Evaluating Large Language Model with Knowledge Oriented Language Specific Simple Question Answering

Anonymous ACL submission

Abstract

We introduce KoLasSimpleQA, the first benchmark evaluating the multilingual factual ability of Large Language Models (LLMs). Inspired by existing research, we created the question set with features such as single knowledge point coverage, absolute objectivity, unique answers, and temporal stability. These questions enable efficient evaluation using the LLM-as-judge paradigm, testing both the LLMs' factual memory and self-awareness ("know what they don't know"). KoLasSimpleQA expands existing research in two key dimensions: (1) **Breadth (Multilingual Coverage)**: It includes 9 languages, supporting global applicability evaluation. (2) **Depth (Dual Domain Design)**: It covers both the general domain (global facts) and the language-specific domain (such as history, culture, and regional traditions) for a comprehensive assessment of multilingual capabilities. We evaluated mainstream LLMs, including traditional LLM and emerging Large Reasoning Models. Results show significant performance differences between the two domains, particularly in performance metrics, ranking, calibration, and robustness. This highlights the need for targeted evaluation and optimization in multilingual contexts. We hope KoLasSimpleQA will help the research community better identify LLM capability boundaries in multilingual contexts and provide guidance for model optimization. KoLasSimpleQA datasets are available at `https://anonymous.4open.science/r/KoLas_anonymous-0D6F` .

1 Introduction

Large Language Models (Grattafiori et al., 2024; Yang et al., 2024; Guo et al., 2025) have advanced significantly, yet hallucination—where models produce unverified or misleading information—remains a major challenge, affecting their reliability and broader use. To tackle this, the SimpleQA (Wei et al., 2024) and ChineseSimpleQA (He et al., 2024b) benchmarks were introduced to evaluate LLMs' factual ability using short, fact-based questions. These questions focus on a single knowledge point, with answers that are objective, stable, and not open to interpretation.

Recent studies (Zhang et al., 2023; Shi et al., 2022; Huang et al., 2023a) indicate that LLM performance varies across languages, particularly in factual ability, with models typically performing better in English. However, the SimpleQA and ChineseSimpleQA benchmarks are limited to English and Chinese. Additionally, most evaluations of LLMs in non-English contexts focus on general knowledge rather than language-specific content like history, culture, and local traditions. While LLMs excel in general knowledge, they often struggle with language-specific facts.

Benchmark	**Lang.**	**Lang. specific**	**Easy to evaluate**
XTREME (Hu et al., 2020)	40 lang.	×	×
Okapi (Lai et al., 2023)	en	×	×
SimpleQA (Wei et al., 2024)	en	×	√
ChineseSimpleQA (He et al., 2024b)	zh	×	√
MINTQA (He et al., 2024a)	en	×	×
BenchMAX (Huang et al., 2025)	17 lang.	×	×
MMLU-ProX (Xuan et al., 2025)	29 lang.	×	√
KoLasSimpleQA(Ours)	9 lang.	√	√

Table 1: Comparison between KoLasSimpleQA and other benchmarks. The comparison is conducted along three dimensions: the range of supported languages, whether the benchmark includes language-specific knowledge, and whether it is easy to evaluate.

To tackle this problem, we introduce **K**nowledge-**O**riented **La**nguage-**S**pecific **Simple Q**uestion **A**nswering (KoLasSimpleQA), a benchmark comprising simple fact-based QA samples grounded in genuinely language-specific knowledge across nine languages. KoLasSimpleQA has three main features: **(1) Foundation:** Inspired by (Wei et al.,

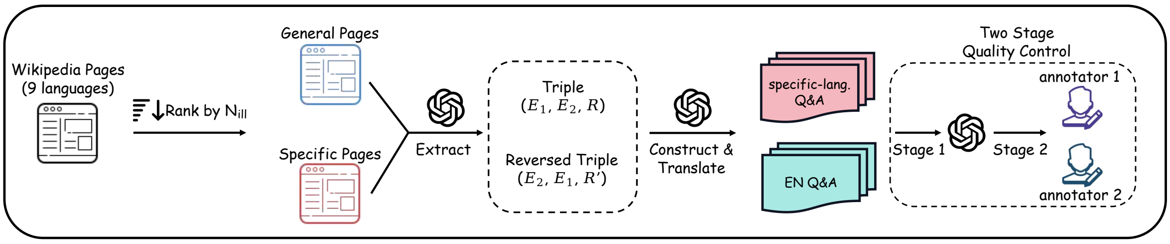

Figure 1: Construction pipeline of KoLasSimpleQA. The process includes Wikipedia entry selection based on inter-language links, triple and QA pair generation using GPT-4o, and a two-stage quality control to ensure question quality and diversity.

2024; He et al., 2024b), we crafted a question set with attributes such as *single knowledge point coverage*, *absolute objectivity*, *unique answers*, and *temporal stability*. These questions enable efficient evaluation using the LLM-as-judge paradigm, assessing both the factual memory and self-awareness of LLMs (i.e., their ability to "know what they don't know"). **(2) Breadth Expansion:** Unlike existing work (Wei et al., 2024; He et al., 2024b) that is limited to a few languages, KoLasSimpleQA includes *9 languages*, allowing for performance evaluation of LLMs in multilingual contexts and supporting assessments of global applicability. **(3) Depth Exploration:** It encompasses both the *general domain* (global facts) and the *language-specific domain* (such as history, culture, and regional traditions). We collected data from Wikipedia, categorizing entries into general and language-specific domains based on the number of inter-language links each article has (as shown in Figure 2; see details in §2.2.1). Using these classifications and filtered entries, we constructed questions for both domains, enabling a comprehensive and in-depth evaluation of LLMs in multilingual settings.

We evaluated mainstream LLMs on KoLasSimpleQA, covering both traditional LLMs and the latest Large Reasoning Models (LRMs). Key insights from our study include:

(1) Performance Disparity Across Domains: LLMs perform much worse in the language-specific domain than in the general domain.

(2) Translating Non-English Queries: Translating queries into English is a common strategy to enhance multilingual performance. While effective in the general domain, it is less so in the language-specific domain.

(3) Calibration Performance: LLMs show significantly poorer calibration in the language-specific domain compared to the general domain.

(4) Knowledge Memorization Robustness: LLMs are notably less robust in the language-specific domain than in the general domain.

	ar	cs	hu	ko	ru	sr	th	vi	zh	total
General	162	142	169	151	92	133	209	121	127	1306
Language Specific	117	90	127	84	57	79	99	92	96	841

Table 2: Distribution of question counts across languages in KoLasSimpleQA.

2 KoLasSimpleQA

2.1 Overview

We created a multilingual QA benchmark including both the general and language-specific domains. The general domain covers global knowledge shared across languages, while the language-specific subset targets knowledge unique to individual linguistic and cultural contexts. Our benchmark includes 9 languages: Hungarian (hu), Czech (cs), Serbian (sr), Russian (ru), Chinese (zh), Korean (ko), Thai (th), Arabic (ar), and Vietnamese (vi), along with their English (en) translations. The data construction process is shown in Figure 1.

2.2 Benchmark Construction

2.2.1 Collection of Specific & General Wikipedia Entries

We crawled all Wikipedia pages in nine languages and extracted their contents. Unlike conventional methods focusing only on entry content, we specifically extracted inter-language link information for each entry and counted the number of these links, denoted as n_{ill}. An inter-language link is a hyperlink that connects a Wikipedia page in one language to the related page in another language, aiding navigation across different language versions. These

Figure 2: Illustration of inter-language links on a Wikipedia page. The number of such links (n_{ill}) is used to distinguish between language-specific and language-general knowledge.

links are usually located in the sidebar at the top right of the webpage, as shown in Figure 2. It's important to note that pages in different language versions of Wikipedia are not machine-translated; they are created by users who are typically native speakers of the respective language. If a page exists in a particular language, the topic or knowledge it represents is likely of interest to native speakers of that language. Thus, if an entity lacks versions in other languages ($n_{\text{ill}} = 0$), it is classified as **language-specific**, indicating the knowledge is unique to that language. Conversely, if an entity appears in many language versions (n_{ill} is large), it likely represents **global knowledge**, making it **general**.

For each of the nine languages, we sample a total of $N_{\text{specific}} + N_{\text{general}}$ entries. Specifically, we randomly select N_{specific} entries from those where $n_{\text{ill}} = 0$, ensuring that they represent language-specific knowledge. Formally, this subset is defined as $\mathbf{E}^{\text{selected}}_{\text{specific}} = \{x \in \mathbf{D} \mid n_{\text{ill}}(x) = 0, |\mathbf{E}^{\text{selected}}_{\text{specific}}| = N_{\text{specific}}\}$. Additionally, we identify the top N_{general} entries with the highest n_{ill} values, as these are more likely to contain language-general knowledge, defined as $\mathbf{E}^{\text{selected}}_{\text{general}} = \{x \in \mathbf{D} \mid \mathsf{Rank}(n_{\text{ill}}(x)) \leq N_{\text{general}}\}$. Here, $\mathbf{D}$ represents the set of all Wikipedia entries, $\mathbf{E}^{\text{selected}}_{\text{specific}}$ and $\mathbf{E}^{\text{selected}}_{\text{general}}$ denote the selected sets of language-specific and language-general entries, respectively. The function $n_{\text{ill}}(x)$ returns the number of inter-language links associated with entity x, and $\mathsf{Rank}(n_{\text{ill}}(x))$ represents the ranking of entity x based on its n_{ill} value. Finally, N_{specific} and N_{general} define the number of sampled language-specific and language-general entries, respectively. We set $N_{\text{specific}} = 400$ and $N_{\text{general}} = 400$.

2.2.2 Construction of Question-Answer

We employ GPT-4o to extract triples (E1, R, E2) that satisfy the requirements from the collected general and specific Wikipedia entries (See prompt in Table 8). Both E1 and E2 are titles of Wikipedia entries, and R represents the core relation between them. Subsequently, we leverage GPT-4o to generate Question-Answer (QA) pairs from these triples and translate them into their corresponding English parallel QA pairs (see prompt in Table 9 and 10). For example, if the title of a Wikipedia entry is “Lombardy,” and one segment of the content is “Lombardy is located between the Alps mountain range and tributaries of the river Po, and includes Milan, its capital, the largest metropolitan area in the country, and among the largest in the EU,” the extracted triple would be (Lombardy, capital is, Milan), and the corresponding QA pair would be “What is the capital city of the Lombardy region?”For each question, if either E1 or E2 in its corresponding triple (E1, R, E2) is a language-specific entity, the question is defined as **language-specific**.

Additionally, we extract the reversed triples (E2, R', E1) and construct QA pairs that explicitly capture inverse relations. This approach allows us to assess whether the model has genuinely internalized the underlying knowledge, as discussed in this study (Berglund et al., 2023). The reversed triple corresponding to the above example is (Milan, is capital of, Lombardy), and the corresponding QA pair would be “What Italian region is Milan the capital of?”

2.2.3 Two-stage Quality Control

For each constructed QA pair, we expect them to meet the following criteria: **(1) Triple Consistency (TC)**: The question must be generated from the head entity and relation of the triple, while the answer must correctly correspond to the tail entity. **(2) Self-Sufficiency (SS)**: The question should contain all necessary information to be answerable on its own, without requiring external context. **(3) Non-Triviality (NT)**: The question should not be overly simplistic, and the answer should not be directly inferable from the question itself. **(4) Objectivity (OBJ)**: The question must be based on verifiable, factual knowledge, avoiding opinion-based, subjective, or interpretive content. **(5) Temporal Stability (TS)**: The answer must remain valid over

General	
zh	**Q:** "伦巴第大区的首府是哪座城市？ " **A:** "米兰" **Q :** "What is the capital city of the Lombardy region? " **A:** "Milan "
vi	**Q :** "Biển Okhotsk nằm dưới sự quản lý của quốc gia nào?" **A:** "Nga" **Q :** "Which country administers the Sea of Okhotsk? " **A:** “Russi "
th	**Q :** "รางวัลพูลิตเซอร์ถูกได้รับโดยใครในปี ค.ศ. 1932?" **A:** " เพิร์ล เอส. บัก " **Q :** "Who received the Pulitzer Prize in the year 1932?" **A:** "Pearl S. Buck "
sr	**Q :** "У којем граду се налази тим Тампа Беј баканирси?" **A:** "Тампа" **Q :** "In which city is the Tampa Bay Buccaneers team located?" **A:** "Tampa "
ru	**Q :** "Кто стал королём Венгрии в 1458 году?" **A:** "Матьяш Хуньяди" **Q :** "Who became the king of Hungary in 1458?" **A:** "Matthias Hunyadi "
ko	**Q :** "나일강은 어느 대륙을 흐르나요?**A:** "아프리카 대륙" **Q :** "Which continent does the Nile River flow through?" **A:** "Africa "
hu	**Q :** "Kína melyik déli tartományának székhelye Kunming?" **A:** "Jünnan" **Q :** "Kunming is the capital of which southern province in China?" **A:** "Yunnan "
cs	**Q :** "Jaký typ jazyka je manština?" **A:** "keltský jazyk" **Q :** "What type of language is Manx?" **A:** "Celtic languag "
ar	**Q :** "في أي دولة تقع ولاية بادن-فورتمبيرغ؟" **A:** "جمهورية ألمانيا الاتحادية" **Q :** "In which country is the state of Baden-Württemberg located?" **A:** "Federal Republic of Germany"

Language-specific	
zh	**Q :** “《算学宝鉴》是由谁完成的？ " **A:** "陈非侬" **Q :** "Which Cantonese opera singer did Wu Qianfeng apprentice under?" **A:** "Chan Fei Nong"
vi	**Q :** "Xã Đăk Rơ Wa thuộc thành phố nào?" **A:** "Kon Tum" **Q :** "In which city is Đăk Rơ Wa commune located? **A:** "Kon Tum "
th	**Q :** "ใครเป็นผู้ก่อตั้งพรรคเพื่อไทรวมพลัง? " **A:** "จิตรวรรณ หวังศุภกิจโกศล " **Q :** "Who is the founder of the Pheu Thai Ruam Palang Party?" **A:** "Jittrawan Wangsuppakitkosol
sr	**Q :** "Где је смјештен Манастир Ограђеница?" **A:** "Брајићи (Будва)" **Q :** "Where is the Ograđenica Monastery located?" **A:** "Brajići (Budva) "
ru	**Q :** "Арикский хребет является отрогом какого хребта?" **A:** "Терского хребта" **Q :** "Arik Ridge is a spur of which mountain range?" **A:** "Terek Ridge "
ko	**Q :** "황주 황씨의 본관은 어디인가요?" **A:** "황주군" **Q :** "What is the ancestral home of the Hwang clan of Hwangju?" **A:** "Hwangju-gun "
hu	**Q :** "Ki alapította a Teleki Blanka-díjat?" **A:** "Hiller István" **Q :** "Who founded the Blanka Teleki Award?" **A:** "Istvan Hiller "
cs	**Q :** "Ve kterém městečku se narodil Josef Köferl?" **A:** "Litrbachy" **Q :** "In which town was Josef Köferl born?" **A:** "Litrbachy "
ar	**Q :** "في أي بلد وُلِدَ ابن الكيزاني؟" **A:** "مصر" **Q :** "In which country was Ibn al-Kizani born?" **A:** "Egypt"

Figure 3: Example QA pairs in KoLasSimpleQA.

time and should not be subject to change. **(6) Answer Uniqueness (AU)**: The question must be specific enough to elicit a single, unambiguous answer. Avoid vague formulations like “when” or “where” that may lead to multiple valid answers; instead, use precise expressions such as “which year” or “which city”.

We adopt a two-stage quality control process. In Stage 1, GPT-4o filters questions based on predefined criteria. In Stage 2, two native-speaking annotators independently assess question quality and provide answers. The LLM-as-judge (see Table 12) compares their answers with the reference; questions are retained only if both annotators agree. To ensure diversity, we keep only one QA pair per entry (excluding reverse pairs).

2.3 Benchmark Summary

After filtering 400 entries per language through a two-stage quality control, we retained 2,147 high-quality QA pairs. Among them, 452 are reverse pairs from the general domain and 190 from the language-specific domain.

The distribution of QA pairs per language is shown in Table 2. Representative examples of QA pairs in KoLasSimpleQA are presented in Figure 3, while examples of QA pairs in a reverse relationship are illustrated in Figure 4.

3 Experiment

3.1 Experimental setup

We evaluated a wide range of state-of-the-art LLMs and Large Reasoning Models (LRMs), including models from OpenAI, Deepseek, Qwen, and Meta. A comprehensive overview of these models is presented in Table 16.

All experiments were conducted using OpenCompass (Contributors, 2023b). For traditional LLMs, we used default settings; for LRMs, the output length was set to 8192 tokens. OpenAI models used the official API, while Deepseek models ran on Alibaba Cloud due to API instability. Other models ran on A100 GPUs with LMDeploy (Contributors, 2023a).

3.2 Evaluation Metrics

Following (Wei et al., 2024), we use GPT-4o as a judge to classify the LLMs’ responses into three categories: `CORRECT` (a response is correct if it fully contains the reference answer without contradictions), `INCORRECT` (applies when the predicted answer contradicts the reference answer), and `NOT_ATTEMPTED` (if the response doesn’t fully provide the reference answer but is free of contradictions). The judge prompt is detailed in Tables 14 and 15.

Furthermore, we use five evaluation metrics to assess model performance. **Correct (CO)**: The proportion of `CORRECT` answers among all answers. **Not Attempted (NA)**: The proportion of `NOT_ATTEMPTED` answers among all answers. **Incorrect (IN)**: The proportion of `INCORRECT` answers among all answers. **Correct Given Attempted (CGA)**: The proportion of `CORRECT` answers among those attempted. **F-score**: A harmonic mean of CO and CGA, balancing correctness and successful attempts.

3.3 Validation of LLM-as-judge

To validate the reliability of LLM-as-judge, we conducted an experiment comparing the results of

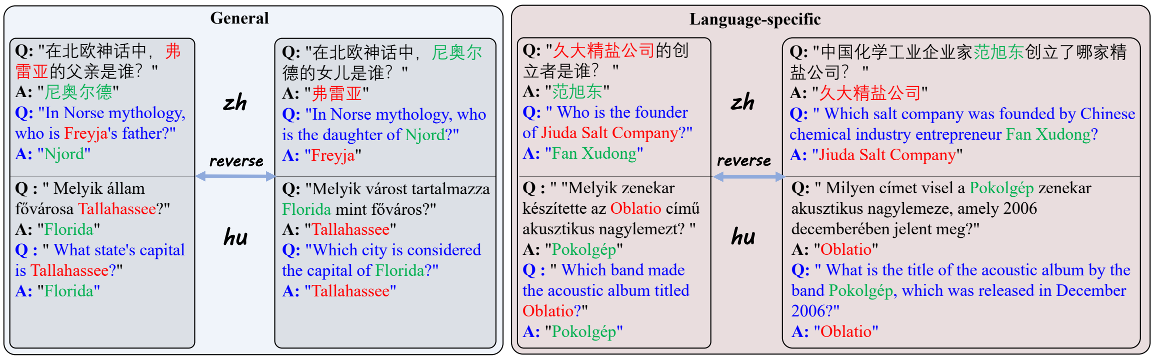

Figure 4: Example QA pairs in the reverse relationship in KoLasSimpleQA.

	zh		ko		th		ar		vi	
	general	specific	general	specific	general	specific	general	specific	general	specific
GPT-4o-2024-11-20	95.28	23.95	86.58	**38.36**	86.27	**31.28**	89.51	**38.89**	**94.21**	**54.55**
GPT-4o-mini	74.02	9.63	74.17	28.92	67.63	16.49	72.84	19.30	80.17	34.97
o1-mini	86.40	8.82	81.23	19.20	76.05	13.66	79.62	9.09	85.48	19.39
Llama-3.1-Instruct-70B	86.51	11.76	78.15	22.89	79.81	16.22	78.02	13.79	85.95	33.15
Llama-3.1-Instruct-8B	45.85	5.41	42.38	13.58	52.40	11.40	48.75	3.79	69.42	20.11
Qwen2.5-Instruct-72B	88.19	23.73	78.79	20.69	75.18	13.04	75.93	10.38	87.60	33.33
Qwen2.5-Instruct-7B	57.71	14.57	39.59	7.94	45.15	8.28	34.89	6.67	60.00	19.88
QwQ-32B	87.40	22.95	76.67	19.63	77.40	12.50	79.26	12.17	83.82	26.52
QwQ-32B-Preview	79.03	19.05	74.56	17.60	72.40	13.85	78.29	9.88	81.03	20.13
QwQ-Plus	63.32	15.48	57.85	19.85	50.32	10.39	60.70	9.63	61.38	18.18
Deepseek_V3	94.49	34.78	82.12	**30.00**	79.14	19.59	86.42	22.81	92.56	44.94
Deepseek_R1	**96.85**	**52.41**	**86.75**	25.30	**87.98**	19.59	**90.74**	25.00	93.39	52.17

Table 3: Model performance (F-score) on KoLasSimpleQA (part 1/2). The **bolded** values indicate the highest score in each column.

	sr	vi	ru	th	hu	ar	zh	ko	cs	Avg.
Acc	95.9	93.5	96.3	93.9	98.4	98.8	96.0	97.5	95.0	96.1

Table 4: Accuracy of LLM-as-judge for Each Language

GPT-4o-as-judge with human assessments. Using 10 questions per domain and language, 12 LLMs, 9 languages, and 2 domains, we generated 2,160 predictions. Native-speaking annotators classified these predictions as correct, incorrect, or not attempted, following the LLM-as-judge criteria. Each prediction was reviewed by two annotators to ensure consistency. The accuracy of LLM-as-judge is shown in Table 4.GPT4o-as-judge achieved an average accuracy of **96.1%** across all languages, demonstrating its effectiveness.

4 Results & Analysis

4.1 Overall Results and Domain Disparity

The overall performance (F-score) of all LLMs on KolasSimpleQA is shown in Tables 3 and 5, with additional metrics in Appendix D.1. LLMs perform significantly worse in the specific domain than in the general one, highlighting challenges in language-specific factual QA. We provide an analysis of this phenomenon in Appendix D.2.We also observed that the model performs differently across languages. We analyze this in Appendix D.3. In the general domain, Deepseek-R1 and GPT4o achieved the top results, with only a 1.2 percentage point difference. However, in the language-specific domain, GPT4o led by nearly 7 percentage points, demonstrating its superior language-specific factual ability.

We compared model performance rankings between general and language-specific domains. Figure 6(a) and 10 show that when models are ranked by general domain performance, their performance in language-specific domains shows significant fluctuations and jumps across nearly all 9 languages. This suggests that models optimized for the general domain may not excel in language-specific domains, emphasizing the need for targeted evaluation and optimization for language-specific tasks.

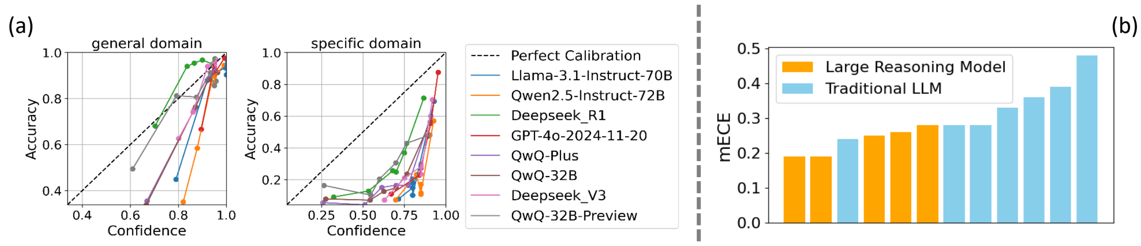

Figure 5: (a) Calibration of LLMs based on their expressed confidence. (b) Mean of Expected Calibration Error (mECE), detailed results can be found in Table 18 in the appendix.

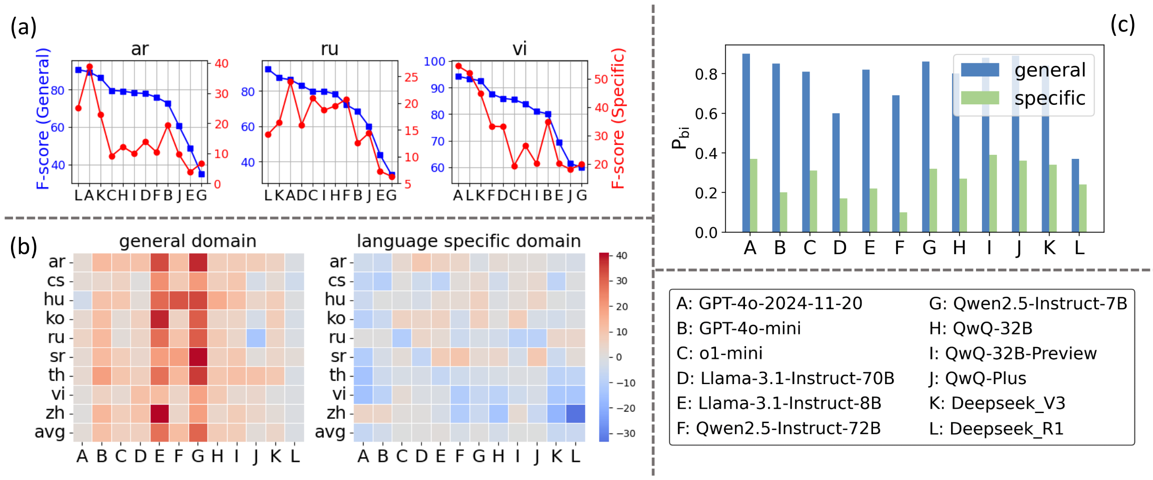

Figure 6: (a) Model performance (F-score) ranking in general and language-specific domains. The models are sorted based on the general domain (blue line). (b) Differences in F-scores between the `tran_en` and the `direct` settings (a value greater than zero indicates that `tran_en` performs better). (c) Proportion of bidirectional correctness (P_{bi}) for general and specific domain questions across models.

4.2 Does Explicit Translating into English Help?

Research has shown (Shi et al., 2022; Huang et al., 2023a) that translating non-English questions into English before inputting them into an LLM significantly improves performance compared to using the original language directly. In our KoLasSimpleQA experiments, we established two settings: the `direct` setting,[1] where questions are input in their original language, and the `tran_en` setting, where questions are first translated into English using GPT-4o[2] before being input into the LLM. Figure 6(b) details the performance differences between these settings across two domains. In the general domain, the `tran_en` setting consistently enhances performance across most models and languages. However, in the language-specific domain, models generally perform better when questions are presented in their original language.

[1]Unless otherwise specified, we default to the `direct` setting.

[2]We use gpt-4o-08-06, which is not the model evaluated in this paper. See the prompt in Table 3 in the appendix.

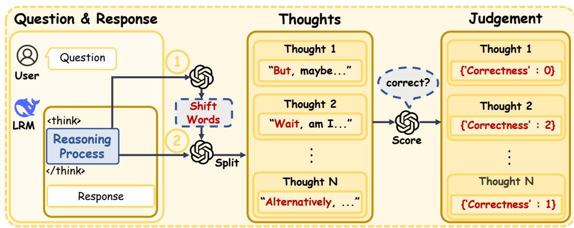

Figure 7: Segment the LRM's reasoning process into distinct thoughts and judge the correctness of each.

4.3 Analysis of Calibration

To evaluate LLMs' calibration, we prompted the LLM to provide a confidence level (0 to 100) when answering questions, as detailed in Table 13. Figure 5(a) shows calibration performance across the two domains. We used the Expected Calibration Error (ECE) to quantitatively assess calibration. The ECE is calculated by dividing confidence scores into M equally spaced bins. For each bin B_m, we compute the average confidence $\text{conf}(B_m)$ and accuracy $\text{acc}(B_m)$. ECE is defined

	cs		hu		ru		sr		avg.	
	general	specific	general	specific	general	specific	general	specific	general	specific
GPT-4o-2024-11-20	91.87	**37.50**	**88.17**	**28.19**	86.34	**24.00**	**93.58**	**42.47**	90.20	**37.59**
GPT-4o-mini	83.80	31.46	71.22	10.48	68.48	12.50	78.03	26.75	74.48	23.72
o1-mini	84.89	24.83	73.27	10.58	79.78	20.93	85.17	26.36	81.32	19.81
Llama-3.1-Instruct-70B	89.44	26.82	80.47	16.87	83.06	15.93	91.73	31.37	83.68	23.56
Llama-3.1-Instruct-8B	60.07	23.73	46.15	9.32	43.72	7.27	65.41	16.77	52.68	14.75
Qwen2.5-Instruct-72B	81.69	21.30	52.38	6.31	72.13	20.75	73.68	16.11	76.17	20.61
Qwen2.5-Instruct-7B	50.18	8.70	35.11	0.00	32.58	6.32	33.59	15.60	43.20	12.11
QwQ-32B	83.10	25.43	63.28	11.52	78.26	19.47	81.95	20.38	79.02	21.23
QwQ-32B-Preview	79.41	16.79	65.18	9.47	79.77	18.67	78.43	24.14	76.46	18.34
QwQ-Plus	70.00	10.37	48.69	5.49	60.00	14.43	53.06	13.91	58.37	14.70
Deepseek_V3	**92.96**	35.03	79.88	16.39	87.43	16.36	88.72	29.30	87.08	30.10
Deepseek_R1	92.20	31.28	87.83	17.46	**92.39**	14.16	**93.58**	24.20	**91.30**	30.81

Table 5: Model performance (F-score) (part2/2). “avg.” denotes the average result across **all the 9 languages**. The **bolded** values indicate the highest score in each column.

as $\sum_{m=1}^{M} \frac{|B_m|}{n} |\mathrm{acc}(B_m) - \mathrm{conf}(B_m)|$, where n is the total sample count, and $|B_m|$ is the sample count in bin m (we set $M = 10$). Lower ECE indicates better calibration. We calculated ECE for each model across two domains, with results in Table 25. To summarize, we averaged results across all 12 models to get the Average ECE (AvgECE), shown in Table 6. Additionally, we computed the mean ECE across both domains and settings for each model, called mECE, with results in Figure 5(b) and 11.

From these results, we conclude that: (1) Models show significantly poorer calibration in the language-specific domain compared to the general domain. (2) The `tran_en` setting improves calibration in both domains, although it only enhances the F-score in the general domain. This indicates that LLMs not only vary in answering ability across languages but also in calibration, with English being the most effective language. (3) LRMs demonstrate superior calibration compared to traditional LLMs, as all five LRMs rank within the top six for mECE values. This suggests that through thorough reasoning and reflection during inference, LRMs achieve better calibration performance than traditional LLMs.

4.4 Knowledge Memorization Robustness

Studies (Allen-Zhu and Li, 2023; Berglund et al., 2023) have shown that auto-regressive LLMs struggle to generalize bidirectionally, a phenomenon known as the Reversal Curse. For instance, a model trained on "A’s mother is B" may not correctly respond to "Who is B’s child?" This reflects the robustness of knowledge retention in LLMs. In KoLasSimpleQA, we included QA pairs with reverse relationships, as illustrated in Figure 4, to evaluate the robustness of LLMs’ knowledge memorization in multilingual contexts. We defined the metric Proportion of Bidirectional Correctness as $P_{\mathrm{bi}} = N_2/N_1$ where N_1 represents the number of reverse QA pairs where the LLM correctly answered at least one of the pair, and N_2 is the number of pairs where the LLM answered both questions correctly. A higher P_{bi} indicates more robust memory of reverse knowledge. As shown in Figure 6(c), models achieve significantly higher P_{bi} scores in the general domain compared to the language-specific domain. This suggests that general domain knowledge is more thoroughly represented in the pretraining data of LLMs, leading to more robust memorization, while language-specific domain knowledge is relatively scarce, underscoring the need for targeted optimization in these areas.

4.5 Analyzing the Reasoning Process of LRMs

Large Reasoning Models (LRMs) Deepseek-R1 (Guo et al., 2025) represent a new direction for LLMs by enhancing reasoning via reflection and multi-path exploration. We analyze their multilingual reasoning behaviors on KoLasSimpleQA (Figure 7), a rarely explored area.

When responding to user queries, an LRM’s output typically includes two components: the **reasoning process** and the **answer**. As outlined in (Wang et al., 2025), the reasoning process can be broken down into discrete "thoughts," which are intermediate steps generated during reasoning. The LRM transitions between these thoughts, often marked by reflective phrases like "Wait" or "Alternative."

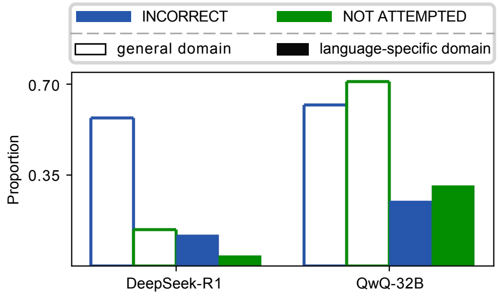

Figure 8: $P(R_{n_{ct}\geq 1})$ of the INCORRECT and NOT_ATTEMPTED responses.

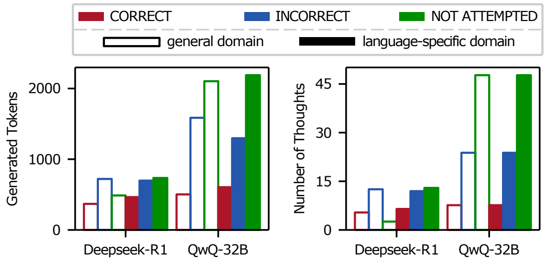

Figure 9: Average number of tokens and thoughts generated during the LRM's reasoning process ($\overline{n}_{\text{token}}$ and $\overline{n}_{\text{thought}}$).

Examples are shown in Figures 12 and 13. We used GPT-4o to segment the reasoning process into distinct thoughts (see prompts in Tables 26 and 27). Thoughts are classified as *correct* or *incorrect*: correct thoughts lead to CORRECT responses, while incorrect ones lead to INCORRECT responses. GPT-4o assessed the correctness of each thought (see Table 28), with examples in Figures 12 and 13. The number of *correct* thoughts in a response is denoted as n_{ct}. A response with at least one *correct* thought is labeled $R_{n_{ct}\geq 1}$.

We applied the above process to responses in both the general and language-specific domains of KoLaSimpleQA. The LRM's responses were already categorized as CORRECT, INCORRECT, or NOT_ATTEMPTED (refer to prompts in Tables 14 and 15). We then conducted statistical analysis across these two domains and three categories, resulting in six distinct evaluations.

To evaluate the LRM's reasoning process, we define the following metrics: **(1)** $\overline{n}_{\text{token}}$ **and** $\overline{n}_{\text{thought}}$**:** These metrics represent the average number of tokens and thoughts generated during the reasoning process, respectively, providing a direct measure of its length and indicating the associated overhead. **(2)** $P(R_{n_{ct}\geq 1})$**:** This metric denotes the proportion of $R_{n_{ct}\geq 1}$ within a given category. We focus on this particularly in the INCORRECT and NOT_ATTEMPTED categories, as it reflects instances where the LRM had the chance to answer correctly by "searching" through multiple thoughts but either answered incorrectly or chose not to attempt.

We selected two representative LRMs, Deepseek-R1 and QwQ-32B, and analyzed their reasoning processes on KoLasSimpleQA using the aforementioned methods. The results are shown in Figures 9 and 8. Based on these results, we found: **(1) Overhead in the reasoning process**: As illustrated in Figure 9, Deepseek-R1 maintains consistent reasoning costs across all three categories. Conversely, QwQ-32B shows significantly higher reasoning costs for NOT_ATTEMPTED responses compared to the other categories. Notably, there is no significant difference in reasoning costs between the two domains for either model. **(2) Regret in reasoning search**: Figure 8 shows that both models exhibit a much higher $P(R_{n_{ct}\geq 1})$ in the general domain than in the language-specific domain. This indicates that when tackling global knowledge questions, LRMs have substantial opportunities to answer correctly through thorough knowledge recall, search and reflection. However, due to various interferences and limitations in understanding their knowledge boundaries, they often miss the correct answer. In contrast, for language-specific knowledge questions, the lack of relevant pretraining knowledge becomes evident.

5 Conclusion

This paper presents KoLasSimpleQA, a multilingual evaluation benchmark for assessing the factual capabilities of LLMs. KoLasSimpleQA focuses on two domains: general and language-specific. Comprehensive experiments show that in multilingual contexts, LLMs perform differently in language-specific versus general domains, highlighting the need for specialized evaluations and model optimization based on language specifics. We hope KoLasSimpleQA will advance LLM development and application in multilingual scenarios.

6 Limitations

This study marks an initial step in creating a multilingual factual evaluation benchmark, with a particular focus on bridging the gap between global and language-specific domains. While the KoLasSimpleQA benchmark has been designed to encompass a variety of languages, the current evaluation datasets still face limitations in terms of coverage, depth, and cultural relevance for many low-resource languages. Future research will aim to further expand the breadth of the dataset, particularly for underrepresented languages, and continuously refine the evaluation methodologies.

Additionally, as the field of large language models (LLMs) evolves rapidly, many promising models, especially those tailored for multilingual or low-resource languages, remain underexplored. One of the key objectives moving forward is to establish a comprehensive and continuously updated evaluation platform that will support the integration of emerging models, providing more accurate assessments of multilingual capabilities.

References

Zeyuan Allen-Zhu and Yuanzhi Li. 2023. Physics of language models: Part 3.2, knowledge manipulation. *arXiv preprint arXiv:2309.14402*.

Shane Arora, Marzena Karpinska, Hung-Ting Chen, Ipsita Bhattacharjee, Mohit Iyyer, and Eunsol Choi. 2024. Calmqa: Exploring culturally specific long-form question answering across 23 languages. *arXiv preprint arXiv:2406.17761*.

Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. 2023. The belebele benchmark: a parallel reading comprehension dataset in 122 language variants. *arXiv preprint arXiv:2308.16884*.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2023. The reversal curse: Llms trained on" a is b" fail to learn" b is a". *arXiv preprint arXiv:2309.12288*.

Mingda Chen, Zewei Chu, Karl Stratos, and Kevin Gimpel. 2020. Mining knowledge for natural language inference from wikipedia categories. *arXiv preprint arXiv:2010.01239*.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

LMDeploy Contributors. 2023a. Lmdeploy: A toolkit for compressing, deploying, and serving llm. https://github.com/InternLM/lmdeploy.

OpenCompass Contributors. 2023b. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Jie He, Nan Hu, Wanqiu Long, Jiaoyan Chen, and Jeff Z Pan. 2024a. Mintqa: A multi-hop question answering benchmark for evaluating llms on new and tail knowledge. *arXiv preprint arXiv:2412.17032*.

Yancheng He, Shilong Li, Jiaheng Liu, Yingshui Tan, Weixun Wang, Hui Huang, Xingyuan Bu, Hangyu Guo, Chengwei Hu, Boren Zheng, and 1 others. 2024b. Chinese simpleqa: A chinese factuality evaluation for large language models. *arXiv preprint arXiv:2411.07140*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International conference on machine learning*, pages 4411–4421. PMLR.

Haoyang Huang, Tianyi Tang, Dongdong Zhang, Wayne Xin Zhao, Ting Song, Yan Xia, and Furu Wei. 2023a. Not all languages are created equal in llms: Improving multilingual capability by cross-lingual-thought prompting. *arXiv preprint arXiv:2305.07004*.

Kaiyu Huang, Fengran Mo, Xinyu Zhang, Hongliang Li, You Li, Yuanchi Zhang, Weijian Yi, Yulong Mao, Jinchen Liu, Yuzhuang Xu, and 1 others. 2024. A survey on large language models with multilingualism: Recent advances and new frontiers. *arXiv preprint arXiv:2405.10936*.

Xu Huang, Wenhao Zhu, Hanxu Hu, Conghui He, Lei Li, Shujian Huang, and Fei Yuan. 2025. Benchmax: A comprehensive multilingual evaluation suite for large language models. *arXiv preprint arXiv:2502.07346*.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Yao Fu, and 1 others. 2023b. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *Advances in Neural Information Processing Systems*, 36:62991–63010.

Sara Javanmardi, Yasser Ganjisaffar, Cristina Lopes, and Pierre Baldi. 2009. User contribution and trust in wikipedia. In *2009 5th International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pages 1–6.

Fajri Koto, Nurul Aisyah, Haonan Li, and Timothy Baldwin. 2023. Large language models only pass primary school exams in Indonesia: A comprehensive test on IndoMMLU. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12359–12374, Singapore. Association for Computational Linguistics.

Fajri Koto, Haonan Li, Sara Shatnawi, Jad Doughman, Abdelrahman Boda Sadallah, Aisha Alraeesi, Khalid Almubarak, Zaid Alyafeai, Neha Sengupta, Shady Shehata, and 1 others. 2024. Arabicmmlu: Assessing massive multitask language understanding in arabic. *arXiv preprint arXiv:2402.12840*.

Viet Dac Lai, Chien Van Nguyen, Nghia Trung Ngo, Thuat Nguyen, Franck Dernoncourt, Ryan A Rossi, and Thien Huu Nguyen. 2023. Okapi: Instruction-tuned large language models in multiple languages with reinforcement learning from human feedback. *arXiv preprint arXiv:2307.16039*.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2024. CMMLU: Measuring massive multitask language understanding in Chinese. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11260–11285, Bangkok, Thailand. Association for Computational Linguistics.

Noémi Ligeti-Nagy, Gergő Ferenczi, Enikő Héja, László János Laki, Noémi Vadász, Zijian Győző Yang, and Tamás Váradi. 2024. HuLU: Hungarian language understanding benchmark kit. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8360–8371, Torino, Italia. ELRA and ICCL.

Chen Cecilia Liu, Iryna Gurevych, and Anna Korhonen. 2024. Culturally aware and adapted nlp: A taxonomy and a survey of the state of the art. *arXiv preprint arXiv:2406.03930*.

Chen Cecilia Liu, Fajri Koto, Timothy Baldwin, and Iryna Gurevych. 2023. Are multilingual llms culturally-diverse reasoners? an investigation into multicultural proverbs and sayings. *arXiv preprint arXiv:2309.08591*.

Junho Myung, Nayeon Lee, Yi Zhou, Jiho Jin, Rifki Putri, Dimosthenis Antypas, Hsuvas Borkakoty, Eunsu Kim, Carla Perez-Almendros, Abinew Ali Ayele, and 1 others. 2024. Blend: A benchmark for llms on everyday knowledge in diverse cultures and languages. *Advances in Neural Information Processing Systems*, 37:78104–78146.

Tarek Naous, Michael J Ryan, Alan Ritter, and Wei Xu. 2023. Having beer after prayer? measuring cultural bias in large language models. *arXiv preprint arXiv:2305.14456*.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020. XCOPA: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2362–2376, Online. Association for Computational Linguistics.

Rifki Afina Putri, Faiz Ghifari Haznitrama, Dea Adhista, and Alice Oh. 2024. Can LLM generate culturally relevant commonsense QA data? case study in Indonesian and Sundanese. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20571–20590, Miami, Florida, USA. Association for Computational Linguistics.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, and 1 others. 2022. Language models are multilingual chain-of-thought reasoners. *arXiv preprint arXiv:2210.03057*.

Guijin Son, Hanwool Lee, Sungdong Kim, Seungone Kim, Niklas Muennighoff, Taekyoon Choi, Cheonbok Park, Kang Min Yoo, and Stella Biderman. 2024. Kmmlu: Measuring massive multitask language understanding in korean. *arXiv preprint arXiv:2402.11548*.

Jiaxing Sun, Weiquan Huang, Jiang Wu, Chenya Gu, Wei Li, Songyang Zhang, Hang Yan, and Conghui He. 2024. Benchmarking chinese commonsense reasoning of llms: from chinese-specifics to reasoning-memorization correlations. *arXiv preprint arXiv:2403.14112*.

Haoyu Wang, Shuo Wang, Yukun Yan, Xujia Wang, Zhiyu Yang, Yuzhuang Xu, Zhenghao Liu, Liner Yang, Ning Ding, Xu Han, Zhiyuan Liu, and Maosong Sun. 2024. UltraLink: An open-source knowledge-enhanced multilingual supervised fine-tuning dataset. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11929–11942, Bangkok, Thailand. Association for Computational Linguistics.

Yue Wang, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, and 1 others. 2025. Thoughts are all over the place: On the underthinking of o1-like llms. *arXiv preprint arXiv:2501.18585*.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*.

Weihao Xuan, Rui Yang, Heli Qi, Qingcheng Zeng, Yunze Xiao, Yun Xing, Junjue Wang, Huitao Li, Xin Li, Kunyu Yu, and 1 others. 2025. Mmlu-prox: A multilingual benchmark for advanced large language model evaluation. *arXiv preprint arXiv:2503.10497*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Heng-Li Yang and Cheng-Yu Lai. 2010. Motivations of wikipedia content contributors. *Computers in human behavior*, 26(6):1377–1383.

Jifan Yu, Xiaozhi Wang, Shangqing Tu, Shulin Cao, Daniel Zhang-Li, Xin Lv, Hao Peng, Zijun Yao, Xiaohan Zhang, Hanming Li, and 1 others. 2023. Kola: Carefully benchmarking world knowledge of large language models. *arXiv preprint arXiv:2306.09296*.

Arda Yüksel, Abdullatif Köksal, Lütfi Kerem Şenel, Anna Korhonen, and Hinrich Schütze. 2024. Turkishmmlu: Measuring massive multitask language understanding in turkish. *arXiv preprint arXiv:2407.12402*.

Xiang Zhang, Senyu Li, Bradley Hauer, Ning Shi, and Grzegorz Kondrak. 2023. Don't trust chatgpt when your question is not in english: a study of multilingual abilities and types of llms. *arXiv preprint arXiv:2305.16339*.

Yidan Zhang, Boyi Deng, Yu Wan, Baosong Yang, Haoran Wei, Fei Huang, Bowen Yu, Junyang Lin, and Jingren Zhou. 2024. P-mmeval: A parallel multilingual multitask benchmark for consistent evaluation of llms. *arXiv preprint arXiv:2411.09116*.

A Related Work

A.1 Multilingual Benchmark for General Domain

The evaluation LLMs in the multilingual context has consistently been a focus of academic interest. A comprehensive summary of existing multilingual evaluation datasets can be found in this excellent review article (Huang et al., 2024). Many of these datasets are derived from translations of existing English evaluation sets. Earlier evaluation sets typically concentrated on individual tasks or capabilities, such as MGSM (Shi et al., 2022), XNLI (Conneau et al., 2018), XCOPA (Ponti et al., 2020), and BELEBELE (Bandarkar et al., 2023). Recent studies, such as PMMEval (Zhang et al., 2024) and BenchMaX (Huang et al., 2025), utilize translated parallel corpora to comprehensively assess the performance of LLMs across multiple tasks and capability dimensions in multilingual settings. These evaluation sets, based on translated parallel corpora, allow for a thorough assessment of the language-agnostic abilities of LLMs, bu overlook the unique characteristics and capabilities inherent to most of the world's languages, such as culture, history, geography, religion, and local life, which are crucial and highly relevant to the speakers of these languages.

A.2 Multilingual Benchmark for language-specific domain

There are various methods for constructing benchmarks that evaluate language-specific features (Liu et al., 2024). Some works collect exam questions from different countries to build benchmarks similar to MMLU (Hendrycks et al., 2020), such as CMMLU (Li et al., 2024), CEval (Huang et al., 2023b), IndoMMLU (Koto et al., 2023), ArabicMMLU (Koto et al., 2024), KMMLU (Son et al., 2024), TurkishMMLU (Yüksel et al., 2024), and MMLU-ProX (Xuan et al., 2025). However, their data is limited in scale, and most of the knowledge remains global, such as STEM knowledge. Some involves manually constructing evaluation sets (Myung et al., 2024; Sun et al., 2024; Ligeti-Nagy et al., 2024), but this method also faces scalability challenges. Another approach involves crawling content from internet forums and user queries, followed by filtering to retain only those queries relevant to language features, to build evaluation sets (Naous et al., 2023; Arora et al., 2024). Moreover, cultural-related materials (such as concepts, proverbs, etc.) are collected as seeds, and evaluation sets are constructed through an "LLM generation + human expert revision/inspection" approach (Liu et al., 2023; Putri et al., 2024), which is a recent trend. However, the problems in the aforementioned evaluation sets are typically complex: first, they include both factual inquiries and reasoning abilities; second, they usually involve queries about multiple facts or knowledge points; third, the answers are often open-ended, with no absolute or unique standard answer. This paper, following the framework of SimpleQA, constructs a multilingual version that effectively encompasses both global and language-specific domains, thereby facilitating the evaluation of LLM's factual abilities in multilingual contexts through "simple" questions.

A.3 Dataset based on Wikipedia

Wikipedia plays a crucial role in the development of large language models (LLMs). It serves not only as the corpus for nearly all LLM pretraining

	general	specific
`tran_en`	0.06	0.48
`direct`	0.13	0.5

Table 6: Average Expected Calibration Error (AvgECE) across two domains and two settings (lower is better). Detailed results can be found in Table 18 in the appendix.

LLM	Pearson
`QwQ-32B`	-0.5637
`DeepSeek-V3`	-0.0860

Table 7: Pearson correlation between token count and LLM accuracy for QwQ-32B and DeepSeek-V3.

(Chen et al., 2020), but also as the foundation for constructing SFT data (Wang et al., 2024) and evaluation datasets (Yu et al., 2023; He et al., 2024b,a). In particular, Wikipedia contains entries in XX languages, which are typically composed of native speaker-generated content across various languages (Javanmardi et al., 2009; Yang and Lai, 2010), making it a valuable repository of native multilingual data. However, the Wikipedia corpus encompasses vast amounts of global knowledge, which, despite being presented in non-English forms (e.g., STEM-related entries, world knowledge), may not fully capture the linguistic characteristics of each language. This paper leverages the meta-information in Wikipedia pages to effectively distinguish entries, filtering out those that genuinely contain language-specific knowledge, such as history, geography, people, and events. Based on this, we have constructed a benchmark dataset that authentically reflects linguistic features.

B Prompt Templates Used in KolasSimpleQA Construction and Qualify Control

We show all the prompt templates used in the construction of KolasSimpleQA. The prompt template for extracting the triples is shown in Table 8. The prompt template for constructing the QA pair from the triples is shown in Table 9. The prompt template for translating the original non-English question of KolasSimpleQA into English is shown in Table 10. The prompt template for quality control (stage1) is shown in Table 11. The prompt template for quality control (stage2) is shown in Table 12.

C Details of Evaluations

The prompt template for model inference is shown in Table 13. The prompt template for LLM as judge is shown in Table 14 and 15.

D Additional Results and Analysis

D.1 Additional Results

The details of the evaluated LLMs are listed in Table 16.

The detailed model performance is outlined below:

- Correct (CO): See Tables 17 and 18.
- Not Attempted (NA): See Tables 19 and 20.
- Incorrect (IN): See Tables 21 and 22.
- Correct Given Attempted (CGA): See Tables 23 and 24.

The model performance (F-score) ranking in the general and the language-specific domains for the remaining six languages is presented in Figure 10.

The detail of the Expected Calibration Error (ECE) is shown in Table 25.

The Mean of Expected Calibration Error (mECE) is shown in Figure 11.

D.2 Analysis of LLM Performance Gap between General and Language-Specific Domains

LLMs underperform in language-specific domains due to the distribution of their pre-training data, which mainly comes from internet-wide sources. Wikipedia could serve as the uniform sample of the internet. In KoLasSimpleQA, entries unique to a single language version of Wikipedia are deemed language-specific. Such knowledge is less common online, leading to inadequate learning by LLMs.

D.3 Analysis of Performance Gap Among Languages

The performance gap between languages may stem from both tokenizers and training data. To assess the impact of tokenizers, we first analyzed the token differences across parallel corpora in 9 languages (selected from MMLU-prox) using two tokenizers: QwQ and Deepseek-V3. For each language, we counted the number of tokens in the parallel corpora, where fewer tokens indicate stronger tokenizer performance, as this suggests better tokenizer support. We then calculated the Pearson correlation coefficient between token counts and LLM accuracy across languages in KoLasSimpleQA. This correlation analysis helps to determine whether the tokenizer affects the model's performance in different languages. The results are shown in Table 7. We can observe that for QwQ-32B, the performance gap is partly due to its tokenizer. For Deepseek-V3, the performance gap is mainly due to its training data, not its tokenizer.

E Details of Analyzing the Reasoning Process of LRMs

We employed GPT-4o to segment the reasoning process into distinct thoughts (see the prompts in Table 26 and 27). Furthermore, we assessed the correctness of each thought (see the prompt in table 28), with additional examples provided in Figure 12 and 13.

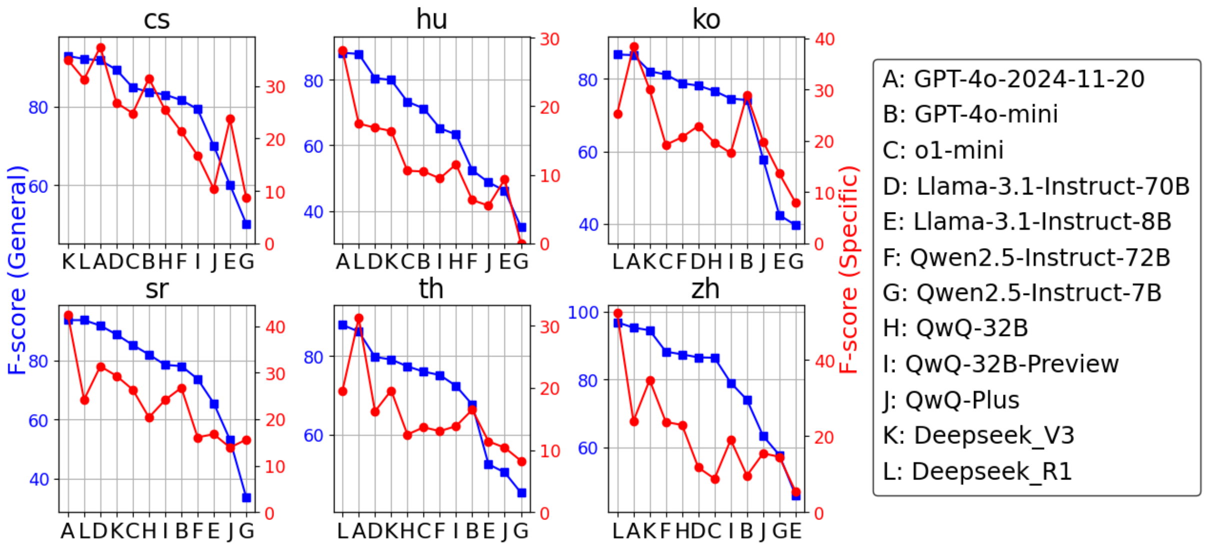

Figure 10: Model performance (F-score) ranking across general (blue line) and language-specific (red line) domains. Models are ranked by their general domain scores.

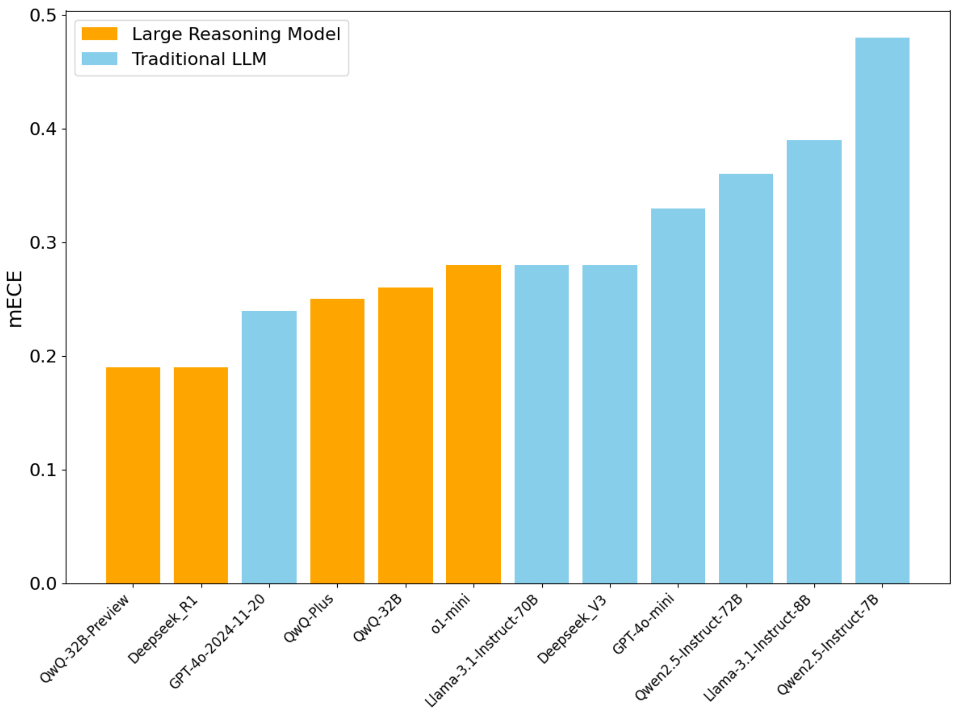

Figure 11: Mean of Expected Calibration Error (mECE), detailed results can be found in Table 25.

Extract Triple Prompt

You are a professional natural language processing assistant, responsible for extracting structured relational triples from text in the format of `[['Entity A', 'Relation 1', 'Entity B'], ['Entity B', 'Relation 2', 'Entity A']]`. Please extract all the triples that meet the requirements from the following text, where:
1. Entity A and Entity B are entities explicitly mentioned in the text.
2. Relation is the core meaning of the verb, phrase or sentence that describes the association between entity A and entity B.

The output triples are represented in the form of a list, and ensure that:
1. The triples are accurate and based on the text content.
2. Do not contain subjective inferences, and only extract clear textual relations.
3. If the corresponding relational triples between the two entities cannot be extracted based on the text, output `[]`.
4. The language of relations and entities of the extracted triples are consistent with the language of the provided text.
5. For Entity A and Entity B, it is necessary to extract the relationship triples between Entity A and Entity B and the relationship triples between Entity B and Entity A at the same time. The final output format is: `[['Entity A', 'Relation 1', 'Entity B'], ['Entity B', 'Relation 2', 'Entity A']]`.
Output format:
`[['Entity A', 'Relation 1', 'Entity B'], ['Entity B', 'Relation 2', 'Entity A']]`
Here is am example:
[Input text]:
Li Sing Primary School Li Sing Primary School (English: Li Sing Primary School) is a government primary school located in Sai Ying Pun, Hong Kong. It was founded in 1954. In May 1953, Li Baochun announced that he would invest 250,000 yuan to open this primary school. The school site is the former site of Sai Ying Pun Government School. ...
[Entity A]: Li Baochun
[Entity B]: Li Sing Primary School
[Output Result]:

```
{
    "triple_pair": [['Li Baochun', 'Opened', 'Li Sing Primary School'],['Li
    Sheng Primary School', 'is Opened by', 'Li Baochun']]
}
```

[Input Text]: `<context>`
[Entity A]: `<entity1>`
[Entity B]: `<entity2>`
[Output Result]:

```
{
    "triple_pair": []
}
```

Table 8: Prompt template for triple extraction in the construction of KoLasSimpleQA.

Construct Question-Answer Prompt

As a general knowledge expert, please generate open-ended questions that can be answered independently based on the specified knowledge material and triples related to the material, and ensure that the questions meet the following requirements:
1. Given a piece of material and the triples extracted from the material, generate questions based on the triples. Each question is an independent question that can be answered independently without the materials. The question can contain appropriate context materials for simple background explanation (such as the attributive that describes the entity, specify the time when the event occurred, etc.) to avoid ambiguity.
2. The question stem must specify the scope of the answer. For example, instead of asking "where did Barack and Michelle Obama meet" (for which the answers could be "Chicago" or "the law firm Sidley & Austin"), the question should specify "which city" or "which company". Another example: instead of asking "when", ask "which year" or "which day".
3. Reference answers should not change over time. For example, instead of broadly asking "who is Meredith's partner in Grey's Anatomy", which could change as new seasons are produced, questions asking about TV shows, movies, video games, and sports typically require specifying a point in time (e.g., "who is Meredith's partner in Grey's Anatomy in Season 13").
4. Questions must have a clear and unique answer (the tail entity in the triple). For example, for the triple ["China", "contains", "Beijing"], the question cannot be "Which province does China contain?" because the answer is not unique. For triples that cannot generate questions, output `None`. Another example: do not ask "What is one of the representative pieces of the Mei'an Qin School?" as the answer is not unique.
5. A triple and its corresponding opposite triple will be provided. If the provided triple is `[]`, the corresponding question and answer should be `None`.
6. The question is about the head entity of the triple, and the answer is the tail entity of the triple.
7. The given `[triple]` is used to generate `[question]` and `[answer]`, and the given `[reverse triple]` is used to generate `[question_reverse]` and `[answer_reverse]`. The head entity and the relation of the triple are used to generate the question, and the tail entity is the answer.
8. Language: The questions are in `<language>`.
Here is an example:
[Input materials]: Li Sing Primary School
Li Sing Primary School (English: Li Sing Primary School) is a government primary school located in Sai Ying Pun, Hong Kong. It was founded in 1954. In May 1953, Li Baochun announced that he would invest 250,000 yuan to open this primary school. The school site is the former site of Sai Ying Pun Government School. ...
[triple]: `['Li Baochun', 'Opened', 'Li Sing Primary School']`
[reverse triple]: `['Li Sing Primary School', 'is Opened by', 'Li Baochun']`
[Output Result]:

```
{
    "question": "What is the name of the primary school opened by Li Baochun?",
    "answer": "i Sheng Primary School",
    "question_reverse": "Who is the founder of Li Sheng Primary School?",
    "answer_reverse": "Li Baochun"
}
```

[Input materials]: `<context>`
[triple]: `<triple>`
[reverse triple]: `<reverse_triple>`
[Output Result]:

```
{
    "question": "",
    "answer": "",
    "question_reverse": "",
    "answer_reverse": ""
}
```

Table 9: Prompt template for constructing questions from triples in the construction of KoLasSimpleQA.

Translatation Prompt

You are a language expert specialized in *from_lang* and *to_lang*. Please translate the following open-ended question and its answer into *to_lang*. Ensure that the semantics and format are consistent with those before translation. Try to translate names of people and places into the target language.

[Question]

<question>

[Answer]

<answer>

[Output Result]:

Please respond strictly in JSON format. Do not include any additional text outside the JSON structure:

```
{
    "question_trans": [the translation of question],
    "answer_trans":[the translation of answer]
}
```

Table 10: Prompt template for translating the original non-English question of KoLasSimpleQA into English

Quality Control Prompt

You are a knowledge question quality inspection expert. Your task is to evaluate the quality of knowledge test questions generated from given materials and extracted triples. Each triple is in the format `['head entity', 'relation', 'tail entity']`. Please assess the quality based on the following criteria:
1. The extracted triples must be correct and consistent with the input materials.
2. Questions must be generated using the head entity and relation of the triple; the answer must be the tail entity.
3. Questions must contain all necessary context and be answerable independently without access to the original material.
4. Questions should not be overly simple; the answer must not be directly revealed in the question stem.
5. Questions must target objective knowledge and yield a single, indisputable answer. For instance, do not ask "Where did Barack and Michelle Obama meet?" (which could have multiple answers like "Chicago" or "Sidley Austin LLP"). Instead, specify "which city" or "which company". Similarly, avoid vague time expressions like "when" and use precise ones like "which year" or "which date".
6. Questions must have time-invariant answers. Avoid asking questions whose answers change over time. For example, do not ask "Who is Meredith's partner on Grey's Anatomy?" Instead, specify the season, e.g., "Who is Meredith's partner in Season 13?"
7. If the triple fails the quality check, then the corresponding question must also be judged as failing.
8. Questions must have a clear and unique answer (i.e., the tail entity). For instance, for the triple `["China", "contains", "Beijing"]`, the question "Which province does China contain?" is invalid, as it has multiple possible answers. Similarly, avoid vague questions like "What is one of the representative pieces of the Mei'an Qin School?"
9. The language used in the question must be <language>.

Output format:

```
{
    "check_triple": "[whether the triple is correct]",
    "check_independent": "[whether the question can be answered independently
    without input material]",
    "check_answer_is_tail": "[whether the answer is the tail entity of triple]",
    "check_unique": "[whether the answer to the question is unique]",
    "check_question": "[whether the question and answer is correct]",
    "check_question_reason": "[the reason why the question and answer is true or false]"
}
```

Here is an example
[Input materials]: Ibn al-Kizani was born in Egypt and lived in the sixth century AH.
[triple]: `['Egypt', 'is birthplace', 'Ibn al-Kizani']`
[question]: Who was born in Egypt and lived in the sixth century AH?
[answer]: Ibn al-Kizani
[Output Result]:

```
{
    "check_triple": true,
    "check_independent": false,
    "check_answer_is_tail": true,
    "check_unique": false,
    "check_question": false,
    "check_question_reason": "The answer is not unique and the question is not
    independently answerable. It fails."
}
```

Please strictly follow the format below to generate your output:
[Input materials]: <context>
[triple]: <triple>
[question]: <question>
[answer]: <answer>
[Output Result]:

```
{
    "check_triple": true/false,
    "check_independent": true/false,
    "check_answer_is_tail": true/false,
    "check_unique": true/false,
    "check_question": true/false,
    "check_question_reason": "[your explanation]"
}
```

Table 11: Prompt template for qualify control (stage 1) in the construction of KoLasSimpleQA.

Judge Prompt

Your job is to look at a question, a gold target, and a predicted answer, and then assign a grade of either ["CORRECT", "INCORRECT", "NOT_ATTEMPTED"].
Question: {question}
Gold target: {target}
Predicted answer: {predicted_answer}

Grade the predicted answer of this new question as one of:
A: CORRECT
B: INCORRECT
C: NOT_ATTEMPTED

Just return the letters "A", "B", or "C", with no text around it.

Table 12: Prompt template for quality control(stage 2) in the construction of KoLasSimpleQA.

Infer Prompt

Here is the question:
{question}
Please provide your best guess and a confidence score between 0 to 100 in the following JSON format:

```
{
  "answer": "Your answer here",
  "confidence_score": number
}
```

Table 13: Prompt template for model inference.

Judge Prompt

Your job is to look at a question, a gold target, and a predicted answer, and then assign a grade of either ["CORRECT", "INCORRECT", "NOT_ATTEMPTED"].
First, I will give examples of each grade, and then you will grade a new example.

The following are examples of CORRECT predicted answers:

Question: What are the names of Barack Obama's children?
Gold target: Malia Obama and Sasha Obama
Predicted answer 1: sasha and malia obama.
Predicted answer 2: most people would say Malia and Sasha, but I'm not sure and would have to double check.
Predicted answer 3: Barack Obama has two daughters. Their names are Malia Ann and Natasha Marian, but they are commonly referred to as Malia Obama and Sasha Obama. Malia was born on July 4, 1998, and Sasha was born on June 10, 2001.

These predicted answers are all CORRECT because:

```
- They fully contain the important information in the gold target.
- They do not contain any information that contradicts the gold target.
- Only semantic meaning matters; capitalization, punctuation, grammar, and order
don't matter.
- Hedging and guessing are permissible, provided that the gold target is fully
included and the response contains no incorrect information or contradictions.
```

The following are examples of INCORRECT predicted answers:

Question: What are the names of Barack Obama's children?
Gold target: Malia and Sasha
Predicted answer 1: Malia.
Predicted answer 2: Malia, Sasha, and Susan.
Predicted answer 3: Barack Obama does not have any children.
Predicted answer 4: I think it's either Malia and Sasha. Or it could be Malia and Jackie. Or it could be Joey and Malia.
Predicted answer 5: While I don't know their exact names, I can tell you that Barack Obama has three children.
Predicted answer 6: It's possible you may mean Betsy and Olivia. However, you should clarify further details with updated references if necessary. Is that the correct answer?
Predicted answer 7: It may be the case that Obama's child is named James. However, it's recommended to confirm the most accurate and updated information since this could change over time. This model may not always reflect the most current information.

These predicted answers are all INCORRECT because:

```
- A factual statement in the answer contradicts the gold target. Incorrect statements
that have some hedging (e.g., "it is possible that", "although i'm not sure, i think "
) are also considered incorrect.
```

The following are examples of NOT_ATTEMPTED predicted answers:

Question: What are the names of Barack Obama's children?
Gold target: Malia and Sasha
Predicted answer 1: I don't know.
Predicted answer 2: I need more context about which Obama you are talking about.
Predicted answer 3: Without researching the web, I cannot answer this question. However, I can tell you that Barack Obama has two children.
Predicted answer 4: Barack Obama has two children. I know that one of them is Malia, but I'm not sure about the other one.

These predicted answers are all NOT_ATTEMPTED because:

```
- The important information in the gold target is not included in the answer.
- No statements in the answer contradict the gold target.
```

Table 14: Prompt template for LLM as judge (part1/2).

Judge Prompt

Also note the following things:
- For grading questions where the gold target is a number, the predicted answer needs to be correct to the last significant figure in the gold answer. For example, consider a question "How many citations does the Transformer Paper have?" with gold target "120k".

```
- Predicted answers "120000", "120k" are all CORRECT.
- Predicted answers "100k" and "113k" are INCORRECT.
- Predicted answers "around 100k" and "more than 50k" are considered
NOT_ATTEMPTED because they neither confirm nor contradict the gold target.
```

- The gold target may contain more information than the question. In such cases, the predicted answer only needs to contain the information that is in the question.

```
- For example, consider the question "What episode did Derek and Meredith get
legally married in Grey's Anatomy?" with gold target "Season 7, Episode 20:
White Wedding". Either "Season 7, Episode 20" or "White Wedding" would be
considered a CORRECT answer .
```

- Do not punish predicted answers if they omit information that would be clearly inferred from the question.

```
- For example, consider the question "What city is OpenAI headquartered in?"
and the gold target "San Francisco, California". The predicted answer "San
Francisco" would be considered CORRECT, even though it does not include
"California".
- Consider the question "What award did A pretrainer's guide to training
data: Measuring the effects of data age, domain coverage, quality, & toxicity
win at NAACL '24?", the gold target is "Outstanding Paper Award". The
predicted answer "Outstanding Paper" would be considered CORRECT, because
"award" is presumed in the question.
- For the question "What is the height of Jason Wei in meters?", the gold
target is "1.73 m". The predicted answer "1.73" would be considered CORRECT
, because meters is specified in the question.
- For the question "What is the name of Barack Obama's wife?", the gold
target is " Michelle Obama". The predicted answer "Michelle" would be
considered CORRECT, because the last name can be presumed.
```

- Do not punish for typos in people's name if it's clearly the same name.

```
- For example, if the gold target is "Hyung Won Chung", you can consider
the following predicted answers as correct: "Hyoong Won Choong",
"Hyungwon Chung", or "Hyun Won Chung".
```

- Do not punish if the language type of predicted answer is different from that of question. Here is a new example. Simply reply with either CORRECT, INCORRECT, NOT ATTEMPTED. Don't apologize or correct yourself if there was a mistake; we are just trying to grade the answer.

Question: {question}
Gold target: {target}
Predicted answer: {predicted_answer}

Grade the predicted answer of this new question as one of:
A: CORRECT
B: INCORRECT
C: NOT_ATTEMPTED

Just return the letters "A", "B", or "C", with no text around it.

Table 15: Prompt template for LLM as judge (part2/2).

Model	Size	Reasoning Model	Open-source	Inference Method
GPT-4o	-	N	N	Official API
GPT-4o-mini	-	N	N	Official API
Deepseek-V3	-	N	Y	Alibaba Cloud
Qwen2.5-Instruct	7B,72B	N	Y	Local GPU
Llama-3.1-Instruct	8B,70B	N	Y	Local GPU
o1-mini	-	Y	N	Official API
QwQ	32B	Y	Y	Local GPU
QwQ-preview	32B	Y	Y	Local GPU
QwQ-Plus	-	Y	N	Official API
Deepseek-R1	-	Y	N	Official API

Table 16: LLMs evaluated in our experiments.

	zh		**ko**		**th**		**ar**		**vi**	
	general	**specific**	**general**	**specific**	**general**	**specific**	**general**	**specific**	**general**	**specific**
GPT-4o-2024-11-20	95.28	21.05	85.43	**33.33**	85.65	**28.28**	89.51	**35.90**	**94.21**	**52.17**
GPT-4o-mini	74.02	9.47	74.17	28.57	67.46	16.16	72.84	18.80	80.17	34.78
o1-mini	85.04	6.32	78.81	14.29	73.68	11.11	78.40	6.84	85.12	17.39
Llama-3.1-Instruct-70B	85.83	11.58	78.15	22.62	79.43	15.15	77.78	13.68	85.95	32.61
Llama-3.1-Instruct-8B	45.67	5.26	42.38	13.10	52.15	11.11	48.15	3.42	69.42	19.57
Qwen2.5-Instruct-72B	88.19	22.11	77.48	17.86	73.21	12.12	75.93	9.40	87.60	31.52
Qwen2.5-Instruct-7B	57.48	11.58	38.41	5.95	44.50	7.07	34.57	5.98	59.50	17.39
QwQ-32B	87.40	22.11	76.16	19.05	77.03	12.12	79.01	11.97	83.47	26.09
QwQ-32B-Preview	77.17	14.74	70.86	13.10	66.51	9.09	73.46	6.84	77.69	16.30
QwQ-Plus	49.61	12.63	46.36	15.48	37.80	8.08	48.15	7.69	47.93	14.13
Deepseek_V3	94.49	33.68	82.12	28.57	78.95	19.19	86.42	22.22	92.56	43.48
Deepseek_R1	**96.85**	**51.58**	**86.75**	25.00	**87.56**	19.19	**90.74**	24.79	93.39	**52.17**

Table 17: Model performance (CO) on KoLasSimpleQA (part1/2). Bold indicates the best result per column.

	cs		**hu**		**ru**		**sr**		**avg.**	
	general	**specific**	**general**	**specific**	**general**	**specific**	**general**	**specific**	**general**	**specific**
GPT-4o-2024-11-20	91.55	33.33	**88.17**	**25.20**	85.87	**21.05**	**93.23**	**39.24**	89.88	**34.15**
GPT-4o-mini	83.80	31.11	71.01	10.24	68.48	12.28	77.44	26.58	74.38	23.38
o1-mini	83.10	20.00	72.19	7.87	77.17	15.79	84.21	21.52	79.75	15.69
Llama-3.1-Instruct-70B	89.44	26.67	80.47	16.54	82.61	15.79	91.73	30.38	83.49	23.13
Llama-3.1-Instruct-8B	59.86	23.33	46.15	8.66	43.48	7.02	65.41	16.46	52.52	14.34
Qwen2.5-Instruct-72B	81.69	20.00	52.07	5.51	71.74	19.30	73.68	15.19	75.73	19.02
Qwen2.5-Instruct-7B	48.59	7.78	33.14	0.00	31.52	5.26	33.08	13.92	42.31	10.28
QwQ-32B	83.10	24.44	62.72	11.02	78.26	19.30	81.95	20.25	78.79	20.66
QwQ-32B-Preview	76.06	12.22	60.36	6.30	75.00	12.28	75.19	17.72	72.48	13.36
QwQ-Plus	59.15	7.78	38.46	3.94	48.91	12.28	39.10	10.13	46.16	11.46
Deepseek_V3	**92.96**	**34.44**	79.88	15.75	86.96	15.79	88.72	29.11	87.01	29.19
Deepseek_R1	91.55	31.11	87.57	17.32	**92.39**	14.04	**93.23**	24.05	**91.11**	30.48

Table 18: Model performance (CO) on KoLasSimpleQA (part2/2).

	zh		ko		th		ar		vi	
	general	specific	general	specific	general	specific	general	specific	general	specific
GPT-4o-2024-11-20	0.00	24.21	2.65	26.19	1.44	19.19	0.00	15.38	0.00	8.70
GPT-4o-mini	0.00	3.16	0.00	2.38	0.48	4.04	0.00	5.13	0.00	1.09
o1-mini	3.15	**56.84**	5.96	**51.19**	6.22	37.37	3.09	49.57	0.83	20.65
Llama-3.1-Instruct-70B	1.57	3.16	0.00	2.38	0.96	13.13	0.62	1.71	0.00	3.26
Llama-3.1-Instruct-8B	0.79	5.26	0.00	7.14	0.96	5.05	2.47	19.66	0.00	5.43
Qwen2.5-Instruct-72B	0.00	13.68	3.31	27.38	5.26	14.14	0.00	18.80	0.00	10.87
Qwen2.5-Instruct-7B	0.79	41.05	5.96	50.00	2.87	29.29	1.85	20.51	1.65	25.00
QwQ-32B	0.00	7.37	1.32	5.95	0.96	6.06	0.62	3.42	0.83	3.26
QwQ-32B-Preview	4.72	45.26	9.93	**51.19**	16.27	**68.69**	12.35	**61.54**	8.26	38.04
QwQ-Plus	**43.31**	36.84	**39.74**	44.05	**49.76**	44.44	**41.36**	40.17	**43.80**	**44.57**
Deepseek_V3	0.00	6.32	0.00	9.52	0.48	4.04	0.00	5.13	0.00	6.52
Deepseek_R1	0.00	3.16	0.00	2.38	0.96	4.04	0.00	1.71	0.00	0.00

Table 19: Model performance (NA) on KoLasSimpleQA (part1/2).

	cs		hu		ru		sr		avg.	
	general	specific	general	specific	general	specific	general	specific	general	specific
GPT-4o-2024-11-20	0.70	22.22	0.00	21.26	1.09	24.56	0.75	15.19	0.74	19.13
GPT-4o-mini	0.00	2.22	0.59	4.72	0.00	3.51	1.50	1.27	0.29	3.17
o1-mini	4.23	**38.89**	2.96	51.18	6.52	49.12	2.26	36.71	3.91	42.28
Llama-3.1-Instruct-70B	0.00	1.11	0.00	3.94	1.09	1.75	0.00	6.33	0.47	3.89
Llama-3.1-Instruct-8B	0.70	3.33	0.00	14.17	1.09	7.02	0.00	3.80	0.67	7.60
Qwen2.5-Instruct-72B	0.00	12.22	1.18	25.20	1.09	14.04	0.00	11.39	1.20	16.59
Qwen2.5-Instruct-7B	6.34	21.11	11.24	**48.03**	6.52	33.33	3.01	21.52	4.47	31.66
QwQ-32B	0.00	7.78	1.78	8.66	0.00	1.75	0.00	1.27	0.61	5.16
QwQ-32B-Preview	8.45	**54.44**	14.79	**66.93**	11.96	**68.42**	8.27	53.16	10.56	**56.63**
QwQ-Plus	**30.99**	50.00	**42.01**	56.69	**36.96**	29.82	**52.63**	**54.43**	**42.28**	45.25
Deepseek_V3	0.00	3.33	0.00	7.87	1.09	7.02	0.00	1.27	0.17	5.89
Deepseek_R1	1.41	1.11	0.59	1.57	0.00	1.75	0.75	1.27	0.41	2.00

Table 20: Model performance (NA) on KoLasSimpleQA (part2/2).

	zh		ko		th		ar		vi	
	general	specific	general	specific	general	specific	general	specific	general	specific
GPT-4o-2024-11-20	4.72	54.74	11.92	40.48	12.92	52.53	10.49	48.72	5.79	39.13
GPT-4o-mini	25.98	**87.37**	25.83	69.05	32.06	79.80	27.16	76.07	19.83	64.13
o1-mini	11.81	36.84	15.23	34.52	20.10	51.52	18.52	43.59	14.05	61.96
Llama-3.1-Instruct-70B	12.60	85.26	21.85	**75.00**	19.62	71.72	21.60	**84.62**	14.05	64.13
Llama-3.1-Instruct-8B	**53.54**	**89.47**	**57.62**	79.76	46.89	83.84	49.38	76.92	30.58	**75.00**
Qwen2.5-Instruct-72B	11.81	64.21	19.21	54.76	21.53	73.74	24.07	71.79	12.40	57.61
Qwen2.5-Instruct-7B	41.73	47.37	55.63	44.05	**52.63**	63.64	**63.58**	73.50	**38.84**	57.61
QwQ-32B	12.60	70.53	22.52	**75.00**	22.01	**81.82**	20.37	**84.62**	15.70	70.65
QwQ-32B-Preview	18.11	40.00	19.21	35.71	17.22	22.22	14.20	31.62	14.05	45.65
QwQ-Plus	7.09	50.53	13.91	40.48	12.44	47.47	10.49	52.14	8.26	41.30
Deepseek_V3	5.51	60.00	17.88	61.90	20.57	76.77	13.58	72.65	7.44	50.00
Deepseek_R1	3.15	45.26	13.25	72.62	11.48	76.77	9.26	73.50	6.61	47.83

Table 21: Model performance (IN) on KoLasSimpleQA (part1/2).

	cs		hu		ru		sr		avg.	
	general	**specific**	**general**	**specific**	**general**	**specific**	**general**	**specific**	**general**	**specific**
GPT-4o-2024-11-20	7.75	44.44	11.83	53.54	13.04	54.39	6.02	45.57	9.39	46.72
GPT-4o-mini	16.20	66.67	28.40	**85.04**	31.52	84.21	21.05	72.15	25.34	73.45
o1-mini	12.68	41.11	24.85	40.94	16.30	35.09	13.53	41.77	16.34	42.03
Llama-3.1-Instruct-70B	10.56	72.22	19.53	79.53	16.30	82.46	8.27	63.29	16.04	72.98
Llama-3.1-Instruct-8B	39.44	73.33	53.85	77.17	55.43	**85.96**	34.59	**79.75**	46.81	**78.06**
Qwen2.5-Instruct-72B	18.31	67.78	46.75	69.29	27.17	66.67	26.32	73.42	23.06	64.39
Qwen2.5-Instruct-7B	**45.07**	71.11	**55.62**	51.97	**61.96**	61.40	**63.91**	64.56	**53.22**	58.06
QwQ-32B	16.90	67.78	35.50	80.31	21.74	78.95	18.05	78.48	20.60	74.18
QwQ-32B-Preview	15.49	33.33	24.85	26.77	13.04	19.30	16.54	29.11	16.97	30.01
QwQ-Plus	9.86	42.22	19.53	39.37	14.13	57.89	8.27	35.44	11.55	43.29
Deepseek_V3	7.04	62.22	20.12	76.38	11.96	77.19	11.28	69.62	12.82	64.91
Deepseek_R1	7.04	**67.78**	11.83	81.10	7.61	84.21	6.02	74.68	8.47	67.52

Table 22: Model performance (IN) on KoLasSimpleQA (part2/2).

	zh		ko		th		ar		vi	
	general	**specific**	**general**	**specific**	**general**	**specific**	**general**	**specific**	**general**	**specific**
GPT-4o-2024-11-20	95.28	27.78	**87.76**	**45.16**	**86.89**	**35.00**	89.51	**42.42**	**94.21**	**57.14**
GPT-4o-mini	74.02	9.78	74.17	29.27	67.79	16.84	72.84	19.82	80.17	35.16
o1-mini	87.80	14.63	83.80	29.27	78.57	17.74	80.89	13.56	85.83	21.92
Llama-3.1-Instruct-70B	87.20	11.96	78.15	23.17	80.19	17.44	78.26	13.91	85.95	33.71
Llama-3.1-Instruct-8B	46.03	5.56	42.38	14.10	52.66	11.70	49.37	4.26	69.42	20.69
Qwen2.5-Instruct-72B	88.19	25.61	80.14	24.59	77.27	14.12	75.93	11.58	87.60	35.37
Qwen2.5-Instruct-7B	57.94	19.64	40.85	11.90	45.81	10.00	35.22	7.53	60.50	23.19
QwQ-32B	87.40	23.86	77.18	20.25	77.78	12.90	79.50	12.39	84.17	26.97
QwQ-32B-Preview	80.99	26.92	78.68	26.83	79.43	29.03	83.80	17.78	84.68	26.32
QwQ-Plus	87.50	20.00	76.92	27.66	75.24	14.55	82.11	12.86	85.29	25.49
Deepseek_V3	94.49	35.96	82.12	31.58	79.33	20.00	86.42	23.42	92.56	46.51
Deepseek_R1	**96.85**	**53.26**	86.75	25.61	88.41	20.00	**90.74**	25.22	**93.39**	52.17

Table 23: Model performance (CGA) on KoLasSimpleQA (part1/2).

	cs		hu		ru		sr		avg.	
	general	**specific**	**general**	**specific**	**general**	**specific**	**general**	**specific**	**general**	**specific**
GPT-4o-2024-11-20	92.20	**42.86**	**88.17**	**32.00**	86.81	27.91	**93.94**	**46.27**	90.53	**41.89**
GPT-4o-mini	83.80	31.82	71.43	10.74	68.48	12.73	78.63	26.92	74.59	24.07
o1-mini	86.76	32.73	74.39	16.13	82.56	31.03	86.15	34.00	82.97	27.23
Llama-3.1-Instruct-70B	89.44	26.97	80.47	17.21	83.52	16.07	91.73	32.43	83.88	24.01
Llama-3.1-Instruct-8B	60.28	24.14	46.15	10.09	43.96	7.55	65.41	17.11	52.85	15.20
Qwen2.5-Instruct-72B	81.69	22.78	52.69	7.37	72.53	22.45	73.68	17.14	76.64	22.55
Qwen2.5-Instruct-7B	51.88	9.86	37.33	0.00	33.72	7.89	34.11	17.74	44.15	14.86
QwQ-32B	83.10	26.51	63.86	12.07	78.26	19.64	81.95	20.51	79.24	21.85
QwQ-32B-Preview	83.08	26.83	70.83	19.05	85.19	**38.89**	81.97	**37.84**	80.96	30.29
QwQ-Plus	85.71	15.56	66.33	9.09	77.59	17.50	82.54	22.22	79.91	20.71
Deepseek_V3	**92.96**	35.63	79.88	17.09	87.91	16.98	88.72	29.49	87.15	31.09
Deepseek_R1	92.86	31.46	88.10	17.60	**92.39**	14.29	**93.94**	24.36	**91.49**	31.14

Table 24: Model performance (CGA) on KoLasSimpleQA (part2/2).

	language-specific domain		general domain		mECE
	tran_en	direct	tran_en	direct	
GPT-4o-2024-11-20	0.42	0.43	0.04	0.05	0.24
GPT-4o-mini	0.56	0.55	0.07	0.14	0.33
o1-mini	0.46	0.48	0.07	0.1	0.28
Llama-3.1-Instruct-70B	0.42	0.57	0.04	0.07	0.28
Llama-3.1-Instruct-8B	0.53	0.63	0.11	0.3	0.39
Qwen2.5-Instruct-72B	0.61	0.62	0.05	0.15	0.36
Qwen2.5-Instruct-7B	0.67	0.71	0.15	0.38	0.48
QwQ-32B	0.45	0.43	0.04	0.1	0.26
QwQ-32B-Preview	0.32	0.35	0.04	0.06	0.19
QwQ-Plus	0.44	0.43	0.05	0.09	0.25
Deepseek_V3	0.51	0.51	0.03	0.06	0.28
Deepseek_R1	0.34	0.33	0.04	0.06	0.19
AvgECE	0.48	0.5	0.06	0.13	0.29

Table 25: LLMs' Expected Calibration Error (ECE) in the general and language-specific domains corresponding to the `tran_en` and `direct` settings. mECE represents the Mean of ECE across the two domains and two settings, AvgECE represents the average across the 12 LLMs.

Shift Expression Extracting Prompt

Given a thinking process for answering a question, follow these steps to extract contrastive expressions from the answer text:
1. Identify the Primary Language:
• First, determine the primary language of the answer text.
2. Extract Contrastive Words, Phrases, or Expressions:
• Identify all the phrases that express a shift in opinion, explanation, or answer, phrases that signal a contrast or change in direction.
• For English: "However," "but," "On the other hand," "Although," "Nevertheless," "Yet," "Despite," "In contrast," "Instead," "Even though."
• Please pay attention that phrases indicating a successive relationship, such as "so", "for example" and the like, must never appear in your answers. Your goal is to find phrases indicating a contrast of viewpoints.

Requirements:
1. Identify and list all the contrastive words or phrases that indicate a shift in meaning, thought, or direction.
2. These expressions should be at the beginning of a sentence to signal a shift.
3. Keep the original text's meaning and context intact.
4. Ensure to maintain the original capitalization of the words (e.g., "However" vs. "however").
5. Provide a clear list of all the identified contrast words or phrases.

[Input text]:

```
{
    "question": <question>,
    "answer": <answer>,
}
```

Please respond strictly in JSON format. Do not include any additional text outside the JSON structure. The output should also include the detected language type.

[Output]:

```
{
    "language": "<detected_language>",
    "shift_expression":[list]
}
```

Table 26: Prompt template for splitting the LRM's reasoning process into thoughts on KoLasSimpleQA (step 1/2)

Shift Expression Confirming Prompt

Given the thinking process, identify all the phrases that express a shift in opinion, explanation, or answer, i.e., phrases that signal a contrast or change in direction (commonly known as “contradiction,” “contrast,” or “transition” phrases). For each identified phrase, wrap it in the format `<shift_phrase_X>word<shift_phrase_X>`, where X is the sequential number for each occurrence of the phrase.
You should provide a list of the sequence numbers corresponding to the phrases that convey a shift in meaning.
If there is no `"<shift_phrase_X>"` tag in text, return an empty list.

[Input text]:`<input_text>`

Please respond strictly in JSON format. Do not include any additional text outside the JSON structure.

[Output]:

```
{
    "shift_phrase_ids": [list of sequential numbers of shifting phrases]
}
```

Table 27: Prompt template for splitting the LRM's reasoning process into thoughts on KoLasSimpleQA (step 2/2)

Thought Access Prompt

Problem P = `<problem>`
Solution Draft S = `<split_solutions>`
Correct Answer A = `<expected_answer>`

1. Please analyze the relevance between the solution S and the problem P, and conduct some verifications to check the correctness of the solution itself. Please think step by step to give an explanation **EXPLANATION**.
2. If you think the solution draft S can lead to the correct answer A of the problem P, please stick to the line of thinking without deviation and carry it through to completion. If you think it cannot yield the correct answer or you're not sure, indicate a confidence score of 0 out of 2.
3. Please tell me honestly how confident you are that you can solve the problem P correctly based on the the solution draft S. Out of 2, please generate your confidence score **CONFIDENT SCORE**.

Please output **EXPLANATION** and **CONFIDENT SCORE** according to the following JSON format:

```
{
    "explanation": "EXPLANATION",
    "confident_score": "CONFIDENT SCORE"
}
```

Table 28: Prompt template for evaluating the correctness of the thoughts of the LRM's reasoning process on KoLasSimpleQA.

Question:何晓钟参与编剧的台视八点档连续剧叫什么名字? What is the name of the TTV primetime drama series that He Xiaozhong participated in writing?
Correct Answer:星星知我心 (Stars Know My Heart)

Thought1 ✗

好的，我需要回答用户关于何晓钟参与编剧的台视八点档连续剧的问题。首先，我得回忆一下何晓钟的资料。他是台湾的编剧，参与过不少电视剧的创作。台视八点档通常指的是台湾电视台黄金时段播出的连续剧，这类剧集很多，比如《意难忘》、《娘家》等。 我记得何晓钟曾参与《意难忘》的编剧工作，这部剧确实是在台视八点档播出的长剧，分很多集，播出时间也很长。
Okay, I need to answer the user's questions about the Taiwan Television's 8 PM slot serial dramas for which He Xiaozhong was involved in the screenwriting. First of all, I have to recall the information about He Xiaozhong. He is a screenwriter from Taiwan and has participated in the creation of quite a few TV dramas. The Taiwan Television's 8 PM slot usually refers to the serial dramas broadcast during the prime time of Taiwan Television. There are many such dramas, such as Unforgettable Love and The Mother's Family, etc. I remember that He Xiaozhong was once involved in the screenwriting work of Unforgettable Love. This drama is indeed a long serial drama broadcast in the 8 PM slot of Taiwan Television. It has many episodes and was broadcast for a long time.

Thought2 ✓

不过，我需要确认这一点是否正确，可能还有其他剧集比如《星星知我心》或者《满庭芳》之类的。

However, I need to confirm whether this point is correct. There may also be other TV dramas, such as The Sentimental Journey or Full Bloom, and the like.

Thought3 ✗

不过《意难忘》是比较有名的，而且确实属于台视的八点档。
However, Unforgettable Love is quite famous, and it is indeed a drama in the 8 PM time slot of Taiwan Television.

......

Thought5 ✗

虽然《意难忘》的可能性较高，但信心可能不是百分之百，因为可能有其他剧集。所以，我需要给出一个信心分数，比如80分，表示比较确定，但可能还有不确定因素。
Although the possibility of Unforgettable Love is relatively high, the confidence may not be 100%, because there may be other dramas. Therefore, I need to give a confidence score, for example, 80 points, which means it is relatively certain, but there may still be some uncertain factors.

Predicted Answer:何晓钟参与编剧的台视八点档连续剧是《意难平》。
The Taiwan Television's 8 PM slot serial drama that He Xiaozhong participated in writing the screenplay for is Unforgettable Love.

Figure 12: Example of the thoughts in Deepseek-R1's reasoning process on KoLasSimpleQA. √ and × indicate the correctness of the thoughts. The original text is in black, while the translation into English is shown in blue..

Question:何晓钟参与编剧的台视八点档连续剧叫什么名字? What is the name of the TTV primetime drama series that He Xiaozhong participated in writing?
Correct Answer:星星知我心 (Stars Know My Heart)

Thought1

嗯，用户问的是何晓钟参与编剧的台视八点档连续剧叫什么名字。首先，我需要确认何晓钟是谁。可能是一位台湾的编剧，参与过八点档电视剧。台视的八点档通常指的是台湾电视公司（Taiwan Television，台视）长期播出的连续剧，通常在晚上八点时段播出，类似中国大陆的长篇连续剧，可能类似家庭伦理剧或者乡土剧。 首先，我需要回想一下台视有哪些著名的八点档连续剧。台视比较有名的可能有《娘家》系列，比如《娘家之新女性》、《娘家之女儿翻身》等。
Okay, the user asked what the name of the Taiwan Television's 8 PM slot serial drama that He Xiaozhong was involved in screenwriting is. First of all, I need to confirm who He Xiaozhong is. He might be a screenwriter from Taiwan who has been involved in 8 PM slot TV dramas. The 8 PM slot of Taiwan Television usually refers to the serial dramas that Taiwan Television (Taiwan Television Enterprise, Ltd., TTV) has been broadcasting for a long time. They are usually broadcast during the 8 PM time period, which is similar to the long serial dramas in mainland China. They might be similar to family ethical dramas or rural dramas. First of all, I need to recall what the famous 8 PM slot serial dramas of Taiwan Television are. Some of the more famous ones of Taiwan Television might include the The Mother‘s Family series, such as The New Women in the Mother’s Family and The Daughters‘ Rise in the Mother’s Family.

✓

Thought2

__不过__，不确定何晓钟是否参与过这些剧集的编剧工作。 另外，可能需要查证何晓钟的编剧作品。不过由于我目前无法实时搜索，只能依靠已有的知识库。记得何晓钟可能参与过《含苞》这部剧？或者《家的脚印》？或者《娘家》系列？
__However__, I'm not sure if He Xiaozhong has ever participated in the screenwriting work of these TV dramas. In addition, it may be necessary to verify He Xiaozhong's screenwriting works. However, since I can't conduct a real-time search at present, I can only rely on the existing knowledge base. I remember that He Xiaozhong might have participated in the TV drama In Bud? Or Footsteps of Home? Or the Mother's Family series?

✗

......

Thought9

__或者__，可能用户的问题中的何晓钟是男性名字，而台湾的编剧可能有女性较多，但不确定。总之，可能需要猜测是《含苞》，但信心分数可能在50-60之间。
__Alternatively__, the He Xiaozhong in the user's question is a male name, and among the screenwriters in Taiwan, there may be more females, but I'm not sure about that. In any case, it's possible that it might be In Bud, but the confidence level might be between 50 and 60.

✗

Predicted Answer:何晓钟参与编剧的台视八点档连续剧是《含苞》。
The Taiwan Television's 8 PM slot serial drama that He Xiaozhong participated in writing the screenplay for is In Bud.

Figure 13: Example of the thoughts in QwQ-32B's reasoning process on KoLasSimpleQA. $\surd$ and $\times$ indicate the correctness of the thoughts. The original text is in black, while the translation into English is shown in blue.