Large Language Models Can Not Perform Well in Understanding and Manipulating Natural Language at Both Character and Word Levels?

Anonymous EMNLP submission

Abstract

Despite their promising performance across various tasks, recent studies reveal that Large language models (LLMs) still exhibit significant deficiencies in handling several word-level and character-level tasks, e.g., word unscrambling and sentence editing, indicating urgent needs for substantial improvements in basic language understanding and manipulation. To address these challenges, it is crucial to develop large-scale benchmarks that can comprehensively assess the performance of LLMs in basic language tasks. In this paper, we introduce a bilingual benchmark, CWUM, to investigate the capabilities and limitations of LLMs in understanding and manipulating natural language at both character and word levels. CWUM consists of 15 simple text editing tasks, e.g., letter counting, word reversing, Chinese character inserting, etc. We conduct extensive experiments on eight advanced LLMs, including base models and instruction-tuned (chat) variants. The experimental results highlight significant failures of existing LLMs on CWUM tasks that humans can solve perfectly with 100% accuracy. On English tasks of CWUM, the average accuracy of GPT-4, LLaMA-3-70B, and Qwen-72B is 66.64%, 39.32%, and 33.16%, respectively, which lags far behind human performance. Instruction-tuning the base model does not lead to a distinct performance improvement, as the average accuracy of LLaMA-3-70B-Instruct on English tasks is only 1.44% higher than that of the base LLaMA-3-70B. Ultimately, we show that supervised fine-tuning (SFT) can enhance model performance on CWUM without compromising its ability to generalize across general tasks.

1 Introduction

Recently, large language models (LLMs) have demonstrated significant capabilities across a wide range of applications, including general natural language processing (NLP) and domain-specific tasks (Bommasani et al., 2021; Wei et al., 2022a; Zhao et al., 2023). Reports indicate that LLMs have matched or even surpassed human performance in several areas. For example, LLMs outperform humans in specific language translation tasks, standardized reading comprehension tests, and logical reasoning assessments. Additionally, LLMs excel at solving complex algebra and calculus problems in standardized mathematics tests and competitions.

Despite the promising performance across various tasks, recent studies propose that LLMs still exhibit significant deficiencies in handling several word-level and character-level tasks, e.g., word unscrambling and sentence editing (Srivastava et al., 2022). In simple tasks such as writing a sentence containing a specific word or choosing which of two words is longer, model performance is worse than that of elementary school students (Efrat et al., 2023). This disparity indicates that while LLMs have made breakthroughs in higher-level language understanding and generation, substantial improvements are still needed for basic language understanding and manipulation.

To address these challenges, it is crucial to develop large-scale benchmarks that can comprehensively assess the performance of LLMs in basic language tasks. A bilingual benchmark is particularly important as it allows for the evaluation of LLMs across different languages, revealing language-specific deficiencies and providing a more comprehensive understanding of their capabilities and limitations. To this end, we propose a bilingual benchmark CWUM, to evaluate the capacities and limitations of LLMs in understanding natural language at both character and word levels. Specifically, CWUM comprises 15 tasks focusing on text edition, including identification, insertion, reversal, and counting. In addition to evaluating model performance on each task of CWUM, we investigate how model performance varies with increasing model size and shots. We also examine the impact

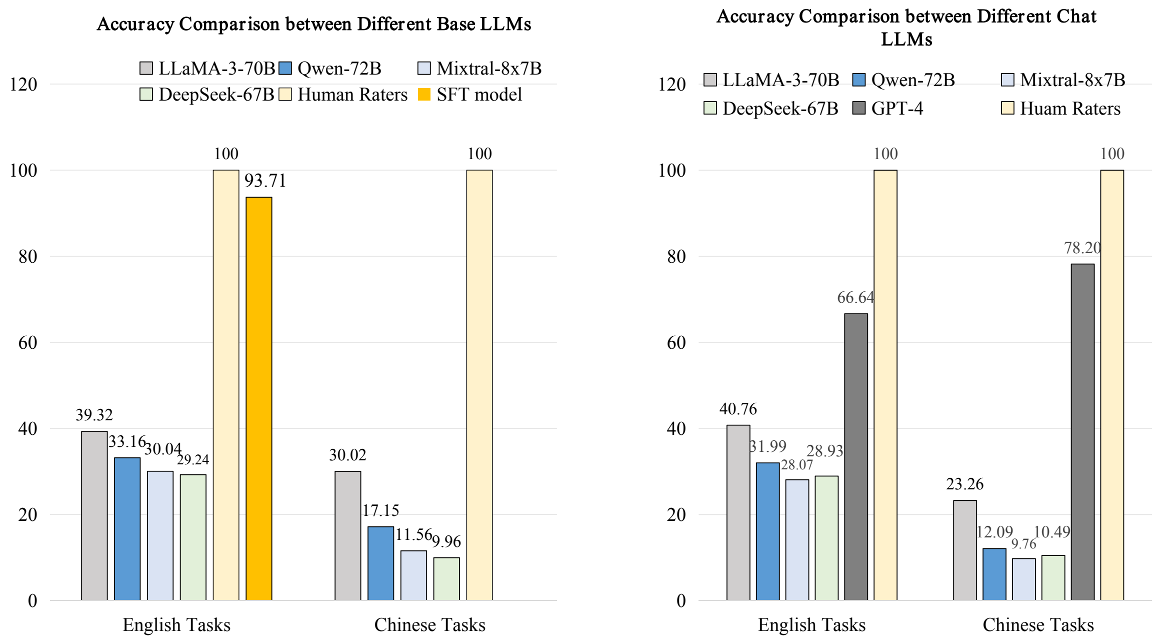

Figure 1: This figure shows the accuracy comparison between base LLMs (left), and the accuracy comparison between chat LLMs (right), on CWUM. GPT-4 performance for each task is computed on 100 uniformly distributed test examples owing to its cost and usage limit. Other model performance is calculated on the full test examples.

of instruction tuning on model performance. To boost the confidence of model predictions, we employ a few-shot Chain-of-Thought (CoT) prompt (Wei et al., 2022b), which encourages the model to follow demonstrations that provide intermediate steps, such as identifying the letters constituting the input word or the words constituting the input sentence, before generating the final output.

We evaluate the performance of eight advanced LLMs including both base models and instruction-tuned (chat) variants, on the CWUM benchmark. These models include LLaMA-2 and LLaMA-3 (Touvron et al., 2023), Qwen (Bai et al., 2023), Mistral (Jiang et al., 2023), Baichuan2 (Baichuan, 2023), ChatGLM3 (Zeng et al., 2023), Yi (AI et al., 2024), DeepSeek (DeepSeek-AI, 2024), and GPT-4 (OpenAI, 2023). Overall, we observe the following phenomena by comparing the testing accuracy of different models. (1) The tasks in the CWUM benchmark pose a huge challenge to all evaluated models, resulting in a significant performance gap compared to human performance. As illustrated in Figure 1, human performance on the CWUM benchmark is perfect (measured at 100%). Even the best-performing model, GPT-4, achieves only 66.64% accuracy on English tasks and 78.20% on Chinese tasks, respectively. The performance of representative open-source LLaMA-3-70B on the English and Chinese tasks is 39.92% and 30.02%, respectively, significantly lower than human performance. (2) Instruction tuning does not lead to substantial performance improvement, e.g., the average accuracy of LLaMA-3-70B-Instruct and the base LaMA-3-70B on English tasks is 40.76% and 39.92%, respectively. (3) While model performance improves with increasing size, it remains unsatisfactory compared to human performance. Additionally, by analyzing model predictions, we attribute the failures of LLMs on CWUM to the following reasons.

- We suggest that the factors contributing to the failure of LLMs in word-level tasks include: 1) limited capacity to understand and process absolute positions, 2) proficiency in handling continuous linguistic information but lacking specialized mechanisms for dealing with discrete data, such as precise numbers and positions, 3) misinterpretation of complex structures or special symbols in a sentence, such as punctuation marks, abbreviations, and numbers.

- The primary factor contributing to the failure of LLMs in character-level tasks is the widespread utilization of the Byte-Pair Encoding (BPE) algorithm to construct vocabulary, which results in the model having never seen individual characters but rather opaque word fragments. These fragments change chaotically based on specific words or the surrounding context, causing the model to struggle with tasks that require precise manipulation of individual characters within words. This is consistent with the study in GPT-3 Creative Fiction [1], which proposes that the BPE encodings result in models bad at phonetic/character-level tasks.

Finally, we conduct experiments to explore whether supervised fine-tuning (SFT) can improve model performance on CWUM while maintaining its generalization ability on general tasks. We collect 160,000 training examples for eight English tasks from CWUM and combine them with 520,000 general-purpose instruction-response pairs to construct the final SFT dataset. Fine-tuning Qwen-7B on this mixed dataset results in an 86% average accuracy improvement on all 10 English CWUM tasks. Additionally, on unseen general tasks including MMLU (Hendrycks et al., 2021a), HellaSwag (Zellers et al., 2019), WinoRrande (Sakaguchi et al., 2020), and Arc (Clark et al., 2018), the performance of the model fine-tuned on mixed data is comparable to that of the model fine-tuned on instruction data alone.

2 Related Works

Large language models (LLMs) are increasingly significant in both research and daily life, making the evaluation of their capabilities a crucial issue. Recently, substantial efforts have been made to develop benchmarks that assess LLMs from various perspectives. LLMs are originally designed to improve performance in natural language processing (NLP) tasks, including understanding, generation, reasoning, multilingual capabilities, factual knowledge, etc. Most evaluation research focuses on specific NLP tasks using datasets such as CommonsenseQA for common sense knowledge (Talmor et al., 2019), SQuAD for reading comprehension (Rajpurkar et al., 2016), and MATH and GSM8K for mathematical reasoning (Hendrycks et al., 2021b; Cobbe et al., 2021). Beyond single-task datasets, large-scale benchmarks like MMLU (Hendrycks et al., 2021a), GLUE (Wang et al., 2019), and C-Eval (Huang et al., 2023) cover a wide range of tasks to provide a comprehensive evaluation. Furthermore, as LLMs are increasingly integrated into everyday activities, studies have begun to examine their robustness (Wang et al., 2021; Nie et al., 2020), ethical considerations and biases (Cao et al., 2023), and trustworthiness (Wang et al., 2023). These evaluations are vital to understanding the broader implications of LLMs and ensuring their reliable and ethical deployment.

In addition to the work mentioned above, several studies focusing on the limitations of LLMs are drawing attention from the research community. Berglund et al. (2023) investigates the Reversal Curse of LLMs, i.e., LLMs trained on "A is B" failing to learn "B is A". Pezeshkpour and Hruschka (2023) aims to study the order sensitivity of LLMs against options of multiple-choice questions and two approaches are presented to calibrate LLMs' predictions including majority vote and multiple evidence calibration (MEC). To explore the limitations and predict the future behavior of LLMs, the Beyond the Imitation Game benchmark (BIG-bench) (Srivastava et al., 2022) compiles 204 tasks believed to exceed current models' capabilities. Similar to our work, LMentry (Efrat et al., 2023) highlights substantial failures of LLMs on 25 tasks that are trivial for humans, e.g., writing a sentence containing a specific word or choosing which of two words is longer. Unlike LMentry, we introduce CWUM, a bilingual benchmark consisting of 15 character and word editing tasks, to evaluate the capabilities and limitations of existing LLMs in understanding and manipulating natural language at both character and word levels. This comprehensive approach aims to identify specific areas where LLMs fall short and provide insights for future model improvements.

3 CWUM

CWUM is a bilingual benchmark designed to assess the basic natural language comprehension abilities of current LLMs. It consists of 15 straightforward character-editing and word-editing tasks such as counting characters, reversing words, and identifying specific characters, which are tasks that an elementary student is generally expected to perform perfectly. Each task consists of a training set and a test set. The simplicity of these tasks highlights the basic language understanding and manipulation capabilities of LLMs, providing a clear measure of their proficiency in handling fundamental linguistic operations.

[1] https://gwern.net/gpt-3#bpes

Language	Task	Samples	Description
Engilish	Count Letters in Word	1000	Count the number of letters comprising the input word
	Count Letters in Sentence	5000	Count the number of letters comprising the word at the specified position in the input sentence
	Count Words in Sentence	1000	Count the number of words comprising the input sentence
	Insert Letters in Word	5000	Insert letters at the specified position in the input word
	Insert Letters in Sentence	5000	Insert letters at the specified position of the word at the specified position in the input sentence
	Insert Words in Sentence	5000	Insert words at the specified position in the input sentence
	Identify Letter in Word	5000	Identify the letter at the specified position in the input word
	Identify Letter in Sentence	5000	Identify the letter at the specified position of the word at the specified position in the input sentence
	Reverse Word	1000	Arrange all the characters of the input word in reverse order
	Reverse Word in Sentence	5000	Arrange all the characters of the word at the specified position in the input sentence in reverse order
Chinese	Count Chinese Characters in Sentence	1000	Count the number of Chinese characters comprising the input sentence
	Reverse Chinese Sentence	1000	Arrange all characters comprising the input sentence in reverse order
	Insert Blank after Each Chinese Characters	1000	Insert blank after each Chinese character in the input sentence
	Insert Chinese Characters in Sentence	5000	Insert Chinese characters at the specified position in the input sentence
	Identify Chinese Character in Sentence	5000	Identify the Chinese character at the specified position in the input sentence

Table 1: An introduction to each task of the CWUM benchmark.

3.1 Task Creation

Our tasks primarily revolve around four types of text-editing operations, including counting, insertion, identification, and reversal. Each task is subject to the following criteria: (1) the answer is readily obtainable, ensuring clear and straightforward solutions; (2) no external tools are necessary, making the tasks accessible and easily implementable; and (3) automatic evaluation is feasible, allowing for efficient and objective assessment. Following these guidelines, we have curated a total of 15 tasks, as depicted in Table 1. Specifically, for the English language, we have designed four single character-editing tasks, four complex character-editing tasks, and two word-editing tasks. For the Chinese language, we have devised five single character-editing tasks.

3.2 Data Construction

In this section, we provide a detailed description of constructing the data of the CWUM benchmark. Each task in CWUM is formulated as an open-ended question, with the input typically consisting of an instruction and a text input. The instruction outlines the task guidelines, providing a foundation for the model's operations. The text input specifies the object for editing operations, which could be an English word (word), an English sentence (sentence), or a Chinese sentence. The answer for each question includes a golden answer and an accompanying natural language rationale. Figure 2 presents the examples for four representative tasks, with additional examples illustrated in Appendix Figures 5 and 6.

Source of input text: For English tasks, we construct a sentence corpus consisting of 100,000 English sentences derived from CommonCrawl dumps from 2020, the C4 Dataset, and Wikipedia dumps (June to August 2022), and a word corpus consisting of 22,000 common words available in the Natural Language Toolkit (NLTK) (Hardeniya et al., 2016) library. For each corpus, 1,000 samples are used as text inputs for the test set, and the remaining sentences are used for the training set. For Chinese tasks, 100,000 Chinese sentences derived from the C4 Dataset and Wikipedia dumps (June to August 2022) are divided into 1,000 for the text inputs of the test data and 99,000 for the training data.

Design the questions: For each task, we meticulously craft 8-10 instructions covering both simple and complex scenarios. The input for a question consists of a randomly sampled instruction. For tasks requiring a specific position in the instruc-

Count Letters in Sentence
Instruction: Examine the 4th word in the following sentence and provide the number of letters it contains. Code prohibited.
Input: In July 2023, they planned to embark on their journey across Europe.
Golden Answer: 7
Rationale: The words contained in the sentence are: ['In', 'July', 'they', 'planned', 'to', 'embark', 'on', 'their', 'journey', 'across', 'Europe']. The 4th word of the given sentence is 'planned'. The letters contained in 'planned' are: ['p', 'l', 'a', 'n', 'n', 'e', 'd']. The total number of letters is 7. Therefore, the answer is 7

Insert Words in Sentence
Instruction: Examine the following sentence and demonstrate the outcome when 'test sample' is added immediately after the last word. Code prohibited.
Input: In July 2023, they planned to embark on their journey across Europe.
Golden Answer: In July 2023, they planned to embark on their journey across Europe test sample.
Rationale: The words contained in the given sentence are: ['In', 'July', 'they', 'planned', 'to', 'embark', 'on', 'their', 'journey', 'across', 'Europe']. The last word of the given sentence is 'Europe'. Therefore, the answer is: In July 2023, they planned to embark on their journey across Europe test sample.

Identify Chinese Character in Sentence
Instruction: 从给定的句子中识别第十个汉字并提供结果。代码被禁止使用。
Input: 2024年9月，他们计划去欧洲旅行，感受这里的独特魅力和风土人情。
Golden Answer: 旅
Rationale: 给定句子包含的汉字列表为：['年', '月', '他', '们', '计', '划', '去', '欧', '洲', '旅', '行', '感', '受', '这', '里', '的', '独', '特', '魅', '力', '和', '风', '土', '人', '情']。其中第十个汉字是"旅"。因此答案是：旅

Reverse Chinese Sentence
Instruction: 请把下列句子的字符顺序颠倒过来并提供结果。代码被禁止使用。
Input: 2024年9月，他们计划去欧洲旅行，感受这里的独特魅力和风土人情。
Golden Answer: 。情人土风和力魅特独的里这受感，行旅洲欧去划计们他，月9年4202
Rationale: 给定句子包含的字符列表为：['2', '0', '2', '4', '年', '9', '月', '，', '他', '们', '计', '划', '去', '欧', '洲', '旅', '行', '，', '感', '受', '这', '里', '的', '独', '特', '魅', '力', '和', '风', '土', '人', '情', '。']。倒着输出该句子包含的字符得到的答案是：。情人土风和力魅特独的里这受感，行旅洲欧去划计们他，月9年4202

Figure 2: Several examples of the benchmark CWUM.

tion, such as the Identify Letter in Word task, we randomly sample a position ranging from 0 to the length of the input text. Each input text is used at five different positions, creating five examples. For insertion tasks, we randomly select combinations of 1 to 10 letters from the lowercase English alphabet ('a' to 'z') or combinations of 1 to 10 words from a set of 20 common words generated by GPT-4 (OpenAI, 2023) as the target for insertion.

Design the answers: For each input, the golden answer is generated using tools and rules, e.g., Python code. For example, the golden answer for the Reverse Word task is simply the reversed word. The rationale provides a detailed breakdown of the input text, e.g., the list of words constituting the input sentence or the list of letters constituting the input word.

In summary, CWUM consists of 51,000 examples designed to evaluate a model's ability to understand natural language at both character and word levels. Each example includes an instruction, an input text, and a golden answer accompanied by the rationale. This diverse and representative benchmark allows for a comprehensive assessment of model performance in various text manipulation tasks.

4 Experiment

In this section, we perform comprehensive evaluation experiments on the proposed benchmark CWUM to achieve the following objectives: evaluate the capability of representative LLMs encompassing both base and chat variants, explore how model performance varies with increasing sizes, increasing shots, and different prompts, and investigate whether supervised fine-tuning (SFT) can improve model performance on CWUM.

4.1 Baselines

We test CWUM on eight models from two families including open-source LLMs and closed-source LLMs. When evaluated on CWUM, all models are prohibited from using codes.

Open-source LLMs include both base and chat ones. For base LLMs, we use Qwen (7B and 72B) (Bai et al., 2023), LLaMA-2 (7B and 70B) and LLaMA-3 (8B and 70B) (Touvron et al., 2023), DeepSeek-67B (DeepSeek-AI, 2024), Mistral-7B (Jiang et al., 2023), Yi (6B and 34B) (AI et al., 2024), Baichuan2-7B (Baichuan, 2023), ChatGLM3-6B (Zeng et al., 2023), and Mixtral-8x7B (Jiang et al., 2024). For chat LLMs, we utilize Qwen-72B-Chat, LLaMA-2-70B-Chat, LLaMA-3-70B-Instruct, Yi-34B-Chat, DeepSeek-67B-Chat, and Mixtral-8x7B-Instruct. For all open-source models, we use the Hugging Face (Wolf et al., 2020) implementation and greedy decoding to generate deterministic answers.

Closed-source LLMs include representative GPT-4[2] (OpenAI, 2023). We set the temperature to 0.2 for generating quality responses.

4.2 Evaluation Metrics

We conduct an automatic evaluation on the CWUM benchmark using the exact string match as the evaluation metric. In addition, a team of human raters is hired to establish a human baseline. Three human annotators, all sixth-grade students, are tasked with generating answers following the instructions provided for each sample. Detailed guidelines are introduced to ensure consistency and clarity before the evaluation process begins. Due to cost considerations, we randomly select a subset of 100 samples from each task. Notably, all annotators achieve a 0% failure rate across all tasks of the CWUM benchmark when evaluated using the automatic evaluation metric. This demonstrates the high proficiency of humans in successfully solving tasks within the CWUM benchmark.

4.3 Overview of Model Performance and Human Rater Performance on CWUM

Although scaling up model sizes leads to noticeable performance enhancements, it remains low in absolute terms compared with human rater performance. Table 2 and Figure 1 display the average accuracy of automatic evaluation results across different LLMs. Notably, on both English and Chinese tasks, average model performance improves with model size (refer to Tables 3 and 6 for a more granular examination of how individual task contributes to the overall performance). Despite these advancements, the top-performing model, GPT-4, achieves an average accuracy of only 66.64% on English tasks and 78.20% on Chinese tasks, falling significantly short of the estimated 100% accuracy of human raters. Instruction-tuning the model does not yield significant performance gains. As illustrated in Figure 1, the average accuracy of Qwen-72B-Chat is 31.99%, slightly lower than that of the base Qwen-72B (33.16%), on English tasks.

Model	English Tasks	Chinese Tasks
LLaMA-2-7B	7.70	-
LLaMA-3-8B	25.13	8.32
Qwen-7B	16.87	4.24
Mistral-7B	12.15	3.17
Baichuan2-7B	12.47	2.30
ChatGLM3-6B	10.54	2.03
Yi-6B	13.87	3.22
LLaMA-2-70B	25.76	-
LLaMA-3-70B	39.32	30.02
Qwen-72B	33.16	17.15
Mixtral-8x7B	30.04	11.56
DeepSeek-67B	29.24	9.96
GPT-4	66.64	78.20
Human Performance	100	100

Table 2: Comparison of average model accuracy on all English and all Chinese tasks of CWUM. GPT-4 performance for each task is computed on 100 uniformly distributed test examples owing to its cost and usage limit. Other model performance is calculated on the full test examples.

All open-source LLMs perform worse on Chinese tasks than on English tasks. For example, on all Chinese tasks, DeepSeek-67B and Mixtral-8x7B achieve average accuracies of only 11.56% and 9.96%, respectively, markedly lower than their performance on all English tasks (30.04% and 29.24%, respectively).

The failures in character-level tasks are primarily due to the widespread adoption of Byte-Pair Encoding (BPE), and the shortcomings in word-level tasks are attributed to the models' inadequate capacity to handle absolute positions, discrete data, and special symbols. Detailed analysis of failure cases of the tested LLMs is presented in Appendix B.

[2]We use gpt4-1106-preview.

Task	LLaMA-2 -70B	LLaMA-3 -70B	Qwen-72B	Mixtral -8x7B	DeepSeek -67B	Avg	Yi-34B	GPT-4
Count Words in Sentence	12.80	20.10	12.60	15.90	15.20	15.32	6.80	41.00
Count Letters in Word	69.40	99.70	97.60	95.60	91.90	90.84	74.50	100
Count Letters in Sentence	25.20	42.52	34.06	25.66	30.38	31.56	21.00	61.00
Insert Words in Sentence	10.14	29.14	15.26	8.66	11.60	14.96	9.92	48.00
Insert Letters in Sentence	4.72	18.72	2.74	6.02	5.76	7.59	3.62	46.00
Insert Letters in Word	19.72	32.66	27.6	20.84	18.82	23.93	18.62	72.00
Identify Letter in Word	57.62	58.14	83.98	56.48	57.54	62.75	39.00	100
Identify Letter in Sentence	19.74	36.42	21.30	19.22	21.56	23.65	12.36	56.00
Reverse Word	6.10	26.20	2.30	11.80	5.80	10.44	4.00	73.00
Reverse Word in Sentence	7.34	29.56	10.18	10.16	8.12	13.07	3.82	49.00
Avg	25.76	39.32	33.16	30.04	29.24	-	22.36	66.64
Count Chinese Characters in Sentence	-	23.30	15.80	12.60	11.80	15.86	13.40	54.00
Insert Chinese Characters in Sentence	-	21.78	15.88	10.06	7.88	13.90	7.36	73.00
Insert Blank after Each Chinese Characters	-	45.20	12.60	19.00	8.70	21.38	17.50	90.00
Reverse Chinese Sentence	-	30.10	10.00	1.90	4.96	11.74	0.70	77.00
Identify Chinese Character in Sentence	-	29.72	31.46	14.26	16.44	22.97	15.26	97.00
Avg	-	30.02	17.15	11.56	9.96	-	9.56	78.20

Table 3: Comparison of testing accuracy by advanced LLMs on each task of CWUM.

4.4 Performance Analysis on Individual Task of CWUM

Models exhibit significant performance differences across various tasks. Table 3 provides a detailed accuracy comparison of different LLMs on each task of CWUM. Focusing on the average accuracy across base models with 56B to 72B parameters, models perform best in the single letter-counting task, achieving the highest average accuracy of 90.84%. In contrast, they exhibit the worst performance in the Insert Letters in Sentence task, achieving the lowest average accuracy of 7.59%. All tested open-source models perform poorly on input-reversing tasks including Reverse Word, Reverse Word in Sentence, and Reverse Chinese Sentence, with average accuracies of 10.44%, 13.07%, and 11.84%, respectively. Also, LLMs emerge with the ability to reverse the input word at specific scales. For example, Yi-6B and Yi-34B achieve accuracies of 0.00% and 4.00% on the task of Reverse Word, respectively. More evaluation results for LLMs with sizes ranging from 5B to 7B and for instruction-tuned LLMs ranging from 56B to 72B are presented in Appendix C and Appendix D, respectively.

Designing different CoT prompts or further increasing the number of shots does not result in distinct performance improvements. Specifically, experiments are conducted to analyze model performance with increasing sizes, increasing shots, and different prompts. Two representative English tasks (Count Letters in Word and Reverse Word), and two representative Chinese tasks (Identify Chinese Character in Sentence and Reverse Chinese Sentence) are taken as examples. As shown in Appendix A Table 5, CoT prompting significantly improves model performance on most tasks, with minimal performance improvements across different CoT prompts. Additionally, model performance shows an overall upward trend as the shot count increases from 0 to 10, but further increases in shots do not yield additional gains, as demonstrated in Appendix A Figure 4.

4.5 Improving model performance on CWUM by Supervised Fine-tuning

In this subsection, we investigate the impact of supervised fine-tuning (SFT) on the performance of the base Qwen-7B model across the 10 English tasks of CWUM. Our analysis aims to answer the following questions:

(1) Can fine-tuning on target training data maintain generalization within in-domain (IND) tasks? We tune the base model on four representative tasks covering four text manipulation operations and incrementally increase the SFT training

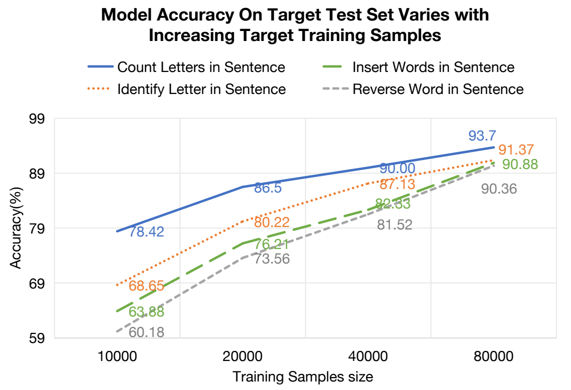

Figure 3: This figure shows the accuracy of the SFT model on the target test set varying with increasing training samples.

data from 10,000 to 80,000 for each task. The results, illustrated in Figure 3, show that each SFT model achieves over 90% testing accuracy when the training data size reaches 80,000. For example, the model accuracy on the Reverse Word in Sentence task improves from 60.18% to 90.36% as the training data increases from 10,000 to 80,000 instances. These results confirm the effectiveness of SFT in enhancing model performance on CWUM. Considering both performance and training costs, the training data size for each task is set to 20,000.

(2) Can multi-task fine-tuning on the part of CWUM tasks generalize to all CWUM tasks? To explore this, we create a mixed training dataset from six tasks (20,000 instances each) covering all types of word-level and character-level tasks (identify letters, insert letters, insert words, reverse words, count words, and count letters). As shown in Table 4, the tuned model achieves an average accuracy of 82.74% on CWUM. Specifically, its average accuracy on six IND tasks is 91.94%, but only 68.94% on four out-of-domain (OOD) tasks from CWUM. This disparity arises because specific task abilities, such as reversing a word within a sentence, do not transfer well to reversing a single word, and inserting letters in a sentence does not transfer to inserting letters in a single word. Extending the training mix to eight CWUM tasks results in an average accuracy of 94.30% on CWUM. However, the performance of the tuned model on four general OOD tasks remains poor, which is significantly worse than that of the base Qwen-7B.

(3) Can comprehensive fine-tuning enhance performance on CWUM tasks while preserving generalization on unseen general tasks? To enhance the model's ability to adhere to general instructions, we merge 520,000 general-purpose instruction-response pairs from Orca (Mukherjee et al., 2023) with the 160,000 training data from step (2) to create the final SFT training dataset. According to Table 4, the fine-tuned model achieves an average accuracy of 93.71% on CWUM, which is 86% higher than the base Qwen-7B. Additionally, its performance on four general OOD tasks averages 61.54%, comparable to the model fine-tuned solely on the 520,000 general-purpose instruction data, which scores 61.61%.

These findings demonstrate that SFT can significantly improve the performance of LLMs on CWUM tasks while maintaining their generalization capability on unseen general tasks.

Training Data	BIBench	General Task			
		MMLU	HellaSwag	WinoGrange	ARC
0 Task	7.70	45.30	77.20	70.20	45.90
6 Tasks	82.74	26.10	28.33	51.95	0.00
8 Tasks	94.30	25.07	28.56	50.67	0.00
8 Tasks + General Data	93.71	50.43	75.65	72.30	47.78
General Data	-	50.93	76.37	70.24	48.90

Table 4: Testing accuracy of the SFT model by fine-tuning Qwen-7B on different training data. 0 Task represents the base Qwen-7B without additional tuning on our constructed training data.

5 Conclusion

In this study, we introduce CWUM, a novel bilingual benchmark designed to evaluate the capabilities and limitations of LLMs in understanding and manipulating natural language at both word and sentence levels. CWUM comprises 15 text-editing tasks, including 10 in English and five in Chinese, which are simple for humans but challenging for current LLMs. Our comprehensive evaluation of eight advanced LLMs, including both base and instruction-tuned (chat) models, reveals significant deficiencies in their performance on these tasks. These findings suggest that while LLMs have made considerable progress, there is still a substantial gap to bridge in terms of achieving human-like proficiency in language understanding and manipulation. Overall, CWUM provides a valuable tool for assessing and guiding the development of future LLMs, emphasizing the need for more sophisticated mechanisms to handle the complexities of natural language at both character and word levels.

Limitations

CWUM primarily focuses on character and word-level editing tasks. Future work should include more complex language understanding tasks, such as paragraph comprehension, text generation, and semantic analysis, to comprehensively evaluate the capabilities and limitations of LLMs. This will provide a more holistic assessment of the language understanding and generation capabilities of LLMs.

Ethics Statement

All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki Declaration and its later amendments or comparable ethical standards. This article does not contain any studies with animals performed by any of the authors. Informed consent was obtained from all individual participants included in the study.

References

01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open foundation models by 01.ai.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Baichuan. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

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

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ B. Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, and et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Yong Cao, Li Zhou, Seolhwa Lee, Laura Cabello, Min Chen, and Daniel Hershcovich. 2023. Assessing cross-cultural alignment between chatgpt and human societies: An empirical study. *arXiv preprint arXiv:2303.17466*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

DeepSeek-AI. 2024. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.

Avia Efrat, Or Honovich, and Omer Levy. 2023. Lmentry: A language model benchmark of elementary language tasks. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 10476–10501. Association for Computational Linguistics.

Nitin Hardeniya, Jacob Perkins, Deepti Chopra, Nisheeth Joshi, and Iti Mathur. 2016. *Natural language processing: python and NLTK*. Packt Publishing Ltd.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and

Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

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 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of GPT-4. *arXiv preprint arXiv:2306.02707*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4885–4901. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Pouya Pezeshkpour and Estevam Hruschka. 2023. Large language models sensitivity to the order of options in multiple-choice questions. *arXiv preprint arXiv:2308.11483*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8732–8740. AAAI Press.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Santilli, Andreas Stuhlmüller, Andrew M. Dai, Andrew La, Andrew K. Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakas, and et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4149–4158. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, Sang T. Truong, Simran Arora, Mantas Mazeika, Dan Hendrycks, Zinan Lin, Yu Cheng, Sanmi Koyejo, Dawn Song, and Bo Li. 2023. Decodingtrust: A comprehensive assessment of trustworthiness in GPT models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021. Adversarial GLUE: A multi-task benchmark for robustness evaluation of language models. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 38–45. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4791–4800. Association for Computational Linguistics.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. GLM-130B: an open bilingual pre-trained model. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

A Analyses of Model Sizes, Shots, and Prompts, on Individual Task

Accuracy gains vary significantly across different tasks with increasing model size. Specifically, the task of Count Letters in Word demonstrates the highest gains in average model accuracy, reaching 50.00% as the model size increases from 6-7B to 56-72B. Conversely, the task of Insert Letters in Sentence exhibits the least gains, with accuracy improvements of merely 6.48%.

LLMs emerge with the ability to reverse the input word at specific scales. Most small-sized LLMs in the range of 6-7B parameters achieve nearly 0.00% accuracy, while larger LLMs with around 34B parameters attain an accuracy of about 4.00%. For example, Yi-6B and Yi-34B achieve accuracies of 0.00% and 4.00%, respectively, on the task of reversing a word.

Model performance tends to stabilize at 10 shots, with further increases in shots not yielding additional gains. We conduct experiments on two representative English tasks (Count Letters in Word and Reverse Word) and two representative Chinese tasks (Identify Chinese Character in Sentence and Reverse Chinese Sentence), to analyze the sensitivity of model performance to increased shots. Using Qwen-72B as the evaluated model, Figure 4 illustrates the model performance with increasing shots. We observe a noticeable performance improvement across three tasks, excluding the Reverse Word task, as the shot increases from 0 to 3. With the shot increasing from 3 to 10, the model performance shows a slow upward trend.

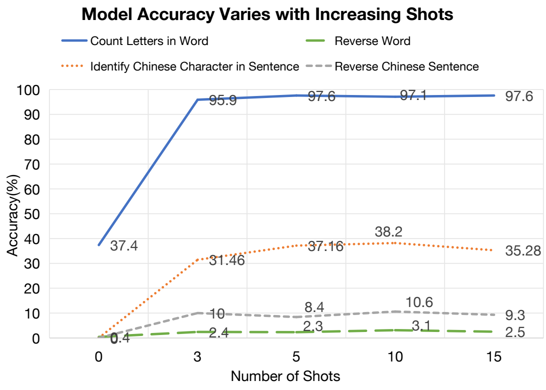

Figure 4: This figure shows the test accuracy of Qwen-72B varying with increasing shots on four representative tasks of CWUM.

When the shot count reaches 10, performance stabilizes. Notably, the Reverse Word task displays the slowest growth trend with increasing shots, with the accuracy consistently below 3%, highlighting the inadequacy of LLMs in handling input-reversing tasks. Subsequently, we conduct experiments on these four tasks to investigate the influence of different prompts on model performance.

Task	CoT1(Ours)	CoT2	Prefix-CoT1	No-CoT
Count Letters in Word	97.60	78.5	100	38.50
Reverse Word	2.43	3.30	2.50	1.00
Identify Chinese Character in Sentence	31.46	30.58	40.34	8.46
Reverse Chinese Sentence	10.00	10.90	9.20	0.00

Table 5: Comparison of testing accuracy of Qwen-72B under different prompts on four representative tasks of CWUM.

CoT prompting can bring huge performance gains on most tasks. As shown in Table 5, CoT2 requires the model to describe the task and explain the answer, while CoT1 (ours) encourages the model first to output the characters or words comprising the input text. Prefix-CoT1 provides the characters or words comprising the queried input text at the end of the prompt. CoT prompting leads to noticeable performance improvements compared to no CoT prompting for tasks excluding Reverse Word. The performance gap led by different CoT prompts for most tasks is minimal, except for the task of Count Letters in Word. Providing the list of letters composing the queried input word enables the model to achieve 100% accuracy in the Count Letters in Word task. However, providing the list of letters comprising the queried input word does not enhance the model performance on the Reverse Word task. Similarly, providing the characters comprising the queried input sentence does not lead to distinct performance improvement in the Reverse Chinese Sentence task.

B Failure Cases

On word-level tasks, we have identified major failure cases of LLMs:

(1) Incorrect word count and positioning: LLMs often underestimate the word count of input sentences and inaccurately position words within specified locations. These issues stem from LLMs' limited capacity to understand and process absolute positions. Additionally, LLMs lack specialized mechanisms for accurately handling discrete data, such as precise numbers and positions.

(2) Inaccurate predictions of word lists: LLMs often produce inaccurate predictions of the word list constituting the input sentence. This inaccuracy arises from the misinterpretation of LLMs to complex structures or special symbols within the sentence, including punctuation marks, abbreviations, numbers, etc.

These failure cases underscore the necessity for enhanced mechanisms within LLMs to better manage absolute positioning and interpret discrete data, thereby ensuring more precise processing of word-level tasks.

On character-level tasks, we have identified major failure cases of LLMs:

(1) Incomplete reversal of common word fragments: Common word fragments within the input word are not correctly reversed. For example, given the input word 'though', the model predicts 'hguoth' instead of the correct reversal 'hguoht', where the fragment 'th' remains unreversed.

(2) Incorrect insertion after common word fragments: The model incorrectly inserts letters after common word fragments. For instance, giving the input word 'though' and a requirement to insert 'abc' after the third character, the model predicts 'thabcough' instead of the correct insertion pattern 'thoabcugh'.

These issues stem from the wide utilization of the Byte-Pair Encoding (BPE) algorithm to construct vocabulary, which results in the model having never seen individual characters but rather opaque word fragments. Consequently, the model struggles with tasks that require precise manipulation of individual characters within words.

C Accuracy Comparison between Different Small Base LLMs

Table 6 provides a detailed overview of the performance of six base LLMs with sizes ranging from 6B to 7B on each task of the CWUM benchmark. For each task, based on the average accuracy of the six tested models, we observe that the model performs best on the Count Letters in Word task and Identify Chinese Character in Sentence task, with an accuracy of 40.84% and 8.70%, respectively. Conversely, they perform worst on the Insert Letters in Sentence and Reverse Chinese Sentence tasks, with an accuracy of 1.11% and 0.12%, respectively. Additionally, based on the average accuracy on 10 English tasks and five Chinese tasks for each model, it can be seen that LLaMA-3-8B performs best on both English and Chinese tasks. LLaMA-2-7B performs worst on English tasks, while ChatGLM3-6B performs worst on Chinese tasks.

D Accuracy Comparison between Different Instruction-tuned LLMs

Table 7 provides a detailed overview of the performance of five instruction-tuned LLMs on each task of the CWUM benchmark. We focus on LLMs with sizes ranging from 56B to 72B. Based on the average accuracy on 10 English tasks and five Chinese tasks for each model, it can be seen that LLaMA3-70B-Instruct performs best on both English and Chinese tasks. LLaMA-2-70B-chat performs worst on English tasks, while Mixtral-8x7B-chat performs worst on Chinese tasks. In particular, LLaMA-2-70B-chat performs worse than Yi-34B-Chat, on English tasks.

Task	LLaMA-2 -7B	LLaMA-3 -8B	Qwen -7B	Mistral -7B	Baichuan2 -7B	ChatGLM3 -6B	Yi-6B	Avg
Count Words in Sentence	5.10	11.10	2.60	3.80	8.30	4.10	9.30	6.33
Count Letters in Word	7.60	94.5	55.00	24.00	40.90	19.00	44.90	40.84
Count Letters in Sentence	15.12	23.94	17.92	13.86	15.00	9.24	12.82	15.41
Insert Words in Sentence	3.88	10.56	6.28	5.50	2.62	1.80	3.60	4.89
Insert Letters in Sentence	0.70	3.90	1.36	0.94	0.22	0.26	0.42	1.11
Insert Letters in Word	8.50	20.94	9.42	10.26	5.94	4.76	8.06	9.70
Identify Letter in Word	13.24	56.34	39.72	38.62	27.88	28.56	28.42	33.25
Identify Letter in Sentence	4.04	15.44	12.18	12.36	9.46	10.16	9.56	10.46
Reverse Word	0.00	8.70	0.30	0.00	0.00	0.00	0.00	1.29
Reverse Word in Sentence	0.50	5.90	2.10	0.42	1.10	1.38	0.48	1.70
Avg	7.70	25.13	16.87	12.15	12.47	10.54	13.87	-
Count Chinese Characters in Sentence	-	11.40	6.30	3.00	0.00	2.70	4.50	4.65
Insert Chinese Characters in Sentence	-	6.90	3.76	1.78	1.48	1.22	2.38	2.92
Insert Blank after Each Chinese Characters	-	10.80	2.00	2.10	1.50	0.20	1.50	3.02
Reverse Chinese Sentence	-	0.70	0.00	0.00	0.00	0.00	0.00	0.12
Identify Chinese Character in Sentence	-	11.78	9.16	8.96	8.52	6.04	7.74	8.70
Avg	-	8.32	4.24	3.17	2.30	2.03	3.22	-

Table 6: Comparison of testing accuracy by small LLMs with sizes ranging from 6B to 7B on each task of CWUM.

Task	LLaMA-2 -70B-Chat	LLaMA-3 -70B-Instruct	Qwen-72B -Chat	Mixtral- 8x7B-Instruct	DeepSeek -67B-Chat	Avg	Yi-34B -Chat
Count Words in Sentence	26.60	12.50	14.00	11.30	13.60	15.60	12.40
Count Letters in Word	68.30	99.50	97.00	76.80	81.10	84.54	57.10
Count Letters in Sentence	16.42	40.04	29.04	29.88	30.36	19.15	26.30
Insert Words in Sentence	1.98	31.66	22.70	8.78	14.64	15.95	5.98
Insert Letters in Sentence	1.64	10.72	4.98	5.04	7.16	5.91	3.26
Insert Letters in Word	14.84	36.94	16.48	21.02	18.62	21.58	17.72
Identify Letter in Word	26.12	97.82	76.82	63.60	54.66	63.80	59.86
Identify Letter in Sentence	12.16	23.36	27.20	20.40	23.92	21.41	17.60
Reverse Word	2.00	32.70	1.70	5.20	2.50	8.82	2.50
Reverse Word in Sentence	1.14	22.34	6.64	12.78	9.78	10.54	4.60
Avg	18.92	40.76	31.99	28.07	28.93	-	24.07
Count Chinese Characters in Sentence	-	19.20	9.80	9.90	13.10	13.00	11.00
Insert Chinese Characters in Sentence	-	25.22	13.60	8.76	11.42	14.75	6.22
Insert Blank after Each Chinese Characters	-	19.90	6.50	13.00	4.00	10.85	1.20
Reverse Chinese Sentence	-	15.30	2.80	0.70	0.80	4.90	0.00
Identify Chinese Character in Sentence	-	36.66	27.76	16.46	23.12	26.00	17.74
Avg	-	23.26	12.09	9.76	10.49	-	7.23

Table 7: Comparison of testing accuracy by advanced instruction-tuned LLMs on each task of CWUM.

Count Words in Sentence
Instruction: Please count the words in the following sentence. Code prohibited.
Input: In July 2023, they planned to embark on their journey across Europe.
Golden Answer: 11
Rationale: The words contained in the sentence are: ['In', 'July', 'they', 'planned', 'to', 'embark', 'on', 'their', 'journey', 'across', 'Europe']. The total number of words is 11. Therefore, the answer is 11

Count Letters in Word
Instruction: Please count the letters in the following word and provide the exact number. Code prohibited.
Input: investor
Golden Answer: 8
Rationale: The letters contained in 'investor' are: ['i', 'n', 'v', 'e', 's', 't', 'o', 'r']. The total number of letters is 8. Therefore, the answer is 8

Insert Letters in Word
Instruction: Please add 'abcdef' before the first letter in the given word. Code prohibited.
Input: investor
Golden Answer: abcdefinvestor
Rationale: The letters contained in 'investor' are: ['i', 'n', 'v', 'e', 's', 't', 'o', 'r']. Inserting 'abcdef' before the first letter get 'abcdefinvestor'. Therefore, the answer is abcdefinvestor

Insert Letters in Sentence
Instruction: Perform the task of inserting 'abcdef' right after the last letter of the 4th word in the following sentence and provide the modified sentence. Code prohibited.
Input: In July 2023, they planned to embark on their journey across Europe.
Golden Answer: In July 2023, they plannedabcdef to embark on their journey across Europe.
Rationale: The words contained in the sentence are: ['In', 'July', 'they', 'planned', 'to', 'embark', 'on', 'their', 'journey', 'across', 'Europe']. The 4th word of the given sentence is 'planned'. The letters contained in 'planned' are: ['p', 'l', 'a', 'n', 'n', 'e', 'd']. Inserting 'abcdef' after the last letter of 'planned' get 'plannedabcdef'. Therefore, the answer is: In July 2023, they plannedabcdef to embark on their journey across Europe.

Identify Letter in Word
Instruction: Employ lexical investigation to identify the 8th letter of the given word. Code prohibited.
Input: investor
Golden Answer: r
Rationale: The letters contained in 'investor' are: ['i', 'n', 'v', 'e', 's', 't', 'o', 'r']. The 8th letter is 'r'. Therefore, the answer is r

Figure 5: Several examples of the benchmark CWUM.

Identify Letter in Sentence
Instruction: Retrieve the first letter of the last word in the given sentence. Code prohibited.
Input: In July 2023, they planned to embark on their journey across Europe.
Golden Answer: E
Rationale: The words contained in the given sentence are: ['In', 'July', 'they', 'planned', 'to', 'embark', 'on', 'their', 'journey', 'across', 'Europe']. The last word of the given sentence is 'Europe'. The letters contained in 'Europe' are: ['E', 'u', 'r', 'o', 'p', 'e']. The first letter of 'Europe' is 'E'. Therefore, the answer is E

Reverse Word
Instruction: Perform the task of reversing the following word and provide the modified word. Code prohibited.
Input: investor
Golden Answer: rotsevni
Rationale: The chars contained in 'investor' are: ['i', 'n', 'v', 'e', 's', 't', 'o', 'r']. Putting the chars in reverse order get 'rotsevni'. Therefore, the answer is rotsevni.

Reverse Word in Sentence
Instruction: Your assignment is to reverse the last word of the following word and furnish the resulting word. Code prohibited.
Input: In July 2023, they planned to embark on their journey across Europe.
Golden Answer: eporuE
Rationale: The words contained in the given sentence are: ['In', 'July', 'they', 'planned', 'to', 'embark', 'on', 'their', 'journey', 'across', 'Europe']. The last word of the given sentence is 'Europe'. The chars contained in 'Europe' are: ['E', 'u', 'r', 'o', 'p', 'e']. Putting the chars in reverse order get 'eporuE'. Therefore, the answer is eporuE.

Count Chinese Characters in Sentence
Instruction: 针对给定的句子，仔细分析它包含的汉字个数。确保你的回答准确无误。代码被禁止使用。
Input: 2024 年 9 月，他们计划去欧洲旅行，感受这里的独特魅力和风土人情。
Golden Answer: 25
Rationale: 给定句子包含的汉字列表为：['年', '月', '他', '们', '计', '划', '去', '欧', '洲', '旅', '行', '感', '受', '这', '里', '的', '独', '特', '魅', '力', '和', '风', '土', '人', '情']。其中总共有 25 个汉字。因此，答案是：25

Insert Blank after Each Chinese Characters
Instruction: 执行在下列句子的每个汉字后插入' '的任务，并提供修改后的句子。代码被禁止使用。
Input: 2024 年 9 月，他们计划去欧洲旅行，感受这里的独特魅力和风土人情。
Golden Answer: 2024 年 9 月 ，他 们 计 划 去 欧 洲 旅 行 ，感 受 这 里 的 独 特 魅 力 和 风 土 人 情 。
Rationale: 给定句子包含的汉字列表为：['年', '月', '他', '们', '计', '划', '去', '欧', '洲', '旅', '行', '感', '受', '这', '里', '的', '独', '特', '魅', '力', '和', '风', '土', '人', '情']。在每个汉字后插入' '后的答案是：2024 年 9 月 ，他 们 计 划 去 欧 洲 旅 行 ，感 受 这 里 的 独 特 魅 力 和 风 土 人 情 。

Insert Chinese Characters in Sentence
Instruction: 你的任务是通过第二个汉字后面插入“测试”来修改下面的句子。在你的回答中展示完整的修改后的句子。代码被禁止使用。
Input: 2024 年 9 月，他们计划去欧洲旅行，感受这里的独特魅力和风土人情。
Golden Answer: 2024 年 9 月测试，他们计划去欧洲旅行，感受这里的独特魅力和风土人情。
Rationale: 给定句子包含的汉字列表为：['年', '月', '他', '们', '计', '划', '去', '欧', '洲', '旅', '行', '感', '受', '这', '里', '的', '独', '特', '魅', '力', '和', '风', '土', '人', '情']。其中第二个汉字是“月”。在第二个汉字后插入“测试”后的答案是：2024 年 9 月测试，他们计划去欧洲旅行，感受这里的独特魅力和风土人情。

Figure 6: Several examples of the benchmark CWUM.