Ning, Boqi Pan, Tonghui Wu, Qipeng Guo, Cheng Deng, Guangsheng Bao, Xiangkun Hu, Zheng Zhang, Qian Wang, and Yue Zhang. 2025. NovelQA: Benchmarking question answering on documents exceeding 200k tokens. In *The Thirteenth International Conference on Learning Representations*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, Junyang Lin, Kai Dang, Kexin Yang, Le Yu, Mei Li, Minmin Sun, Qin Zhu, Rui Men, Tao He, and 9 others. 2025. Qwen2.5-1m technical report. *Preprint*, arXiv:2501.15383.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. 2025. Helmet: How to evaluate long-context language models effectively and thoroughly. In *International Conference on Learning Representations (ICLR)*.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2024. ∞Bench: Extending long context evaluation beyond 100K tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15262–15277, Bangkok, Thailand. Association for Computational Linguistics.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. QMSum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5905–5921, Online. Association for Computational Linguistics.

A Additional Results on LiteraryQA

In this section we present additional details on LiteraryQA and our filtering and cleaning approach. Figure 2 shows the length difference in tokens between the 138 shared documents in LiteraryQA and NarrativeQA.

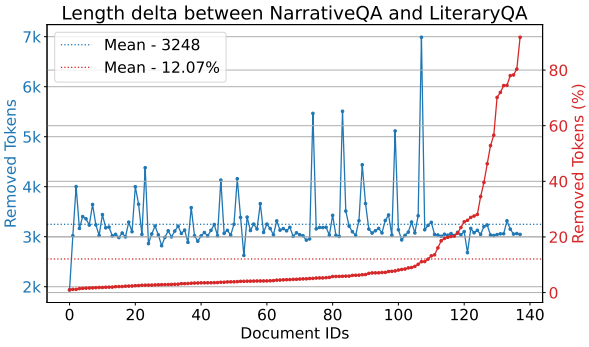

Figure 2: Token difference across books between NarrativeQA and LiteraryQA.

We also report the classification performance of the document-level steps of our pipeline carried out with Llama-3.1-8B-Instruct on the test set. This classification task is abstract enough to allow for perfect agreement.

Category	Precision	Recall	F1-score	κ
Mismatched	0.99	0.73	0.81	1.00
Plays	1.00	1.00	1.00	1.00
Non-narrative	0.86	0.79	0.82	1.00

Table 11: Classification performance of Llama-3.1-8B-Instruct on the documents categorized by the annotator. We also report the Inter-Annotator Agreement through Cohen's Kappa.

The complete list of the 10 annotated documents (298 QA samples) can be found in Table 12. We chose these books because they span over multiple genres, authors, styles, and languages (#2 and #10 were originally written in French, although we only work on the English versions).

1. *The Variable Man* (Philip K. Dick)
2. *Father Goriot* (Honoré de Balzac)
3. *Youth* (Joseph Conrad)
4. *A Portrait of the Artist as a Young Man* (Joyce)
5. *Tarzan of the Apes* (Edgar Rice Burroughs)
6. *The Vampyre* (John Polidori)
7. *Lothair* (Benjamin Disraeli)
8. *The House on the Borderland* (W. H. Hodgson)
9. *Uncle Silas* (Joseph S. Le Fanu)
10. *The Gods Are Athirst* (Anatole France)

Table 12: Subset of annotated documents for the evaluation of the data refinement pipeline.

In Figure 3 we further show how the models performance assessed through n-gram-based metrics decrease when the length difference (in tokens) between the generated answer and the reference answer increases.

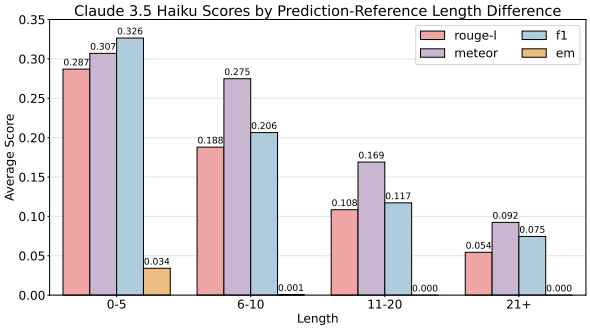

Figure 3: Metrics scores of Claude 3.5 Haiku grouped by prediction and reference answer length absolute difference.

Finally, we report the prompt we used throughout the data refinement pipeline Tables 13 to 17.

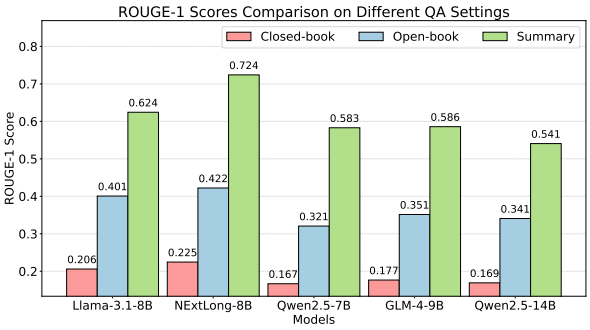

Figure 4: ROUGE-1 scores of the models across different settings.

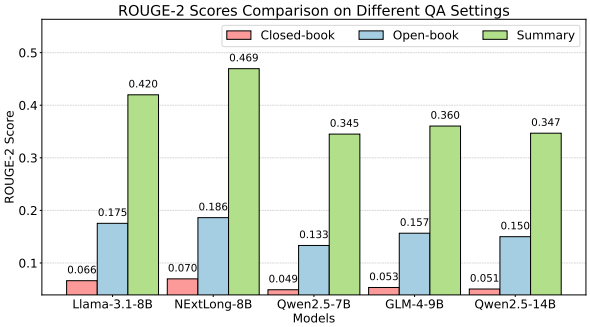

Figure 5: ROUGE-2 scores of the models across different settings.

B Licenses

We note that NarrativeQA is distributed under the Apache-2.0 License, which permits distributions

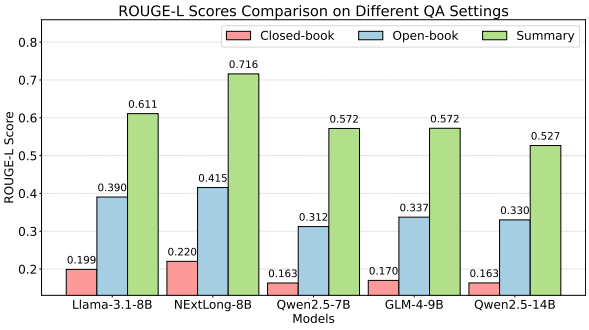

Figure 6: ROUGE-L scores of the models across different settings.

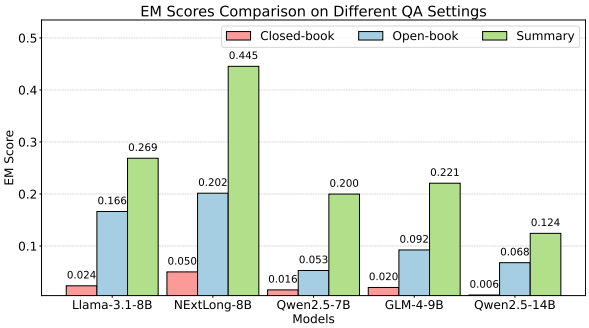

Figure 7: EM scores of the models across different settings.

and modifications. We adopt the same license when distributing LiteraryQA. Regarding models, we used closed-sourced options only to evaluate their performance, which complies with their Terms-of-Service (ToS). The only exception is Claude 3.5 Haiku, which we used through API in our data pipeline. According to their ToS, this is a legitimate use of their product as we are not developing a competing product and our dataset cannot be classified as harful.

System Prompt

Your task is to determine whether a question is not acceptable (grammatically malformed and/or ill-posed with respect to the reference summary). The question may refer to unusual, made-up, or technical words found in the reference summary – this is acceptable **only if they are spelled consistently**.

A question is **malformed** if it contains *any* common grammatical or misspellings errors, for example (non-exhaustive list):

- Misspelled words (including names and summary terms spelled inconsistently)

- Redundant or conflicting auxiliary verbs (e.g., 'was can not')

- Incorrect verb tense or verb form after auxiliaries (e.g., 'did played', 'does belives')

- Subject-verb disagreement (e.g., 'whose runs')

- Fat-finger errors (e.g., too many or missing whitespaces, letters inversions)

- Include proper contractions and possessives (e.g., 'who's', 'it's', 'he's')

- Faulty structure (e.g., missing auxiliaries, incorrect use of question words)

A question is **ill-posed** if (non-exhaustive list):

- It refers to something (an event, a character, etc.) that is not present in the summary

- It misunderstands the summmary or misrepresents its content

- It does not have a clear answer in the summary

A question is **well-posed** if it is clear, unambiguous, and has a specific answer in the summary.

If the question is not acceptable, rewrite it so to keep it as close as possible to the original question, while making it well-formed and well-posed. Respond in **JSON format** with exactly this structure:

{ "label": "acceptable" or "not acceptable", "correction": "..." // rewrite the question with the least amount of edits if it is not acceptable, otherwise write an empty string }

Only output this JSON. Do not add any commentary, do not explain your changes.

User Prompt

Reference summary: {summary}

Question: {question}

Is the question acceptable or not? Follow the rules above and respond with a JSON object as specified.

Table 13: System prompt used with Claude 3.5 Haiku to identify and correct invalid question samples.

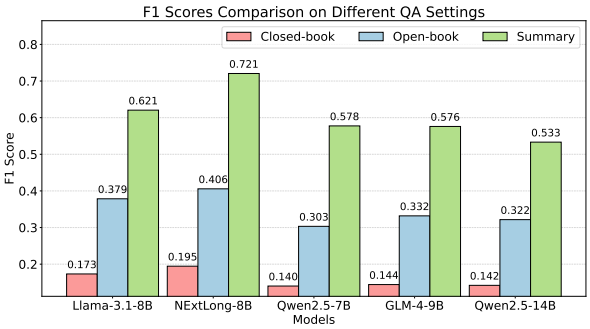

Figure 8: F1 scores of the models across different settings.

System Prompt

You are an English teacher evaluating answers about a narrative.

Your task is to determine whether an answer is acceptable (grammatically well-formed and valid).

The answer may refer to unusual, made-up, or technical words found in the reference summary — this is acceptable **only if they are spelled consistently**.

An answer is **malformed** if it contains *any* common grammatical or misspellings errors, for example (non-exhaustive list):

- Misspelled words (including names and summary terms spelled inconsistently)

- Redundant or conflicting auxiliary verbs (e.g., 'was can not')

- Incorrect verb tense or verb form after auxiliaries (e.g., 'did played', 'does belives')

- Fat-finger errors (e.g., too many or missing whitespaces, letters inversions)

- Include proper contractions and possessives (e.g., 'who's', 'it's', 'he's')

- Faulty structure (e.g., missing auxiliaries, incorrect use of question words)

A question is valid according to the following criteria:

- The answer must be factually correct, i.e. it must be supported by the reference summary, AND

- The answer must be complete (include all necessary entities for a complete response), AND

- The answer must provide a single precise response, not multiple possibilities or vague statements, AND

- The answer must be properly scoped, i.e. it must concisely address the question using the information found in the summary and without speculating or adding information.

Finally, the answer may consist of only one or two words — this is acceptable provided that there are no grammatical errors and the above criteria are met.

Respond in **JSON format** with exactly this structure:

{

"label": "acceptable" or "not acceptable",

"correction": "..." // if "not acceptable", rewrite the answer with the smallest amount of edits to make it acceptable, otherwise write an empty string

}

Only output this JSON. Do not add any commentary, do not explain your changes.

User Prompt

Reference summary: {summary}

Question: {question}

Answer: {answer}

Is the answer acceptable or not? Follow the rules above and respond with a JSON object as specified.

Table 14: System prompt used with Claude 3.5 Haiku to identify and correct invalid answers samples.

System Prompt

You are an expert literature analyst. Given a book description, you extract its category (novel or play). You rely **ONLY** on the text provided and do not make up any information.

User Prompt Description: {description} Is this a novel or a play? Reply with one word and do not include any other information.

Table 15: System prompt used with Llama-3.1-8B-Instruct to identify theatrical plays.

System Prompt

`You are an expert literature analyst. Given a book description, you extract its category (novel or non-fiction). You rely **ONLY** on the text provided and do not make up any information.`

User Prompt `Description: {description} Is this a novel or a non-fiction? Reply with one word and do not include any other information.`

Table 16: System prompt used with Llama-3.1-8B-Instruct to identify non-fiction books.

System Prompt

`You are an expert literature analyst. Given a book summary and its first paragraphs, you identify whether the two refer to the same literary work. You rely **ONLY** on the text provided and do not make up any information.`

User Prompt `Summary: {summary} Paragraphs: {paragraphs} Do they refer to the same literary work? Reply with yes/no and do not include any other information.`

Table 17: System prompt used with Llama-3.1-8B-Instruct to identify mismatched samples.

Score	Criteria
1	The response is completely wrong.
2	The output generally deviates from the original question, but there is some information related to the reference answer.
3	The response is partially correct, but the generated answer contains some errors, omits key information, or adds **major extra information** that cannot be validated (in the summary or the references, according to the setting).
4	The response is correct **but** it includes **minor** details that cannot be verified against the references or summary (according to setting)
5	Either exactly the same as one of the references, or a paraphrase of a reference that does not alter its meaning

Table 18: Likert Scale Grading Rubric