ZiCiEval: Challenging Large Language Models with Seemingly Basic Chinese Character-Word Questions

Anonymous ACL submission

Abstract

Large language models (LLMs) have made significant progress in handling complex tasks, while some seemingly basic questions remain unexpectedly unsolved. In practice, LLMs are prone to hallucinate on free-form questions about Chinese characters and words, which causes inconvenience for ordinary users or language learners who use LLMs to acquire Chinese knowledge. To quantitatively investigate the issue, we introduce **ZiCiEval**, a dataset covering five types of real-world Chinese character-word questions. For reliable automatic evaluation, we develop an LLM-as-judge framework enhanced with adaptive tool use. Empirical results demonstrate substantial performance gaps among advanced language models. In some tasks, the top-performing models only reach 70% accuracy. The resources will be publicly available to facilitate further research.

1 Introduction

Large Language Models (LLMs) are making rapid strides in conquering difficult problems that require high-level intelligence (Chang et al., 2024). However, the most advanced models still occasionally stumble over some seemingly simple tasks that are common in everyday use. For example, the famous "strawberry problem" (counting some letters in a word) becomes an unexpected challenge (Fu et al., 2024), revealing that LLMs process language in a way fundamentally different from humans.

Similarly, some basic tasks related to Chinese characters and words could be surprisingly problematic. For example, the question "What character is formed by combining 木 and 乞" has become a popular test on Chinese social networks. LLMs are prone to severely hallucinate when faced with such questions, as shown in Figure 1.

Why are such basic Chinese tasks difficult to resolve? Tokenization and knowledge reporting bias might be part of the reason. Tokens do not inherently encode phonetic and structural information. Models learn their representations through their occurrences and context in corpora. Crucially, certain intuitive knowledge for humans (e.g. components of a character) is rarely directly expressed. Such knowledge sparsity exacerbates language models' hallucination when handling related queries. Worse still, wrong outputs are contaminating public corpora, thus adversely affecting both model training and retrieval-augmented systems.

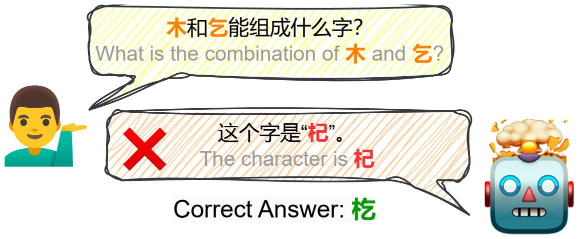
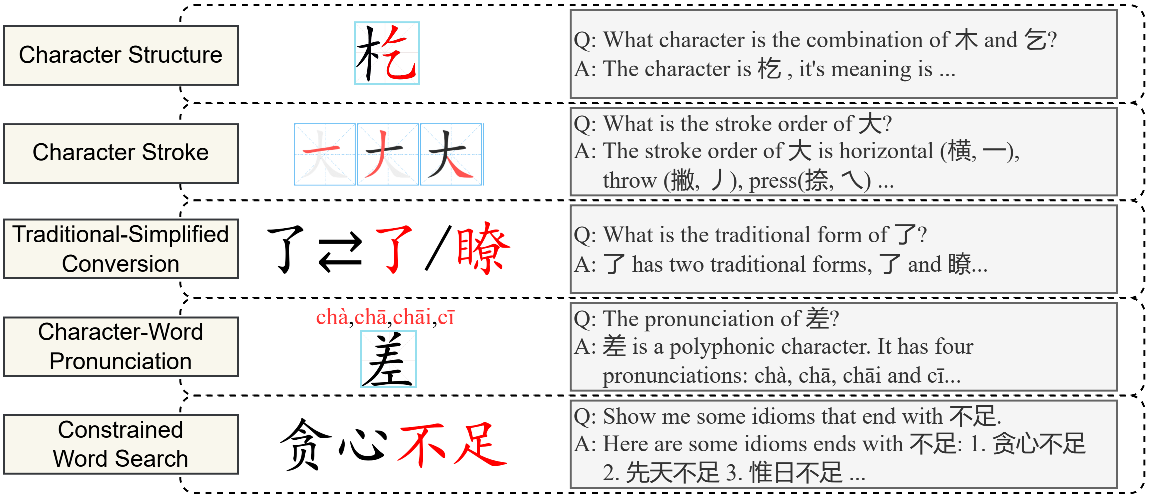


Figure 1: A Chinese character question that stumps most LLMs.

The issue is not negligible. Due to the vast number of Chinese characters and their profound cultural significance, both native Chinese speakers and language learners frequently have free-form querying requirements about characters, leveraging character structures, strokes and pronunciations for naming decisions and other creative endeavors. For such needs, today's users prefer to seek help from LLMs rather than cumbersome dictionaries. However, most Chinese LLM evaluation benchmarks (Xu et al., 2020, 2023; Li et al., 2024) focus on general knowledge test or traditional NLP tasks, leaving a gap in evaluating the performance on such real-world Chinese character-word questions.

In this paper, we present **ZiCiEval**[1], an open-ended QA dataset for quantitatively evaluating Chinese character-word capabilities of LLMs. We first identify five representative task types through a pilot study on mainstream models. Based on the task taxonomy, we carefully curate a 500-sample

[1] "字词" (**ZiCi**) means characters and words in Chinese.



Q: What character is the combination of 木 and 乞?
A: The character is 杚 , it's meaning is ...

Q: What is the stroke order of 大?
A: The stroke order of 大 is horizontal (横, 一), throw (撇, 丿), press(捺, ㇏) ...

Q: What is the traditional form of 了?
A: 了 has two traditional forms, 了 and 瞭...

Q: The pronunciation of 差?
A: 差 is a polyphonic character. It has four pronunciations: chà, chā, chāi and cī...

Q: Show me some idioms that end with 不足.
A: Here are some idioms ends with 不足: 1. 贪心不足 2. 先天不足 3. 惟日不足 ...

Figure 2: An illustration of the five sub-tasks in ZiciEval. The labels on the left are task types. The corresponding knowledge is illustrated in the middle. The sample QA pairs are on the right (translated to English).

evaluation dataset based on real-world anonymous search logs and authentic volunteer user queries, ensuring diversity in complexity and questioning styles. Concurrently, we collect 3.5k training samples from the same source to facilitate research.

To address the challenge of automatically evaluating free-form responses from LLMs, we implement a tool-enhanced LLM-As-Judge evaluation framework. Specifically, through optimized prompt-engineering, we use a model to verify the correctness of responses with human-validated reference answers. To enhance the reliability of judgment when the responses contain content outside the scope of reference answers, we develop a character knowledge toolkit that the model can invoke as needed. The combined approach achieves nearly 90% agreement with human evaluation.

Based on the evaluation protocol, we perform a benchmark of 15 open-weights and proprietary LLMs. Empirical results demonstrate substantial performance gaps among the advanced models. In some tasks, the top-performing systems still only solve 70+% questions, indicating further efforts are still needed to improve the capability of LLMs.

2 Dataset Construction

2.1 Identifying the Tasks

First, we aim to identify the character-word questions that are frequently posed by users but current language models struggle to answer. Therefore, we conduct a pilot study before the formal dataset construction. We sample 140 questions from anonymous search logs and analyze their involved knowledge types. These questions cover a wide range of knowledge, including orthography (the structure and strokes of a character), phonology (the phonetic notation, a.k.a *pinyin* of a character or word), and lexicology (the meaning and compositional structure of a word).

After that, we obtain the responses to these questions from three LLMs (GPT-4o, ERNIE-4, and Doubao Pro). We manually review the results. All models achieve 90% accuracy in directly explaining meaning and identifying semantic relationships (i.e. synonyms and antonyms), while other tasks still have larger error rates (refer to Appendix A).

Based on this, we focus on five representative task types, as illustrated in Figure 2. The tasks include **character structure** (questions related to the combination and disassembly of characters), **character stroke** (questions related to the character strokes count and order), **traditional-simplified conversion** (converting Chinese character forms), **character-word pronunciation** (questions related to pinyin annotation and polyphonic character analysis), and **constrained word search** (finding words that meet the constraints).

2.2 Data Collection and Annotation

After determining the task taxonomy, we further collect more real-world questions from both anonymous logs and volunteer users' feedback. These questions are de-duplicated and then automatically classified by LLMs. After that, we obtain two different responses from models for which we have licenses to utilize their outputs. We sample 5k questions and their responses for subsequent annotation.

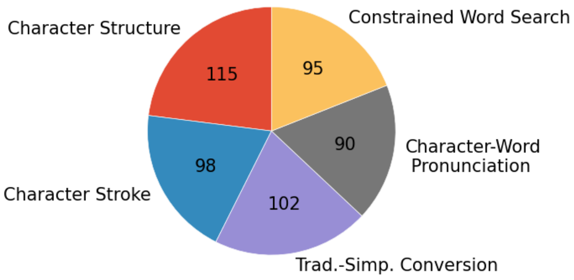


Figure 3: The distribution of the evaluation data.

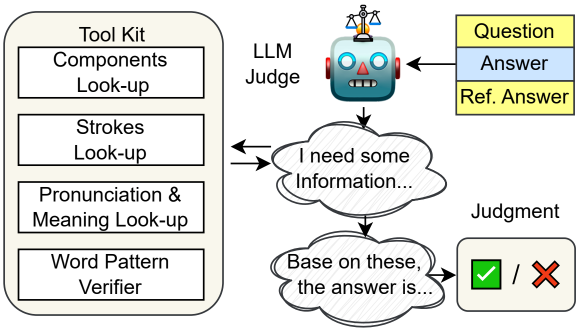


Figure 4: Automatic evaluation with tool-enhanced LLM judge. Refer to Appendix B for more details.

During the annotation, we recruit native Chinese annotators to create a reference answer for each question based on the provided responses. The reference answer should have correct content and completely meet the requirements of the question. For questions without appropriate answers, a reasonable refusal response should be given. The annotators are encouraged to use dictionaries and other recommended tools to obtain necessary background knowledge to verify the correctness. If a question exceeds the verification ability of the annotator, it can be discarded. Also, the annotators are asked to identify the task type of each question. Notably, a question could involve multiple tasks, we only ask annotators to provide a main task type. After one round of annotation is completed, each instance is checked by a different annotator to ensure the quality. From a sampling inspection on 1% of the results, the qualified rate reached 95%.

For evaluation dataset construction, we sample 500 instances from the valid annotated results. To ensure a higher level of quality, each instance is checked and corrected again by our research team members. The remaining 3.5k annotated results serve as a training dataset.

Figure 3 shows the evaluation data distribution.

2.3 Automatic Evaluation

In practice, LLMs can give free-form responses, which contain markups and knowledge hints for better readability and helpfulness. This poses challenges for evaluation. Therefore, we utilize the curated reference answer and turn to the LLM-As-Judge paradigm for automatic accuracy evaluation. Specifically, we use Qwen2.5-72B-Instruct (Yang et al., 2025b) as the judge model. Through prompt-engineering optimization, we highlight the key evaluation aspects for different tasks.

However, some cases may exceed the model's capabilities. For open questions such as list words that meet specific criteria, the reference answer may not cover all correct results. Also, for fixed-answer questions, the model responses may include additional information beyond the key answer. Therefore, we create a database of Chinese characters and words based on open resources, and develop a Python toolkit for querying background knowledge and pattern verification. Equipped with the toolkit, the evaluation method becomes a two-round process. As shown in Figure 4, the LLM judge uses multiple tools in parallel to acquire necessary information in the first round, and make the final judgment in the second round.

To verify the reliability of the automatic evaluation, we randomly sample 450 test responses from different models and manually annotate their correctness. After that, we apply automatic evaluation on these instances, and conduct an agreement analysis. The agreement rate between human and the automatic judge is 89.9% and the cohen's κ is 0.76, indicating substantial agreement. Furthermore, we find that the agreement rate on samples deemed correct by humans are much higher than the agreement on samples where human identify errors (94.8% vs 76.6%), which indicates the system ignores some nuanced mistakes and thus it may slightly overestimate the model performance during evaluation.

3 Experiments and Results

3.1 Implementation Details

In the experiments, we evaluate a series of open-weights LLMs and proprietary LLMs (refer to Appendix C for details). The models are grouped into non-reasoning models (e.g. DeepSeek-V3) and reasoning models (e.g. DeepSeek-R1). For open-weights models, we use the official recommended generation setting. For proprietary models, we use the default setting of the official API.

Model	Character Structure	Character Stroke	Trad.-Simp. Conversion	Char.-Word Pronunciation	Constrained Word Search	Macro Acc.	Micro Acc.
Non-Reasoning Models							
Qwen2.5-72B-Instruct	24.35	9.18	70.59	46.67	56.84	41.53	41.00
GPT-4o-20241120	35.65	20.41	75.49	56.67	51.58	47.96	47.60
GLM-4-Plus	39.13	29.59	69.61	57.78	52.63	49.75	49.40
Qwen3-235B (Non-Thinking)	47.83	19.39	79.41	65.56	71.58	56.75	56.40
DeepSeek-V3	55.65	26.53	81.37	64.44	60.00	57.60	57.60
Qwen-Max-0125	53.04	35.71	79.41	68.89	63.16	60.04	59.80
Doubao-1.5-Pro-32k-250115	56.52	48.98	71.57	70.00	77.89	64.99	64.60
Hunyuan-TurboS-20250416	66.96	**72.45**	63.73	77.78	69.47	70.08	69.80
ERNIE-4.0-Turbo	67.83	68.37	77.45	76.67	70.53	72.17	72.00
DeepSeek-V3-0324	**72.17**	50.00	**82.35**	**85.56**	**78.95**	**73.81**	**73.60**
Reasoning Models							
QwQ-32B	20.87	11.22	67.65	47.78	57.89	41.08	40.40
Qwen3-235B (Thinking)	46.09	24.49	73.53	70.00	70.53	56.93	56.40
OpenAI O3	59.13	25.51	87.25	81.11	80.00	66.60	66.20
Hunyuan-T1-20250403	**82.61**	**73.47**	**90.20**	80.00	68.42	78.94	79.20
Doubao-1.5-Thinking-Pro-250415	70.43	69.39	**90.20**	83.33	**92.63**	81.20	80.80
DeepSeek-R1	76.52	**72.45**	86.27	**91.11**	84.21	**82.11**	**81.80**

Table 1: The evaluation results of non-reasoning LLMs and reasoning LLMs. The best and second-best results are highlighted with **Bold** and **Underline** respectively. Macro/Micro Acc. = Macro/Micro Averaged Accuracy.

3.2 Main Results

The evaluation results are shown in Table 1. From the results, we have the following findings. (**1**) Some advanced models show a significant performance gap in this evaluation (e.g. Qwen3), indicating the Chinese character-word questions may unexpectedly beat top models. (**2**) Reasoning models reach a higher performance upper bound. The self-reflection in chain-of-thoughts could improve their reliability. (**3**) Among different tasks, character structure and stroke are relatively more difficult for most LLMs. It reveals a common weakness of LLMs in grasping orthography-related knowledge. (**4**) Each model has underperforming tasks, where the error rates reach close to 30% or more. These results highlight that Chinese character-word tasks remain a challenge for advanced LLMs.

3.3 Fine-tuning Experiments

To obtain more insights, we fine-tune an open-weights model (Qwen2.5-72B-Instruct) on the in-domain annotated training data. According to the results, the in-domain training only brings marginal improvement in underperforming tasks (character structure 24.35->25.22, character stroke 9.18->20.41, refer to Appendix D for details). It suggests that there is a lack of some foundational knowledge and the model performance is difficult to improve solely through post-training, which should be taken into account when designing training strategies.

4 Related Work

A growing number of Chinese-language evaluation datasets for LLMs have emerged within the research community. Most of them are derived from traditional NLP tasks (Xu et al., 2020), or standardized exams (Huang et al., 2023; Li et al., 2024). Recent benchmarks (Xu et al., 2023; Contributors, 2023) pay more attention to open-ended and real-world tasks. Nevertheless, there are few resources covering the basic Chinese character-word questions investigated in this paper. The most related work to this paper is AlignBench (Liu et al., 2024), which collects 28 Chinese character questions in its Chinese understanding subset. In comparison, ZiCiEval provides a more detailed character-word task taxonomy and larger amounts of instances.

5 Conclusion

In this paper, we introduce ZiciEval, a dataset for evaluating the capabilities of LLMs on basic Chinese character-word questions. We propose a tool-enhanced LLM-As-Judge framework for reliable automatic evaluation. Empirical results show that advanced models can still be stumped on these tasks, especially for orthography-related tasks. Future work should explore better pre-training and post-training data strategies to tackle the challenges. ZiciEval can serve as a strong resource for these research scenarios.

Limitations

This work focus on the evaluation of basic Chinese character-word questions. The main limitations include: (1) We mainly collect direct questions about Chinese characters and words. Some domains which indirectly involve Chinese character knowledge, such as riddles and rhyme creation, are not included in this work. (2) We pay more attention to the Chinese usage in mainland China, which may have subtle differences from the Chinese standards in other regions. (3) Although the automatic evaluation framework has high agreement with human judgment, it sometimes fails to detect subtle errors, as discussed in Section 2.3.(4) Due to the lack of available access to retrieval-augmented generation (RAG) API and the complexities of building a reproducible RAG system, we do not systematically evaluate the LLMs under the RAG settings. While qualitative tests on the web interfaces of some proprietary models show the RAG results are also prone to errors.

Ethics Statement

The dataset is constructed based on real-world questions from human users. These questions are either collected from anonymous logs that users authorized us to utilize, or directly contributed by active feedback from volunteer users. The data are carefully checked to ensure that there is no personal information or other sensitive content included. During the annotation process, we make use of the responses from LLMs. To avoid intellectual property and ethical legal disputes, we only use the models which have an open license to use their outputs. All annotators are native Chinese people and they receive corresponding compensation and rewards. They are also informed that their annotation results will be used for language model optimization purposes. We release the evaluation dataset for evaluation and research purposes. It is not recommended to use the dataset in model training.

References

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2024. A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol.*, 15(3).

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

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025a. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025b. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Tairan Fu, Raquel Ferrando, Javier Conde, Carlos Arriaga, and Pedro Reviriego. 2024. Why do large language models (llms) struggle to count letters? *Preprint*, arXiv:2412.18626.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, jiayi lei, Yao Fu, Maosong Sun, and Junxian He. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. In *Advances in Neural Information Processing Systems*, volume 36, pages 62991–63010. Curran Associates, Inc.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

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

Xiao Liu, Xuanyu Lei, Shengyuan Wang, Yue Huang, Andrew Feng, Bosi Wen, Jiale Cheng, Pei Ke, Yifan Xu, Weng Lam Tam, Xiaohan Zhang, Lichao Sun, Xiaotao Gu, Hongning Wang, Jing Zhang, Minlie Huang, Yuxiao Dong, and Jie Tang. 2024. AlignBench: Benchmarking Chinese alignment of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11621–11640, Bangkok, Thailand. Association for Computational Linguistics.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, and 13 others. 2020. CLUE: A Chinese language understanding evaluation benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4762–4772, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Liang Xu, Anqi Li, Lei Zhu, Hang Xue, Changtai Zhu, Kangkang Zhao, Haonan He, Xuanwei Zhang, Qiyue Kang, and Zhenzhong Lan. 2023. Superclue: A comprehensive chinese large language model benchmark. *Preprint*, arXiv:2307.15020.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2025b. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

A Pilot Study Details

To identify the tasks that current language models struggle to solve, we conduct a pilot study before the formal dataset construction. We sample 140 questions from anonymous logs, categorize them and use them to test three models. The results are shown in Table 2. All the models do well in the synonym-antonym task and word explanation task. Therefore, we pay attention to other tasks in the evaluation construction.

Task	GPT-4o	ERNIE-4	Doubao-Pro
Character Structure	22.7%	70.8%	79.2%
Character Stroke	5.6%	53.3%	66.7%
Trad.-Simp. Conversion	87.5%	75%	81.3%
Pronunciation	66.7%	67.1%	80.6%
Constrained Word Search	50%	45%	65%
Synonym-Antonym	95%	90%	90%
Word Explanation	95%	100%	95%

Table 2: Pilot study results. We report the manually checked accuracy here.

B Prompt Engineering for Evaluation

B.1 Prompt Designs

For interested readers, here we translate some key parts of the LLM-As-Judge prompt templates into English. At the beginning of the prompt, we list some key rules of the judgment, as shown in Figure 5. In the first round, the judge model is asked to generate Python codes of tool use rather than give direct judgment. The key instruction is shown in Figure 6. Given the tool results, the judge model is asked to give the final judgment. The prompt template is shown in Figure 7.

B.2 Toolkit Design

For toolkit construction, we collect resources from several open projects about Chinese characters and words, including cnchar[2], zhHanSequence[3], yibai-ids[4], chinese-dictionary[5], and pinyin-data[6]. These resources are licensed under MIT license or BSD-2 license. We merge and reorganize these resources into a character database and a word database. After that, we develop several tool functions based on the databases, as well as some string pattern verification functions. The descriptions are shown in Figure 8.

C Evaluated Model Details

In the experiments, we evaluate the following open-weights models: DeepSeek-V3 (DeepSeek-AI et al., 2025b), DeepSeek-V3-0324, DeepSeek-R1 (DeepSeek-AI et al., 2025a), Qwen2.5-72B-Instruct (Yang et al., 2025b), Qwen3-235B (Yang et al., 2025a), and the following proprietary models: GPT-4o (OpenAI et al., 2024), OpenAI O3[7], Qwen-Max-0125[8], Doubao-1.5-Pro[9], Doubao-1.5-Thinking-Pro[10], GLM-4-Plus[11], ERNIE-4-Turbo[12], Hunyuan-TurboS[13], Hunyuan-T1 [14]. These are recent mainstream models used by Chinese users.

For open-weights models, we use VLLM (Kwon et al., 2023) for inference, and set the generation

[2] https://github.com/theajack/cnchar

[3] https://github.com/DongSky/zhHanSequence

[4] https://github.com/yi-bai/ids

[5] https://github.com/mapull/chinese-dictionary

[6] https://github.com/mozillazg/pinyin-data

[7] https://openai.com/index/o3-o4-mini-system-card/

[8] https://qwenlm.github.io/zh/blog/qwen2.5-max/

[9] https://seed.bytedance.com/zh/special/doubao_1_5_pro

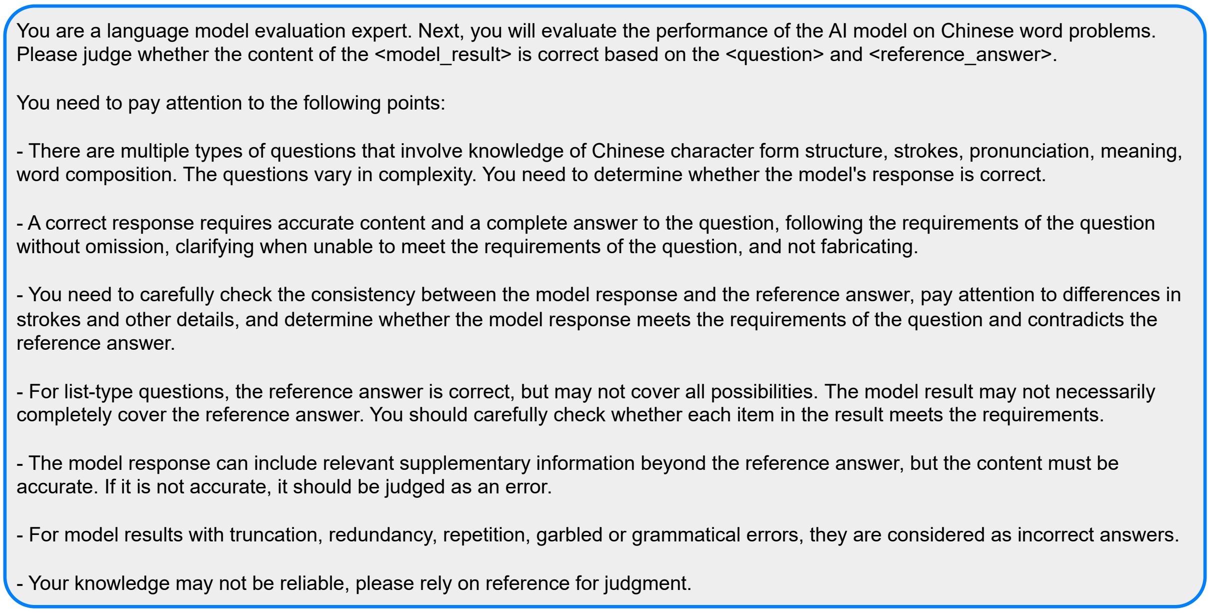
[10] https://github.com/bytedance-seed/seed-thinking-v1.5

[11] https://open.bigmodel.cn/dev/howuse/glm-4

[12] https://aistudio.baidu.com/modelsdetail/714

[13] https://github.com/Tencent/Hunyuan-TurboS

[14] https://github.com/Tencent/llm.hunyuan.T1



You are a language model evaluation expert. Next, you will evaluate the performance of the AI model on Chinese word problems. Please judge whether the content of the <model_result> is correct based on the <question> and <reference_answer>.

You need to pay attention to the following points:

- There are multiple types of questions that involve knowledge of Chinese character form structure, strokes, pronunciation, meaning, word composition. The questions vary in complexity. You need to determine whether the model's response is correct.

- A correct response requires accurate content and a complete answer to the question, following the requirements of the question without omission, clarifying when unable to meet the requirements of the question, and not fabricating.

- You need to carefully check the consistency between the model response and the reference answer, pay attention to differences in strokes and other details, and determine whether the model response meets the requirements of the question and contradicts the reference answer.

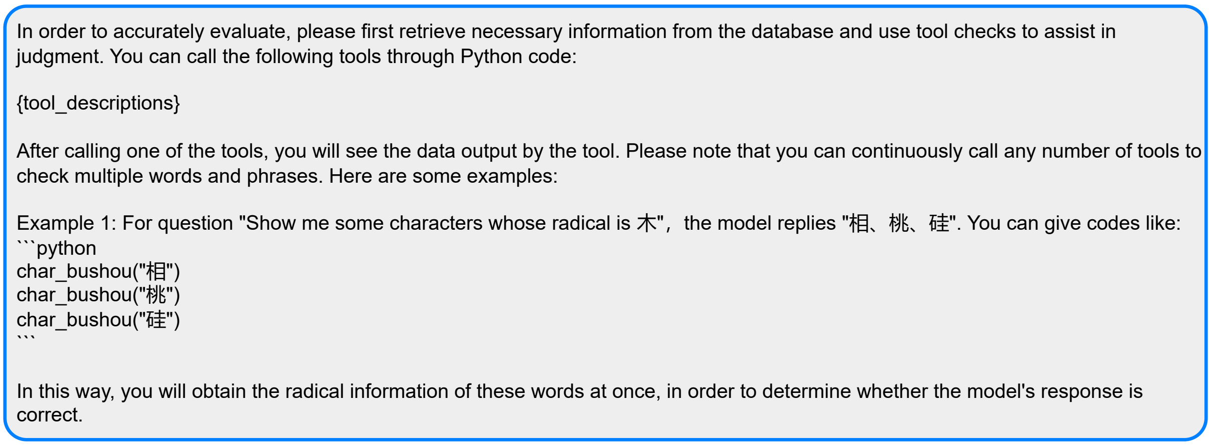
- For list-type questions, the reference answer is correct, but may not cover all possibilities. The model result may not necessarily completely cover the reference answer. You should carefully check whether each item in the result meets the requirements.

- The model response can include relevant supplementary information beyond the reference answer, but the content must be accurate. If it is not accurate, it should be judged as an error.

- For model results with truncation, redundancy, repetition, garbled or grammatical errors, they are considered as incorrect answers.

- Your knowledge may not be reliable, please rely on reference for judgment.

Figure 5: The key instruction about the judgment rules in the prompt template. The original prompt is in Chinese.



In order to accurately evaluate, please first retrieve necessary information from the database and use tool checks to assist in judgment. You can call the following tools through Python code:

{tool_descriptions}

After calling one of the tools, you will see the data output by the tool. Please note that you can continuously call any number of tools to check multiple words and phrases. Here are some examples:

Example 1: For question "Show me some characters whose radical is 木"，the model replies "相、桃、硅". You can give codes like:

```python
char_bushou("相")
char_bushou("桃")
char_bushou("硅")
```

In this way, you will obtain the radical information of these words at once, in order to determine whether the model's response is correct.

Figure 6: The key instruction for generating tool-use codes. The original prompt is in Chinese.

Model	Character Structure	Character Stroke	Trad.-Simp. Conversion	Char.-Word Pronunciation	Constrained Word Search	Macro Acc.	Micro Acc.
Qwen2.5-72B-Instruct	24.35	9.18	70.59	46.67	56.84	41.53	41.00
+ Annotated Data	25.22	20.41	60.78	46.67	43.16	39.25	38.80
+ Synthetic Data	36.52	21.43	67.65	52.22	41.05	43.77	43.60
+ Annotated Data and Synthetic Data	40.00	28.57	60.78	47.78	43.16	44.06	44.00

Table 3: Fine-tuning experiments results on Qwen2.5-72B-Instruct.

```
The following is the current question:

<Question>
{question}
</Question>

Here is the reference answer:

<Reference Answer>
{ref_answer}
</Reference Answer>

Here is the model result:

<Model Resultt>
{model_result}
</Model Result>

The following are auxiliary materials found through tools, which could help you to determine whether the model result meets the
requirements of the question:

<Auxiliary Information>
{tool_result}
</Auxiliary Information>

Next, please provide the judgment result and only output "correct" or "incorrect"
```

Figure 7: The key instruction for generating final judgment. The original prompt is in Chinese.

```
char_structure(char)
    Provide structural information and structure descriptions (Ideographic Description Sequence)
    of a character. For example, the character 相 has a left-right structure, and the components
    composition can be represented as ⿰木目.

char_bushou(char)
    Describe the bushou (character radical) of a character. For example, the radical of 张 is 弓.

char_strokes(char)
    Provide the stroke count and stroke order information of Chinese characters, for example, the
    total number of strokes of 大 is 3. Its stroke order is: horizontal (横), throw (撇), press(捺).

char_pinyin_and_explain(char)
    Provide the pinyin (pronunciation notation) and corresponding meaining explaination of a
    character.

word_pinyin_and_explain(word)
    Provide the pinyin (pronunciation notation) and corresponding meaining explaination of a word.

word_pattern(word)
    Provide the composition pattern of a word. For example, 开开心心 belongs to AABB-pattern words。

startswith(word, prefix)
    Verify whether a word starts with the prefix。

endswith(word, suffix)
    Verify whether a word ends with the suffix。

contains(word, infix)
    Verify whether a word contains the prefix。
```

Figure 8: The description of the tools, which is also provided to the judge model. Note that the original description is in Chinese.

parameters recommended in official repositories, reporting the average results from three runs. Note that Qwen3-235B is a “hybrid thinking” model. It can be controlled to generate the thinking process or directly answer. Therefore we evaluate it both in thinking mode and non-thinking mode, by switching a /no_think instruction in the prompt. For proprietary models, we use the default generation settings of their official APIs, and only report the result from a single run.

D Fine-Tuning Experiments

To obtain more insights about the tasks, we fine-tune an open-weights model (Qwen2.5-72B-Instruct) on the in-domain annotated 3.5k training data. Also, to verify the effectiveness of large-scale synthetic data, we create a template-based synthetic dataset with 500k instances, which are converted from the character databases of evaluation toolkit. The training instances are packed into sequences with a length of 4096. We set the training batch size to 4 when only using annotated data, 64 when synthetic data are used. The results are shown in Table 3 According to the results, the in-domain training only bring marginal improvement on weak tasks. By adding large-scale synthetic data, we observe larger performance improvement, yet the accuracy is still relatively low. It proves that there is a lack of some foundational knowledge and the model performance cannot be simply improved through post-training.