M^{3}TQA: Massively Multilingual Multitask Table Question Answering

Daixin Shu[1], **Jian Yang**[1]*, **Zhenhe Wu**[1], **Xianjie Wu**[1], **Xianfu Cheng**[1], **Xiangyuan Guan**[1], **Yanghai Wang**[1], **Pengfei Wu**[3], **Tingyang Yang**[1], **Hualei Zhu**[1], **Wei Zhang**[1], **Ge Zhang**[2], **Jiaheng Liu**[3], **Zhoujun Li**[1]
[1]CCSE, Beihang University, [2]M-A-P, [3]Nanjing University
{shudx,jiayang}@buaa.edu.cn

Abstract

Tabular data is a fundamental component of real-world information systems. However, existing multilingual table benchmarks suffer from geolinguistic imbalance - overrepresenting certain languages and lacking sufficient scale for rigorous cross-lingual analysis. To address these limitations, we introduce M^{3}TQA, which is a comprehensive framework for massively multilingual multitask table question answering, including subsequent datasets M^{3}TQA-BENCH and M^{3}TQA-INSTRUCT, featuring tables expanded to 97 languages from Chinese and English sources. M^{3}TQA-BENCH includes 6,606 professionally annotated question-answering pairs across four tasks designed to evaluate nuanced table reasoning capabilities. Additionally, we synthesized the training set M^{3}TQA-INSTRUCT in 97 languages using Large Language Model (LLM). Experiments on state-of-the-art LLMs reveal critical insights into cross-lingual generalization, demonstrating that synthetically generated, unannotated training data can significantly boost performance, particularly for low-resource languages. M^{3}TQA establishes a new standard for multilingual table understanding, providing both a challenging evaluation platform and a scalable methodology for future research. The dataset and code are available at https://m3tqa.github.io.

1 Introduction

Tabular data serves as a cornerstone of real-world information systems, underpinning critical applications from recommender systems (Guo et al., 2017) to financial analytics (Clements et al., 2020; Yang et al., 2025b). The advent of LLMs has catalyzed breakthroughs in table reasoning tasks, including table question answering (Table QA) (Ye et al., 2023; Nahid and Rafiei, 2024b; Sui et al., 2024b; Zhang et al., 2025a, 2024a), fact verification (Ye et al., 2023; Zhang et al., 2024b; Nahid and Rafiei, 2024a), and semantic retrieval (Pourreza and Rafiei, 2023; Talaei et al., 2024).

Figure 1: Partial language visualization from M^{3}TQA. Spatially proximate languages belong to the same language family. Widely-used language families and their representative languages are shown below (Korean, Japanese, and other language isolates are categorized separately).

However, a critical linguistic bias persists: Current research predominantly targets English-language tables, neglecting the complexities of multilingual table understanding. While recent efforts have begun addressing this gap—such as MULTITAT (Zhang et al., 2025b), XINFOTABS (Minhas et al., 2022), and TATA (Gehrmann et al., 2023)—these initiatives remain limited. Existing datasets exhibit two fundamental limitations: (1) **Geolinguistic imbalance:** Existing datasets focus predominantly on Indo-European languages (Europe/North America/Middle East/Indian subcontinent) and high-impact languages (e.g., Chinese, Arabic), with inadequate coverage of most language families. (2) **Scale constraints:** Current cross-lingual datasets lack sufficient scale for granular linguistic feature analysis.

To address these gaps, we present the Massive Multilingual Multitask Table Question Answering framework(M^{3}TQA). To ensure experimental generalizability, we collect 50 real-world tables from various sources and languages (41 Chinese and 9 English) and adopt the four types of table structures from IM-TQAS (Zheng et al., 2023) as se-

* Corresponding Author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 22578–22602

lection criteria. We establish a comprehensive translation and verification pipeline to extend the dataset to 97 languages, utilizing DeepSeek and GPT-4o as translation engines through a six-step process. Translation accuracy is rigorously validated via back-translation, with a median BLEU score of 60.19 achieved after excluding low quality translations. We construct training set M^{3}TQA-INSTRUCT and test set M^{3}TQA-BENCH respectively using LLMs. Specifically, M^{3}TQA-BENCH contains 2,916 LLM-generated question-answer (QA) pairs and 3,690 manually constructed QA pairs, and all data in M^{3}TQA-BENCH has undergone professional manual annotation, while M^{3}TQA-INSTRUCT contains 39,982 automatically generated QA pairs without human annotation. We design four types of Table QA tasks and conduct a series of experiments on them to evaluate diverse LLMs' tabular comprehension and cross-lingual capabilities. The results show that LLMs exhibit significant performance variations across different language families, and training on synthetic data without manual annotation can also yield notable performance improvements.

Our contributions are summarized as follows:

- We introduce a large-scale multilingual table understanding dataset covering 97 languages. This significantly extends linguistic coverage beyond existing benchmarks.
- We propose an efficient translation pipeline using advanced LLMs. This pipeline achieves high-quality cross-lingual table conversion (median BLEU: 60.19).
- We design four tasks on M^{3}TQA-BENCH to systematically evaluate various models' capabilities in interpreting and processing tabular data. Our experiments demonstrate that synthetically generated, unannotated QA pairs enhance cross-lingual comprehension. Crucially, this finding reveals a viable path toward optimizing performance for low-resource languages.

2 M^{3}TQA

We present M^{3}TQA (Massively Multilingual Multitask Table Question Answering), a curated dataset designed for Multilingual Multitask Table QA. The dataset is built on 50 source tables in English and Chinese. Through a six-step procedure, we translate these tables to 97 languages. QA pairs are then constructed via a hybrid approach combining human annotation and LLM generation, as shown in Figure 2. This yields a linguistically diverse resource that authentically reflects real-world multilingual challenges in Table QA tasks.

Table Size			
	Max	Min	Mean
Rows	57	5	15.52
Columns	15	5	8.30
Table Type			
		Number	Proportion
Relational Tables		25	50%
Entity Tables		6	12%
Matrix Tables		13	26%
Composite Tables		6	12%
Table Features			
Ratio of Numerical Cells			56.66%
English Tables			9
Chinese Tables			41
Multilingual Tables			4,349

Table 1: Data statistics of Tables. Relational tables have a vertical layout while Entity tables arrange data horizontally. Matrix tables show relationships in both directions. Composite tables contains headers in variable positions.

2.1 Data Collection

To ensure the authenticity and complexity of the data, we collect 50 tables from real-world international network sources, including both Chinese and English language tables. Table selection follows the structural taxonomy established in prior work (Wang et al., 2021; Zheng et al., 2023), categorizing tables into four types: (1) **Relational tables:** This type features a vertical layout, where the top rows serve as column headers and the rest of the rows contain the tabular main content. (2) **Entity tables:** This type is structured to organize data horizontally, with the first few columns acting as row headers and the remaining cells composing the tabular body. (3) **Matrix tables:** This type illustrates relationships between rows and columns (i.e., in both directions), typically featuring multi-level headers. (4) **Composite tables:** This type includes tabular headers in variable positions (e.g., beyond the top/left edges, or in middle rows), mirroring complex real-world tabular layouts.

The detailed strategy for table selection is described in Appendix A.

2.2 Table translation

To reduce translation costs, we leverage multilingual LLMs (DeepSeek and GPT-4o) for table translation through a six-phase pipeline. The translation targets all cells containing non-numerical content, while purely numerical numerical cells are con-

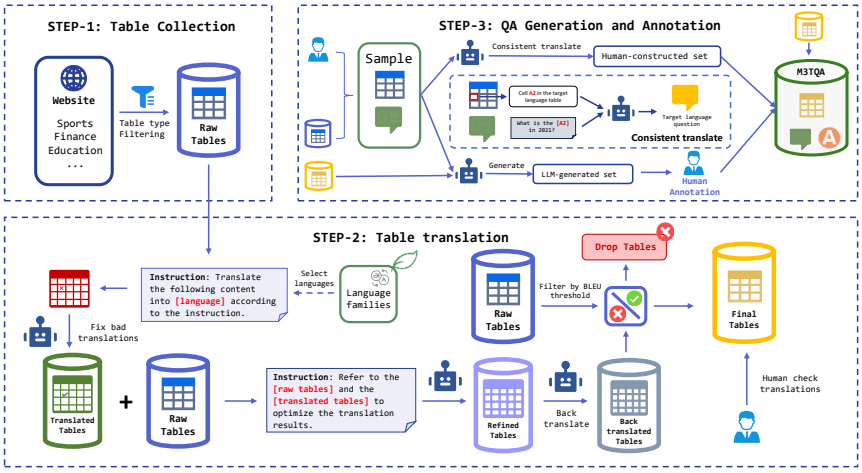

Figure 2: The framework of constructing M^{3}TQA.

verted to Arabic numeral representations. Each stage is implemented using a single type of LLM, with the model used indicated in parentheses.

(1) **Initial Translation**: To preserve structural semantics during translation, we serialize tables as two-dimensional lists, where each element of the outer list is a list of cells from a single row. We then construct prompts to indicate LLM(GPT-4o) to translate content while strictly maintaining this matrix structure .

(2) **Cell Complement Translation**: This stage rectifies translation omissions from the initial translation to ensure the integrity of the final target-language table (GPT-4o).

(3) **Global Refinement**: We leverage LLM(GPT-4o) to analyze both the original table and its translated version, determining whether the translation requires refinement and generating the refined output if necessary. The objective of this stage is to rectify discrepancies arising from the translation process, with corrections guided by the overall semantics of the table.

(4) **Cell Refinement**: Similar to the previous stage, both the original cell content and its corresponding translated content are provided to the LLM(DeepSeek) to produce the refined translation.

(5) **Back-Translation Validation**: To evaluate translation quality, we back-translate the tables into the source language using LLM(DeepSeek). We then compute the average adaptive BLEU between the original and back-translated content for all non-numeric table cells. To address the limitations of BLEU metrics on short texts, we employ an adaptive scoring method: BLEU-1 for cells with 1-3 characters, the average of BLEU-1 and BLEU-2 for 4–7 characters, and the average of BLEU-1, BLEU-2, and BLEU-4 for longer texts.

Properties	LLM-generated test set	Human-constructed test set	Entire test set
M^{3}TQA-Instruct			
Numerical Computation	28,555		28,555
Cell Extraction	3,157		3,157
Factual Verification	1,103		1,103
Open-Ended Questions	7,167		7,167
Total	39,982		39,982
M^{3}TQA-Bench			
Numerical Computation	2,258	1,441	3,699
Cell Extraction	337	1,102	1,439
Factual Verification	48	174	222
Open-Ended Questions	273	973	1,246
Total	2,916	3,690	6,606
Max Prompt Length	11,248	10,932	11,248
Mean prompt Length	3,888.03	3,767.47	3,820.69

Table 2: Data statistics of M^{3}TQA

After filtering, approximately 10% of tables with low BLEU scores are excluded. The final dataset contains 4,349 tables, with a median BLEU score of 60.19 and a minimum BLEU score of 34.06.

(6) **Human Check**: We manually check the tables and correct any errors. This task is performed by nine computer science graduate students (master's and Ph.D. candidates) from the research team. Finally, professional linguists are engaged to conduct a manual sampling inspection of the translation results, which demonstrate a cell-level translation accuracy exceeding 93%.

Appendix B provides information on the annotators involved in the task, and Appendix C presents in detail the prompt design for each step of the table translation process.

2.3 QA Generation and Annotation

Hybrid QA Generation QA pairs are generated through a hybrid human-LLM approach. Human annotators first create a seed corpus by manually generating one QA pair per table according to TableBench's(Wu et al., 2025a) question classification schema. These seed examples are then incor-

porated into prompts to guide LLMs in expanding the dataset. Appendix D provides details for generating QA pairs. Using this approach, 10 initial QA pairs are generated for each table-language combination.

Data Partitioning and Annotation We construct a full-language test set following the methodology outlined in Appendix E. For the LLM-generated test set, we randomly select one QA pair per table-language combination, under the condition that the content of each QA pair is complete and formulated entirely in the target language. The annotators then screen, inspect and refine these pairs to meet task requirements. This process results in a curated test set of 2,916 QA pairs, averaging approximately 30 QA pairs per language. On the other hand, we employ the entity-linking-based translation approach to extend the human-constructed English QA set to all target languages. After manual screening, a refined test set of 3,690 QA pairs is generated.

M^{3}TQA-BENCH is jointly composed of the two test sets mentioned above. Table 2 presents relevant information regarding the constructed QA pairs, and Appendix F elaborates on the necessity of establishing diverse testing scenarios and delineates the performance advantage intervals of different models.

Additionally, we leverage LLMs to generate 39,982 QA pairs in batches as a training set M^{3}TQA-INSTRUCT. This batch of data uses the same generation method as M^{3}TQA-BENCH but is not manually annotated. M^{3}TQA-INSTRUCT is used to verify whether unannotated training sets contribute to the multilingual table comprehension capability of LLM. Table 8 presents the size of the training and test sets included for each language.

Quality Control To ensure M^{3}TQA-BENCH's reliability, we implement rigorous quality controls throughout the 50-day annotation cycle. The profile of annotators is provided in Appendix B. Each annotator is equipped with multiple professional translation tools to facilitate cross-verification of the translation results. Quality control strategies include comprehensive training with open question channels, two interim quality checks with real-time issue resolution, and final random sampling verification. This multi-stage oversight guarantees consistent adherence to annotation standards while addressing challenges inherent in multilingual table comprehension.

Following the completion of the annotation process, we engage native speakers to verify both the accuracy and reliability of the annotations for major languages. The evaluation indicates that the annotation accuracy for the test set exceeded 95%.

2.4 Task Definition

Our task is defined as: Given the k-th language $L_k \in \{L_i\}_{i=1}^{K}$, (where $K = 97$, the total number of languages). For a Table QA task, we provide a table T^{L_k} of the language L_k, as well as the description of the question q^{L_k}, as input of LLM M, so that M generates the corresponding result a^{L_k}. The process can be expressed as $a = M(T^{L_k}, q^{L_k})$. Then we use the evaluation function $I(\cdot)$ to evaluate the answer of M, where $s^{L_k} = I(a^{L_k}, y^{L_k})$ and y^{L_k} denotes the ground truth answer for q^{L_k}. In summary, the process can be described as:

$$s^{L_k} = I(M(T^{L_k}, q^{L_k}), y^{L_k})$$

Specifically, Table QA tasks can be categorized into four types based on answer characteristics:

(1) **Numerical Computation**: This category involves numerical operations (e.g., aggregation, counting, calculations) based on tabular data. To address potential variations in numerical representations across languages, responses a^{L_k} must be converted to Arabic numerals and rounded to two decimal places. For numerical answers that include units (e.g., thousand, million), the answers must be converted to the ones place based on the respective units. The evaluation function computes the Jaccard similarity coefficient between the model-generated answer (A_{gen}) and ground truth answer (A_{gt}), defined as:

$$\mathcal{J}(A_{\text{gen}}, A_{\text{gt}}) = \frac{|A_{\text{gen}} \cap A_{\text{gt}}|}{|A_{\text{gen}} \cup A_{\text{gt}}|}$$

(2) **Cell Extraction**: Answers require retrieving one or multiple cells from the table. To mitigate language differences in interpretation, the model also needs to answer the coordinates of the cell when returning the cell content. The evaluation function computes the Jaccard similarity coefficient of the full coordinates between the model-generated answer and the ground truth.

(3) **Factual Verification**: Models are required to output standardized responses "T" or "F" according to the table and question. The evaluation employs the F1-score to assess prediction accuracy against ground truth labels.

(4) **Open-Ended Questions**: For open-ended questions requiring descriptive answers, we employ the ROUGE-L metric to assess the similarity between the model answer a^{L_k} and the ground truth y^{L_k}. To ensure fairness during the tokenization process, a uniform tokenization scheme was employed across all languages: languages with spaces in their text are tokenized by spaces, while languages without spaces are tokenized by characters.

The overall system performance S aggregates all QA scores:

$$S = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{(q,a)\in\mathcal{D}_{\text{test}}} s^{(q,a)}$$

where $\mathcal{D}_{\text{test}}$ is the test set, and $s^{(q,a)}$ is the score for an individual QA pair in the test set.

3 Method

In recent years, numerous task-specific optimization strategies have been proposed for LLMs, such as Supervised Fine-Tuning and Reinforcement Learning, and demonstrate exceptional performance in various tasks. This section describes multiple learning and training methods for the M^{3}TQA task that enable lower-parameter LLMs to generalize across multiple languages.

3.1 Thinking

Research (Wei et al., 2022; Yeo et al., 2025) shows that thinking process can significantly enhance the comprehension capabilities of LLMs. We integrate the model's thinking process into training, utilizing the thinking paradigms of large-parameter multilingual LLMs to optimize the thinking logic of smaller-parameter LLMs.

Specifically, we utilize prompts to direct large-parameter LLMs to output their thinking traces within dedicated `<think>` tags. These traces are then concatenated with corresponding model-generated outputs enclosed within `<answer>` tags. This composite structure forms the training objective for subsequent model fine-tuning in Figure 3.

3.2 Supervised Fine-Tuning

We perform instruction Supervised Fine-Tuning (SFT) on M^{3}TQA-INSTRUCT for Table QA, minimizing $\mathcal{L}$:

$$\mathcal{L} = \mathbb{E}_{(q^{L_k},a^{L_k})\sim\mathcal{D}}\left[-\log P(a^{L_k} \mid T^{L_k}, q^{L_k}; \theta)\right]$$

```
<think>
1.Identify the relevant data for Work in Process:
   - March 31, 2021
   - Carrying Amount: 54,755,852.09
   - Allowance for Inventory Obsolescence: 459,951.51
   - Carrying Amount (net): 54,295,900.58
   - December 31, 2020:
   - Carrying Amount: 30,924,258.73
   - Allowance for Inventory Obsolescence:
216,271.99
   - Carrying Amount (net): 30,707,986.74
2. The question asks for the percentage change in the
carrying amount considering the allowance for inventory
obsolescence, which means we should use the net carrying
amounts:
   - March 31, 2021 net carrying amount: 54,295,900.58
   - December 31, 2020 net carrying amount:
30,707,986.74
3. Calculate the change amount:
   Change = March 31, 2021 net - December 31, 2020 net =
54,295,900.58 - 30,707,986.74 = 23,587,913.84
4. Calculate the percentage change:
   Percentage Change = (Change / December 31, 2020 net)
* 100 = (23,587,913.84 / 30,707,986.74) * 100 = 76.81%
</think>
<answer>
[76.81%]
</answer>
```

Figure 3: An example of our thinking training paradigm.

where q^{L_k} and a^{L_k} denote the question and answer in language L_k, K represents the total number of languages, and $\mathcal{D}$ is the training data distribution

Experiments are conducted for both thinking and non-thinking modes: (1) **Non-Thinking Mode**: a^{L_k} consists of the answer label `<answer>` followed by the ground truth. (2) **Thinking Mode**: a^{L_k} contains the thinking label `<think>`, followed by the thinking process, and concludes with `<answer>` and the ground truth.

3.3 Group Relative Policy Optimization

Building upon the optimal SFT checkpoint, we apply Group Relative Policy Optimization (GRPO) (Shao et al., 2024) optimization for reinforcement learning. The scoring mechanism for both thinking and non-thinking modes consistently employs the evaluation function $I(\cdot)$ established in task definition.

4 Experiments

We design unified prompt templates for each task category, while model outputs are subjected to strict normalization procedures to extract final answers. We fine-tune multiple open-source LLMs using M^{3}TQA-INSTRUCT, with further optimization via GRPO.

4.1 Experiment Setup

LLMs. We evaluate 20 models encompassing both open-source and proprietary architectures. The open-source models range in scale from 7B to 34B parameters, including Baichuan2 (Yang et al., 2025a), DeepSeek-Chat (DeepSeek-AI et al., 2024), Seed-Coder (ByteDance Seed et al., 2025),

	Uralic languages					Indo-European languages					Austronesian languages					Dravidian languages				
	NC	CE	FV	OEQ	Avg.	NC	CE	FV	OEQ	Avg.	NC	CE	FV	OEQ	Avg.	NC	CE	FV	OEQ	Avg.
Open-source Methods																				
Baichuan2-7B-Chat	0.00	0.21	0.00	1.03	0.20	0.11	0.49	5.26	2.88	0.84	0.00	1.30	5.26	1.87	0.86	0.00	0.00	11.11	1.28	0.58
Deepseek-llm-7B-Chat	0.02	0.00	0.00	2.10	0.33	0.70	0.23	8.42	3.17	1.27	0.91	0.30	5.26	4.31	1.61	0.77	0.00	0.00	0.97	0.61
Seed-Coder-8B-Instruct	9.36	2.51	62.50	14.63	10.61	9.18	4.56	55.79	23.60	12.11	7.17	4.55	47.37	26.95	12.10	5.37	3.20	44.44	18.01	8.38
Llama-3-8B-Instruct	3.30	1.13	12.50	20.54	5.82	6.82	1.32	22.11	30.80	10.47	4.43	1.04	21.05	35.52	10.38	6.95	0.88	0.00	26.14	8.80
Llama-3.1-8B-Instruct	5.54	1.06	12.50	23.42	7.62	5.34	2.18	12.63	30.19	9.44	2.59	1.74	26.32	34.38	9.51	2.04	0.58	0.00	31.71	6.82
Qwen3-8B-nothinking	5.21	1.32	50.00	25.48	9.06	6.89	4.23	69.47	31.37	12.55	5.81	4.44	57.89	36.31	13.56	5.89	5.48	66.67	37.81	13.31
Qwen3-8B-thinking	45.65	5.74	62.50	21.64	34.56	40.15	10.54	68.42	25.47	31.76	33.80	5.99	68.42	29.89	28.61	32.10	10.29	66.67	30.36	28.43
GLM-4-9B-0414-nothinking	15.74	1.99	12.50	20.29	13.58	17.00	2.61	8.42	25.44	15.19	9.13	2.92	21.05	29.85	12.29	9.13	2.77	0.00	26.98	10.62
GLM-4-9B-0414-thinking	31.50	3.37	50.00	15.52	24.06	26.86	2.39	55.79	21.47	21.32	23.62	3.47	52.63	22.16	20.29	12.15	0.88	11.11	17.72	10.76
Phi-4-14B	38.22	3.19	50.00	26.67	29.84	37.84	2.27	53.68	28.51	28.80	33.35	3.25	47.37	31.88	27.27	38.50	0.94	66.67	28.89	30.02
Moonlight-16B-A3B-Instruct	4.20	0.71	25.00	2.27	3.92	3.57	3.09	17.89	6.65	4.43	3.34	2.18	31.58	3.93	4.39	0.36	1.05	33.33	2.78	1.99
ERNIE-4.5-21B-A3B-PT	24.91	0.00	62.50	23.30	20.96	26.19	4.59	63.16	29.54	23.11	24.91	5.87	52.63	36.06	24.18	26.95	4.48	44.44	30.64	23.53
GLM-4-32B	48.17	11.70	62.50	20.04	37.06	42.95	10.73	73.68	24.49	33.36	41.50	10.02	84.21	27.35	33.90	32.74	13.53	88.89	24.50	29.18
Qwen3-32B-nothinking	15.34	6.11	100.00	22.14	17.41	20.84	8.24	80.00	26.47	20.75	11.58	6.46	63.16	30.46	16.28	28.83	8.65	66.67	31.52	26.37
Qwen3-32B-thinking	58.73	26.76	87.50	17.73	47.03	54.74	30.76	84.21	23.10	44.47	48.11	25.12	89.47	27.24	40.96	47.94	31.30	100.00	26.56	42.50
Close-source Methods																				
GPT-3.5-turbo	11.67	7.87	87.50	28.56	16.08	11.54	8.01	75.79	32.77	16.45	8.12	9.90	57.89	39.84	16.68	10.04	3.12	100.00	33.28	15.56
GPT-4o	40.46	25.32	37.50	30.51	35.81	44.52	25.11	42.86	29.08	37.36	44.87	19.22	42.11	34.25	37.28	42.44	33.89	77.78	33.33	40.25
Gemini-2.0	31.48	30.59	62.50	9.05	28.92	32.79	28.15	63.16	18.10	29.89	37.18	24.81	73.68	18.68	32.54	40.51	33.98	100.00	28.45	39.01
Gemini-2.5	64.59	55.04	100.00	29.62	58.53	61.63	53.76	86.17	27.27	54.23	62.96	47.50	89.47	31.44	54.74	60.49	47.46	100.00	32.37	54.22
Deepseek-Chat-V3	59.28	17.63	87.50	34.07	48.05	57.41	21.05	73.68	32.59	45.34	58.76	20.20	63.16	38.90	46.96	50.60	23.26	88.89	38.61	44.14
Open-Source Training Methods																				
Llama-3.1-8B-Instruct-SFT-GRPO	58.02	46.61	62.50	39.77	52.86	54.96	50.87	48.83	45.60	51.22	42.32	50.53	46.23	46.86	44.98	50.73	48.26	55.56	38.11	46.97
Qwen3-8B-SFT-GRPO	55.54	30.83	58.16	20.56	43.61	53.37	39.22	59.04	36.36	49.17	48.56	24.15	73.68	8.04	36.86	52.08	42.88	44.44	34.14	47.80

	Turkic languages					Language isolate					Austroasiatic languages					Mongolic languages				
	NC	CE	FV	OEQ	Avg.	NC	CE	FV	OEQ	Avg.	NC	CE	FV	OEQ	Avg.	NC	CE	FV	OEQ	Avg.
Open-source Methods																				
Baichuan2-7B-Chat	0.00	0.79	0.00	1.13	0.39	0.00	0.10	7.14	0.20	0.32	0.00	0.00	0.00	0.32	0.06	0.00	0.00	0.00	0.65	0.13
Deepseek-llm-7B-Chat	0.00	0.79	0.00	0.36	0.23	0.59	0.79	7.14	1.32	1.02	0.00	0.00	0.00	1.70	0.30	0.00	0.00	0.00	1.54	0.31
Seed-Coder-8B-Instruct	4.55	3.85	66.67	21.76	9.13	7.70	5.82	50.00	11.64	9.61	4.60	1.33	0.00	7.59	4.49	3.38	2.69	60.00	16.07	8.02
Llama-3-8B-Instruct	8.77	2.70	16.67	24.08	10.86	4.60	1.23	28.57	14.66	6.76	4.02	2.22	0.00	12.01	5.04	5.18	2.27	20.00	21.18	8.44
Llama-3.1-8B-Instruct	5.74	2.19	0.00	20.87	8.03	7.51	1.55	21.43	15.79	8.39	5.17	0.00	0.00	13.87	5.74	3.60	8.33	40.00	28.16	10.75
Qwen3-8B-nothinking	5.91	3.62	33.33	26.34	10.18	9.28	4.36	71.43	16.89	11.98	9.45	1.33	50.00	23.34	11.44	10.81	6.06	0.00	23.86	12.14
Qwen3-8B-thinking	40.52	11.06	66.67	18.33	30.58	35.85	14.07	50.00	14.49	27.35	28.81	13.33	100.00	19.87	26.25	39.86	18.18	80.00	21.69	34.07
GLM-4-9B-0414-nothinking	12.68	3.37	0.00	20.05	12.10	15.81	3.73	14.29	15.15	13.01	10.06	5.50	0.00	14.59	9.88	9.91	0.52	20.00	21.95	11.06
GLM-4-9B-0414-thinking	26.61	2.96	50.00	15.00	19.95	26.57	4.56	57.14	9.06	19.39	20.40	0.00	50.00	13.50	16.48	23.42	0.00	40.00	18.40	18.99
Phi-4-14B	35.98	4.71	25.00	23.59	26.77	34.60	2.98	57.14	15.62	24.75	27.10	1.79	50.00	26.81	22.47	27.03	0.00	60.00	17.29	21.68
Moonlight-16B-A3B-Instruct	1.87	1.64	33.33	4.94	3.05	3.08	1.89	28.57	6.26	4.38	0.09	1.11	0.00	0.00	0.24	0.34	1.52	60.00	7.83	4.40
ERNIE-4.5-21B-A3B-PT	25.35	5.18	62.50	25.03	22.09	22.98	2.86	71.43	13.44	18.45	18.75	1.79	100.00	25.95	19.08	19.37	4.80	20.00	25.11	17.99
GLM-4-32B	42.39	6.49	62.50	19.12	30.69	42.83	13.04	50.00	13.34	30.71	34.71	8.93	75.00	22.79	28.17	35.50	4.55	100.00	20.81	29.74
Qwen3-32B-nothinking	13.89	10.95	66.67	21.81	15.94	16.02	8.09	85.71	13.65	16.34	19.56	8.89	100.00	17.02	19.12	16.22	12.16	40.00	21.77	17.56
Qwen3-32B-thinking	57.10	29.10	83.33	19.74	44.33	49.60	33.01	71.43	12.19	39.27	36.09	45.40	100.00	19.57	36.13	51.85	35.98	100.00	20.81	44.83
Close-source Methods																				
GPT-3.5-turbo	10.52	6.62	75.00	25.70	14.47	11.83	10.01	64.29	16.91	14.35	9.53	3.69	75.00	31.18	14.59	10.72	2.53	80.00	25.01	14.87
GPT-4o	46.12	30.93	12.50	25.53	37.90	46.96	28.90	42.86	16.42	36.76	40.72	23.21	25.00	26.90	33.86	52.07	17.88	60.00	25.71	41.19
Gemini-2.0	30.20	36.52	50.00	18.01	29.38	29.56	28.43	28.57	14.60	26.27	27.38	33.67	75.00	16.91	28.00	34.46	37.27	80.00	12.22	32.34
Gemini-2.5	65.78	40.75	87.50	25.93	52.81	60.74	49.72	78.57	14.01	49.60	52.87	56.58	100.00	21.11	48.69	52.48	51.52	100.00	26.58	49.06
Deepseek-Chat-V3	59.28	23.32	50.00	30.92	45.79	49.09	22.83	71.43	18.17	37.99	47.22	16.54	75.00	26.18	37.47	56.35	24.83	80.00	28.30	46.22
Open-Source Training Methods																				
Llama-3.1-8B-Instruct-SFT-GRPO	41.75	42.86	66.67	48.52	44.39	52.13	50.70	50.79	40.87	49.26	51.35	38.92	50.00	48.44	46.92	48.73	46.53	68.21	46.46	47.92
Qwen3-8B-SFT-GRPO	51.46	31.56	62.50	36.63	44.41	43.01	28.99	44.53	31.58	37.44	33.31	25.12	45.28	35.50	35.79	56.99	36.69	87.50	49.89	50.55

Table 3: The main results of the 8 language families with relatively high comprehensive scores.

LLaMA3, LLaMA3.1 (AI@Meta, 2024), Qwen3 (Team, 2025b), GLM4 (GLM et al., 2024), Phi-4 (Abdin et al., 2024), Moonlight (Liu et al., 2025), and ERNIE (Team, 2025a). For proprietary models, we assessed Gemini-2.0, Gemini-2.5 (Comanici et al., 2025), GPT-3.5-turbo, and GPT-4o (Achiam et al., 2023). Additionally, we perform fine-tuning and reinforcement learning on Qwen3-8B and LLaMA3.1-8B to further investigate LLMs' multilingual table comprehension capabilities.

Implementation Details. For locally deployable models, we conduct experiments on an 8×NVIDIA A800 GPU cluster (80GB memory) using the PyTorch Transformer framework. The proprietary models are accessed through the official APIs. During SFT, all model parameters are optimized via the Adam optimizer with a global batch size of 64 achieved through gradient accumulation. The training uses cosine annealing scheduling with an initial learning rate of $1e^{-5}$ over 5 epochs, processing sequences up to 12,000 tokens, which requires approximately 20 hours of training time. The GRPO phase utilizes a per-device batch size of 2 (aggregating to global batch size of 16 via gradient accumulation), with a constant learning rate of $1e^{-6}$ over 2 epochs. The maximum prompt word length and maximum response length are 12000 and 4096 respectively, and the group number for each question is 5, requiring approximately 36 hours.

4.2 Main Results

Table 3 and Table 4 show the results of current advanced LLMs on M^{3}TQA. Tasks include Numerical Computation (NC), Cell Extraction (CE), Factual Verification (FV), and Open-Ended Question (OEQ). The average scores (Avg.) across all test cases are also presented. For LLMs that include thinking modes, we test them both with(-thinking) and without(-nothinking) thinking mode enabled.

Among open-source LLMs, the Qwen3 series demonstrate superior overall performance, while Gemini-2.5 maintains the highest performance level in proprietary models. Regarding task categories, "Factual Verification" achieves optimal results. As model scale increases, the most sub-

	Sino-Tibetan languages					Kra-Dai languages					Afroasiatic languages					Niger-Congo languages				
	NC	CE	FV	OEQ	Avg.	NC	CE	FV	OEQ	Avg.	NC	CE	FV	OEQ	Avg.	NC	CE	FV	OEQ	Avg.
Open-source Methods																				
Baichuan2-7B-Chat	1.19	0.00	0.00	1.43	0.88	0.00	0.57	0.00	0.00	0.14	0.00	0.45	4.55	1.09	0.48	0.33	0.88	0.00	0.75	0.53
Deepseek-llm-7B-Chat	2.18	0.00	0.00	0.00	1.21	0.00	0.00	40.00	3.49	2.46	0.37	0.02	4.55	3.12	0.95	0.00	0.00	18.18	2.79	1.85
Seed-Coder-8B-Instruct	12.10	3.51	71.43	5.88	11.71	5.03	4.31	60.00	16.92	9.80	4.39	2.28	54.55	19.51	8.72	1.43	2.72	50.00	19.34	8.93
Llama-3-8B-Instruct	7.14	0.00	14.29	7.28	5.70	6.32	0.00	0.00	17.47	6.96	5.28	1.35	9.09	26.08	8.21	4.14	1.10	31.82	23.71	9.51
Llama-3.1-8B-Instruct	2.58	0.00	28.57	10.61	4.31	2.84	3.45	60.00	10.41	7.08	2.64	1.75	13.64	23.70	6.59	3.95	2.72	27.27	23.03	9.35
Qwen3-8B-nothinking	2.38	2.41	71.43	5.09	5.97	8.45	2.99	80.00	8.97	10.25	7.18	2.46	77.27	28.31	12.85	3.49	2.04	63.64	23.50	11.58
Qwen3-8B-thinking	34.33	5.26	57.14	4.81	23.52	34.77	7.47	40.00	17.36	24.45	27.19	8.41	50.00	20.32	22.90	17.53	9.63	36.36	15.19	16.45
GLM-4-9B-0414-nothinking	13.96	1.56	14.29	3.56	9.34	8.53	1.23	20.00	8.25	7.16	9.99	1.96	4.55	22.89	10.26	5.26	2.70	13.64	18.05	7.96
GLM-4-9B-0414-thinking	20.36	2.63	28.57	4.74	14.01	8.33	6.90	40.00	10.90	9.89	17.78	1.35	36.36	19.08	15.24	7.22	3.45	45.45	13.54	10.32
Phi-4-14B	25.78	1.54	42.86	4.98	17.44	27.64	2.07	60.00	11.86	19.25	22.52	0.75	54.55	23.94	19.42	17.35	2.95	61.54	20.20	17.21
Moonlight-16B-A3B-Instruct	3.37	0.53	14.29	2.25	2.99	0.37	1.20	20.00	0.00	1.33	2.59	0.71	18.18	8.70	3.92	1.06	1.59	18.18	6.39	3.51
ERNIE-4.5-21B-A3B-PT	22.02	1.32	85.71	6.76	17.40	31.24	1.72	80.00	14.00	22.25	16.89	6.60	77.27	28.31	19.25	18.60	2.68	69.23	22.61	18.76
GLM-4-32B	38.10	6.32	57.14	4.53	26.09	26.72	0.00	60.00	11.41	18.19	32.94	11.11	68.18	20.58	27.56	27.31	9.30	50.00	18.54	22.16
Qwen3-32B-nothinking	12.80	7.32	85.71	5.40	13.61	18.97	8.74	60.00	10.61	16.35	18.99	4.79	63.64	21.94	18.34	10.35	4.58	68.18	21.15	15.28
Qwen3-32B-thinking	45.36	23.68	71.43	6.03	35.00	47.70	21.84	60.00	14.33	34.51	38.29	31.18	68.18	21.49	35.11	30.67	17.17	50.00	19.67	26.43
Close-source Methods																				
GPT-3.5-turbo	7.74	5.79	85.71	6.54	10.63	7.76	4.60	60.00	13.62	10.49	9.70	7.88	63.64	28.08	14.82	7.32	10.64	53.85	24.32	15.14
GPT-4o	27.98	19.89	42.86	5.50	23.12	32.96	15.69	20.00	15.38	24.29	38.50	21.50	45.45	24.96	32.75	39.75	21.05	50.00	25.86	32.42
Gemini-2.0	28.17	28.16	85.71	5.39	27.23	27.01	29.43	40.00	6.58	23.65	29.08	28.64	77.27	25.58	30.45	37.90	24.38	50.00	18.35	30.69
Gemini-2.5	51.02	44.74	85.71	5.60	43.92	51.87	72.07	40.00	9.91	47.09	51.30	41.06	77.27	20.83	44.67	51.22	47.36	69.23	17.23	43.57
Deepseek-Chat-V3	44.84	16.83	85.71	7.30	33.86	54.37	16.21	60.00	11.11	35.70	47.14	15.21	72.73	29.61	38.25	41.02	20.87	65.38	28.50	34.56
Open-Source Training Methods																				
Llama-3.1-8B-Instruct-SFT-GRPO	51.64	43.51	52.33	45.85	48.13	54.23	41.47	40.00	35.36	45.87	52.43	45.14	63.48	36.08	49.85	57.12	53.83	60.32	34.41	52.33
Qwen3-8B-SFT-GRPO	55.59	37.08	66.97	42.24	49.60	51.06	27.81	60.00	19.92	38.53	55.70	32.26	47.62	30.30	47.56	59.93	37.88	38.26	31.40	47.92

Table 4: The main results of the 4 language families with relatively low comprehensive scores

stantial improvements are observed in "Numerical Computation" tasks.

We implement SFT and GRPO on two base models: Qwen3-8B and Llama-3.1-8B. Results demonstrate that these approaches substantially enhance performance across most language families and tasks. The fully-trained Llama-3.1-8B model yields state-of-the-art performance among open-source LLMs, ranking second only to Gemini-2.5 among all LLMs, which further validates the efficacy of our proposed training methodology and dataset curation strategy.

Impact of language family on LLM reasoning The results across language families reveal that Indo-European languages achieve the highest scores. This advantage stems from abundant high-quality corpora (e.g., English) produced by its extensive and widely distributed speaker base, whose generated data effectively supports LLM learning. However, the Niger-Congo language family - although it has 800 million native speakers predominantly in underdeveloped regions - yields limited corpora, reflected in lower LLM performance. In contrast, Uralic languages (25 million speakers) exhibit comparatively stronger performance. This disparity arises from Uralic speakers' concentration in developed regions, generating disproportionately greater digital resources than Niger-Congo languages. For example, Finnish (Uralic) has 598,717 Wikipedia articles versus Swahili's (Niger-Congo) 100,179 articles (Data as of July 2025). This resource asymmetry highlights significant imbalances in linguistic resource allocation that challenge equitable LLM development.

4.3 Ablation study

We employ Qwen3-8B as the base model for ablation studies to examine the impact of thinking mode, SFT, and GRPO on cross-lingual performance. As shown in Table 5, both SFT and the activation of thinking mode yield substantial and comparable performance improvements over the base model. Specifically, SFT achieves a relative improvement of 133.80%, while the activation of thinking mode leads to a 142.04% relative improvement. Following the completion of SFT, we further optimize the model using the GRPO algorithm. Experimental results demonstrate that this optimization yields a comprehensive performance improvement of approximately 6 points across both the thinking mode and non-thinking mode settings.

In conclusion, the systematic incorporation of thinking capabilities yields significant incremental gains at different training stages: Base pretraining (17.23 points), SFT (11.18 points), and GRPO (10.11 points). These results demonstrate its essential role in enhancing complex multilingual table understanding.

5 Analysis

Effectiveness of training data generated by LLM. We observe that although M^{3}TQA-INSTRUCT is entirely generated by LLMs without human annotation, models fine-tuned on this dataset and further optimized via reinforcement learning exhibit substantial improvements when evaluated on M^{3}TQA-BENCH. This finding potentially informs LLM optimization strategies, suggesting that task-relevant synthetic training data generated by LLMs can effectively boost base model performance at lower operational costs.

Language family	Afroasiatic	Indo-European	Austronesian	Sino-Tibetan	Austroasiatic	Niger-Congo	Uralic
Qwen3-8B w/o Thinking	12.85	12.55	13.56	5.97	11.44	11.58	9.06
w/ SFT	23.93	30.65	29.10	22.86	21.65	22.02	27.63
w/ GRPO	32.19	38.93	25.63	35.10	34.25	35.77	33.60
Qwen3-8B w/ Thinking	22.90	31.76	28.61	23.52	26.25	16.45	34.56
w/ SFT	34.84	41.28	41.04	30.24	36.59	31.27	42.81
w/ GRPO	47.56	49.17	36.86	49.60	35.79	47.92	43.61
Language family	Mongolic	Dravidian	Turkic	Kra-Dai	Language isolate	Avg.	/
Qwen3-8B w/o Thinking	12.14	13.31	10.18	10.25	11.98	12.13	/
w/ SFT	29.56	29.29	27.67	20.37	24.20	28.36	/
w/ GRPO	40.82	38.70	34.21	31.51	28.39	35.06	/
Qwen3-8B w/ Thinking	34.07	28.43	30.58	24.45	27.35	29.36	/
w/ SFT	43.46	44.25	41.22	33.98	35.23	39.54	/
w/ GRPO	50.55	47.80	44.41	38.53	37.44	45.17	/

Table 5: The ablation study about thinking mode, SFT, and GRPO.

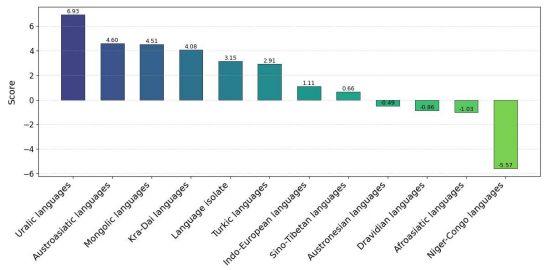

Figure 4: Difference between thinking mode and SFT score (thinking mode score - SFT score)

Thinking mode and SFT Figure 4 reveals distinct cross-linguistic response patterns to thinking mode versus SFT, despite comparable overall performance gains. We identify a pronounced geographical correlation: Thinking mode yields superior enhancement for languages predominantly spoken in developed regions (Indo-European, Sino-Tibetan, Uralic), while SFT demonstrates greater efficacy for languages concentrated in underdeveloped regions (Niger-Congo, Afro-Asiatic, Austronesian). This divergence reflects fundamental disparities in linguistic resource availability within LLMs' knowledge bases. For high-resource languages, inherent high-quality representations enable effective task resolution through structured reasoning alone. Conversely, low-resource languages benefit from SFT's capacity to mitigate knowledge gaps through task-aligned external supervision.

6 Related Work

The advent of artificial intelligence (AI) has driven breakthroughs in a wide range of tasks (Yang et al., 2026b,c, 2025d, 2026a, 2025c,e). To quantitatively evaluate the performance of AI models in diverse scenarios, a variety of high-quality datasets have been proposed.

Development of Table QA Datasets. To advance research on Table QA, several English benchmark datasets have been proposed (Akhtar et al., 2025; Chen et al., 2020; Zhong et al., 2017; Pasupat and Liang, 2015). These datasets play a crucial role in evaluating models' semantic understanding capabilities regarding tabular structures. Currently, benchmarks such as (Chang et al., 2023; Zhao et al., 2023; Sui et al., 2024a; Wu et al., 2025a) specifically assess LLMs' table comprehension abilities (Tiwari et al., 2025; Ye et al., 2023; Wang et al., 2024; Wu et al., 2025b; Zheng et al., 2024; Titiya et al., 2025). However, these datasets remain exclusively English-centric, failing to meet the evaluation needs of modern multilingual LLMs.

Exploration of Multilingual Datasets. To address the multilingual evaluation gap, researchers have introduced non-English QA datasets (Liu et al., 2019; Clark et al., 2020; Liu et al., 2024). In the tabular data domain, IM-TQA (Zheng et al., 2023) construct benchmarks for table classification and QA using Chinese corporate annual reports and encyclopedic web pages. (Zhang et al., 2025b; Minhas et al., 2022) extending INFOTABS (Gupta et al., 2020) to 10 languages (covering 3 billion native speakers) through translation techniques, enabling cross-language family and cross-script evaluation. For low-resource languages, recent studies (Cho et al., 2025; Pal et al., 2024) address data scarcity through selective question translation and cross-lingual adaptation methods, while analyzing the impact of linguistic features (e.g., word order) on model performance.

7 Conclusion

In this work, we introduce M^{3}TQA, a comprehensive and linguistically complex Table QA dataset to evaluate multilingual tabular reasoning capabilities,This dataset covers 97 languages across 12 language families. M^{3}TQA-BENCH contains

2,916 LLM-generated QA pairs and 3,690 human-constructed QA pairs with meticulous annotations, while M^{3}TQA-INSTRUCT contains 39,982 LLM-generated QA pairs with no human annotation. We benchmark 20+ models on M^{3}TQA-BENCH to quantify cross-lingual performance variations under diverse task constraints. Experimental results demonstrate that models trained on M^{3}TQA-INSTRUCT achieve significant performance gains despite the absence of human annotation. M^{3}TQA establishes a new evaluation standard for multilingual table understanding and provides a rigorous benchmarking platform for assessing model robustness across linguistic typologies.

Limitations

Several limitations should be acknowledged. Although we employed two large language models and manual verification to minimise translation artifacts, the original data sources are exclusively English and Chinese. As a result, the translated instances cannot fully capture the unique cultural nuances, idioms, or social norms of other languages—a form of cultural bias that persists despite our best efforts. Beyond this, the entire training set was generated by large language models, with numerical computation problems accounting for over 70% of the data. Such skew likely encourages models to specialise in numeric computation while under-learning other reasoning patterns, potentially biasing performance estimates and limiting generalisability.

A further limitation concerns our language analysis: we compared across language families (e.g., Sino-Tibetan vs. Indo-European) but overlooked within-family diversity. For instance, within Indo-European, languages using Latin, Cyrillic, or other writing systems exhibit substantial orthographic and morphological variation that we did not control for. Lastly, our study design does not address cross-lingual evaluation scenarios—such as querying non-English tabular data using English prompts—despite this being a common practical use case.

These limitations point to important directions for future work, including native-source data collection, more balanced task generation, fine-grained family-internal analysis, and cross-lingual evaluation protocols.

Ethics Statement

In this study, we utilize tabular data derived from real-world, publicly available sources online. To rigorously uphold ethical standards and protect individual privacy, all personally identifiable and sensitive information within the dataset has been thoroughly de-identified and anonymized. Consequently, the processed data used for analysis presents no risk of information leakage or compromise to personal privacy, ensuring that our research complies with established ethical principles for data handling.

Acknowledgments

This work is supported by the Fundamental Research Funds for the Central University (Grant No. GW2025-19) and supported by State Key Laboratory of Complex & Critical Software Environment (Grant No. SKLCCSE-2025ZX-26).

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024. Phi-4 technical report. *Preprint*, arXiv:2412.08905.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

AI@Meta. 2024. Llama 3 model card.

Mubashara Akhtar, Chenxi Pang, Andreea Marzoca, Yasemin Altun, and Julian Martin Eisenschlos. 2025. Tanq: An open domain dataset of table answered questions. *Transactions of the Association for Computational Linguistics*, 13:461–480.

ByteDance Seed, Yuyu Zhang, Jing Su, Yifan Sun, Chenguang Xi, Xia Xiao, Shen Zheng, Anxiang Zhang, Kaibo Liu, Daoguang Zan, Tao Sun, Jinhua Zhu, Shulin Xin, Dong Huang, Yetao Bai, Lixin Dong, Chao Li, Jianchong Chen, Hanzhi Zhou, and 8 others. 2025. Seed-Coder: Let the code model curate data for itself. *Preprint*, arXiv:2506.03524.

Shuaichen Chang, Jun Wang, Mingwen Dong, Lin Pan, Henghui Zhu, Alexander Hanbo Li, Wuwei Lan, Sheng Zhang, Jiarong Jiang, Joseph Lilien, and 1 others. 2023. Dr. spider: A diagnostic evaluation benchmark towards text-to-sql robustness. *arXiv preprint arXiv:2301.08881*.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036.

Sanghyun Cho, Minho Kim, Hye-Lynn Kim, Jung-Hun Lee, and Hyuk-Chul Kwon. 2025. Multilingual table question answering for low-resource languages via selective question translation and cross-lingual adaptation. In *2025 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 264–272. IEEE.

Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi qa: A benchmark for information-seeking question answering in ty pologically di verse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Jillian M Clements, Di Xu, Nooshin Yousefi, and Dmitry Efimov. 2020. Sequential deep learning for credit risk monitoring with tabular financial data. *arXiv preprint arXiv:2012.15330*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

DeepSeek-AI, :, Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, and 69 others. 2024. Deepseek llm: Scaling open-source language models with longtermism. *Preprint*, arXiv:2401.02954.

Sebastian Gehrmann, Sebastian Ruder, Vitaly Nikolaev, Jan Botha, Michael Chavinda, Ankur Parikh, and Clara Rivera. 2023. Tata: A multilingual table-to-text dataset for african languages. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1719–1740.

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

Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. Deepfm: a factorization-machine based neural network for ctr prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 1725–1731.

Vivek Gupta, Maitrey Mehta, Pegah Nokhiz, and Vivek Srikumar. 2020. Infotabs: Inference on tables as semi-structured data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2309–2324.

Jiahua Liu, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2019. Xqa: A cross-lingual open-domain question answering dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2358–2368.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, and 9 others. 2025. Muon is scalable for llm training. *Preprint*, arXiv:2502.16982.

Yang Liu, Meng Xu, Shuo Wang, Liner Yang, Haoyu Wang, Zhenghao Liu, Cunliang Kong, Yun Chen, Maosong Sun, and Erhong Yang. 2024. Omgeval: An open multilingual generative evaluation benchmark for large language models. *arXiv preprint arXiv:2402.13524*.

Bhavnick Minhas, Anant Shankhdhar, Vivek Gupta, Divyanshu Aggarwal, and Shuo Zhang. 2022. Xinfotabs: Evaluating multilingual tabular natural language inference. In *Proceedings of the Fifth Fact Extraction and VERification Workshop (FEVER)*, pages 59–77.

Md Nahid and Davood Rafiei. 2024a. Normtab: Improving symbolic reasoning in llms through tabular data normalization. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3569–3585.

Md Nahid and Davood Rafiei. 2024b. Tabsqlify: Enhancing reasoning capabilities of llms through table decomposition. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5725–5737.

Vaishali Pal, Evangelos Kanoulas, Andrew Yates, and Maarten Rijke. 2024. Table question answering for low-resourced indic languages. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 75–92.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480.

Mohammadreza Pourreza and Davood Rafiei. 2023. Din-sql: Decomposed in-context learning of text-to-sql with self-correction. *Advances in Neural Information Processing Systems*, 36:36339–36348.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2024a. Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 645–654.

Yuan Sui, Jiaru Zou, Mengyu Zhou, Xinyi He, Lun Du, Shi Han, and Dongmei Zhang. 2024b. Tap4llm: Table provider on sampling, augmenting, and packing semi-structured data for large language model reasoning. In *EMNLP (Findings)*.

Shayan Talaei, Mohammadreza Pourreza, Yu-Chen Chang, Azalia Mirhoseini, and Amin Saberi. 2024. Chess: Contextual harnessing for efficient sql synthesis. *CoRR*.

Baidu ERNIE Team. 2025a. Ernie 4.5 technical report.

Qwen Team. 2025b. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Prasham Yatinkumar Titiya, Jainil Trivedi, Chitta Baral, and Vivek Gupta. 2025. Mmtbench: A unified benchmark for complex multimodal table reasoning. *arXiv preprint arXiv:2505.21771*.

Aman Tiwari, Shiva Krishna Reddy Malay, Vikas Yadav, Masoud Hashemi, and Sathwik Tejaswi Madhusudhan. 2025. Auto-cypher: Improving llms on cypher generation via llm-supervised generation-verification framework. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 623–640.

Zhiruo Wang, Haoyu Dong, Ran Jia, Jia Li, Zhiyi Fu, Shi Han, and Dongmei Zhang. 2021. Tuta: Tree-based transformers for generally structured table pre-training. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1780–1790.

Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, and 1 others. 2024. Chain-of-table: Evolving tables in the reasoning chain for table understanding. In *ICLR*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Xianjie Wu, Jian Yang, Linzheng Chai, Ge Zhang, Jiaheng Liu, Xeron Du, Di Liang, Daixin Shu, Xianfu Cheng, Tianzhen Sun, and 1 others. 2025a. Tablebench: A comprehensive and complex benchmark for table question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25497–25506.

Zhenhe Wu, Jian Yang, Jiaheng Liu, Xianjie Wu, Changzai Pan, Jie Zhang, Yu Zhao, Shuangyong Song, Yongxiang Li, and Zhoujun Li. 2025b. Table-r1: Region-based reinforcement learning for table understanding. *arXiv preprint arXiv:2505.12415*.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Haizhou Zhao, Hang Xu, Haoze Sun, and 36 others. 2025a. Baichuan 2: Open large-scale language models. *Preprint*, arXiv:2309.10305.

Jian Yang, Xianglong Liu, Weifeng Lv, Ken Deng, Shawn Guo, Lin Jing, Yizhi Li, Shark Liu, Xianzhen Luo, Yuyu Luo, and 1 others. 2025b. From code foundation models to agents and applications: A comprehensive survey and practical guide to code intelligence. *arXiv preprint arXiv:2511.18538*.

Jian Yang, Jiaxi Yang, Wei Zhang, Ke Jin, Yibo Miao, Lei Zhang, Liqun Yang, Zeyu Cui, Yichang Zhang, Zhoujun Li, Binyuan Hui, and Junyang Lin. 2025c. Codearena: Evaluating and aligning codellms on human preference. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 9672–9683. Association for Computational Linguistics.

Jian Yang, Wei Zhang, Shawn Guo, Zhengmao Ye, Lin Jing, Shark Liu, Yizhi Li, Jiajun Wu, Cening Liu, X Ma, and 1 others. 2026a. Iquest-coder-v1 technical report. *arXiv preprint arXiv:2603.16733*.

Jian Yang, Wei Zhang, Yizhi Li, Shawn Guo, Haowen Wang, Aishan Liu, Ge Zhang, Zili Wang, Zhoujun Li, Xianglong Liu, and 1 others. 2025d. Codesimpleqa: Scaling factuality in code large language models. *arXiv preprint arXiv:2512.19424*.

Jian Yang, Wei Zhang, Yibo Miao, Shanghaoran Quan, Zhenhe Wu, Qiyao Peng, Liqun Yang, Tianyu Liu, Zeyu Cui, Binyuan Hui, and Junyang Lin. 2025e. Qwen2.5-xcoder: Multi-agent collaboration for multilingual code instruction tuning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 13121–13131. Association for Computational Linguistics.

Jian Yang, Wei Zhang, Jiajun Wu, Junhang Cheng, Shawn Guo, Haowen Wang, Weicheng Gu, Yaxin Du, Joseph Li, Fanglin Xu, and 1 others. 2026b. Incoder-32b: Code foundation model for industrial scenarios. *arXiv preprint arXiv:2603.16790*.

Jian Yang, Wei Zhang, Jiajun Wu, Junhang Cheng, Tuney Zheng, Fanglin Xu, Weicheng Gu, Lin Jing, Yaxin Du, Joseph Li, and 1 others. 2026c. Incoder-32b-thinking: Industrial code world model for thinking. *arXiv preprint arXiv:2604.03144*.

Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. 2023. Large language models are versatile decomposers: Decomposing evidence and questions for table-based reasoning. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pages 174–184.

Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. 2025. Demystifying long chain-of-thought reasoning in llms. *arXiv preprint arXiv:2502.03373*.

Han Zhang, Yuheng Ma, and Hanfang Yang. 2025a. Alter: Augmentation for large-table-based reasoning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 179–198.

Xuanliang Zhang, Dingzirui Wang, Keyan Xu, Qingfu Zhu, and Wanxiang Che. 2025b. Multitat: Benchmarking multilingual table-and-text question answering. *arXiv preprint arXiv:2502.17253*.

Yunjia Zhang, Jordan Henkel, Avrilia Floratou, Joyce Cahoon, Shaleen Deep, and Jignesh M Patel. 2024a. Reactable: enhancing react for table question answering. *Proceedings of the VLDB Endowment*, 17(8):1981–1994.

Zhehao Zhang, Yan Gao, and Jian-Guang Lou. 2024b. e5: Zero-shot hierarchical table analysis using augmented llms via explain, extract, execute, exhibit and extrapolate. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1244–1258.

Yilun Zhao, Chen Zhao, Linyong Nan, Zhenting Qi, Wenlin Zhang, Xiangru Tang, Boyu Mi, and Dragomir Radev. 2023. Robut: A systematic study of table qa robustness against human-annotated adversarial perturbations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6064–6081.

Mingyu Zheng, Xinwei Feng, Qingyi Si, Qiaoqiao She, Zheng Lin, Wenbin Jiang, and Weiping Wang. 2024. Multimodal table understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9102–9124.

Mingyu Zheng, Yang Hao, Wenbin Jiang, Zheng Lin, Yajuan Lyu, Qiaoqiao She, and Weiping Wang. 2023. Im-tqa: A chinese table question answering dataset with implicit and multi-type table structures. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5074–5094.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*.

A Table Selection

Our dataset incorporates tables from multiple sources, including public annual reports from the Shanghai Stock Exchange and Shenzhen Stock Exchange in China, as well as statistical reports from Statistics Canada and the U.S. National Science Foundation. An initial set of 150 tables is selected from these sources, consisting of 100 Chinese tables and 50 English tables, covering all table types described in Section 2.1. The proportion of each table type is determined based on the prevalence of such tables in real-world scenarios: relational tables account for 50%, matrix tables for 25%, and entity tables and composite tables each make up 12.5%. To increase task complexity, a number of rich-text tables—where most cells contain natural language descriptions—are also included. Due to the high degree of structural redundancy observed in the collected English tables (primarily relational tables, and mostly numerical in nature), the final selected tabular dataset is predominantly composed of Chinese tables. After this selection process, we retain a final set of 50 tables for dataset construction, which comprised 41 Chinese tables and 9 English tables. The tables span a wide range of domains, encompassing statistical data from fields such as finance, sports, education, and public affairs, along with related data on industrial production and entity information.

B Human Annotators

The annotation team consists of nine male computer science students (Master's and Ph.D. candidates), all fluent in both Chinese and English. To ensure quality control, we also recruit native speakers of other languages and linguistic professionals to perform sampled validation of the annotation results. All annotators are compensated at a rate of $5 per QA pair, while validators receive $3 per validated QA pair. We provide comfortable working conditions throughout the annotation and validation process, including complimentary meals, as well as necessary equipment and LLM API access.

C Table Translation Prompts

Figure 5 to Figure 9 illustrate the prompt design for table translation, where few-shot examples were employed during certain translation steps to enhance the output quality.

D Table QA generation prompt

Figure 10 to Figure 11 show the prompt for Table QA generation. The seed examples are manually annotated and consist entirely of English data. Following the question–answer taxonomy defined in TableBench, we create one English QA pair per table, covering 17 question types across multiple domains such as numerical reasoning, fact checking, and data analysis. During the QA generation phase, the prompt used for generating QA pairs is language-agnostic, which can automatically produce QA pairs in any target language directly from its corresponding table. To ensure the diversity of generated QA pairs, we adopt a type rotation strategy when sampling from the seed corpus. Specifically, we cycle through all 17 question types in a fixed order. In each sampling step, one QA pair is randomly selected from the currently chosen type. Moreover, the rotation state is preserved across different table-language combinations and does not reset when switching between them. For instance, if the previous combination is sampled from types 1–10, the next combination starts sampling from type 11.

E QA Consistency

In constructing the multilingual test sets, multiple strategies are employed to ensure translation accuracy. For the QA pairs generated by the LLM, we instruct the model to produce them directly based on the source-language tables. This approach is adopted because first constructing QA pairs in the source language and then translating both the tables and the QA pairs into target languages can introduce inconsistencies. Without shared context, identical content in the table and the QA pairs may be translated differently. The direct generation method reduces error propagation inherent in the translation process.

For the human-constructed QA pairs, however, the translation step is unavoidable. To maintain consistency, we manually align and annotate corresponding entities in both the tables and the QA pairs. This ensures that their semantic meaning remains unchanged before and after translation and that they correctly correspond to the content in the target-language tables. In effect, only grammatical elements such as particles and connectives are generated by the LLM itself, while the core substantive content of the QA pairs comes from manual annotation.

F LLMs performance on different test sets

During the annotation of QA pairs generated by the LLM, we observe a discernible difference between the questioning style of the LLM and that of human annotators. For instance, the LLM demonstrates a stronger tendency to pose questions focused on numerical data within tables. In tables describing entity information—even those containing only a single row of numerical cells—the majority of questions generated by the LLM still pertain to comparisons and computations involving these figures, while questions targeting other textual descriptions are relatively scarce. To bridge this gap, we introduce human-constructed and annotated QA pairs. Furthermore, the inclusion of human-annotated data allows for an evaluation of the generalizability of the LLM-generated training data, testing whether its effectiveness extends beyond the scenario of "self-generated question–self-generated answer".

Figure 16 presents the performance of different models on the two source test sets. Scores on both test sets exhibit an upward trend as model parameter count increases. However, a distinct pattern emerges: smaller-scale models (below 16B parameters) generally achieve superior performance on the human-constructed test set, whereas larger-scale models (above 16B parameters) tend to perform better on the LLM-generated test set. These results delineate distinct performance advantage zones associated with model scale and setting, which underscores the necessity of utilizing dual test sets. Consequently, this dual-test-set approach renders the evaluation leaderboard more comprehensive and challenging.

G LLM Prompt Examples

Figure 12 to Figure 15 show the prompt for different tasks, where `{Table data}` is the serialized table data and `{Question}` is the table question.

H Results of different types of tables

Table 6 shows the performance of different models on each type of table. Overall, most models perform better on entity tables, which may be related to the horizontal arrangement of information.

I All Language Results

Table 7 shows the scores of all models on each language. Including test results of open source models, closed source models, and open source training models.

For Qwen3-8B, the introduction of SFT produces substantial performance gains in most languages. The average score improvement reaches 9.77 points in general, with particularly pronounced enhancements exceeding 15 points for low-resource languages, including Maori (Austronesian), Xhosa (Niger-Congo) and Somali (Afro-Asiatic).

Subsequent GRPO training further increases comprehensive performance (from 38.09 to 43.51). However, language-specific analysis reveals divergent outcomes: Performance degradation occurs in Cambodian during SFT, while GRPO induces regression across multiple additional languages.This may be due to the cross-influence between multiple languages in M^{3}TQA-INSTRUCT, and indirectly reflects the complexity of multilingual tasks. These limitations underscore the need for more robust cross-lingual generalization techniques.

J Statistics for all languages

Table 8 summarizes the dataset by language, including the language family to which each language belongs and the amount of training and test data it has.

K About Chinese in Sino-Tibetan languages

The underperformance of Sino-Tibetan languages is anomalous given Chinese's status as a high-resource language(Table 4). This outcome arises from our evaluation methodology: when calculating the family's aggregate score, we use unweighted averaging across constituent languages. As a result, the dominant influence of Chinese (spoken by over 1.4 billion users) is diluted by numerous low-resource members, and the lack of resource-weighted scoring consequently depresses the family's overall performance metrics.

System:
You are a professional linguist skilled in translating **{source language}** into other languages. Your task is to translate the given text into the corresponding language while preserving the original semantics unchanged.

[Precautions]:
1. Do not output prompt words;
2. If the text content is difficult to translate (such as personal names, special company names, etc.), it should be translated by transliteration;
3. Strictly translate according to the original meaning, only output the translation result, do not fabricate any content, do not output unnecessary content, do not output the original content;
4. The translation content should maintain the original two-dimensional list structure and not be changed;
5. All elements in the output content are represented by double quotation marks. When special meanings need to be expressed within the text, single quotation marks are used uniformly.

The following is extracted content from a table. Your task is to translate the content in this table into the corresponding language and ensure that its meaning remains unchanged.
Translate the following content into **{target language}** according to the above instructions.

User:
{Source language table examples}

Assistant:
{Target language table examples}

User:
{Source language table to be translated}

Figure 5: Instruction of Initial Translation

System:
You are a professional linguist skilled in translating **{source language}** into other languages. Your task is to translate the given text into the corresponding language while preserving the original semantics unchanged.

[Precautions]:
1. Do not output prompt words;
2. If the text content is difficult to translate (such as personal names, special company names, etc.), it should be translated by transliteration;
3. Strictly translate according to the original meaning, only output the translation result, do not fabricate any content, do not output unnecessary content, do not output the original content, do not output **{source language}** characters;
4. All information is fictitious and has no practical significance.

Translate the following content into **{target language}** according to the above instructions. Do not include **{source language}** characters

User:
{Source language cell examples}

Assistant:
{Target language cell examples}

User:
{Source language cell to be translated}

Figure 6: Instruction of Cell Complement Translation

System:
You are a professional linguist skilled in translating **{source language}** into other languages.
Your task is to meticulously review the original **{source language}** table alongside its **{target language}** translation, evaluate the precision of the translation, contemplate potential enhancements, and ultimately, based on your critical analysis, deliver an optimized version of the translation.

[Precautions]:
1. The table is input in the form of a two-dimensional list, and the output strictly follows the original format and can be parsed by JSON.
2. Do not output any prompt words, do not output the thought process, only output the translation result.
3. If a cell contains a large amount of duplicate text, please reconsider the translation result for that cell.
4. If there are English prompts in a non-English cell, such as "translations," "text," "optimized," etc, please reconsider the translation result for that cell.

User:
{Source language} table:
{Source language table}
{Target language} table:
{Target language table}

Figure 7: Instruction of Global Refinement

System:
You are a professional linguist skilled in translating **{source language}** into other languages.
Your task is to meticulously review the original **{source language}** text alongside its **{target language}** translation, evaluate the precision of the translation, contemplate potential enhancements, and ultimately, based on your critical analysis, deliver an optimized version of the translation.
[Precautions]: Do not output any prompt words, do not output the thought process, only output the translation result.

User:
{Source language} text:
{Source language cell}
{Target language} text:
{Target language cell}

Figure 8: Instruction of Cell Refinement

System:

You are a linguistics expert. Your task is to translate the given text into the corresponding language's text, while preserving the original semantics unchanged.

The following is content extracted from a table. Your task is to translate the content of this table except for numerical values into the corresponding language's text, and ensure that the expressed meaning remains unchanged.

[Precautions]:

1. Do not output prompt words;
2. If the text content is difficult to paraphrase (such as personal names, special company names, etc.), then translate it using the transliteration method;
3. Strictly translate according to the original meaning, only output the translation result, do not output descriptions such as 'the translation of XXX into XX language is XXX', do not fabricate any content, do not output extra content, do not output the original content;
4. Maintain the original structure of the translation content, do not change it;
5. If the cell content is 'nan', then change its content to empty;
6. Use double quotes to represent all elements in the output content, and when quotes are needed inside the text to express special meanings, always use single quotes.

Translate the following **{target language}** content into **{Source language}** based on the above instructions.

User:

{Target language table}

Figure 9: Instruction of Back-Translation Validation

System:

You are an expert data analyst tasked with generating question-answer pairs from tabular data.

Your task is to refer to the given table and give one pair of QA you set. The table is in **{target language}** , and you give the questions and answers in **{target language}**.

Your mission is to:

1. Carefully analyze the input table structure and content
2. Study the provided example QA pair to understand:
   - Question type and phrasing style
   - Answer depth and formatting
   - Data interpretation approach
3. Generate 1 new QA pair that:
   - Cover different aspects of the table data
   - Maintain consistent style with the example
   - Demonstrate varied question types (comparisons, calculations, trends)
4. The given problem needs to be difficult. Generate challenging QA pairs from tabular data with controlled difficulty levels.

   Difficulty Control Parameters

   a. **Cognitive Complexity**

      Must include at least two items:

      - The correlation analysis must involve at least 3 data points in the table
      - Multi step reasoning (2+logical steps)
      - Need to include cross-row/cross-column comparison or calculation
      - Implicit relationship derivation
      - Conditional hypothesis verification
      - Contains multiple layers of logical relationships

   b. **Technical Depth**

      Mandatory inclusion of at least 2 of the following:

      - Ratio/rate of change calculation
      - Time series trend prediction
      - Abnormal value recognition
      - Data conflict detection
      - Multi dimensional cross comparison

Figure 10: Table QA Generation Prompt Part 1

5. The answer must meet the following requirements

answer type 1: If the problem type is numerical calculation, perform necessary mathematical operations. Provide answers aligned with the units in the question (such as thousands, millions, etc.), rounded to two decimal places. The final result must be presented in the form of a list and can only contain numbers and percentages (if any), and the order of the answers must be the same as the order in which they appear in the question. Do not add a thousand separator in numerical values and use a '.' to indicate the decimal point. Example answer: [15600.23, 8.61%]

answer type 2: If the answer to the question is the value of several cells in the table, then directly give the cell contents and cell coordinates of the Excel type. The output format is as follows: [{{"answer": "sports", "locate": "B4"}},{{"answer": "89%", "locate": "D7"}}]

answer type 3: If the question is a true or false judgment, the output is based on the answer. If it is true, output T, if it is false, output F.

answer type 4: For open questions, just give the answer according to the question requirements.

6. Format output strictly as:

```
[
  {
    "question": "...",
    "answer": "...",
    "difficulty": "...",
    "answer_type": "..."
  },
]
```

7. Quality Check:

- Ensure that each question requires at least two steps of reasoning to answer
- Verify that the numerical value in the answer corresponds accurately to the table data
- Eliminate questions that can be directly answered with a single data point
- Check whether the complexity of the question meets the standard
- Check the accuracy of the answer calculation
- Confirm the correctness of the data reference

8. Output one set of QA pairs. Only output JSON data, do not output any prompt words.

Example table QA:

User:

table content: **{Target language table example}**

Giving **{Question type}** type QA pairs.

Assistant:

{TQA example}

User:

table content: **{Target language table}**

Giving **{Question type}** type QA pairs.

Figure 11: Table QA Generation Prompt Part 2

<Data input>
<Table data>
{Table data}
</Table data>
<Question>
{Question}
</Question>
</Data input>

Processing steps:
1. Read the table carefully and identify the correspondence between the table header and the data row
2. When analyzing the problem, pay special attention to:
- Numerical problems: the calculation object, formula, and unit must be clearly defined
3. QA requirements:
- Perform necessary mathematical operations
- Check the consistency of units during calculation
- Verify the numerical units (e.g., thousand, million) mentioned in the question and provide an answer that aligns with the specified units, rounded to two decimal places.
- The final result can only contain numbers and percent signs (if any)
- The final results must be given in the form of a list, and the order of the answers must be the same as the order in which they appear in the question.
- No additional text description. Do not add thousands separators to the answer value, and use '.' to represent the decimal point.
- Sample output: <answer>[15600.23, 8.61%]</answer>

Accuracy verification:
- Numerical calculations require the reasoning process to be written in the <verify> tag (not displayed to the user)

Final output requirements:
- Only include the final result in the <answer> tag
- Do not add explanatory text
- Strictly follow the sample format
- The value is accurate to two decimal places

Now start to deal with the problem, and execute the corresponding analysis process.

Figure 12: Instruction of Numerical Computation

<Data input>
<Table data>
{Table data}
</Table data>
<Question>
{Question}
</Question>
</Data input>

Processing steps:
1. Read the table carefully and identify the correspondence between the table header and the data row
2. When analyzing the problem, pay special attention to:
- Positioning problems: all cells that meet the conditions must be located
3. QA requirements:
- Traverse all relevant cells in the table
- Check whether each candidate cell meets the question conditions
- Output format: The answer and coordinates form a dict, where the value of "answer" represents the content of the answer and the value of "locate" represents the coordinates of the answer
- Use the Excel-style coordinates (e.g., "A1", "B2") directly from the original table without any operations (such as transposing or rearranging), and maintain the exact cell references as they appear in the source data.
- Sample output:
<answer>[{{"answer": "Best sales in the quarter", "locate": "B4"}},{{"answer": "89%", "locate": "D7"}}]</answer>

Accuracy verification:
- The positioning result must be traceable in the original table

Final output requirements:
- Only include the final result in the <answer> tag
- Do not add explanatory text
- Strictly follow the sample format

Now start to deal with the problem, and execute the corresponding analysis process.

Figure 13: Instruction of Cell Extraction

<Data input>
<Table data>
{Table data}
</Table data>
<Question>
{Question}
</Question>
</Data input>

Processing steps:
1. Read the table carefully and identify the correspondence between the table header and the data row
2. When analyzing the problem, pay special attention to:
- True or false problems: confirmation of whether the judgment condition is established
3. QA requirements:
- Verify the truth of the proposition based on the table data
- If it is true, output T, if it is false, output F.
- Sample output: <answer>F</answer>

Accuracy verification:
- The right or wrong judgment needs to refer to the clear basis in the table

Final output requirements:
- Only include the final result in the <answer> tag
- Do not add explanatory text
- Strictly follow the sample format

Now start to deal with the problem, and execute the corresponding analysis process.

Figure 14: Instruction of Factual Verification

<Data input>
<Table data>
{Table data}
</Table data>
<Question>
{Question}
</Question>
</Data input>

Processing steps:
1. Read the table carefully and identify the correspondence between the table header and the data row
2. When analyzing the problem, pay special attention to:
- Descriptive questions: Identify key information to formulate natural language responses
3. QA requirements:
- Provide answers in complete English sentences based on table data
- Keep responses concise
- Maintain neutral, factual tone
- Sample output: <answer>The population of Tokyo is 37 million according to the data.</answer>

Accuracy verification:
- All responses must be directly supported by table data
- Avoid speculation or interpretation beyond visible data

Final output requirements:
- Enclose only the final natural language answer in <answer> tags
- Do not include analysis steps or explanations
- Maintain consistent formatting
- The tables and questions will be given in {} language and you answer questions in the same language.

Now start to deal with the problem, and execute the corresponding analysis process.

Figure 15: Instruction of Open-Ended Questions

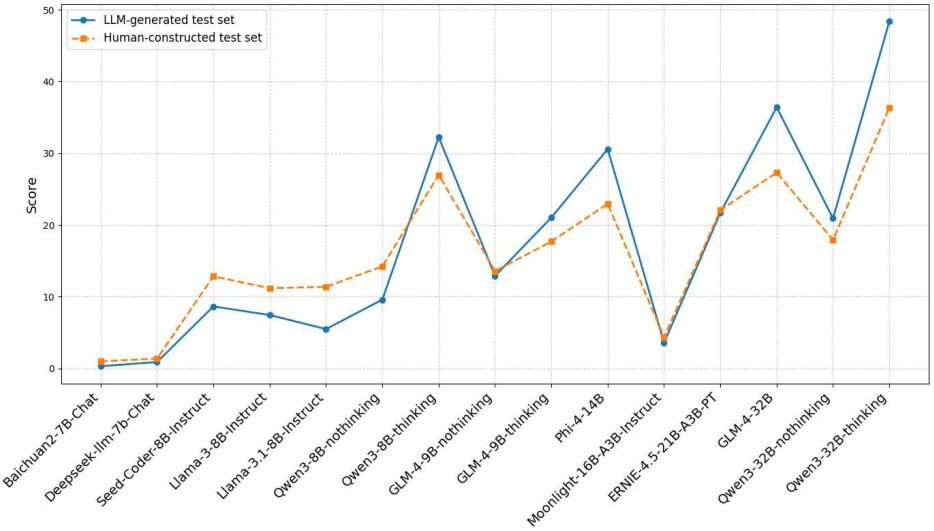

Figure 16: The performance of the LLM on test sets derived from two distinct sources.

	Relational table	Entity table	Matrix table	Composite table
Open-source Methods				
Baichuan2-7B-Chat	0.56	0.57	0.83	0.86
Deepseek-llm-7B-Chat	0.76	2.25	1.03	1.82
Seed-Coder-8B-Instruct	11.57	11.64	8.94	12.52
Llama-3-8B-Instruct	7.06	16.24	9.05	12.76
Llama-3.1-8B-Instruct	6.09	16.15	8.39	11.63
Qwen3-8B-nothinking	11.45	12.73	10.50	18.02
Qwen3-8B-thinking	26.99	41.72	27.96	28.51
GLM-4-9B-0414-nothinking	11.06	20.38	12.10	16.91
GLM-4-9B-0414-thinking	17.66	30.08	15.76	21.61
Phi-4-14B	25.04	36.44	23.27	27.63
Moonlight-16B-A3B-Instruct	3.53	5.75	3.72	4.60
ERNIE-4.5-21B-A3B-PT	21.18	25.19	20.18	25.34
GLM-4-32B	30.43	42.93	28.89	28.08
Qwen3-32B-nothinking	18.51	21.21	19.15	20.35
Qwen3-32B-thinking	41.39	49.32	41.86	34.86
Close-source Methods				
GPT-3.5-turbo	14.59	23.77	12.66	18.35
GPT-4o	35.03	46.97	33.45	35.69
Gemini-2.0	29.53	45.56	24.82	28.27
Gemini-2.5	55.23	52.97	50.43	42.89
Deepseek-Chat-V3	42.50	51.49	42.96	38.40
Open-Source Training Methods				
Llama-3.1-8B-Instruct-SFT	33.51	44.32	34.27	33.99
Qwen3-8B-SFT	37.89	47.71	39.39	37.43

Table 6: Performance of each model on different types of tables.

Method	Aar	Af	Sq	Arq	Am	Ar	As	Ban	Bal	Eu	Be	Bn	Bho	Bs	Bul	My	Csx	Ca	Nya	Zh
Open-source Methods																				
Baichuan2-7B-Chat	0.00	0.22	0.00	0.22	0.00	0.00	3.02	1.99	0.17	0.00	0.17	1.70	0.15	1.71	0.32	0.00	0.08	0.31	0.64	1.58
Deepseek-llm-7B-Chat	1.05	2.97	0.76	2.79	0.00	0.00	0.00	0.50	0.64	1.34	0.11	1.39	2.79	2.22	1.27	0.76	0.00	0.20	0.38	1.57
Seed-Coder-8B-Instruct	14.42	14.44	9.39	7.05	2.89	12.08	7.01	17.07	9.45	8.16	9.84	7.83	8.15	10.12	11.52	7.47	3.08	15.27	5.40	15.10
Llama-3-8B-Instruct	11.53	14.06	11.06	7.51	5.00	9.08	5.19	12.94	11.23	5.01	6.99	7.40	9.52	10.85	11.02	3.44	0.94	11.57	7.69	7.45
Llama-3.1-8B-Instruct	7.28	13.91	6.86	4.26	1.10	7.79	6.74	11.92	10.47	7.36	5.75	11.89	9.36	10.59	6.60	6.31	4.40	13.46	13.48	2.75
Qwen3-8B-nothinking	17.59	14.96	10.79	10.95	9.65	11.11	12.67	9.92	9.90	10.26	10.26	12.70	14.47	8.28	11.19	7.56	9.93	14.14	7.88	4.71
Qwen3-8B-thinking	16.22	33.12	28.66	20.58	23.69	33.59	29.25	31.43	26.24	39.22	32.16	30.24	36.49	26.94	29.25	14.20	24.74	33.68	19.70	30.98
GLM-4-9B-0414-nothinking	7.96	19.77	14.11	9.35	8.26	16.08	7.83	11.93	9.26	11.94	13.66	13.86	14.04	15.92	12.99	5.96	4.40	11.89	9.86	11.96
GLM-4-9B-0414-thinking	18.72	21.30	18.38	18.95	4.68	18.71	16.31	17.54	20.21	23.36	21.16	17.52	18.99	24.14	24.83	4.27	9.72	20.35	9.20	21.57
Phi-4-14B	17.83	27.06	36.63	16.76	9.08	28.54	28.71	24.35	29.68	24.08	19.90	27.74	27.37	26.63	31.47	11.61	11.39	32.63	12.94	21.96
Moonlight-16B-A3B-Instruct	8.28	4.90	1.89	4.83	1.55	2.51	3.19	2.21	2.33	4.03	3.24	1.22	5.78	8.06	3.02	0.42	0.38	5.59	3.08	5.04
ERNIE-4.5-21B-A3B-PT	16.05	25.39	18.44	11.65	18.62	23.88	22.13	24.69	20.56	22.72	24.29	21.38	23.46	20.19	17.38	16.60	15.93	19.47	28.11	18.04
GLM-4-32B	19.00	36.04	39.78	24.27	25.11	31.64	25.18	32.93	18.82	37.25	24.10	31.45	32.30	33.44	36.83	17.42	20.50	36.12	25.71	32.82
Qwen3-32B-nothinking	21.63	15.61	27.70	15.43	19.63	17.33	20.72	15.75	14.47	18.65	20.25	25.08	28.60	22.52	14.55	16.12	12.18	19.65	13.76	11.61
Qwen3-32B-thinking	23.68	50.96	52.55	36.71	36.53	48.59	32.16	46.91	35.63	43.93	38.77	46.47	46.99	32.93	49.47	31.20	33.61	50.66	20.57	38.04
Close-source Methods																				
GPT-3.5-turbo	19.69	22.20	18.95	16.97	8.70	19.65	9.13	16.28	10.65	17.20	12.23	14.05	15.22	18.20	16.43	10.15	6.13	17.50	10.08	11.02
GPT-4o	26.49	38.82	38.69	27.37	25.51	31.29	31.53	36.32	30.53	44.07	33.77	30.35	41.94	33.81	42.11	17.02	25.07	41.54	27.05	28.01
Gemini-2.0	26.60	29.72	38.22	31.08	30.76	26.43	24.89	41.85	31.93	28.61	30.43	21.54	41.14	24.21	28.43	21.80	22.59	31.53	31.54	31.57
Gemini-2.5	31.59	49.37	58.48	38.99	49.01	52.33	44.95	49.82	54.78	61.07	54.98	52.94	54.45	49.26	54.88	33.38	36.13	55.38	49.18	52.35
Deepseek-Chat-V3	32.87	40.74	52.46	31.67	33.23	44.49	35.98	43.94	41.64	42.28	46.56	51.82	44.11	41.50	46.38	29.50	24.73	46.23	39.04	37.35
Open-Source Training Methods																				
Llama-3.1-8B-Instruct-SFT	26.83	34.79	39.13	26.48	19.56	35.23	40.86	39.81	39.65	37.00	29.75	34.98	37.23	34.63	41.61	22.70	15.44	39.66	25.96	25.39
Llama-3.1-8B-Instruct-SFT-GRPO	50.39	48.62	52.96	42.55	46.35	53.01	50.20	46.16	49.98	58.23	48.70	52.49	45.20	45.52	60.17	42.97	39.12	49.30	41.93	45.67
Qwen3-8B-SFT	23.11	34.58	40.75	29.40	31.76	36.32	35.26	47.33	41.98	40.52	30.86	46.98	41.72	39.38	39.18	27.60	22.59	43.73	34.85	32.35
Qwen3-8B-SFT-GRPO	25.92	33.34	48.04	38.38	38.05	46.58	45.74	52.03	41.24	46.20	43.07	52.33	49.77	46.53	50.75	27.74	32.41	45.89	32.78	39.78

Method	Hr	Cs	Da	Nl	En	Et	Fil	Fi	Fr	Ka	De	El	Kal	Gu	Ha	He	Hi	Hu	Is	Id
Open-source Methods																				
Baichuan2-7B-Chat	0.00	0.68	0.27	1.27	3.05	0.00	1.16	0.21	0.00	0.05	1.88	0.23	0.17	1.66	0.25	0.20	0.23	0.38	0.00	0.31
Deepseek-llm-7B-Chat	0.46	0.50	1.90	1.77	0.53	0.28	1.88	0.73	1.21	1.49	2.11	0.08	1.02	0.80	2.58	0.73	4.03	0.00	1.62	1.93
Seed-Coder-8B-Instruct	16.24	18.59	16.47	13.53	9.40	10.25	16.22	8.81	14.07	10.97	15.39	11.03	9.26	9.46	10.36	6.38	14.97	12.71	9.84	11.82
Llama-3-8B-Instruct	11.69	14.83	5.18	15.39	11.04	3.81	16.36	7.29	9.89	11.07	5.35	12.54	8.53	11.10	7.60	6.30	10.70	6.51	7.61	9.79
Llama-3.1-8B-Instruct	5.55	10.73	14.23	11.08	7.95	4.27	10.45	11.99	10.48	12.01	8.03	9.25	8.63	7.68	3.75	7.73	13.01	6.95	5.29	7.38
Qwen3-8B-nothinking	10.73	12.34	13.16	12.28	8.99	5.30	18.10	8.64	17.12	14.67	7.88	15.99	17.87	12.85	13.39	15.37	15.59	13.42	11.85	16.22
Qwen3-8B-thinking	31.94	35.85	37.63	35.17	26.66	33.16	34.71	32.76	36.57	29.74	31.41	25.17	17.11	35.71	6.85	36.15	38.11	37.77	28.75	37.56
GLM-4-9B-0414-nothinking	19.22	17.92	19.67	13.20	20.23	12.42	15.84	11.56	14.93	9.42	16.93	13.40	14.61	9.87	7.09	11.85	15.69	16.73	13.20	14.21
GLM-4-9B-0414-thinking	19.94	26.21	24.45	31.77	21.47	19.48	24.53	28.11	22.28	14.69	23.14	16.95	15.29	11.84	9.00	18.46	22.32	24.99	15.52	27.12
Phi-4-14B	27.71	30.97	24.10	36.80	23.35	30.08	31.30	30.10	30.82	22.29	30.04	27.53	17.96	27.64	15.73	27.13	28.07	29.35	33.66	31.34
Moonlight-16B-A3B-Instruct	3.83	4.79	6.86	2.30	5.89	3.24	7.98	3.47	3.55	3.72	3.05	5.23	4.59	1.40	4.89	2.03	4.18	5.05	2.90	6.06
ERNIE-4.5-21B-A3B-PT	23.33	27.31	21.33	33.01	16.04	22.30	31.11	20.80	17.65	23.37	14.68	20.73	11.98	19.41	14.41	20.58	25.08	19.71	23.61	28.05
GLM-4-32B	35.21	35.33	39.25	41.53	27.32	40.87	38.62	34.36	34.54	26.77	33.28	31.96	23.71	31.62	25.10	39.95	33.13	35.64	36.71	42.69
Qwen3-32B-nothinking	16.35	20.81	17.37	15.38	22.02	12.14	22.09	14.66	18.63	21.83	18.13	17.93	16.24	24.51	15.78	23.90	23.02	25.61	15.15	14.52
Qwen3-32B-thinking	48.72	46.93	48.91	52.86	41.65	44.97	45.64	38.72	39.31	47.19	44.26	42.19	30.64	39.85	19.84	49.46	47.23	57.19	40.02	42.47
Close-source Methods																				
GPT-3.5-turbo	13.89	18.48	17.87	17.76	14.62	12.73	20.36	16.96	17.98	9.90	19.15	18.16	16.63	9.05	12.59	12.74	14.27	18.76	18.07	15.07
GPT-4o	37.75	38.36	35.97	42.20	33.55	30.86	46.47	37.87	32.76	38.88	35.28	35.78	28.70	45.83	29.09	39.45	33.85	39.03	39.91	40.65
Gemini-2.0	34.14	23.13	31.91	32.41	32.07	34.14	29.11	27.98	27.05	33.19	28.65	21.70	22.92	37.31	28.30	28.19	24.90	24.35	23.37	31.07
Gemini-2.5	65.01	50.88	62.82	62.15	57.65	56.00	57.26	55.95	57.85	52.45	44.70	59.80	35.04	59.32	38.21	42.70	48.51	63.66	56.59	52.61
Deepseek-Chat-V3	51.05	42.88	49.23	45.79	41.53	43.76	45.57	46.27	39.53	40.38	41.97	47.58	32.88	49.79	34.68	48.27	42.85	54.28	52.41	51.60
Open-Source Training Methods																				
Llama-3.1-8B-Instruct-SFT	41.88	41.57	42.10	39.70	35.69	35.58	38.02	31.80	34.22	35.15	36.57	38.97	26.75	30.67	29.00	34.91	36.74	40.68	40.73	41.14
Llama-3.1-8B-Instruct-SFT-GRPO	58.56	55.63	47.32	52.67	45.94	44.49	40.23	48.18	47.43	48.84	44.74	52.93	51.91	46.08	38.19	59.94	49.25	54.86	53.73	51.41
Qwen3-8B-SFT	45.86	40.24	42.20	46.78	39.79	37.20	47.83	40.07	39.53	36.09	39.48	42.70	28.31	42.68	26.77	47.00	35.33	51.35	49.40	48.52
Qwen3-8B-SFT-GRPO	52.15	51.44	48.95	46.66	43.87	49.12	50.95	41.64	45.72	49.82	42.73	46.94	22.41	49.15	25.93	46.13	44.45	52.58	48.56	60.56

Method	Ga	It	Ja	Jav	Xal	Kn	Kk	Rw	Kon	Ko	Ky	Lao	Lat	Lv	Li	Mk	Mg	Ms	Ml	Mt
Open-source Methods																				
Baichuan2-7B-Chat	1.73	1.51	1.11	0.14	0.13	1.45	0.32	0.89	0.22	0.12	0.21	0.28	0.59	0.00	0.99	1.20	0.00	0.74	0.00	0.18
Deepseek-llm-7B-Chat	0.00	2.12	1.14	2.63	0.62	2.05	0.28	0.00	5.41	0.00	0.00	2.00	1.24	0.38	0.71	0.48	0.43	1.49	0.64	0.25
Seed-Coder-8B-Instruct	5.80	19.35	10.34	11.63	7.76	5.55	12.61	14.32	9.71	8.25	8.17	10.51	6.72	15.25	7.84	8.98	2.44	15.00	5.17	8.94
Llama-3-8B-Instruct	11.67	7.22	3.79	7.78	6.39	7.52	19.04	8.55	17.23	6.96	7.60	0.00	11.22	7.91	10.18	11.29	6.07	10.30	10.20	11.01
Llama-3.1-8B-Instruct	7.95	11.01	6.58	10.40	12.06	4.86	8.75	6.51	7.38	9.63	5.54	6.15	6.50	6.54	7.44	11.36	5.71	11.89	5.19	8.52
Qwen3-8B-nothinking	12.71	13.79	11.98	10.93	11.73	11.26	12.70	11.50	21.83	12.78	7.15	8.59	11.07	13.14	11.30	14.40	6.91	18.92	10.89	14.54
Qwen3-8B-thinking	28.93	30.31	31.10	33.04	37.03	31.76	38.40	12.42	18.61	30.35	22.62	19.78	24.74	37.75	35.16	35.35	10.62	35.91	24.15	24.04
GLM-4-9B-0414-nothinking	10.76	16.11	11.48	9.92	12.08	5.85	17.99	6.81	7.80	23.17	7.44	5.17	5.88	10.90	11.58	9.49	8.84	17.09	8.46	9.42
GLM-4-9B-0414-thinking	22.99	18.85	26.14	23.09	21.40	8.86	22.89	6.86	7.86	21.70	14.80	5.31	11.47	24.54	20.43	18.15	5.41	32.23	10.79	21.41
Phi-4-14B	24.44	29.61	28.79	28.69	25.41	25.00	33.93	11.61	15.31	32.20	26.25	11.89	38.27	23.21	35.44	24.41	18.50	37.39	31.91	21.79
Moonlight-16B-A3B-Instruct	4.06	3.65	5.51	5.83	2.67	2.88	4.76	4.12	6.99	2.47	2.74	2.39	5.16	1.57	4.90	6.26	1.33	4.47	0.50	4.94
ERNIE-4.5-21B-A3B-PT	18.17	20.97	21.42	20.17	22.77	21.77	27.62	17.83	20.57	16.80	23.21	18.65	27.18	22.45	24.74	22.32	19.36	22.98	25.56	25.56
GLM-4-32B	36.82	32.90	35.17	35.67	38.70	28.32	34.48	12.27	13.88	39.21	33.15	10.47	30.01	29.76	30.68	36.46	20.90	42.90	33.21	30.27
Qwen3-32B-nothinking	26.34	20.87	16.80	11.21	13.37	22.31	16.35	14.65	22.99	14.51	13.86	13.18	25.45	13.24	18.41	22.01	8.70	19.50	24.67	13.10

Table 7 The performance of all models in various languages.

Qwen3-32B-thinking	57.86	44.79	44.55	40.82	49.40	40.16	50.32	19.56	19.63	43.45	48.78	26.75	43.06	50.47	46.68	47.81	20.28	48.00	45.86	30.48
Close-source Methods																				
GPT-3.5-turbo	13.47	15.16	16.21	14.94	11.58	15.50	17.30	14.27	16.45	15.45	10.96	3.34	17.40	19.63	13.91	16.25	13.42	20.42	17.55	16.93
GPT-4o	46.85	38.32	39.02	30.80	38.04	39.29	42.49	22.42	39.71	37.95	37.09	14.59	37.08	31.63	43.34	38.17	30.10	38.08	39.74	41.00
Gemini-2.0	32.27	27.38	31.17	30.45	36.70	36.20	42.68	26.31	31.46	19.54	26.59	20.86	31.41	31.60	29.38	30.75	25.03	33.20	38.73	36.20
Gemini-2.5	65.57	53.44	51.44	54.61	51.00	44.43	63.84	33.97	42.17	54.29	48.49	45.10	53.36	54.37	55.51	49.43	60.65	54.99	57.23	48.52
Deepseek-Chat-V3	57.39	49.99	43.24	43.60	46.55	36.73	48.00	24.09	41.98	39.83	44.15	33.28	45.57	49.55	48.00	46.31	47.63	52.31	47.73	34.19
Open-Source Training Methods																				
Llama-3.1-8B-Instruct-SFT	41.34	37.79	32.68	29.30	41.02	36.83	43.69	29.04	29.11	32.21	37.34	25.51	38.83	39.18	36.24	40.59	22.63	39.02	33.81	32.21
Llama-3.1-8B-Instruct-SFT-GRPO	57.60	54.83	51.02	51.02	48.16	50.10	52.09	52.39	44.68	45.36	50.53	51.98	54.14	48.58	53.65	45.58	43.64	48.91	48.93	51.67
Qwen3-8B-SFT	50.88	38.94	35.80	37.05	50.26	37.55	49.59	23.93	32.91	39.89	38.03	31.05	46.50	44.64	47.65	39.91	27.33	41.41	40.06	37.33
Qwen3-8B-SFT-GRPO	52.12	47.30	45.65	45.59	48.57	41.15	53.52	37.50	31.71	48.71	43.93	32.54	45.32	52.31	49.53	48.49	40.96	48.69	49.23	44.89

Method	Mri	Mr	Mon	Mos	Ne	No	Ps	Fa	Pl	Pt	Pa	Que	Ro	Rom	Ru	San	Sr	Si	Sk	Sl
Open-source Methods																				
Baichuan2-7B-Chat	1.23	0.08	0.22	0.00	0.69	0.25	1.30	0.78	0.87	0.53	0.00	0.08	0.54	1.41	1.18	0.70	0.38	0.10	0.21	1.88
Deepseek-llm-7B-Chat	2.37	0.27	0.05	0.00	1.42	2.73	0.65	1.66	1.42	1.32	1.12	1.13	3.00	0.08	0.20	1.84	1.05	0.00	0.82	0.58
Seed-Coder-8B-Instruct	9.40	5.89	8.24	3.03	10.17	11.82	8.23	14.54	14.52	10.00	6.28	10.68	18.00	10.10	14.09	9.02	18.80	1.71	14.00	13.87
Llama-3-8B-Instruct	8.72	4.98	10.13	6.92	5.23	10.80	13.21	12.54	12.07	12.12	7.50	6.47	14.17	8.04	10.02	8.55	15.85	4.85	9.04	11.95
Llama-3.1-8B-Instruct	8.94	6.11	9.66	9.66	4.36	12.06	7.08	14.30	15.00	8.43	7.41	6.06	15.76	5.91	7.63	11.21	6.94	1.86	12.58	4.86
Qwen3-8B-nothinking	11.07	10.90	12.49	10.18	11.60	12.17	12.11	17.44	12.60	10.91	13.27	3.53	18.07	10.60	9.70	10.75	12.62	6.42	14.97	10.94
Qwen3-8B-thinking	9.90	23.22	31.62	7.77	26.84	37.86	24.59	35.02	30.91	36.97	29.34	7.66	38.04	21.49	29.21	32.61	38.63	19.43	36.89	29.83
GLM-4-9B-0414-nothinking	5.38	10.63	10.22	10.04	11.21	23.34	6.96	23.74	20.23	21.69	15.96	6.34	19.96	7.51	15.58	17.90	23.44	4.48	17.56	12.65
GLM-4-9B-0414-thinking	5.25	13.20	17.00	10.94	23.23	21.79	18.07	26.35	22.40	24.63	16.57	9.33	30.62	13.81	20.91	19.27	28.47	7.21	26.76	21.63
Phi-4-14B	13.89	23.65	18.61	10.41	24.61	31.20	28.02	35.63	31.19	34.39	30.79	18.34	31.54	15.59	26.15	32.54	29.44	11.67	31.33	25.01
Moonlight-16B-A3B-Instruct	1.39	1.00	5.82	1.32	3.57	6.79	5.70	4.49	7.50	6.19	1.93	6.17	6.02	2.56	3.87	6.63	6.34	1.23	4.49	4.31
ERNIE-4.5-21B-A3B-PT	20.58	23.12	14.04	12.35	24.63	27.37	25.76	26.76	21.58	28.95	20.99	9.57	24.94	18.17	25.43	24.86	33.70	20.97	25.34	19.73
GLM-4-32B	16.43	25.97	22.34	13.09	36.05	36.76	30.66	39.93	32.76	39.04	33.76	14.63	30.68	34.97	32.82	30.77	36.26	26.73	36.91	28.15
Qwen3-32B-nothinking	22.14	21.74	21.01	10.30	25.55	20.83	22.35	27.51	20.76	16.58	22.04	7.55	22.95	21.85	21.05	25.82	26.17	22.09	18.57	17.36
Qwen3-32B-thinking	40.53	43.75	41.05	13.38	48.11	44.73	34.73	40.98	37.15	44.66	42.54	15.72	43.14	32.16	43.66	46.35	45.13	39.38	47.23	50.68
Close-source Methods																				
GPT-3.5-turbo	15.38	8.67	17.60	7.54	11.47	26.80	14.00	21.11	18.52	16.09	13.51	9.39	25.56	8.20	14.62	11.99	19.85	7.84	16.20	17.42
GPT-4o	35.97	27.37	43.79	18.98	38.10	37.04	41.13	43.50	39.13	35.83	43.94	26.57	38.57	23.65	35.18	33.94	37.53	39.61	39.28	34.72
Gemini-2.0	39.13	22.30	28.74	25.29	34.47	26.25	43.68	25.84	20.99	26.99	27.62	17.61	39.15	31.46	26.45	29.38	34.06	35.76	29.19	23.86
Gemini-2.5	53.02	48.56	47.45	20.11	48.35	56.63	56.49	51.50	53.66	55.77	56.96	35.33	57.11	50.06	47.39	49.31	47.68	50.92	58.97	52.56
Deepseek-Chat-V3	40.96	39.22	45.94	15.91	43.52	42.34	38.62	51.24	44.39	46.31	42.39	22.21	40.06	30.02	46.43	45.64	50.89	40.04	49.86	43.29
Open-Source Training Methods																				
Llama-3.1-8B-Instruct-SFT	37.59	33.15	29.63	14.87	33.74	34.22	36.10	39.27	35.40	44.37	40.99	17.97	38.46	27.50	39.64	38.71	41.15	32.49	41.78	41.28
Llama-3.1-8B-Instruct-SFT-GRPO	51.25	49.92	48.72	40.93	52.59	58.50	50.81	55.91	51.28	48.59	42.34	45.05	44.58	54.69	46.05	55.82	49.28	46.48	50.60	47.52
Qwen3-8B-SFT	34.07	34.98	37.83	12.42	34.08	38.42	38.24	46.69	38.32	42.82	46.96	26.49	42.17	39.67	37.89	39.29	44.86	33.14	47.17	41.49
Qwen3-8B-SFT-GRPO	46.13	44.56	48.12	12.89	41.94	46.45	47.46	52.61	40.35	45.21	46.11	27.88	50.49	38.83	42.34	44.55	46.76	48.52	55.13	45.41

Method	Som	Es	Swa	Sv	Tg	Ta	Te	Th	Tr	Uk	Ur	Ug	Uz	Vi	Xh	Yor	Zu
Open-source Methods																	
Baichuan2-7B-Chat	2.63	0.18	0.25	0.28	1.02	0.13	1.92	0.22	0.22	2.16	1.39	1.15	0.30	0.00	2.00	0.00	0.00
Deepseek-llm-7B-Chat	0.61	2.34	1.71	2.58	0.17	0.00	0.00	2.79	0.54	0.72	3.62	0.00	0.30	0.80	4.38	5.56	0.00
Seed-Coder-8B-Instruct	8.46	23.96	10.01	11.64	6.73	10.99	11.14	9.26	6.75	16.27	15.66	9.50	8.94	6.84	10.32	16.67	2.04
Llama-3-8B-Instruct	8.39	8.90	9.50	15.14	12.40	10.12	7.28	12.25	8.67	12.28	17.82	7.29	11.27	11.93	8.28	12.96	3.87
Llama-3.1-8B-Instruct	11.49	14.12	13.80	15.77	8.87	7.99	8.78	7.80	5.01	8.19	14.27	11.78	11.10	7.98	4.17	11.11	1.95
Qwen3-8B-nothinking	10.95	12.06	9.42	12.78	11.60	18.35	12.38	11.52	13.63	12.50	23.66	9.37	7.42	13.98	6.51	25.97	3.21
Qwen3-8B-thinking	13.35	39.44	22.87	30.94	22.76	31.33	26.89	28.00	37.75	31.65	39.11	15.92	35.57	28.78	11.93	18.52	13.61
GLM-4-9B-0414-nothinking	9.38	25.03	7.01	19.71	15.56	16.89	10.47	8.68	17.33	19.85	13.53	6.54	9.41	19.08	12.30	5.56	4.28
GLM-4-9B-0414-thinking	9.88	26.71	14.54	25.70	16.41	11.77	11.32	13.37	30.12	25.83	29.44	6.15	22.47	27.80	13.33	18.52	0.92
Phi-4-14B	13.06	30.56	27.56	32.69	22.28	23.75	38.39	24.85	29.46	28.15	34.77	20.39	22.51	30.56	16.85	24.79	12.51
Moonlight-16B-A3B-Instruct	4.00	6.10	1.86	9.36	3.67	4.40	0.32	0.52	2.16	8.02	1.65	3.46	2.18	0.00	9.80	1.39	0.57
ERNIE-4.5-21B-A3B-PT	20.58	20.47	21.03	28.57	20.40	29.13	17.72	25.00	19.51	26.93	30.00	20.33	19.62	21.38	16.82	17.83	15.48
GLM-4-32B	18.86	45.12	36.44	33.12	15.40	32.73	22.78	24.07	29.89	37.91	34.72	22.56	30.93	33.77	20.11	28.67	19.47
Qwen3-32B-nothinking	19.67	18.84	16.08	22.61	10.75	30.68	27.12	18.76	19.67	24.30	23.13	11.12	17.69	30.76	10.28	30.56	5.28
Qwen3-32B-thinking	25.02	51.71	39.34	42.13	37.15	44.45	39.45	40.43	42.07	42.41	45.87	29.99	46.74	40.35	20.90	63.89	23.48
Close-source Methods																	
GPT-3.5-turbo	11.02	21.01	17.86	24.83	9.29	14.41	14.90	15.93	17.47	19.62	24.45	10.17	15.38	20.77	10.57	29.26	14.03
GPT-4o	38.71	35.77	37.65	34.18	40.18	44.18	37.72	31.68	36.31	39.74	43.24	27.99	42.56	40.29	37.00	43.95	29.16
Gemini-2.0	36.65	31.54	26.44	24.19	32.47	46.49	34.37	25.78	24.03	33.40	40.11	28.53	25.86	31.96	41.56	40.53	24.91
Gemini-2.5	52.27	59.51	44.04	53.42	49.72	60.69	53.20	48.60	50.66	54.16	55.25	43.78	55.16	57.86	42.96	61.25	53.02
Deepseek-Chat-V3	41.23	51.04	41.68	41.97	38.79	44.42	46.61	37.53	47.45	42.92	50.37	40.87	47.06	46.79	38.57	47.41	25.01
Open-Source Training Methods																	
Llama-3.1-8B-Instruct-SFT	33.27	38.45	28.73	36.34	25.60	42.92	35.30	29.10	29.38	31.27	43.20	30.83	39.24	42.82	25.26	58.74	29.87
Llama-3.1-8B-Instruct-SFT-GRPO	50.65	49.95	37.03	45.50	48.07	46.80	43.48	40.11	55.26	43.09	59.12	45.05	53.95	44.94	49.72	49.38	44.50
Qwen3-8B-SFT	40.72	45.06	37.06	38.25	32.10	52.28	45.85	36.22	39.43	43.29	45.74	38.73	40.20	46.81	33.53	45.03	27.21
Qwen3-8B-SFT-GRPO	37.56	50.85	47.33	47.16	39.56	55.27	52.39	41.04	45.06	45.30	48.41	42.01	50.83	50.44	29.42	43.39	33.03

Table 7: The performance of all models in various languages

Code	Language	Language Family	Train	Test	Code	Language	Language Family	Train	Test
aar	Afar	Afroasiatic	369	49	ky	Kyrgyz	Turkic	405	74
af	Afrikaans	Indo-European	423	77	lao	Lao	Kra-Dai	349	51
sq	Albanian	Indo-European	395	76	lat	Latin	Indo-European	404	63
arq	Algerian Arabic	Afroasiatic	414	63	lv	Latvian	Indo-European	422	67
am	Amharic	Afroasiatic	367	60	lt	Lithuanian	Indo-European	423	74
ar	Arabic	Afroasiatic	431	79	mk	Macedonian	Indo-European	414	78
as	Assamese	Indo-European	412	70	mg	Malagasy	Austronesian	369	61
ban	Balinese	Austronesian	422	58	ms	Malay	Austronesian	404	72
bal	Baluchi	Indo-European	405	59	ml	Malayalam	Dravidian	411	69
eu	Basque	isolate	405	66	mt	Maltese	Afroasiatic	404	64
be	Belarusian	Indo-European	404	61	mri	Maori	Austronesian	351	53
bn	Bengali	Indo-European	413	72	mr	Marathi	Indo-European	430	73
bho	Bhojpuri	Indo-European	423	67	mon	Mongolian	Mongolic	404	69
bs	Bosnian	Indo-European	441	78	mos	Mossi	Niger-Congo	270	41
bul	Bulgarian	Indo-European	414	79	ne	Nepali	Indo-European	421	79
my	Burmese	Sino-Tibetan	396	68	no	Norwegian	Indo-European	441	76
csx	Cambodian	Austroasiatic	349	57	ps	Pashto	Indo-European	386	62
ca	Catalan	Indo-European	414	77	fa	Persian	Indo-European	403	64
nya	Chichewa	Niger-Congo	387	41	pl	Polish	Indo-European	423	72
zh	Chinese	Sino-Tibetan	448	85	pt	Portuguese	Indo-European	423	76
hr	Croatian	Indo-European	422	85	pa	Punjabi	Indo-European	411	63
cs	Czech	Indo-European	405	76	que	Quechua	isolate	306	49
da	Danish	Indo-European	405	73	ro	Romanian	Indo-European	414	70
nl	Dutch	Indo-European	422	73	rom	Romany	Indo-European	414	52
en	English	Indo-European	432	87	ru	Russian	Indo-European	413	77
et	Estonian	Uralic	423	82	san	Sanskrit	Indo-European	439	68
fil	Filipino	Austronesian	404	69	sr	Serbian	Indo-European	413	79
fi	Finnish	Uralic	423	75	si	Sinhalese	Indo-European	404	63
fr	French	Indo-European	421	82	sk	Slovak	Indo-European	405	73
ka	Georgian	isolate	423	67	sl	Slovenian	Indo-European	432	86
de	German	Indo-European	423	74	som	Somali	Afroasiatic	405	61
el	Greek	Indo-European	396	67	es	Spanish	Indo-European	423	77
kal	Greenlandic	isolate	288	50	swa	Swahili	Niger-Congo	394	74
gu	Gujarati	Indo-European	405	67	sv	Swedish	Indo-European	414	70
ha	Hausa	Afroasiatic	396	58	tg	Tajik	Indo-European	369	54
he	Hebrew	Afroasiatic	405	77	ta	Tamil	Dravidian	405	72
hi	Hindi	Indo-European	403	68	te	Telugu	Dravidian	414	75
hu	Hungarian	Uralic	405	78	th	Thai	Kra-Dai	422	67
is	Icelandic	Indo-European	432	84	tr	Turkish	Turkic	423	71
id	Indonesian	Austronesian	423	79	uk	Ukrainian	Indo-European	423	80
ga	Irish	Indo-European	396	67	ur	Urdu	Indo-European	422	73
it	Italian	Indo-European	432	86	ug	Uyghur	Turkic	359	49
ja	Japanese	isolate	431	88	uz	Uzbek	Turkic	432	75
jav	Javanese	Austronesian	404	61	vi	Vietnamese	Austroasiatic	429	78
xal	Kalmyk	Mongolic	395	57	xh	Xhosa	Niger-Congo	394	53
kn	Kannada	Dravidian	396	61	yor	Yoruba	Niger-Congo	368	43
kk	Kazakh	Turkic	423	67	zu	Zulu	Niger-Congo	395	54
rw	Kinyarwanda	Niger-Congo	340	50					
kon	Kongo	Niger-Congo	288	44					
ko	Korean	isolate	385	68					

Table 8: Statistics of the training and test sets for all languages. The languages are ranked in alphabet.