DocBench: A Benchmark for Evaluating LLM-based Document Reading Systems

Anonymous ARR submission

Abstract

Recent advancements in proprietary large language models (LLMs), such as those from OpenAI and Anthropic, have led to the development of document reading systems capable of handling raw files with complex layouts, intricate formatting, lengthy content, and multimodal information. However, the absence of a standardized benchmark hinders objective evaluation of these systems. To address this gap, we introduce DocBench, a benchmark designed to simulate real-world scenarios, where each raw file consists of a document paired with one or more questions. DocBench uniquely evaluates entire document reading systems and adopts a user-centric approach, allowing users to identify the system best suited to their needs.

1 Introduction

Recent advancements made by proprietary LLM developers, such as OpenAI and Anthropic, have led to the release of several LLM-based document reading systems (Achiam et al., 2023; Lee et al., 2024). Unlike standalone LLMs designed solely for reading comprehension, these systems allow users to upload raw files and require the capability to parse complex layouts, navigate intricate formatting, retrieve relevant context, manage lengthy content, and integrate multi-modal information (Cheng et al., 2023). However, despite widespread claims of excellent performance in public blogs, the lack of a standardized benchmark makes it difficult to objectively evaluate and compare the document reading performance across these systems, thereby leaving a critical gap in the fair and fine-grained assessment of their capabilities.

To fill this gap, we introduce DocBench, a benchmark designed to evaluate LLM-based document reading systems. DocBench is developed to mirror real-world scenarios where each input consists of a raw document file paired with one or multiple questions, each of which is annotated with a golden answer. Our benchmark undergoes a meticulous development process, incorporating human annotation and synthetic question generation. To the end, DocBench features 229 real-world files and 1,102 questions. We evaluate several proprietary LLM-based systems that are accessible via web interfaces. However, these proprietary systems are close-sourced, thus leading to the limited disclosure of their detailed operational strategies.

In summary, DocBench introduces two key features that set it apart from previous benchmarks:

1. DocBench evaluates LLM-based systems rather than just standalone LLMs. This approach ensures that, regardless of underlying black-box designs or backbone LLMs, the system's overall performance is fairly evaluated.

2. DocBench is a user-centric benchmark, allowing users to identify which system best suits their specific needs. This perspective is often absent from traditional machine reading comprehension benchmarks, which primarily assess the LLM ability to extract answers from given passages.

2 Related Works

Document reading is a critical area where LLM-based systems have shown significant advancements. Proprietary developers such as OpenAI[1] and Anthropic[2] have introduced advanced systems that can take a raw document file as input. While these systems build upon the fundamental capabilities of their underlying LLMs (Zeng et al., 2022; Bai et al., 2023; Achiam et al., 2023; Anthropic, 2024), they differ in design and implementation, with some excelling in long-context reading and others focusing on retrieval-augmented methods to enhance document reading. Despite claims of effectiveness and efficiency in online public blogs, the lack of a standardized benchmark makes it diffi-

[1] OpenAI's ChatGPT: https://chat.openai.com
[2] Anthropic's Claude: https://claude.ai/chats

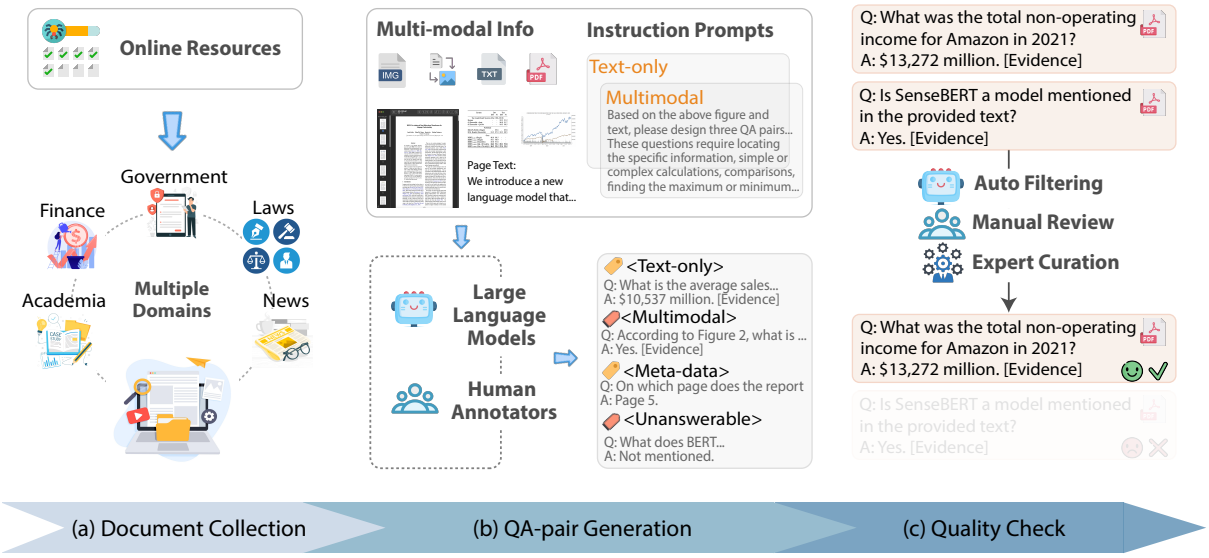

Figure 1: Construction pipeline of DocBench. (a) Document Collection: gathering PDF files from five different domains; (b) QA-pair Generation: creating diverse and comprehensive QA pairs through a combination of LLMs and human effort; (c) Quality Check: ensuring data quality through a multi-step process.

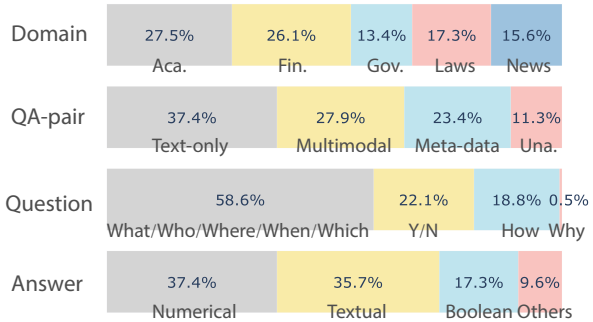

Figure 2: Data distribution of DocBench based on four distinct classification criteria.

cult to objectively evaluate and compare document reading performance across these systems. Existing benchmarks relevant to document reading fail to adequately reflect the real performance of these systems. Datasets focusing on traditional machine reading comprehension, such as SQuAD (Rajpurkar, 2016), HotpotQA (Yang et al., 2018), and those specifically focusing on long-context reading like NarrativeQA (Kočiskỳ et al., 2018), primarily use text as input only, overlooking the complex nature of document structure, meta-data, and multi-modal information. On the other hand, multi-modal document reading datasets like DocVQA (Mathew et al., 2021), MMLongBench-Doc (Ma et al., 2024), and M-longdoc (Chia et al., 2024) incorporate multi-modal inputs and preserve the original document structure and layout. However, these datasets treat document pages exclusively as images, neglecting other question types and inadvertently complicating text understanding.

3 The DocBench

3.1 Dataset Construction

Our dataset construction pipeline consists of three phases. First, we crawl documents across various domains from publicly accessible online resources (§3.1.1). Second, we generate corresponding QA pairs with the help of GPT-4 and a team of human annotators (§3.1.2). Finally, we conduct auto filtering followed by a manual review to validate the quality of the generated instances (§3.1.3).

3.1.1 Document Collection

To establish a practical and constructive benchmark for document reading, we concentrate on scenarios where it is crucial to read documents. We standardize the documents to PDF format due to its high compatibility and stability. We identify five domains where documents are frequently utilized: *Academia*, *Finance*, *Government*, *Laws*, *News*. For Academia, papers are downloaded from arXiv within the range of top-100 citations on Google Scholar. [3] For *Finance*, we crawl the annual reports of companies with top-100 global market capitalization up to 2024-02-23 from AnnualReports. [4] For *Government*, we manually download official governmental reports in 2023 from the U.S. Department of State and GovInfo. [5] For *Laws*, files are

[3] https://scholar.google.com/; https://arxiv.org/.

[4] https://companiesmarketcap.com; http://www.annualreports.com.

[5] https://www.state.gov/department-reports/; https://www.govinfo.gov/.

Methods	Form	File size	Domain					Type				Overall Acc.
			Aca.	Fin.	Gov.	Laws	News	Text.	Multi.	Meta.	Una.	
Human	-	-	83.0	82.2	77.8	75.0	86.4	81.4	83.3	77.5	82.2	81.2
GPT-4	*Web*	100M	65.7	**65.3**	75.7	69.6	79.6	**87.9**	**74.7**	50.8	37.1	69.8
GPT-4o	*Web*	100M	56.4	56.3	73.0	65.5	75.0	85.0	62.7	50.4	17.7	63.1
GLM-4	*Web*	20M	55.8	35.4	61.5	62.8	82.0	73.1	50.3	48.8	33.1	56.5
KimiChat	*Web*	100M	62.4	61.8	**77.0**	78.5	**87.2**	87.6	65.3	50.4	**71.8**	**70.9**
Claude-3.5	*Web*	10M	**73.9**	40.6	70.3	**79.1**	86.6	80.8	64.6	**54.3**	58.9	67.6
Gemini-1.5	*Web*	30M	60.4	42.5	57.4	71.7	74.3	74.0	30.8	53.8	60.2	55.4
Qwen-2.5	*Web*	150M	42.9	29.9	51.4	55.5	69.2	61.7	31.8	36.0	58.1	46.9
ERNIE-3.5	*Web*	10M	56.4	37.5	54.7	58.1	58.1	63.6	47.7	36.8	54.0	51.8

Table 1: System and human Performance on DocBench across various types and domains. We did not test more recent o1-like models because OpenAI o1-series models do not yet support document uploads.

gathered from an official online collection of publications from the Library of Congress, within the years ranging from 2020 to 2024. [6] For *News*, we collect front-page scanned documents of the New York Times, covering dates from 2022-02-22 to 2024-02-22. [7] After skipping damaged files, we eventually obtained 229 PDF files.

3.1.2 QA-pair Generation

We deliver extracted text to GPT-4 (*gpt-4-0125-preview*) for generating *text-only* QA pairs and resort to GPT-4v (*gpt-4-1106-vision-preview*) for yielding multi-modal ones based on tables, figures, and their related textual descriptions. On the other hand, we further request a set of human annotators to manually elaborate 350 QA pairs based on the given document files. Their primary task is to focus on types that are rarely covered in the previous generation stage but are frequent in daily usage, such as meta-data and unanswerable instances. Details of the annotation process and instruction prompts are attached in Appendix B.

3.1.3 Quality Check

We begin by instructing GPT-4 to automatically filter out questions that are excessively lengthy, unnatural, or impractical. We then conduct a manual review following the automatic filtering to ensure both the quality of questions and the accuracy of answers. To further align our data with real-world user scenarios, we engage 7 practitioners from distinct domains to review and refine the data within their areas of expertise. In this way, our data quality is validated from multiple perspectives.

3.2 Dataset Statistics

DocBench has a total of 229 PDF documents sourced from publicly accessible online repositories along with 1,102 questions, spanning across 5 domains: Academia, Finance, Government, Law, and News. Figure 2 shows data distribution in DocBench based on various criteria.

Specifically, the types of QA pairs can be mainly divided into four groups: *text-only* (37.4%), *multi-modal* (27.9%), *meta-data* (23.4%), and *unanswerable* (11.3%). The *text-only* and *multimodal* types collectively account for over half (65.3%), centering on the abilities to comprehend long contexts and interpret information from different modalities. Besides, we incorporate approximately one-third (34.7%) of questions to more closely fit the actual scenarios as well as assess the robustness of the document reading systems, including 23.4% inquiring about metadata (e.g., page numbers, word counts) and 11.3% that cannot be answered based on the given document. The types of questions can be primarily separated into four categories according to the inquiry focus: *what / who / where / when / which* (58.6%), *Y/N* (22.1%), *how* (18.8%), and *why* (0.5%). And the types of answers can be partitioned into four classes: *numerical* (37.4%), *textual* (35.7%), *boolean* (17.3%), and *others* (9.6%).

3.3 Evaluation Setup

Since different LLMs and systems exhibit substantial variations in the organization and style of their outputs, potentially leading to biases in traditional evaluation approaches, such as exact match (EM) or F1 score that were commonly used in reading

[6] https://www.loc.gov/collections/publications-of-the-law-library-of-congress.

[7] https://static01.nyt.com/images/.

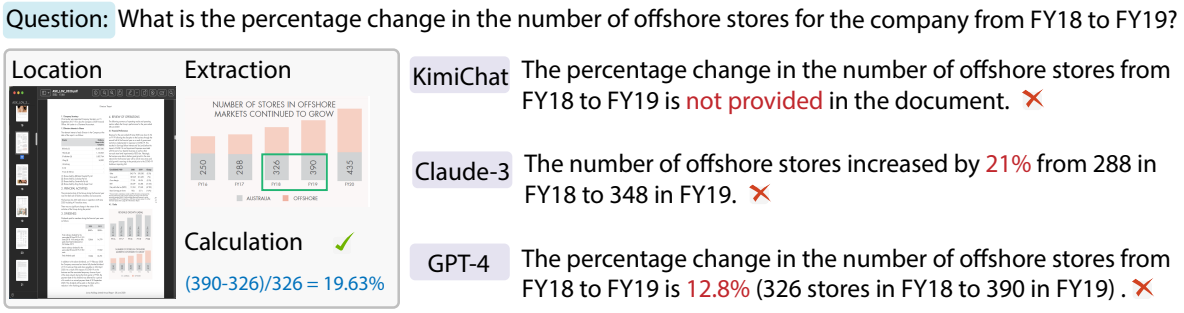

Figure 3: To address multi-modal questions in DocBench, it is essential to: (i) identify the relevant figure/table (Location); (ii) extract specific data (Extraction); (iii) perform necessary calculations (Calculation). In this case study, KimiChat fails to locate the figure, Claude-3.5 retrieves incorrect data, and GPT-4, despite succeeding in the first two steps, struggles with the calculation.

comprehension tasks. Following Liu et al. (2023), we instruct GPT-4 to assign a score of 0 (incorrect) or 1 (correct). After evaluating 200 examples by both human evaluators and GPT-4, we found that the GPT-4 automatic evaluator shows a 98% agreement with human annotators, significantly exceeding the traditional string matching approach. Details of this experiment are shown in Table 2, and the instruction prompts for evaluation are attached in Appendix B. As mentioned above, we instruct GPT-4 to assign a score of 0 (incorrect) or 1 (correct), thus using Accuracy (abbreviated as Acc.) to measure system performance. We report accuracy across all instances, as well as for each domain and QA-pair type in Table 1.

4 Experiments and Analysis

4.1 Experimental Setup

We conduct evaluation of 8 popular LLM-based proprietary systems that support document uploads, including GPT-4 and GPT-4o[8] from OpenAI, GLM-4[9] from ZhipuAI, Kimi[10] from Moonshot AI, Claude-3.5[11] from Anthropic, Qwen-2.5[12] from Alibaba Cloud, and ERNIE-3.5[13] from Baidu.

4.2 Results and Discussion

Table 1 showcases the performance of various document reading systems on DocBench. Our findings reveal substantial variations in document reading capabilities among these systems, driven by differences in their backbone LLMs, context length limitations, diverse design and implementation approaches, and etc. Figure 3 presents a case study illustrating the unique challenge of answering multi-modal questions in DocBench. We observe that leading proprietary LLM-based systems often fail due to errors in one of the steps in the *Location→Extraction→Calculation* sequence. Take the case study as an example, in the first step, KimiChat fails to locate the relevant chart on page 17. In the extraction phase, Claude-3.5 misidentifies the data as *288 & 348*, instead of the correct *326 & 390*. Finally, while GPT-4 locates and extracts the correct information, it errs in calculating the percentage change, demonstrating the complexity of these questions. Besides, most existing document reading systems falter when faced with unanswerable questions based on the provided document. Intriguingly, despite the commonly-shared base model on GPT-4, there is a notable low performance for handling unanswerable questions (i.e., 37.1%). We analyze that this may be due to: (i) the proprietary LLM-based system have undergone optimizations on the base model, potentially causing overfitting; (ii) GPT-4 tends to adhere more closely to the in-context learning information. Such phenomenon thus underscores a critical challenge for future document reading systems on enhancing fidelity to the given documents.

5 Conclusion

In this paper, we introduce DocBench, a novel benchmark designed to assess LLM-based document reading systems in a comprehensive and granular manner. DocBench comprises 229 documents and 1,102 questions. We evaluate several proprietary LLM systems and uncover significant disparities in their document reading capabilities, highlighting the current limitations and presenting key challenges in this field.

[8] https://chatgpt.com
[9] https://chatglm.cn/main/doc
[10] https://kimi.moonshot.cn
[11] https://claude.ai/chats
[12] https://tongyi.aliyun.com/qianwen
[13] https://yiyan.baidu.com

6 Limitation

While DocBench aims to cover a broad spectrum of real-world document-related questions, it is not exhaustive. Our benchmark focuses primarily on the four most common question types, leaving other potential types unaddressed. Furthermore, our evaluation of proprietary LLM-based document reading systems is limited. Many such systems, including OpenAI-o1, are accessible only through web interfaces with restricted access and lack APIs, making the evaluation process slow and challenging.

References

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

Anthropic. 2024. Claude 3 haiku: our fastest model yet.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Daixuan Cheng, Shaohan Huang, and Furu Wei. 2023. Adapting large language models via reading comprehension. In *The Twelfth International Conference on Learning Representations*.

Yew Ken Chia, Liying Cheng, Hou Pong Chan, Chaoqun Liu, Maojia Song, Sharifah Mahani Aljunied, Soujanya Poria, and Lidong Bing. 2024. M-longdoc: A benchmark for multimodal super-long document understanding and a retrieval-aware tuning framework. *arXiv preprint arXiv:2411.06176*.

Tomáš Kočiskỳ, 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.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. 2024. A human-inspired reading agent with gist memory of very long contexts. *arXiv preprint arXiv:2402.09727*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, et al. 2024. Mmlongbench-doc: Benchmarking long-context document understanding with visualizations. *arXiv preprint arXiv:2407.01523*.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209.

P Rajpurkar. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

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.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

A Annotation Process

Since the QA-pair generation process requires data annotators to deeply understand the motivations behind our benchmark construction, and considering the initial training costs and the need to manually annotate about 350 QA pairs, we've decided to assign 2 annotators to this task.

The annotation process presents as follows:

- We first communicate the motivation behind our work to the annotators and explain the concepts of meta-data and unanswerable questions in detail.
- Next, we provide 10 example QA pairs for reference (5 for each type).
- Finally, each annotator generates 170 QA pairs. They then exchange their annotations for double-checking and review.

B Instruction Prompts

B.1 Response Evaluation

Detailed instruction prompts for response evaluation are shown in Table 4.

B.2 QA-pair Generation

Details of instruction prompts for generating QA pairs are attached in Table 5. We discover that simply passing diagrams to GPT-4V leads to subpar question quality. This issue likely stems from the fact that figures or tables without accompanying text descriptions typically lack sufficient information, thus causing the generated QA pairs to deviate from their intended meanings. In addition, we observe that adding difficulty settings for QA generation (e.g., *Easy*, *Medium*, *Hard*) in the instruction prompt can result in higher quality. We analyze that this may be due to the model being able to favor higher generation quality in potential comparisons.

C Analysis of Input Sources

Table 6 presents the impact of different input sources on model performance. We provide questions to GPT-4 and GPT-4o, both with and without attached files. Remarkably, even without files, the models correctly answer a portion of the questions (19.1% for GPT-4 and 21.7% for GPT-4o). Our analysis reveals that the correctly answered questions are predominantly textual and are largely associated with government, law, and news domains. This trend suggests that the models' underlying training data is heavily skewed towards these categories, enabling them to answer some questions accurately without additional files. Moreover, as GPT-4o is an optimized version of GPT-4, it likely benefits from a broader and more training data.

Sources	# Correct / Wrong by different evaluators				Agreement (human and automatic evaluators)		
	Human	GPT-4	GPT-3.5	StrMatch	GPT-4	GPT-3.5	StrMatch
KimiChat	24 / 16	23 / 17	33 / 7	0 / 40	97.5%	75.0%	40.0%
Qwen-2.5	17 / 23	18 / 22	31 / 9	0 / 40	97.5%	57.5%	57.5%
Gemma (7B)	19 / 21	18 / 22	18 / 22	0 / 40	97.5%	75.0%	52.5%
Mixtral (7B)	14 / 26	14 / 26	26 / 14	0 / 40	100.0%	65.0%	65.0%
Llama-3 (70B)	16 / 24	15 / 25	28 / 12	0 / 40	97.5%	62.5%	60.0%
Total	90 / 110	88 / 112	136 / 64	0 / 200	98.0%	67.0%	55.0%

Table 2: The GPT-4 automatic evaluator shows a 98% agreement with human annotators. We randomly sample 40 questions and answers from five systems, asking human annotators to assess their accuracy. We then employ string matching (StrMatch), GPT-3.5, and GPT-4 as automatic evaluators. Finally, we measure the agreement between the human and these automatic evaluators.

Table 3: Examples of instances from DocBench, with multiple labels indicating our data diversity.

Question	Answer	Labels	Document
Why does the model not perform as well in German compared to Spanish and Dutch?	Due to its **complex morphology and compound words**...	<Aca.><Why> <Text-only> <Textual>	When and Why are Pre-trained Word Embeddings Useful for Machine Translation [clickable file link]
By **how much** did the number of Erica users increase from 2018 to 2019?	The number increased by **5.5 million**...	<Fin.><How> <Multimodal> <Numerical>	Bank of America Annual Report 2020 [clickable file link]
What is the primary focus of Bureau Objective 3.4?	The report **does not contain** such objective.	<Gov.> <Wh-> <Unanswerable> <Others>	Governmental report from *Secretary's Office of Global Women's Issues* 2022 [clickable file link]
How many times does the report mention "scientific ethics"?	The report mentions "scientific ethics" **11** times.	<Laws><How> <Meta-data> <Numerical>	Report on *Regulation of Stem Cell Research* from Library of Congress 2023 [clickable file link]
Is the article about Hurricane Ian's impact in Florida written by multiple authors?	**Yes**, the article is about Hurrican Ian's impace in Florida...	<News><Y/N> <Meta-data> <Boolean>	New York Times front page on 2022-09-30 [clickable file link]

Table 4: Instruction Prompts in Response Evaluation.

`System Content:`

You are a helpful evaluator.

`Prompt:`

Task Overview:

You are tasked with evaluating user answers based on a given question, reference answer, and additional reference text. Your goal is to assess the correctness of the user answer using a specific metric.

Evaluation Criteria:

1. Yes/No Questions: Verify if the user's answer aligns with the reference answer in terms of a "yes" or "no" response.
2. Short Answers/Directives: Ensure key details such as numbers, specific nouns/verbs, and dates match those in the reference answer.
3. Abstractive/Long Answers: The user's answer can differ in wording but must convey the same meaning and contain the same key information as the reference answer to be considered correct.

Evaluation Process:

1. Identify the type of question presented.
2. Apply the relevant criteria from the Evaluation Criteria.
3. Compare the user's answer against the reference answer accordingly.
4. Consult the reference text for clarification when needed.
5. Score the answer with a binary label 0 or 1, where 0 denotes wrong and 1 denotes correct.

NOTE that if the user answer is 0 or an empty string, it should get a 0 score.

Question: `{{question}}`
User Answer: `{{sys_ans}}`
Reference Answer: `{{ref_ans}}`
Reference Text: `{{ref_text}}`

Evaluation Form (score ONLY):

- Correctness:

Table 5: Instruction Prompts in QA-pair Generation.

System Content:

You are a helpful assistant that can generate question-answer pairs.

Text-only QA:

Based on the above text, please design three question-answer pairs with different levels of difficulty: Easy, Medium, Hard.

The questions should be close-ended and should be answered based on the provided text.

The answer form should be as diverse as possible, including [Yes/No, Short Answer, Long Answer, Abstractive Answer].

You should provide the reference in the text and the answer form if possible.

The output should be formalized as: "'Q: | A: | Reference: | Difficulty Level: | Answer Form:"'

Multimodal QA (w/table+text):

Based on the above table and text, please design three question-answer pairs with different levels of difficulty: Easy, Medium, Hard.

The text provided is text related to the table, which can provide more reference for question generation, but the focus is still on the table itself.

These questions require locating the specific information, simple or complex calculations, comparisons, finding the maximum and minimum, reading across rows and columns, etc.

Note that these questions also need to be realistic. You should provide the reason if possible.

The output should be formalized as: "'Q: | A: | Reference: | Difficulty Level: | Answer Form:"'

Multimodal QA (w/figure+text):

Based on the above figure and text, please design three question-answer pairs with different levels of difficulty: Easy, Medium, Hard.

The text provided is text related to the figure, which can provide more reference for question generation, but the focus is still on the figure itself.

These questions require a deep reading of the meaning of the image.

Note that these questions also need to be realistic. You should provide the reason if possible.

The output should be formalized as: "'Q: | A: | Reason: | Difficulty Level: | "'

Multimodal QA (w/table):

Based on the above image, please design three question-answer pairs with different levels of difficulty: Easy, Medium, Hard.

These questions require locating the specific information, simple or complex calculations, comparisons, finding the maximum and minimum, reading across rows and columns, etc.

Note that these questions also need to be realistic. You should provide the reason if possible.

The output should be formalized as: "'Q: | A: | Reason: | Difficulty Level: | "'

Multimodal QA (w/figure):

Based on the above image, please design three question-answer pairs with different levels of difficulty: Easy, Medium, Hard.

These questions require a deep reading of the meaning of the image. Note that these questions also need to be realistic. You should provide the reason if possible.

The output should be formalized as: "'Q: | A: | Reason: | Difficulty Level: | "'

Table 6: Analyzing the Influence of Input Sources: We deliver questions with attached files and without files to GPT-4 and GPT-4o for evaluation, respectively.

Methods	**Domain**					**Type**				**Overall** Acc.
	Aca.	**Fin.**	**Gov.**	**Laws**	**News**	**Text.**	**Multi.**	**Meta.**	**Una.**	
GPT-4										
w/ file	65.7	65.3	75.7	69.6	79.6	87.9	74.7	50.8	37.1	69.8
w/o file	10.9	10.8	23.0	29.3	32.6	40.8	8.1	1.6	10.5	19.1
GPT-4o										
w/ file	56.4	56.3	73.0	65.5	75.0	85.0	62.7	50.4	17.7	63.1
w/o file	11.2	13.5	29.1	31.9	36.0	46.6	10.7	2.3	6.5	21.7