

LiteraryQA: Towards Effective Evaluation of Long-document Narrative QA

Tommaso Bonomo^{1*}, Luca Gioffré^{1*}, and Roberto Navigli^{1,2}

¹Sapienza NLP Group, Sapienza University of Rome ²Babelscape, Italy
{bonomo, gioffre, navigli}@diag.uniroma1.it

Abstract

Question Answering (QA) on narrative text poses a unique challenge to 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 clarify how systems should be evaluated on LiteraryQA. This analysis reveals 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-weight 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 github.com/SapienzaNLP/LiteraryQA.

1 Introduction

Question Answering (QA) has long been a core task in Natural Language Processing, supported by a large number of datasets that differ one from another across several dimensions (Rogers et al., 2023): question type and objective, answer format, and given context. These datasets have enjoyed widespread adoption by the community, making up an important part of the evaluations of current models (Team OLMo et al., 2024; Anthropic, 2024b; Qwen et al., 2025; 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 very long text, as outlined by characters or by an external narrator (Piper et al.,

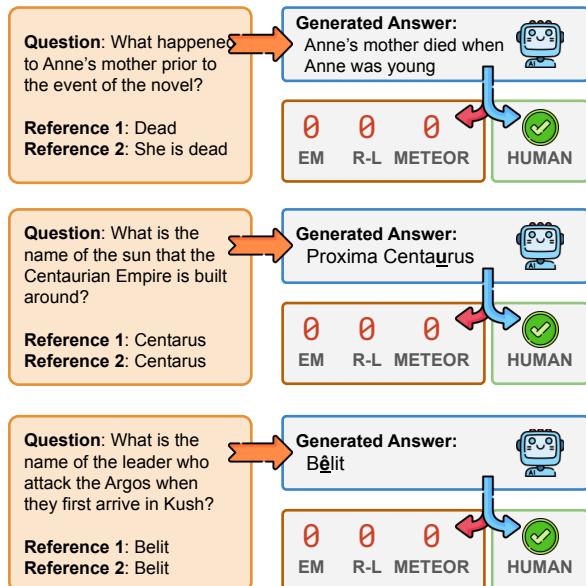


Figure 1: Illustrative example of the failure modes that automatic metrics incur when evaluating predictions on the original NarrativeQA.

2021). Even the latest Large Language Models (LLMs) find this setting challenging, as they have to understand the underlying plot and link information from different parts of the (long) story (Pang et al., 2022; Wang et al., 2025).

In this context, NarrativeQA (Kočiský et al., 2018) is arguably the most established benchmark for the evaluation of long-context models' abstractive QA 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., 2024, 2025), and HELMET (Yen et al., 2025). NarrativeQA is 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). As we show in our work, NarrativeQA also contains *noisy* content: there are instances of misaligned summaries and source

*Equal contribution.

texts, questions and answers that are grammatically and semantically incorrect with respect to the reference summary, and incorrect or malformed reference answers.

Moreover, model performance on NarrativeQA is comparatively low: it is unclear if the underlying reason stems from the inherent difficulty of abstractive questions on narrative text, the flaws present in the dataset, or the unsuitable metrics used to evaluate predictions with respect to 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 (Shaham et al., 2022; 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 n -gram-based metrics rely on an exact match between the words appearing in the reference answers and in the prediction of a system. Figure 1 exemplifies this kind of issue. These metrics can assign low scores, or even zero, to semantically correct responses that contain minor typos, missing diacritics, or are paraphrases of the reference answers. In contrast, human readers would easily recognize the correctness of such responses beyond surface-level variations. To the best of our knowledge, there has not been a comprehensive study to ascertain which automatic metric is most correlated with human judgment in the context of abstractive, narrative QA pairs.

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; Mohta 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 judgment, 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 these types of datasets pose 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 and a choice between multiple possible answers to a question, although it is restricted to 60 publicly available books. Unfortunately, to avoid data leakage, they do not publicly release the correct answers or the pieces of evidence required to correctly answer a question. We include a full analysis of the differences between our work and NovelQA in Appendix appendix A.

2.2 Long-document Resources

Various resources have annotated narrative text for tasks besides QA: LitBank (Bamman et al., 2019, 2020; Sims et al., 2019) provides annotations for Literary Event Detection, Literary Entities and Coreference Resolution on the first 2000 tokens of 100 literary works, while BOOKCOREF (Martinelli et al., 2025) provides silver- and gold-quality Coreference Resolution data on the full text of 53 books. 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 (Huang et al., 2021, GovReport), TV transcripts (Chen et al., 2022, SummScreenFD), and meeting notes (Zhong et al., 2021, QMSum). 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 (Kooreeda 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 they are 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.

2.3 Meta-evaluation of QA Metrics

Many works have carried out a meta-evaluation of metrics for free-form, generative QA. Kamaloo et al. (2023) and Wang et al. (2023) focused on factual, Wikipedia-based QA datasets, reporting contrasting findings regarding the correlation of metrics based on lexical overlap with human judgments. Instead, Chen et al. (2019) carry out a study in the narrative domain, where they find a moderately high correlation for n -gram- and neural-based metrics. However, their study analyses predictions from older models equipped with a copying mechanism, which makes them lexically similar to the references. Moreover, using the summary instead of the full text of a book makes the setting significantly easier than long-document narrative QA.

To the best of our knowledge, no prior work has conducted a comprehensive meta-evaluation of QA metrics in the narrative domain. We address this gap by providing the first systematic analysis of how standard evaluation metrics perform when applied to narrative question answering.

3 LiteraryQA

We hypothesize that the challenging aspect of NarrativeQA can be partly ascribed to inconsistencies in text quality and formatting (including HTML artifacts and unrelated content), and to problematic QA samples containing wrong and misspelled reference answers or unanswerable questions, which could artificially lower the performance of the systems. To mitigate these problems, we develop a human-curated data refinement pipeline that, applied to NarrativeQA, creates an improved high-quality dataset, LiteraryQA. Since our main goal is to provide a benchmark for narrative QA, we manually validate only the test set. We run the whole pipeline also on the train and validation sets in order to release the full dataset.

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.

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 may 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 to the book categories in order to develop a structurally and stylistically homogeneous narrative dataset.

Document Filtering First, we manually annotate all documents in the book category¹ of the test set so as to identify and exclude mismatched documents, theatrical plays, and non-narrative texts. Our annotation process reveals that, out 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 remove from the dataset, resulting in 138 documents kept in the filtered test set. In addition to the manual annotations, we also develop an automatic approach employing an LLM to classify the training and validation sets' samples. For each document, we prompt an LLM either with the Wikipedia page of the document or with the starting paragraphs of the text (Tables 16 to 18) and validate this approach on the test set.

Text Cleaning When examining the filtered documents, we discovered that many documents contain text unrelated to the book, which inflates document length and could confuse models. Such text

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

included 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 URLs included in the dataset². Then, we devised an algorithm to isolate the narrative content of each book through a set of heuristics. This algorithm was iteratively refined on the manually cleaned test set, with each iteration undergoing manual validation. 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. More details can be found in Appendix B.

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 produced documents of 3K tokens shorter than the original texts in NarrativeQA (Figure 4 in Appendix C).

3.1.2 QA-level Steps

Our second refinement phase focuses on individual question-answer pairs. Upon manual inspection, we found duplicated questions within the same book, grammatical errors, and issues with the semantic correctness of question-answer pairs. Given the high number (4223) of QA samples in the filtered NarrativeQA test split, we employed an LLM to identify and correct QA issues, validating its outputs on a set of 20 documents spanning different genres and authors. At the end of the pipeline, 1608 samples (38%) were modified. Appendix C lists these annotated documents, along with additional statistics on the test set and the full guidelines and prompts for QA refinement.

Question Deduplication We implemented a simple ROUGE filtering mechanism in which we identify and remove 125 (1.2%) duplicate questions (i.e., questions exceeding a ROUGE-L similarity threshold) after manual validation³.

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 typographical mistakes, as well

²We used Project Gutenberg’s mirrors as some of the original URLs are no longer available.

³We repeated this step at the end of the pipeline to remove new duplicates introduced with the LLM corrections.

as grammatical issues. *Ill-posed questions*, on the other hand, were defined as questions containing false assumptions, misrepresenting facts presented in the summary, or being fundamentally unanswerable based on the available information (Table 10, Appendix C).

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 14, Appendix C).

Answers Refinement We evaluated the reference answers following a similar approach. We applied the exact criteria used for malformed questions to identify *malformed answers*, and we defined *invalid answers* as answers failing to be either i) factually accurate, ii) complete in addressing all aspects of the question, or iii) directly relevant to the information requested (Table 10, Appendix C).

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 evaluate⁴ (Table 15, Appendix C).

3.2 Evaluation of Pipeline Steps

When designing our pipeline, we prioritize precision over recall to ensure that 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.

Regarding document-filtering step for the training and validation sets, we use a Llama 3.1 8B Instruct model (Grattafiori et al., 2024). We validate the model classification outputs on the test set, resulting in good performance (Table 11, Appendix C). We believe this automated approach offers a promising foundation for addressing issues in the training and validation datasets. However, we defer this work to future research due to the prohibitive scale of manual refinement required and focus only on the test set for the last part of the evaluation.

For the QA-level steps, which pose a greater evaluation challenge compared to the document-level phase, we employ Claude 3.5 Haiku (Anthropic, 2024a). We evaluate the quality of the QA-level steps of our pipeline by examining and

⁴We evaluated each reference answer independently.

Acceptability	κ	A1 %	A2 %
Questions	0.75	80.24	85.25
Answer #1	0.71	86.60	84.54
Answer #2	0.68	81.96	74.70
Average	0.71	82.93	81.50

Table 1: Inter-annotator agreement on the classification of 583 QA samples (20 documents) in the original NarrativeQA test set, before refinement. Values in columns A1 and A2 are the percentage of accepted modifications according to the respective annotators.

Corrections	κ	Acc. A1	Acc. A2
Only in Questions	0.88	0.95	0.96
Only Answers	1.00	0.96	0.96
Both	1.00	0.65	0.65

Table 2: Analysis on 176 QA samples of the annotated subset modified by Claude 3.5 Haiku. We report the accuracy of the corrections based on the judgments of the two annotators (A1 and A2), and the inter-annotator agreement with Cohen’s Kappa.

annotating the outputs of the LLM. Except for the question deduplication step, for which all identified duplicates are examined, we perform our evaluation on a selected subset of 20 documents from the test set, comprising a total of 583 QA samples (15%). These documents are listed in Table 12, Appendix C. Two of the authors performed the annotations necessary to assess pipeline quality, analyzing the question subset described above. The annotation process required approximately 30 hours total for each annotator.

Initially, annotators validate the original questions and two reference answers according to the criteria established in our methodology (Table 10, Appendix C). Inter-annotator agreement was measured using Cohen’s Kappa coefficient, yielding an average $\kappa = 0.83$, which indicates excellent agreement (Table 1).

Then, to assess the quality of the LLM corrections, we evaluated the samples of the annotated subset modified by the LLM. This qualitative analysis reveals that many false positives (instances classified as acceptable by human annotators but rejected and subsequently corrected by the LLM) involved only minor modifications. These corrections typically produce paraphrases that preserve the essential meaning of the original samples.

We also observe that samples with corrections to either the question or the answer result in predominantly correct instances. However, samples

Step	# Docs	# QAs
NarrativeQA (original)	355	10557
– Movies	-178	-5207
– Plays	-20	-573
– Other	-11	-234
– Mismatched	-8	-320
NarrativeQA (filtered)	138	4223
– QA duplicates	-	-125
– Double Correction	-	-308
– QA duplicates	-	-5
LiteraryQA	138	3785

Table 3: Breakdown of the impact of our data refinement pipeline on the LiteraryQA test set. The first document-level phase produces a ‘filtered’ version of NarrativeQA, whilst the QA-level refinement yields the final LiteraryQA test set (last row). The ‘QA duplicates’ step is run twice to remove duplicates introduced by the LLM, and ‘Double Correction’ stands for the samples where the LLM modified both the question and the answer, which we chose to exclude.

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

Table 4: Average length of questions and answers in the original NarrativeQA dataset compared to LiteraryQA after applying our Data Refinement pipeline. We also report statistics for only the modified samples.

with corrections to both the question and answer often contain compounding errors that render the QA pairs invalid. Based on this finding, we exclude all QA samples with double corrections from our dataset. We present the quantitative results of this last analysis on the 176 modified samples of the annotated subset in Table 2 and some examples in Table 13, Appendix C.

Table 3 presents an overall breakdown of the impact of each step of our pipeline on the test set of NarrativeQA. We also show the length distribution of question-answer pairs before and after processing in Table 4. While modified answers show greater length variability, the mean length remains consistent across both versions.

4 Metrics Analysis

We analyze several metrics on LiteraryQA, ranging from traditional n -gram-based approaches to neural-based methods, to measure their system-level correlation with human annotations on LiteraryQA. This step is necessary to establish the most suited evaluation metric for QA on narrative text, as previous approaches have simply adopted existing metrics from other QA domains. Following recent work that confirmed LLMs to be capable evaluators (Li et al., 2025; Gu et al., 2024), we also include LLM-as-a-judge as a metric to compare and contrast its agreement with n -gram- and neural-based measures. The LLM-as-a-judge paradigm involves querying an LLM with a question, multiple reference answers, a context, and a candidate answer, obtaining a score that the LLM generates according to a rubric provided through system instructions. By measuring the system-level correlation between a metric and human judgment, we assess how closely the metric’s ranking aligns with the human preferences.

We also compare the differences in system-level correlation of n -gram-based and neural-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). As our neural-based metric, we use BERTScore (Zhang et al., 2020), which provides a score between 0 and 1 that represents the semantical similarity of two pieces of text. Regarding the LLM-as-a-judge setup, we use state-of-the-art LLMs accessed through APIs (GPT 4.1⁵ and Claude 3.7 Sonnet⁶) 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. Our benchmark requires models to process entire books for each question, so our computational budget⁷ limited our LLM selection to the above models.

⁵gpt-4.1-2025-04-14

⁶claude-3-7-sonnet-20250219

⁷A single node equipped with 4 NVIDIA A100 GPUs.

Model	Size	Context
Qwen 2.5 (Yang et al., 2025)	7B	1M
Qwen 2.5 (Yang et al., 2025)	14B	1M
Llama 3.1 (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 5: Instruction-finetuned models tested, categorized as open-weight (first 5) or API-based (last 2). Inputs exceeding the context window are truncated.

4.1 Experimental Setup

Human judgments We collect human judgments on the quality of generated responses. We randomly sample $N = 500$ QA pairs from LiteraryQA’s test split and input the question to each of the $M = 7$ systems in Table 5, obtaining their prediction. We repeat the same process for NarrativeQA, resulting in 7000 (QA, prediction) pairs collected across the two datasets. Annotators then evaluate the quality of each prediction according to the rubric in Table 19, Appendix C. Given a question, its two reference answers, and an answer produced by a system, each annotator is required to score the automatic answer in a range between 1 and 5 in this **reference-based setting** following the rubric. Annotators also evaluate each automatic answer in a separate **summary-based setting**, where they have access to the book summary as additional context.

Two of the authors of this paper annotated 7000 predictions each, with an estimated annotation time of 50 hours across multiple sessions. The inter-annotator agreement between them, measured through Kendall’s τ correlation, is 0.7876 for LiteraryQA and 0.8098 for NarrativeQA.

Correlation measurement We compute the system-level correlation (r) to see how well a metric’s ranking of systems aligns with human judgments. The calculation uses outputs from M systems on N documents, following the notation of Deutch et al. (2024). Specifically,

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

Metric	NarrativeQA		LiteraryQA	
	Reference-based	Summary-based	Reference-based	Summary-based
EM	0.0325 [-0.48, 0.48]	-	0.0614 [-0.24, 0.43]	-
F1	0.0328 [-0.48, 0.48]	-	0.0574 [-0.24, 0.39]	-
ROUGE-L	0.0291 [0.48, 0.42]	-	0.0580 [-0.24, 0.39]	-
METEOR	<u>0.1519</u> [-0.33, 0.62]	-	<u>0.4444</u> [0.14, 0.81]	-
BERTScore	-0.0477 [-0.52, 0.39]	-	0.0677 [-0.24, 0.43]	-
Prometheus 2 7B	0.2195 [-0.24, 0.68]	0.3155 [-0.05, 0.68]	0.4499 [-0.05, 0.88]	0.6881 [0.33, 0.98]
Sonnet 3.7	0.3114 [-0.14, 0.78]	0.5829 [0.20, 0.90]	0.3651 [-0.14, 0.81]	0.5243 [0.09, 0.90]
GPT 4.1	0.3517 [-0.10, 0.81]	0.5080 [0.09, 0.90]	0.3282 [-0.14, 0.71]	0.5593 [0.20, 0.90]

Table 6: System-level Kendall’s τ correlation with human judgments on NarrativeQA and LiteraryQA. Bold and underline mark the best LLM-as-a-Judge and n-gram metrics, respectively. 95% confidence intervals are in brackets. The summary-based setting is exclusive to the LLM-as-a-Judge method.

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

Regarding n -gram-based metrics, we calculate the correlation of EM, F1, ROUGE-L and METEOR on two sets of $N \cdot M = 3500$ samples, one from LiteraryQA and one from NarrativeQA, which are our human-annotated judgments. As a neural-based metric, we calculate the correlation of BERTScore equipped with DeBERTa-XLARGE (He et al., 2021) finetuned for NLI.

For the LLM-as-a-judge paradigm, we evaluate three LLMs to see how closely they correlate with human judgment, specifically GPT 4.1, Claude 3.7 Sonnet, and an open-weight option, Prometheus 2 7B. 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 Table 19, Appendix C). Contrary to n -gram-based metrics, LLMs can also incorporate extra context when assigning the score of a predicted answer. We make use of this characteristic, as we did during the annotation process, and devise two settings in which we measure the system-level LLM-as-a-judge correlation: **reference-based**, where the LLM is given only the question, the reference answers, and the candidate answer; and **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.

⁸We 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`.

4.2 Results

Table 6 presents the system-level correlations between our chosen metrics and human judgment, measured using Kendall’s τ . The metrics include four n -gram-based measures, one neural-based metric, and three LLMs used as judges. In this section, we examine each category of metrics to discuss the results.

N -gram-based metrics The results of n -gram-based metrics show poor correlations in general, especially in NarrativeQA: except for METEOR, all other metrics are poorly 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 (except METEOR) have a slightly better 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. METEOR actually achieves a good system-level correlation with human judgment of 0.44, indicating that it should be preferred among all n -gram-based metrics.

LLM-as-a-judge LLMs used as judges show the highest correlation with human judgments. 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 0.36 for Sonnet, 0.32 for GPT 4.1 and

Model	Context	R-L	METEOR	EM	F1	BERTScore	Prometheus 2
Llama3.1-8B	128K	0.3904	0.3669	0.1663	0.3785	0.7105	2.981
NExtLong-8B	512K	0.4155	0.3617	0.2015	0.4057	0.7195	2.836
Qwen2.5-7B	1M	0.3123	0.3311	0.0529	0.3033	0.6689	2.843
GLM-4-9B	1M	0.3372	0.3849	0.0924	0.3319	0.6705	3.149
Qwen2.5-14B	1M	0.3300	0.3632	0.0679	0.3216	0.6764	3.123
Claude 3.5 Haiku	200K	0.2534	0.2988	0.0425	0.2818	0.6569	3.296
Gemini2-Flash-L	1M	0.2299	0.2825	0.0158	0.2574	0.6440	2.860

Table 7: Performance of seven open-weight and closed-source models on LiteraryQA using four automatic n -gram-based metrics, a neural-based metric (BERTScore) and an LLM-as-a-Judge (Prometheus 2). Best scores are in bold.

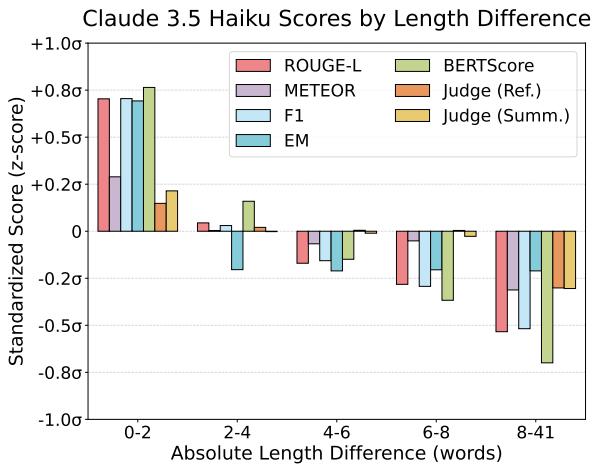


Figure 2: Standardized metrics of Claude 3.5 Haiku’s generated responses grouped by the absolute length difference between the prediction and the reference answers. Each bin contains an equal number of samples.

0.45 for Prometheus 2. This indicates that our pipeline has succeeded in improving the quality of QA pairs. Notably, every LLM shows *consistently higher correlations* on LiteraryQA compared to NarrativeQA in this setting. This gap is particularly evident in Prometheus 2, with an increase of 25 percentage points. It is a stark contrast with NarrativeQA, where we hypothesize that its smaller 7B parameter size may hinder its ability to handle the dataset’s noisy reference answers. Instead, when provided with LiteraryQA’s clean reference answers, it reaches a higher correlation with human judgments than Sonnet 3.7 or GPT 4.1.

When considering the summary-based setting, all system-level correlations increase drastically, arriving at a maximum value of 0.68 for Prometheus 2. 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, as represented by the summary. In this setting as

well, correlations on LiteraryQA are higher than on NarrativeQA, further confirming the effectiveness of our refinement process.

We conclude that summary-based LLM-as-a-judge has a higher correlation than any of the analyzed n -gram- or neural-based metrics on LiteraryQA, and advocate for its use in future work.

Finally, in Figure 2 we show how the length difference in words between the generated answers and the references impacts the (standardized⁹) metric scores. Most metrics show greater score variability at extreme length differences. BERTScore, ROUGE-L, and F1 are particularly sensitive, assigning higher scores for small length mismatches and lower scores for large ones. EM scores remain consistently low due to its binary nature. Among n -gram metrics, only METEOR maintains stable performance across all bins, behaving similarly to Prometheus 2 7B, which has the lowest variability and highest correlation with human judgments.

5 LLM benchmarking

In this section, we report the performance on the test set of LiteraryQA of the models in Table 5 across three distinct settings. 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 7. Three out of the four n -gram-based metrics (ROUGE-L, EM, F1) rank the systems in the same order, with NExtLong-8B achieving higher scores than all other models, including closed-source ones. Perhaps surprisingly, BERTScore follows the same trend as these metrics. As described in the previous section, we note that a lower score in n -gram-based metrics does not

⁹We standardize a metric’s scores by subtracting its the mean and dividing by its the standard deviation.

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 8: Performance increase of closed models responses (Claude 3.5 Haiku and Gemini 2.0 Flash Lite) evaluated through n -gram-based metrics across NarrativeQA, NarrativeQA Filtered, and LiteraryQA.

necessarily imply a wrong output, but merely that the generated answer was *syntactically different* from the references. METEOR, which was identified as the best n -gram-based metric, identifies GLM-4-9B as the best model. The closed-source models are consistently ranked below their counterparts, which contrast with a comparative analysis of a sample of all models’ answers. In fact, according to Prometheus 2 judgments on a sample of 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 ones (scores reported as “open-book” in Figure 3).

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 8 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 suggests a reduction in noise in LiteraryQA compared to NarrativeQA.

We also test the models in two other settings. In the **closed-book** setting, models receive only the literary work’s title, without additional context, requiring them to rely entirely on their pre-training knowledge and limiting their overall performance. Instead, in the **summary setting**, models receive story summaries. This is the easiest setting, as summaries are brief (typically <500 words) and many answers appear nearly verbatim. Performance differences across settings, according to Prometheus 2, are shown in Figure 3.

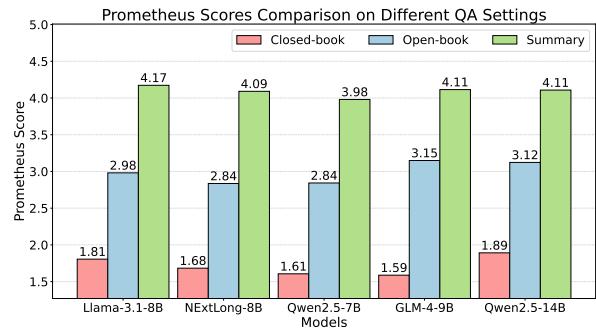


Figure 3: Prometheus-as-a-judge scores of the models across different settings.

6 Conclusions

In this work, we introduced LiteraryQA, a human- and LLM-improved subset of NarrativeQA focused exclusively on literary works, addressing the limitations of existing narrative QA datasets. The resulting dataset exhibits improved question clarity, reduced ambiguity, and better alignment between questions and reference answers. Our extensive benchmarking demonstrates that the higher 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 when provided with the book summaries. However, despite these improvements, overall performance remains below that observed in other QA settings, indicating that LiteraryQA (and in general the open-ended 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 relies partly on an LLM to support human validation, which can introduce 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.

Acknowledgments

The authors acknowledge the support of the PNRR MUR project PE0000013-FAIR.  

The authors gratefully acknowledge the support of the AI Factory IT4LIA project and the CINECA award IsCc8_CRAFT under the ISCRA initiative

for granting access to high-performance computing resources. Finally, the authors would also like to thank Alessandro Scirè for his insights and contributions in the initial phase of the project.

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. 2024. [LongBench: A bilingual, multi-task 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. 2025. [LongBench v2: Towards deeper understanding and reasoning on realistic long-context multitasks](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3639–3664, Vienna, Austria. Association for Computational Linguistics.

David Bamman, Olivia Lewke, and Anya Mansoor. 2020. [An annotated dataset of coreference in English literature](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 44–54, Marseille, France. European Language Resources Association.

David Bamman, Sejal Popat, and Sheng Shen. 2019. [An annotated dataset of literary entities](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2138–2144, Minneapolis, Minnesota. Association for Computational Linguistics.

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, Chen-hui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Han-lin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jia-dai 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.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Yuanzhuo Wang, and Jian Guo. 2024. [A Survey on LLM-as-a-Judge](#). *CoRR*, abs/2411.15594.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. [Deberta: decoding-Enhanced Bert with Disentangled Attention](#). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

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.

Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. 2023. [Evaluating open-domain question answering in the era of large language models](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5591–5606, Toronto, Canada. 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.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu,

Kai Shu, Lu Cheng, and Huan Liu. 2025. *From Generation to Judgment: Opportunities and Challenges of LLM-as-a-judge*.

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.

Giuliano Martinelli, Tommaso Bonomo, Pere-Lluís Huguet Cabot, and Roberto Navigli. 2025. *BOOK-COREF: Coreference resolution at book scale*. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24526–24544, Vienna, Austria. Association for Computational Linguistics.

Jay Mohta, Kenan Ak, Yan Xu, and Mingwei Shen. 2023. *Are large language models good annotators?* In *Proceedings on "I Can't Believe It's Not Better: Failure Modes in the Age of Foundation Models" at NeurIPS 2023 Workshops*, volume 239 of *Proceedings of Machine Learning Research*, pages 38–48. PMLR.

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.

Qwen, :, 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, and 25 others. 2025. *Qwen2.5 technical report*. *Preprint*, arXiv:2412.15115.

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.

Matthew Sims, Jong Ho Park, and David Bamman. 2019. *Literary event detection*. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3623–3634, Florence, Italy. Association for Computational Linguistics.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneweld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, and 21 others. 2024. *2 OLMo 2 Furious*. *Preprint*, arXiv:2501.00656.

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, 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)*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. [BERTScore: Evaluating Text Generation with BERT](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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 Comparison with NovelQA

To better contextualize LiteraryQA’s contributions, we compare it with NovelQA (Wang et al., 2025), a contemporary work that introduces a benchmark for long-form reading comprehension. While both datasets target literary understanding, they differ significantly in their design philosophy and evaluation focus (Table 9).

Aspect	LiteraryQA	NovelQA
Length ($\mu \pm \sigma$)	$74K \pm 59K$	$179K \pm 145K$
# Documents	138 (test set)	89
QA source	Summaries	Paragraphs
Availability	Full	Partial

Table 9: Breakdown of the main differences between LiteraryQA and NovelQA. We take into account only the test set of LiteraryQA and the public domain books of NovelQA for the length stats in this Table.

NovelQA features documents with an average length approximately 2.5 times longer than those

in LiteraryQA. However, LiteraryQA contains 50% more documents than NovelQA, providing broader coverage across different literary works. Moreover, the datasets employ fundamentally different annotation approaches. NovelQA generates QA pairs through a combination of templates and free-form generation, allowing for systematic coverage but potentially limiting question diversity and introducing biases. In contrast, LiteraryQA adopts a fully manual QA generation process (inherited from NarrativeQA) followed by an automatic QA pairs refinement (detailed in Section 3), which is fundamental in order to obtain more nuanced and varied questions.

Most significantly, the datasets test different aspects of reading comprehension. NovelQA focuses on detailed questions supported by specific annotated evidence paragraphs, emphasizing precise information extraction and evidence-based reasoning. On the other hand, LiteraryQA derives its questions from book summaries, requiring models to i) synthesize information across multiple passages, ii) avoid full reliance on surface-level pattern matching, and iii) demonstrate broader narrative understanding. This distinction makes LiteraryQA particularly suited for evaluating synthesis and reasoning capabilities rather than fine-grained information extraction.

Finally, NovelQA’s answers and evidence paragraphs are not publicly released to prevent data contamination in model training, an increasingly legitimate concern for keeping the benchmark integrity. Moreover, approximately one-fourth of the dataset is composed of copyrighted books, which are not freely accessible. However, this design choice also limits researchers’ ability to conduct experiments, detailed error analysis, and iteration on evaluation methods. In contrast, LiteraryQA, following the open-access approach of its parent dataset NarrativeQA, prioritizes transparency and reproducibility, enabling more comprehensive model analysis and community engagement.

These complementary approaches suggest that both datasets serve important but distinct roles in evaluating literary reading comprehension, with LiteraryQA particularly well-suited for assessing synthesis and narrative understanding capabilities.

B Text Extraction Algorithm Details

Algorithm 1 shows how we parse the raw HTML Gutenberg data into clean text documents. We only

committed changes to this extraction algorithm when they maintained complete preservation of all narrative samples. This conservative approach guaranteed that no valuable narrative content was inadvertently removed.

The algorithm takes as input an HTML document H and a set of parameters θ . The θ parameters contain lists and mappings of HTML tags categorized by their processing needs, such as tags to decompose, unwrap, replace, or remove attributes from. We use the Python library BeautifulSoup¹⁰ for parsing the HTML into a tree structure S . Each tag t in the tree S is processed based on its category. We define the categories for tags to keep or remove after comparing the source HTML and the rendered page, defining the following cases:

- If t is in the Decompose list, it is removed entirely from S .
- If t is in the Unwrap list, it is replaced by its children nodes in S , effectively removing the tag but preserving its content.
- If t is in the Replace mapping, the tag t is replaced by the corresponding substitute defined in θ (e.g., to keep it with a specific formatting).
- If t is in the Remove attributes list, specified attributes are removed from t without deleting the tag itself.

After modifying the HTML structure in this way, the algorithm extracts text content only from the modified tree structure S and normalizes the text by removing multiple spaces and line breaks and filtering out empty or invalid strings.

T is then processed by Algorithm 2, which filters remaining noise that passed the initial step. The algorithm processes each line using RegEx patterns to determine whether to skip the line, keep it, reinitialize the clean text buffer, or terminate processing. Since terminating sequences are more difficult to handle reliably than starting patterns, we introduce a parameter α to control when the algorithm may terminate. Specifically, when an ending pattern (e.g., "THE END.") is detected, the algorithm stops only if it occurs in the final portion of the text; otherwise, it retains the line and continues processing. We set $\alpha = 0.9$ in our experiments, allowing termination only when such patterns appear in the last 10% of the text¹¹.

¹⁰<https://beautiful-soup-4.readthedocs.io>

¹¹Our implementation also allows certain specific, unam-

Algorithm 1: Extract structured text from HTML.

Input: HTML document H , options θ

Output: Raw text T

$S \leftarrow \text{parse}(H);$

for tag $t \in S$ **do**

if $t \in \theta_{\text{decompose}}$ **then**

$S \leftarrow S \setminus \{t\};$

if $t \in \theta_{\text{unwrap}}$ **then**

$S \leftarrow S \cup \text{children}(t);$

$S \leftarrow S \setminus \{t\};$

if $t \in \theta_{\text{replace}}$ **then**

$t \leftarrow \theta_{\text{replace}}(t);$

if $t \in \theta_{\text{remove_attrs}}$ **then**

$t \leftarrow t \setminus \theta_{\text{remove_attrs}}(t);$

$T \leftarrow \emptyset;$

for tag $t \in S$ **do**

$s \leftarrow \text{normalize}(t_{\text{text}});$

if $s \neq \emptyset \wedge \text{valid}$ **then**

$T \leftarrow T \cup \{s\};$

return T

C Additional Results on LiteraryQA

In this Section we present additional details on final version of the dataset, LiteraryQA. Figure 4 shows the length difference in tokens between the 138 shared documents in LiteraryQA and NarrativeQA. Our cleaning procedure removes an average of 3K tokens per document, representing 12% of the original text.

Table 11 presents the classification performance of our pipeline’s document-level steps using Llama-3.1-8B-Instruct on the test set. The clear-cut nature of this classification task enables perfect agreement between evaluators.

The complete list of the 20 annotated documents (583 QA samples) can be found in Table 12. We chose these books because they span over multiple genres, authors, styles, and languages (a few were originally written in French, although we only work on the English versions).

Figures 5 and 6 show the most represented authors and the publication years within the LiteraryQA test set, respectively.

Finally, we report the prompt we used throughout the data refinement pipeline in Tables 14 to 18,

biguous patterns to terminate the process regardless of the α threshold.

Algorithm 2: Clean extracted text with start, end, and skip patterns.

Input: Raw text T , patterns Φ , threshold α
Output: Cleaned text lines C

$$C \leftarrow [];$$

$$L \leftarrow \text{split}(T);$$

$$\text{for } \text{line } l_i \in L \text{ do}$$

$$\quad \text{if } l_i \in \Phi_{\text{start}} \text{ then}$$

$$\quad \quad C \leftarrow [];$$

$$\quad \text{else if } l_i \in \Phi_{\text{skip}} \text{ then}$$

$$\quad \quad \text{continue;}$$

$$\quad \text{else if } l_i \in \Phi_{\text{end}} \wedge i \geq \alpha \cdot |L| \text{ then}$$

$$\quad \quad \text{break;}$$

$$\quad \text{else}$$

$$\quad \quad C \leftarrow C \cup \{l_i\};$$

$$\text{return } C$$

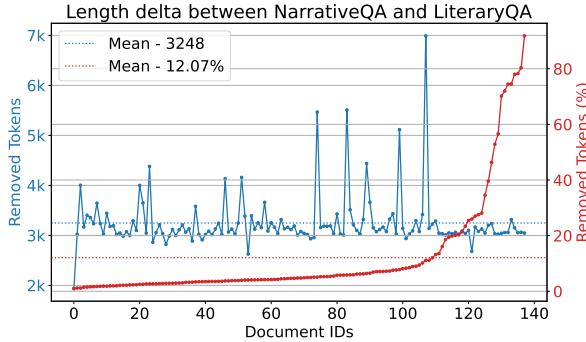


Figure 4: Comparison of document lengths between NarrativeQA and LiteraryQA for the same books. Blue line indicates absolute token differences; red line shows percentage differences.

and some examples of the corrections made by Claude 3.5 Haiku in Table 13.

D Licenses

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



Figure 5: List of the most represented authors in LiteraryQA (test set).

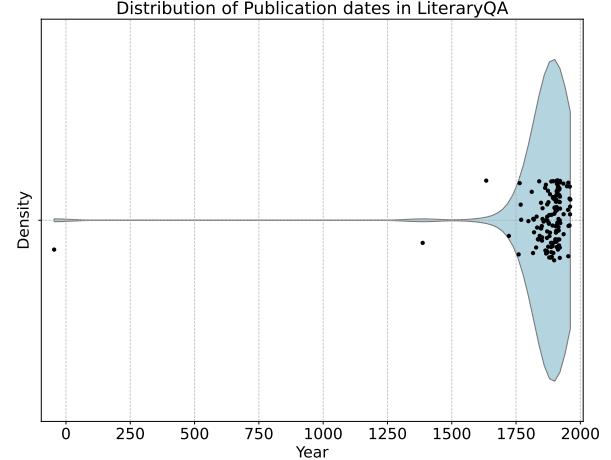


Figure 6: Publication year distribution in LiteraryQA (test set).

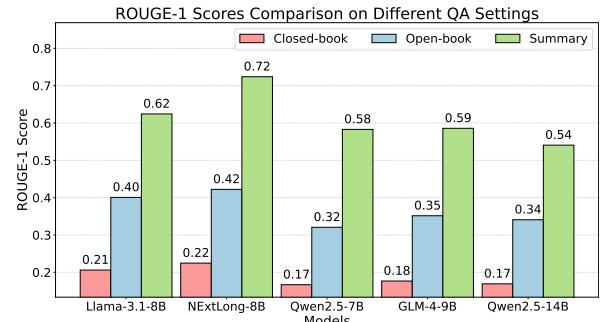


Figure 7: ROUGE-1 scores of the models across different settings.

Question Criteria
An acceptable question must:
<ol style="list-style-type: none"> 1. Be grammatically correct in relation to the summary 2. Be unambiguous and have a clear answer within the summary context 3. Be answerable using only information present in the summary
Answer Criteria
An acceptable answer must:
<ol style="list-style-type: none"> 1. Be grammatically correct, specifically: <ol style="list-style-type: none"> (a) Free of typos (b) Free of misspellings (c) Free of mistakes due to accidental key presses (d) Include proper contractions and possessives (e.g., <i>don't</i>, <i>wasn't</i>, <i>John's</i>) (e) Use correct subject-verb agreement 2. Be factually correct and complete according to the summary: <ol style="list-style-type: none"> (a) Contain no information contradicting the summary (b) Include all relevant entities (people, locations, dates, etc.) when applicable (c) Provide a single, precise response (not multiple possibilities or vague statements) 3. Be properly scoped: <ol style="list-style-type: none"> (a) Include only information found in the summary (b) Be concise while addressing the full question (c) Avoid speculation beyond what's stated in the summary

Table 10: Guidelines used throughout the annotation process.

Category	Pr	Re	F1	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 showing Precision (Pr), Recall (Re) and F1-score (F1). We also report the Inter-Annotator Agreement through Cohen's Kappa.

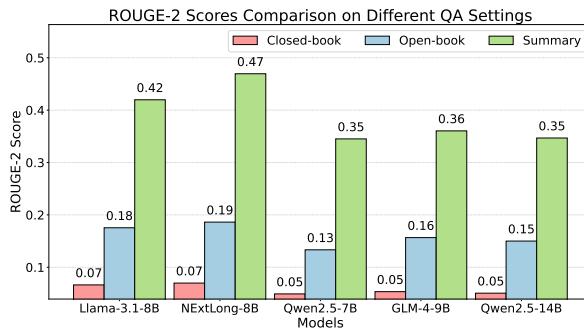


Figure 8: ROUGE-2 scores of the models across different settings.

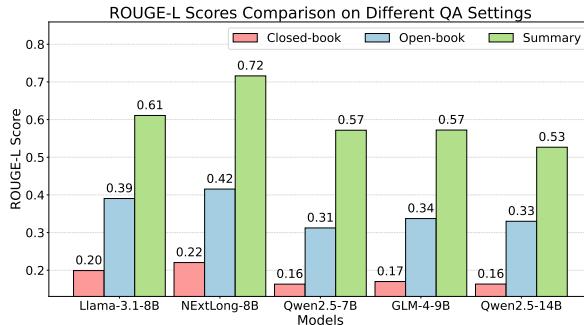


Figure 9: ROUGE-L scores of the models across different settings.

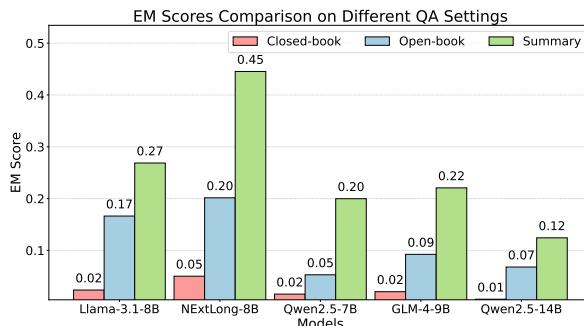


Figure 10: EM scores of the models across different settings.

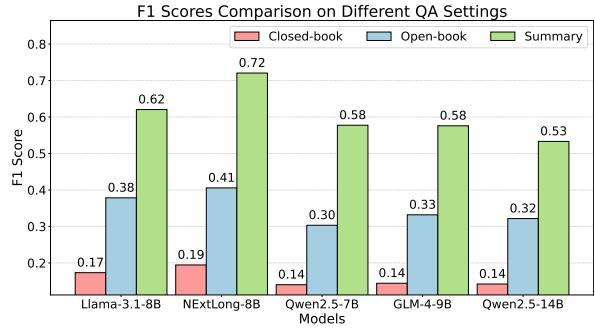


Figure 11: F1 scores of the models across different settings.

Title	Author	Nationality
1. <i>A Portrait of the Artist as a Young Man</i>	James Joyce	Irish
2. <i>A Voyage to Arcturus</i>	David Lindsay	Scottish
3. <i>Father Goriot</i>	Honoré de Balzac	French
4. <i>Lisbeth Longfrock</i>	Hans Aanrud	Norwegian
5. <i>Lothair</i>	Benjamin Disraeli	British
6. <i>Peter Pan in Kensington Gardens</i>	J. M. Barrie	Scottish
7. <i>Tarzan and the Jewels of Opar</i>	Edgar Rice Burroughs	American
8. <i>Tarzan of the Apes</i>	Edgar Rice Burroughs	American
9. <i>The Adventures of the Dying Detective</i>	Arthur Conan Doyle	Scottish
10. <i>The Black Dwarf</i>	Walter Scott	Scottish
11. <i>The Call of the Wild</i>	Jack London	American
12. <i>The Children of the New Forest</i>	Frederick Marryat	British
13. <i>The House of the Seven Gables</i>	Nathaniel Hawthorne	American
14. <i>The House on the Borderland</i>	W. H. Hodgson	British
15. <i>The Gods Are Athirst</i>	Anatole France	French
16. <i>The Vampyre</i>	John Polidori	British
17. <i>The Variable Man</i>	Philip K. Dick	American
18. <i>Uncle Silas</i>	Joseph S. Le Fanu	Irish
19. <i>Voodoo Planet</i>	Andre Norton	American
20. <i>Youth</i>	Joseph Conrad	Polish-British

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

Original Sample	Refined Sample
<p>Summary: [...] The rats abandon the reshipped barque and a new crew is brought in from Liverpool</p> <p>Q: Why is a new ship brought in from Liverpool?</p> <p>A1: because no man will stay on a ship abandoned by rats</p> <p>A2: The ship had been abandoned by rats.</p>	<p>Q: Why was a new crew brought in from Liverpool?</p> <p>A1: because no man will stay on a ship abandoned by rats</p> <p>A2: The ship had been abandoned by rats.</p>
<p>Summary: [...] Indefer Jones' niece, Isabel Brodrick, has lived with him for years after the remarriage of her father, and endeared herself to everyone. However, according to his strong traditional beliefs, the estate should be bequeathed to a male heir. His sole male blood relative is his nephew Henry Jones [...]</p> <p>Q: Why did Indefer Jones originally leave his estate to his nephew when he wanted to leave it to his niece?</p> <p>A1: Because that was tradition</p> <p>A2: Because that was tradition</p>	<p>Q: Why did Indefer Jones initially plan to leave his estate to his male heir Henry despite preferring his niece Isabel?</p> <p>A1: Because that was tradition</p> <p>A2: Because that was tradition</p>
<p>Summary: [...] While Oxford's academic staff barely notice that nearly all of their undergraduates have vanished, Zuleika decides to order a special train for the next morning [...]</p> <p>Q: After many of the students have died how does Zuleika choose to travel away from Oxford?</p> <p>A1: On a train</p> <p>A2: All of the undergraduate students.</p>	<p>Q: After many of the students have died how does Zuleika choose to travel away from Oxford?</p> <p>A1: On a train</p> <p>A2: She orders a special train to Cambridge.</p>
<p>Summary: [...] The book begins with the death of Helen Carey, the much beloved mother of nine-year-old Philip Carey [...] he is sent to live with his aunt Louisa and uncle William Carey [...]</p> <p>Q: Who was Phillip sent to live with after his mother died?</p> <p>A1: His aunt and uncle</p> <p>A2: Aunt and uncle</p>	<p>Q: Who was Phillip sent to live with after his mother died?</p> <p>A1: His aunt and uncle</p> <p>A2: Philip was sent to live with his aunt Louisa and uncle William Carey</p>

Table 13: Examples of QA samples from different books after being corrected by Claude. In the first two samples, Claude corrected the question, while in the last two one of the reference answers. Factual and semantic errors are highlighted in red, while their correction and evidence is in green. Although the second and fourth original samples contained no major errors, Claude improved their grammatical fluency and specificity (highlighted in blue).

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

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 believes')
- 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 15: 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 16: 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 17: 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 18: 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 19: Likert Scale Grading Rubric