Revisiting Compositional Generalization Capability of Large Language Models Considering Instruction Following Ability

Yusuke Sakai, Hidetaka Kamigaito, Taro Watanabe
Nara Institute of Science and Technology (NAIST), Japan
{sakai.yusuke.sr9, kamigaito.h, taro}@is.naist.jp

Abstract

In generative commonsense reasoning tasks such as CommonGen, generative large language models (LLMs) compose sentences that include all given concepts. However, when focusing on instruction-following capabilities, if a prompt specifies a concept order, LLMs must generate sentences that adhere to the specified order. To address this, we propose Ordered CommonGen, a benchmark designed to evaluate the compositional generalization and instruction-following abilities of LLMs. This benchmark measures ordered coverage to assess whether concepts are generated in the specified order, enabling a simultaneous evaluation of both abilities. We conducted a comprehensive analysis using 36 LLMs and found that, while LLMs generally understand the intent of instructions, biases toward specific concept order patterns often lead to low-diversity outputs or identical results even when the concept order is altered. Moreover, even the most instruction-compliant LLM achieved only about 75% ordered coverage, highlighting the need for improvements in both instruction-following and compositional generalization capabilities.

1 Introduction

With the maturity of instruction-tuning techniques such as supervised fine-tuning (Wei et al., 2022a; Sanh et al., 2022; Ghosal et al., 2023; Wang et al., 2023c) and human-preference tuning (Ziegler et al., 2020; Ouyang et al., 2022; Winata et al., 2024), generative large language models (LLMs) are capable of producing high-quality, fluent responses aligned with user instructions. Leveraging their instruction-following capabilities and advanced text generation abilities, LLMs are applied not only to general downstream tasks such as dialogue systems (Wang et al., 2023a; Yi et al., 2024; Li et al., 2024a; Zhao et al., 2024a) but also to creative domains, including dataset construction through complex prompts tailored to specific purposes (Tan et al., 2024; Sakai et al., 2024b; Makinae et al., 2024), as well as crafting catchy slogans (Kim et al., 2023; Brigham et al., 2024), advertisements (Lei et al., 2022; Mita et al., 2024), and poetry (Yu et al., 2024; Zhang and Eger, 2024). However, the instruction-following abilities of LLMs are still developing, and they sometimes produce outputs that deviate from intended instructions (Palmeira Ferraz et al., 2024; Qian et al., 2024). This highlights the need to benchmark their ability to consistently generate text that strictly adheres to given instructions.

Figure 1: Overview of our proposed Ordered CommonGen. Unlike CommonGen (Lin et al., 2020), we evaluate whether the composed sentences include the concepts in the specified order. To create the Ordered Concept Sets, we use CommonGen's Concept Sets containing four concepts and generate all permutations, resulting in a total of 24 permutations per set.

Generative commonsense reasoning (GCR) (Lin et al., 2020; Nan et al., 2021; Seo et al., 2022; Liu et al., 2023) is a type of constrained text generation task that requires compositional generalization capabilities, where LLMs are tasked with creating natural, commonsense sentences that include all given concepts in Figure 1. Traditional GCR tasks evaluate coverage, focusing on whether the given concepts are included in the composed sentences with-

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 31219–31238

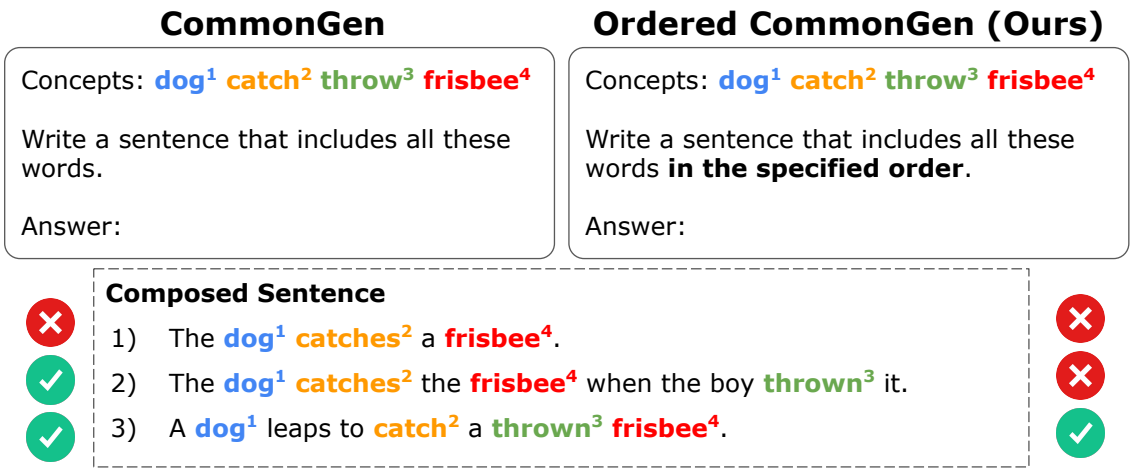

Figure 2: Our evaluation methodology for Ordered CommonGen. The left side shows the instruction template for the standard CommonGen task, while the right side shows the template used in our Ordered CommonGen. All templates are provided in Appendix B. In Ordered CommonGen, the phrase **"in the specified order"** is inserted to explicitly instruct the LLM to compose sentences following the given concept order. For example, Given a concept set (dog, catch, throw, frisbee), the composed sentence (1) does not include all the concepts, making it incorrect for both tasks; (2) includes all the concepts but involves reordering, making it correct only for CommonGen; and (3) includes all the concepts in the specified order, making it correct for both tasks.

out considering their order. Consequently, LLMs often rearrange the order of concepts to produce more natural sentences (Ou et al., 2022; Zhang et al., 2023). However, when instructed to follow a specific order, LLMs must compose sentences that adhere to the user-specified order of concepts.

Such potential needs are particularly important in creative domains, such as describing events in chronological order (Mostafazadeh et al., 2016; Lu et al., 2023; Chen et al., 2023; Pawłowski and Walkowiak, 2024), planning actions (Sakib and Sun, 2023; Lin et al., 2024; Singh et al., 2024), or composing lyrics (Fan et al., 2019; Hu and Sun, 2020; Qian et al., 2023), where a change in the order of concepts can significantly alter the meaning or nuance of the output. According to generative grammar (Chomsky, 1965; Jackendoff, 2002), natural sentences can be composed following syntactic rules without arbitrary rearrangements, as humans are capable of processing and generating sentences in grammatically permissible orders. Therefore, by simultaneously addressing instruction-following ability and compositional generalization ability, LLMs can be applied to more creative text generation tasks.

To address this, we propose Ordered CommonGen, a framework designed to evaluate both compositional generalization and instruction-following abilities by reorganizing CommonGen (Lin et al., 2020), a representative GCR task. As shown in Figure 1, Ordered CommonGen permutes the order of concepts and, as illustrated in the prompts in Figure 2, inserts the phrase **"in the specified order"** into the instruction. This setup enables the evaluation of whether LLMs can compose sentences while adhering to the given instructions.

Through a comprehensive benchmarking across 36 LLMs, we found that while LLMs can compose sentences containing the given concepts, their ability to compose sentences in the specified order, as required by Ordered CommonGen, is limited to about 75%, even for the best-performing model. Interestingly, inserting the phrase **"in the specified order"** into the instruction improves the ability to follow the specified order, indicating that LLMs understand the intent of the instructions. However, biases toward specific concept order patterns and identical outputs, even when the order of concepts is altered, remain significant challenges. These findings highlight challenges in the compositional generalization and instruction-following capabilities of LLMs within GCR tasks.

2 Ordered CommonGen

2.1 Task Definition

Generative Commonsense Reasoning (GCR) tasks such as CommonGen (Lin et al., 2020) involve generating a natural and grammatically correct sentence that incorporates all given concepts in a concept set. Formally, the input consists of a concept

set $X = \{c_1, c_2, \ldots, c_k\}$, where each concept c_i is a lemmatized common object (noun) or action (verb). The task requires LLMs to generate a sentence Y such that all concepts in X are included, allowing for morphological inflections or variations. The inclusion of concepts is judged by verifying if the lemmatized forms of all concepts in X are present in Y. Formally, the output is represented as $Y = \{y_1, y_2, \ldots, y_n\}$, where y_i is the i-th sentence generated given the input X. A generated sentence Y is considered valid if it contains all concepts in X in any order, satisfying $X \subseteq Y$.

In Ordered CommonGen, beyond the traditional requirements of GCR tasks, we introduce an additional dimension to evaluate the instruction-following ability of LLMs. Specifically, we assess whether the generated sentence adheres to the **specified order of the concepts** in the input concept set X, checking them sequentially from the beginning.

Humans, capable of the *"infinite use of finite means"* (Chomsky, 1965), can compose sentences following any specified order of concepts, regardless of the arrangement of the words themselves (Bever, 2013; Levelt, 1989; Ferreira and Engelhardt, 2006; Goldberg, 1995; Jackendoff, 2002). This expanded evaluation enables us to explore both compositional generalization, which refers to the systematic combination of known elements in novel ways (Lake et al., 2017; Lake and Baroni, 2018), and instruction-following abilities, which reflect the capacity to align generated outputs with explicit external directives (Lupyan and Clark, 2015). This evaluation highlights the gap between human linguistic understanding and the ability of LLMs.

2.2 Dataset Construction

As shown in Figure 2, two key components are required for dataset creation: **concept sets**, which are collections of word lists used as seeds for text generation, and **instruction templates**, which incorporate the concept sets and serve as input prompts for LLMs. First, we used CommonGen-lite[1], a subset of the CommonGen test data, as a seed dataset. We extracted all 192 seed concept sets, each consisting of four concepts. By generating all possible permutations (4! = 24) of the concepts in each case, we created a total of 192 $\times$ 4! = 4,608 concept sets.

Next, we curated the instruction templates from FLAN (Wei et al., 2022a; Longpre et al., 2023), a collection of manually created templates for various task datasets. Specifically, we utilized all six instruction templates designated for CommonGen in FLAN as base templates. By averaging scores across the six multi-templates, our evaluation accounts for variance in template performance (Sakai et al., 2024c). We modified the base instruction templates by appending the phrase **"in the specified order"**. This template requires LLMs to generate sentences that adhere to the order of the concept set. All examples of instruction templates are provided in Appendix B. Finally, we obtained a total of 6 $\times$ 4,608 = 27,648 instances.

3 Experimental Settings

3.1 Evaluation Metrics

Concepts Coverage: We report three types of coverage rates. First, following Lin et al. (2020)[2], we report the average percentage of input concepts included in the lemmatized outputs, disregarding their order (**Coverage w/o order**). Next, aligned with the aim of Ordered CommonGen, we evaluate whether the generated sentences include all input concepts in the specified order (**Coverage w/ order**). Finally, we report the average percentage of generated sentences that include all input concepts and adhere to the specified order (**Ordered Rate**).

Sentence-level Similarity: When the given order of concepts varies, it influences the structure and coherence of composed sentences, as certain syntactic and semantic constraints govern how concepts can be naturally combined (Goldberg, 1995; Jackendoff, 2002; Steedman, 2000). Therefore, it is important to produce diverse outputs that adhere to different orders of the concept set. Following the idea of Pairwise-BLEU (Shen et al., 2019), we evaluate the diversity of generated sentences by computing the average pairwise similarity across all 24 sentences obtained from permutations of the same four concepts. A lower similarity score suggests greater diversity, as it indicates less overlap among generated sentences. To assess similarity at different levels, we employ two metrics: Pairwise-BLEU (**pBLEU**), which measures surface-level n-gram overlap using BLEU (Papineni et al., 2002; Post, 2018), and Pairwise-BLEURT (**pBLEURT**), which evaluates semantic similarity using BLEURT (Sel-

[1] https://hf.co/datasets/allenai/commongen_lite

[2] We referred to the implementation at https://github.com/allenai/CommonGen-Eval. For lemmatizing the generated sentences, we similarly used spaCy (Honnibal et al., 2020) with the EN_CORE_WEB_LG model: https://spacy.io/models/en#en_core_web_lg.

lam et al., 2020). For BLEU, we set $n = 4$, and for BLEURT, we used the BLEURT-20 model.

Corpus-level Diversity: We also evaluate the overall diversity of the generated sentences using distinct-n (Li et al., 2016)[3] (**Distinct**), which is calculated as the ratio of unique n-grams to the total number of n-grams in all texts formulated as:

$$\text{distinct-}n = \frac{\#\text{unique } n\text{-grams}}{\#\text{total } n\text{-grams}}. \quad (1)$$

We set $n = 2$. Furthermore, we observed in our preliminary study that some outputs remain identical despite the different order of concepts. This occurs because LLMs have the ability to rearrange concepts to generate more natural sentences (Ou et al., 2022; Zhang et al., 2023). However, from the perspective of instruction-following ability, this behavior is inappropriate. Therefore, we introduce a new diversity metric, **Diverse Rate**, to quantify variations in generated sentences. The Diverse Rate is calculated as the ratio of unique sentences to the total number of sentences formulated as:

$$\text{Diverse Rate} = \frac{\#\text{unique sentences}}{\#\text{total sentences}}. \quad (2)$$

If all generated sentences are unique, the Diverse Rate is 1.00. For instance, if 24 sentences are generated but only one is unique, the Diverse Rate would be $\frac{1}{24} \fallingdotseq 0.04$. As these metrics represent the ratio of unique n-grams or sentences, a higher score indicates greater diversity.

Perplexity: In GCR tasks such as CommonGen, the quality of generated sentences is typically evaluated by comparing them to human references. However, our focus is on generating diverse sentences, and the references do not account for all permutations of the concept sets. Therefore, these evaluation methods are not aligned with our goal. Instead, we employ GPT2-XL (Radford et al., 2019)[4] as an evaluation model and measure perplexity via LMPPL[5] to perform a relative assessment of the quality of the generated sentences. Since perplexity is equivalent to the information content of a sentence, lower perplexity generally indicates higher-quality sentences. Furthermore, LLMs capable of generating sentences with lower perplexity while adhering to the specified order are more likely to produce commonsensical and coherent sentences.

[3] https://github.com/neural-dialogue-metrics/Distinct-N

[4] https://hf.co/openai-community/gpt2-xl

[5] https://github.com/asahi417/lmppl

3.2 Evaluation Setup

Settings for LLMs. We primarily selected a total of 36 well-known instruction-tuned LLMs for evaluation: Llama3-3.2-1B, 3.2-3B, 3.1-8B, 3.1-70B, 3.3-70B, 3.1-405B (Dubey et al., 2024), Qwen2-0.5B, 1.5B, 7B, 72B (Yang et al., 2024), Qwen2.5-0.5B, 1.5B, 3B, 7B, 14B, 32B, 72B (Qwen et al., 2024), Gemma2-2B, 9B, 27B (Team et al., 2024b), Phi3-mini, small, medium (Abdin et al., 2024), Tülu3-8B, 70B (Lambert et al., 2024), OLMo2-7B, 13B (OLMo et al., 2025), Mistral-7B, Small, Large (Jiang et al., 2023), Mixtral-8x7B, 8x22B (Jiang et al., 2024), Gemini-Flash, Pro (Team et al., 2024a), GPT-3.5 (Ouyang et al., 2022), and GPT-4o (OpenAI et al., 2024). Following the evaluation settings by Sakai et al. (2024a), for proprietary models (Gemini, GPT-3.5, and GPT-4o), the temperature was set to 0 to achieve as deterministic outputs as possible. For the other open LLMs, outputs were generated using greedy decoding with a fixed seed value and 4-bit quantization (Dettmers et al., 2023)[6]. Further details are provided in Appendix A.

Settings for Datasets. We compared two instruction prompts: one using our Ordered CommonGen template with the phrase **"in the specified order"**, and the other using the original CommonGen template without this phrase. If including the phrase in the prompt improves order-considered coverage, it indicates that the model successfully follows the given instructions. We report average scores across all six templates, with Ordered CommonGen values shown alongside their differences from the original CommonGen.We primarily conducted evaluations in zero-shot settings to highlight differences in inductive reasoning ability.

4 Results and Discussions

Table 1 shows the main evaluation results.

Finding 1: LLMs understand the intent of instruction prompts. Focusing on the increase in the w/ order scores in Table 1, specifying the usage order of concepts in the prompt through Ordered CommonGen improves coverage rates for most LLMs. This is further supported by the rise in Ordered Rate scores. Notably, for certain LLMs

[6] The 4-bit quantization models yield optimal performance (Dettmers et al., 2023; Liu et al., 2024a; Dettmers and Zettlemoyer, 2023). Therefore, we apply 4-bit quantization to save computational costs.

	Concepts Coverage (↑)			Similarlity (↓)		Diversity (↑)		Perplexity (↓)
	w/o order	w/ order	Ordered Rate	pBLEU	pBLEURT	Distinct	Diverse Rate	
Llama3.2-1B	61.82(+2.56)	17.22(-1.02)	27.86(-2.92)	17.62(+2.37)	41.89(+1.95)	92.79(-1.17)	93.66(-2.35)	52.46(+7.95)
Llama3.2-3B	63.81(+2.64)	19.47(+1.33)	30.52(+0.86)	18.49(+2.02)	44.38(+3.08)	94.18(-0.84)	90.53(-2.88)	53.85(+8.47)
Llama3.1-8B	72.10(+3.32)	23.35(+3.80)	32.39(+3.96)	19.61(+0.66)	47.30(+2.83)	93.45(-1.14)	87.91(-2.24)	61.72(+13.47)
Llama3.1-70B	95.69(+4.12)	41.19(+25.66)	43.04(+26.08)	21.89(-7.30)	52.40(-1.92)	93.03(-0.61)	83.84(+8.73)	60.24(+11.39)
Llama3.3-70B	97.25(+2.17)	66.79(+47.34)	68.68(+48.22)	15.24(**-11.17**)	48.04(-4.45)	93.53(-0.50)	94.70(+14.13)	67.50(+21.19)
Llama3.1-405B	**98.91**(+2.84)	**74.44**(**+55.41**)	**75.26**(**+55.46**)	12.54(-11.12)	42.90(**-5.72**)	94.81(-0.20)	98.28(+13.43)	61.49(+17.73)
Qwen2-0.5B	53.78(-0.61)	30.84(-0.89)	57.34(-0.98)	**10.48**(+0.57)	39.61(+1.35)	93.31(-0.72)	96.60(-1.01)	171.27(+81.35)
Qwen2-1.5B	73.06(+0.91)	24.12(-0.65)	33.01(-1.32)	15.72(+1.27)	43.70(+1.17)	93.71(-0.32)	91.81(-2.86)	53.52(+2.06)
Qwen2-7B	88.52(+4.16)	27.49(+2.72)	31.06(+1.70)	13.72(-0.08)	43.93(+0.98)	94.26(-0.69)	95.49(-0.82)	70.46(+12.74)
Qwen2-72B	94.99(+3.51)	32.09(+7.13)	33.78(+6.50)	19.13(-1.53)	48.58(+0.25)	93.47(-0.72)	85.49(+0.35)	**48.68**(+3.45)
Gemma2-2B	80.07(**+10.79**)	23.66(+0.44)	29.55(-3.97)	21.28(+3.84)	51.04(+5.22)	91.09(-1.69)	82.45(-7.27)	159.99(+80.40)
Gemma2-9B	90.10(+6.76)	23.56(+3.82)	26.15(+2.47)	23.95(+1.06)	53.74(+2.53)	91.87(-0.63)	78.94(-1.86)	88.74(+11.56)
Gemma2-27B	88.49(+8.58)	28.24(+7.73)	31.91(+6.24)	22.14(-0.86)	51.35(+0.88)	92.16(-0.58)	80.84(+0.42)	94.27(+17.05)
Phi3-mini	79.85(+4.90)	49.54(+2.73)	62.04(-0.41)	10.72(+0.04)	41.70(+0.39)	94.36(-0.56)	97.53(-0.72)	100.71(+11.19)
Phi3-small	89.79(+3.38)	49.93(+14.13)	55.61(+14.17)	10.99(-1.22)	40.81(-0.28)	94.22(-0.63)	97.01(-0.38)	77.02(+15.27)
Phi3-medium	85.84(+1.77)	50.21(+6.85)	58.49(+6.92)	10.91(-0.77)	39.69(-0.46)	**95.16**(**-0.07**)	98.23(+0.48)	80.62(+4.38)
Mistral-7B	81.26(+3.82)	40.71(-2.42)	50.10(-5.60)	18.13(+2.14)	48.79(+2.66)	92.56(-1.36)	86.49(-5.64)	238.96(+157.60)
Mistral-small	95.36(+2.68)	37.12(+17.12)	38.92(+17.34)	22.93(-5.86)	55.55(-2.86)	91.64(-0.35)	78.39(+7.90)	97.88(+21.38)
Mistral-large	87.07(+5.05)	23.67(-0.73)	27.19(-2.56)	25.27(+2.80)	54.61(+3.13)	92.58(-0.66)	76.54(-4.98)	94.91(+9.18)
Mixtral-8x7B	77.36(+2.90)	19.67(+0.09)	25.43(-0.87)	11.29(+0.36)	**38.50**(+0.31)	95.06(-0.31)	**98.82**(-0.17)	52.35(**+1.51**)
Mixtral-8x22B	90.96(+4.52)	36.18(+9.11)	39.77(+8.46)	13.40(-0.03)	42.34(+0.68)	94.55(-0.43)	96.08(-0.58)	56.51(+2.19)
Qwen2.5-0.5B	64.50(+3.27)	28.12(+0.86)	43.60(-0.92)	10.86(-0.09)	40.38(-0.59)	93.27(**-0.07**)	97.12(+0.15)	94.22(+19.41)
Qwen2.5-1.5B	57.09(+5.97)	14.67(+0.02)	25.69(-2.96)	14.65(+0.80)	43.22(+2.39)	94.28(-0.63)	91.71(-1.64)	64.83(+7.50)
Qwen2.5-3B	86.01(+4.98)	22.60(+3.36)	26.28(+2.53)	14.85(+1.22)	47.51(+2.38)	91.94(-0.99)	92.75(-2.79)	115.35(+26.75)
Qwen2.5-7B	87.80(+4.46)	28.58(+6.26)	32.55(+5.76)	15.19(+0.59)	45.97(+1.76)	92.94(-0.73)	92.35(-1.56)	75.97(+6.24)
Qwen2.5-14B	94.49(+3.84)	40.65(+15.76)	43.02(+15.56)	13.96(-2.15)	44.38(-0.14)	93.10(-0.53)	93.21(+1.31)	68.24(+8.90)
Qwen2.5-32B	96.50(+3.18)	47.23(+22.82)	48.94(+22.78)	12.94(-4.11)	43.56(-2.85)	92.80(-0.51)	93.20(+3.18)	71.03(+7.20)
Qwen2.5-72B	97.17(+3.91)	50.26(+24.66)	51.73(+24.27)	17.24(-5.01)	47.98(-1.64)	93.19(-0.50)	88.51(+6.41)	72.02(+13.93)
OLMo2-7B	91.11(+2.09)	28.10(-3.97)	30.84(-5.18)	19.05(+2.21)	52.44(+2.82)	91.65(-0.69)	85.25(-3.67)	112.93(+15.01)
OLMo2-13B	95.22(+3.29)	27.20(+1.68)	28.56(+0.80)	24.33(+1.51)	58.05(+2.69)	90.01(-1.32)	77.20(-3.28)	433.50(+301.35)
Tülu3-8B	89.79(+4.04)	27.55(-1.63)	30.69(-3.35)	16.09(+1.99)	45.97(+3.48)	92.61(-1.16)	90.42(-3.39)	103.14(+34.83)
Tülu3-70B	93.46(+3.58)	33.46(+12.28)	35.80(+12.24)	21.55(-3.26)	52.25(-0.44)	92.23(-0.57)	80.04(+3.52)	79.78(+22.72)
GPT-3.5	92.74(+2.15)	34.14(+8.22)	36.82(+8.20)	23.56(-1.75)	56.83(+0.34)	90.88(-0.99)	76.16(+0.68)	236.08(+155.67)
GPT-4o	96.70(+3.12)	53.34(+30.25)	55.16(+30.49)	19.95(-9.37)	50.94(-4.80)	92.91(-0.25)	86.51(**+14.47**)	58.15(+7.27)
Gemini-Flash	91.79(+7.68)	33.29(+14.51)	36.27(+13.94)	24.03(-1.78)	56.16(+2.08)	91.37(-0.89)	75.92(+1.40)	92.75(+29.04)
Gemini-Pro	89.16(+8.03)	32.68(+16.52)	36.66(+16.74)	18.67(-2.79)	50.88(-0.11)	91.51(-0.82)	84.48(+2.40)	117.09(+41.66)

Table 1: Evaluation results for the generated sentences of each LLM. The average scores for Ordered CommonGen, where the order of concepts is specified, are reported across all six templates as the main results, with performance differences from the original CommonGen, which does not consider concept order, shown in parentheses. The bold scores highlight the LLM with the best performance for each metric, while the underlined scores indicate the second-best. Higher scores indicate better performance for Concepts Coverage and Corpus-level Diversity, whereas lower scores indicate better performance for Sentence-level Similarity and Perplexity.

such as Llama3.3-70B, Llama3.1-405B, and GPT-4o, w/o order scores remain high while w/ order scores increase significantly, indicating that these models attempt to align with the specified order. This suggests that LLMs strive to produce outputs consistent with the intent of instruction prompts. Interestingly, even in the w/o order scores, Ordered CommonGen improves coverage rates for most LLMs. This suggests that explicit instructions to include concepts in the output encouraged LLMs to use them effectively, even when they struggled to arrange them in the specified order.

Finding 2: LLMs tend to generate natural sentences while trying to follow the instructions. When focusing on perplexity in Table 1, specifying the order of concepts in the input prompt generally worsens perplexity compared to when no order is specified. However, the degradation is relatively limited, except for certain LLMs such as OLMo2-

13B and GPT-3.5. This indicates that LLMs are capable of following instructions while still generating natural sentences. Furthermore, LLMs respond to the phrase "in the specified order" included in Ordered CommonGen, demonstrating their ability to switch generation behavior accordingly while maintaining compositional generalization abilities. As the increase in perplexity is generally minimal, yet the Ordered Rate improves, this suggests that LLMs can generate sentences that adhere to the specified order while maintaining a degree of naturalness. It suggests behavior resembling semantic compositionality (Jackendoff, 2002).

Finding 3: Yet LLMs often struggle to generate sentences that follow instructions precisely. On the other hand, even the LLM with the highest Ordered Rate, Llama3.1-405B, followed instructions in only about 75% of cases, leaving over 20% of outputs that did not adhere to the given instructions. Furthermore, w/o order scores did not reach 100%, indicating that LLMs fail to fully comply with instructions. For example, in the case of Llama3.1-405B, given the concept set (*number, wear, shirt, run*), the generated sentence *"The large **number** on the back of my **shirt** seemed to motivate me to **run** even faster."* excludes the concept *wear*. In contrast, humans would compose sentences such as *"A large **number** of people **wear** a yellow **shirt** to **run** in the charity marathon."*, naturally including all concepts in the specified order. Such omissions of concepts highlight a critical challenge in GCR tasks (Lin et al., 2020; Zhao et al., 2022; Zhang et al., 2023). Moreover, the additional constraint of maintaining the specified order in Ordered CommonGen makes this task even more challenging.We provide a more detailed analysis of such cases in Appendix C.3, where we show, based on statistical evidence, that at least one of the 36 LLMs we tested is capable of composing each sentence correctly. This reflects the fact that many LLMs still have some room in their compositional capabilities.

Finding 4: LLMs generate more diverse outputs when concept order is considered. Focusing on Sentence-level Similarity in Table 1, LLMs such as Llama3.1-405B, Qwen2.5-72B, and GPT-4o, which show significant improvements in Ordered Rate through Ordered CommonGen, also demonstrate substantial gains in both pBLEU and pBLEURT. In contrast, some LLMs, such as GPT-3.5 and Gemini-Flash, show improvements in pBLEU but a decline in pBLEURT. Interestingly, in some LLMs, such as Qwen, smaller models tend to generate more diverse outputs than their larger ones at the expense of Concepts Coverage. This suggests that while adhering to instructions improves syntactic diversity, it does not necessarily enhance semantic diversity. These results indicate that LLMs prioritize syntactic compositionality, as reflected in consistent pBLEU improvements (Goldberg, 1995; Steedman, 2000; Lake and Baroni, 2018), while semantic compositionality remains more challenging due to its reliance on deeper contextual understanding (Jackendoff, 2002; Bresnan, 2001; Bender and Koller, 2020). This suggests that LLMs with higher instruction-following abilities, such as Llama3.1-405B, better balance syntactic and semantic compositionality, whereas lower-performing LLMs rely more heavily on syntactic patterns.

Finding 5: LLMs sometimes generate identical sentences even when the concepts are shuffled. According to the Diverse Rate in Table 1, even the highest-performing LLM, Mixtral-8x7B, does not reach 100%, indicating that identical sentences are sometimes generated despite changes in the order of concepts. Furthermore, LLMs such as Gemma, Gemini, and OLMo2 achieve only around 80%, suggesting a tendency to reorder concepts into sequences they consider most natural, overriding instructions to follow the specified order. For example, in the case of Gemma2-2B with the concept set (*dock, dog, jump, water*), the output remains the same as ***"The dog jumped off the dock into the water."*** across all permutations of the input concepts for a given prompt. This behavior reflects the influence of frequent concept set orders in the training data (Zhang et al., 2023; Ou et al., 2022; Zhao et al., 2022; Yang et al., 2023; Bender and Koller, 2020), which often take precedence over given instructions. Interestingly, models like GPT-4o and Llama3.3-70B tend to produce duplicate outputs when instructions do not enforce a specified order. In such cases, LLMs default to generating sentences in the most natural sequence, regardless of input order. This tendency to default to natural sequences aligns with usage-based theories (Bybee, 2006, 2010), which argue that frequent patterns shape human language production, establishing the 'default' or 'natural' order. Similar tendencies are observed in downstream tasks like neural machine translation (Raunak and Menezes, 2022) and grammatical error correction (Cao et al., 2023), where minor input variations are absorbed, and frequent

POS	Concepts Coverage			Diverse
pattern	w/o order	w/ order	Ordered Rate	Rate
NNNN	**91.13**$_{\pm 11.86}$	40.92$_{\pm 15.70}$	44.88$_{\pm 16.39}$	91.97$_{\pm 6.78}$
NNNV	88.41$_{\pm 10.70}$	29.39$_{\pm 14.26}$	33.04$_{\pm 14.57}$	86.94$_{\pm 8.57}$
NNVN	88.06$_{\pm 10.46}$	37.34$_{\pm 13.69}$	42.09$_{\pm 13.40}$	86.94$_{\pm 8.57}$
NNVV	84.26$_{\pm 12.61}$	27.45$_{\pm 14.55}$	32.34$_{\pm 15.69}$	89.58$_{\pm 6.81}$
NVNN	87.38$_{\pm 10.67}$	39.80$_{\pm 12.83}$	45.23$_{\pm 12.14}$	86.94$_{\pm 8.57}$
NVNV	83.73$_{\pm 12.50}$	29.55$_{\pm 13.44}$	34.91$_{\pm 13.78}$	89.58$_{\pm 6.81}$
NVVN	83.16$_{\pm 13.11}$	29.44$_{\pm 14.89}$	34.70$_{\pm 14.96}$	89.58$_{\pm 6.81}$
NVVV	80.35$_{\pm 17.03}$	35.10$_{\pm 17.84}$	42.11$_{\pm 16.27}$	92.81$_{\pm 5.09}$
VNNN	88.01$_{\pm 11.46}$	39.79$_{\pm 13.33}$	44.92$_{\pm 12.58}$	86.94$_{\pm 8.57}$
VNNV	84.77$_{\pm 13.40}$	34.81$_{\pm 12.86}$	40.69$_{\pm 11.85}$	89.58$_{\pm 6.81}$
VNVN	84.83$_{\pm 13.02}$	**46.88**$_{\pm 14.38}$	54.50$_{\pm 11.61}$	89.58$_{\pm 6.81}$
VNVV	80.36$_{\pm 17.12}$	42.59$_{\pm 18.82}$	51.35$_{\pm 15.51}$	92.81$_{\pm 5.09}$
VVNN	83.54$_{\pm 13.84}$	29.50$_{\pm 14.50}$	34.75$_{\pm 14.20}$	89.58$_{\pm 6.81}$
VVNV	79.63$_{\pm 17.33}$	34.03$_{\pm 17.22}$	41.31$_{\pm 15.98}$	92.81$_{\pm 5.09}$
VVVN	80.38$_{\pm 18.05}$	33.08$_{\pm 19.02}$	39.61$_{\pm 17.69}$	92.81$_{\pm 5.09}$
VVVV	37.38$_{\pm 28.54}$	24.31$_{\pm 20.38}$	**63.84**$_{\pm 27.89}$	**98.17**$_{\pm 2.48}$

Table 2: Results of Concepts Coverage and Diverse Rate for each part-of-speech (POS) order of the given four concepts. Higher scores indicate better performance. **N** denotes nouns, and **V** denotes verbs. We report the score of mean and standard deviation across 36 LLMs.

		Concepts Coverage			Diverse
ID		w/o order	w/ order	Ordered Rate	Rate
0	w/o	81.39$_{\pm 12.99}$	26.76$_{\pm 9.30}$	33.35$_{\pm 11.97}$	90.77$_{\pm 7.16}$
	w/	**85.05**$_{\pm 12.57}$	**39.66**$_{\pm 16.64}$	**45.81**$_{\pm 15.94}$	**92.13**$_{\pm 5.98}$
1	w/o	83.60$_{\pm 12.75}$	27.67$_{\pm 9.27}$	33.50$_{\pm 11.40}$	88.86$_{\pm 7.79}$
	w/	**86.34**$_{\pm 13.06}$	**39.10**$_{\pm 15.10}$	**44.69**$_{\pm 14.17}$	**90.90**$_{\pm 6.04}$
2	w/o	77.62$_{\pm 12.19}$	25.52$_{\pm 10.51}$	33.85$_{\pm 15.31}$	**86.97**$_{\pm 9.18}$
	w/	**83.04**$_{\pm 12.66}$	**36.91**$_{\pm 13.67}$	**44.21**$_{\pm 14.47}$	88.62$_{\pm 7.37}$
3	w/o	82.00$_{\pm 12.25}$	22.77$_{\pm 8.83}$	28.45$_{\pm 11.88}$	89.18$_{\pm 8.80}$
	w/	**85.43**$_{\pm 12.54}$	**31.78**$_{\pm 15.84}$	**36.62**$_{\pm 15.63}$	**89.75**$_{\pm 8.06}$
4	w/o	77.09$_{\pm 12.36}$	27.85$_{\pm 7.82}$	**37.17**$_{\pm 12.48}$	**89.79**$_{\pm 7.76}$
	w/	**83.89**$_{\pm 11.79}$	**30.58**$_{\pm 9.97}$	36.62$_{\pm 10.78}$	86.77$_{\pm 8.97}$
5	w/o	85.84$_{\pm 12.65}$	17.77$_{\pm 5.68}$	21.09$_{\pm 7.24}$	82.02$_{\pm 13.21}$
	w/	**87.85**$_{\pm 12.79}$	**28.88**$_{\pm 15.98}$	**32.09**$_{\pm 15.06}$	**84.25**$_{\pm 11.04}$

Table 3: The mean and standard deviation of Concepts Coverage and Diverse Rate across 36 LLMs for each prompt. Details of the prompt templates are provided in Appendix B. The terms **w/o** and **w/** indicate the absence or presence of the phrase **"in the specified order"** in the prompt template, respectively.

patterns influence outputs in the training data. Addressing these limitations may require improved training methods or increased model parameters. Additionally, well-instruction-following LLMs like GPT-4o and Llama3.1-405B demonstrate the ability to switch behaviors based on the instructions.

5 Analysis

We will focus on important aspects in the following sections and defer more discussions to Appendix C.

5.1 Which Part-of-Speech (POS) Patterns do LLMs Struggle with?

In GCR tasks, the concepts are categorized into two types: common objects (nouns) and actions (verbs). To investigate the impact of part-of-speech (POS) order on LLM performance, we report the mean and standard deviation for each POS pattern across all 36 LLMs in Table 2. Table 2 shows that the NNNN pattern (noun-only) achieves the highest Concepts Coverage. In contrast, the VVVV pattern (verb-only) results in outputs where all concepts are included in only approximately 37% of cases. Additionally, the VVVV pattern exhibits the highest variance in Concepts Coverage, indicating significant compositional generalization challenges for verb-heavy concept sets in GCR tasks. These challenges also contribute to performance differences among LLMs. Interestingly, the VVVV pattern achieves the highest scores in both the Ordered Rate and the Diverse Rate. This suggests that, despite difficulties in composing sentences with actions, LLMs actively attempt to do so. Furthermore, when all concepts are included in the output for the VVVV pattern, LLMs demonstrate relatively strong instruction-following capabilities in these scenarios.

5.2 Impact of Prompt Template Variations on Instruction-Following Performance

LLM performance is affected by slight differences in prompt phrasing (Sakai et al., 2024c). To measure the impact of such variations on instruction-following ability, we report the mean and standard deviation of all metrics across 36 LLMs for each of the six prompt templates in Table 3. The results in Table 3 indicate that while LLM performance varies depending on prompt phrasing, specifying **"in the specified order"** generally improves w/ order scores across all templates. However, performance differences persist across templates, and some metrics, such as Ordered Rate, do not always improve with this phrasing. Furthermore, performance varies significantly among LLMs for each prompt template, making it difficult to identify a universally effective prompt. These findings highlight the importance of evaluating and reporting averaged results across multiple templates. While prompt-tuning is necessary to achieve optimal performance, the results confirm that LLMs can follow specific instructions when explicitly stated.

	Concepts Coverage (↑)			Similarlity (↓)		Diversity (↑)		Perplexity (↓)
	w/o order	w/ order	Ordered Rate	pBLEU	pBLEURT	Distinct	Diverse Rate	
Llama3.2-1B	67.05(+5.23)	13.82(-3.40)	20.61(-7.25)	23.60(+5.98)	47.63(+5.73)	92.75(-0.04)	85.52(-8.14)	**43.97**(-8.49)
Llama3.2-3B	69.67(+5.86)	13.57(-5.90)	19.48(-11.03)	28.22(+9.73)	52.50(+8.12)	93.84(-0.34)	78.46(-12.07)	48.03(-5.82)
Llama3.1-8B	85.26(**+13.16**)	18.31(-5.05)	21.47(-10.92)	28.36(+8.76)	54.16(+6.86)	93.29(-0.15)	76.43(-11.48)	48.22(-13.50)
Llama3.1-70B	98.57(+2.88)	41.62(+0.43)	42.22(-0.82)	27.06(+5.17)	56.12(+3.72)	93.00(-0.03)	78.43(-5.41)	49.63(-10.61)
Llama3.3-70B	99.01(+1.76)	75.81(+9.03)	76.58(+7.90)	18.01(+2.77)	49.72(+1.68)	93.71(+0.19)	95.72(+1.01)	55.80(-11.70)
Llama3.1-405B	**99.56**(+0.65)	**87.79**(**+13.35**)	**88.18**(**+12.92**)	14.65(+2.12)	46.01(+3.11)	94.10(-0.71)	**99.37**(**+1.09**)	65.66(+4.18)
Qwen2-0.5B	35.39(-18.40)	16.32(-14.52)	46.12(-11.23)	15.16(+4.68)	43.73(+4.12)	93.10(-0.20)	91.31(-5.30)	57.46(-113.81)
Qwen2-1.5B	63.50(-9.57)	13.06(-11.05)	20.57(-12.43)	21.87(+6.16)	50.41(+6.71)	92.99(-0.72)	84.31(-7.50)	49.56(-3.96)
Qwen2-7B	89.82(+1.30)	17.44(-10.05)	19.41(-11.64)	19.75(+6.04)	48.71(+4.79)	94.00(-0.25)	89.68(-5.82)	52.69(-17.76)
Qwen2-72B	96.77(+1.79)	27.97(-4.12)	28.90(-4.88)	24.28(+5.16)	53.95(+5.37)	92.80(-0.67)	77.41(-8.07)	47.27(-1.40)
Gemma2-2B	85.94(+5.87)	9.23(-14.43)	10.74(-18.81)	38.36(+17.07)	62.05(+11.01)	91.25(+0.16)	58.32(-24.13)	76.64(-83.35)
Gemma2-9B	93.34(+3.24)	13.77(-9.80)	14.75(-11.40)	34.82(+10.87)	61.50(+7.77)	91.73(-0.13)	61.24(-17.70)	61.94(-26.80)
Gemma2-27B	93.72(+5.24)	16.78(-11.45)	17.91(-14.00)	31.81(+9.67)	57.35(+6.00)	92.35(+0.20)	65.62(-15.22)	55.07(-39.20)
Phi3-mini	78.87(-0.98)	41.68(-7.86)	52.85(-9.19)	**13.59**(+2.87)	43.34(+1.64)	93.44(-0.92)	94.31(-3.22)	90.59(-10.11)
Phi3-small	90.64(+0.85)	42.05(-7.87)	46.40(-9.21)	14.27(+3.28)	**42.06**(**+1.25**)	**94.22**(+0.00)	94.75(-2.26)	58.47(-18.55)
Phi3-medium	92.24(+6.40)	47.83(-2.38)	51.85(-6.64)	16.21(+5.31)	45.43(+5.74)	93.71(-1.45)	92.63(-5.60)	85.77(+5.15)
Mistral-7B	86.75(+5.49)	22.12(-18.59)	25.50(-24.60)	31.88(+13.75)	61.74(+12.94)	91.69(-0.87)	66.29(-20.20)	93.86(-145.09)
Mistral-small	97.24(+1.89)	29.85(-7.27)	30.70(-8.23)	30.94(+8.01)	62.80(+7.25)	90.70(-0.93)	64.38(-14.01)	109.91(+12.03)
Mistral-large	91.58(+4.52)	14.48(-9.19)	15.81(-11.37)	41.04(+15.77)	67.42(+12.81)	90.81(-1.77)	48.60(-27.94)	74.12(-20.79)
Mixtral-8x7B	83.64(+6.28)	14.44(-5.23)	17.26(-8.17)	16.15(+4.87)	43.52(+5.02)	93.97(-1.09)	93.61(-5.21)	48.99(-3.36)
Mixtral-8x22B	91.59(+0.63)	21.43(-14.74)	23.40(-16.37)	20.96(+7.56)	49.03(+6.69)	90.55(-4.00)	81.77(-14.31)	45.93(-10.58)
Qwen2.5-0.5B	64.51(+0.01)	29.23(+1.11)	45.31(+1.71)	15.75(+4.89)	47.91(+7.53)	90.89(-2.38)	89.56(-7.56)	175.63(+81.41)
Qwen2.5-1.5B	62.76(+5.67)	10.52(-4.14)	16.76(-8.93)	25.50(+10.86)	55.05(+11.83)	92.16(-2.12)	76.82(-14.89)	72.33(+7.51)
Qwen2.5-3B	84.05(-1.96)	12.41(-10.19)	14.77(-11.51)	24.93(+10.08)	55.18(+7.68)	91.61(-0.33)	80.78(-11.97)	82.97(-32.38)
Qwen2.5-7B	90.08(+2.27)	17.61(-10.97)	19.55(-13.00)	22.23(+7.04)	52.50(+6.53)	92.64(-0.30)	83.96(-8.39)	63.78(-12.19)
Qwen2.5-14B	95.31(+0.82)	24.92(-15.73)	26.14(-16.88)	20.56(+6.60)	51.17(+6.79)	92.71(-0.39)	86.16(-7.05)	56.11(-12.14)
Qwen2.5-32B	97.12(+0.63)	40.02(-7.20)	41.21(-7.73)	14.61(+1.67)	44.89(+1.33)	92.15(-0.65)	92.47(-0.73)	72.87(+1.84)
Qwen2.5-72B	97.63(+0.46)	39.70(-10.57)	40.66(-11.07)	22.44(+5.20)	51.62(+3.64)	93.32(+0.13)	82.47(-6.03)	57.46(-14.56)
OLMo2-7B	93.32(+2.22)	16.61(-11.49)	17.80(-13.04)	25.95(+6.89)	56.29(+3.85)	91.84(+0.19)	76.18(-9.07)	75.27(-37.66)
OLMo2-13B	95.74(+0.52)	18.11(-9.08)	18.92(-9.64)	31.63(+7.30)	61.68(+3.63)	90.48(+0.47)	66.07(-11.13)	117.15(**-316.35**)
Tülu3-8B	94.20(+4.41)	16.68(-10.88)	17.70(-12.98)	26.23(+10.13)	54.80(+8.82)	92.41(-0.20)	76.87(-13.55)	55.90(-47.23)
Tülu3-70B	96.43(+2.96)	30.09(-3.38)	31.20(-4.60)	27.98(+6.43)	55.32(+3.07)	92.59(+0.36)	71.97(-8.07)	51.83(-27.95)
GPT-3.5	96.03(+3.28)	24.31(-9.83)	25.32(-11.50)	35.70(+12.14)	65.28(+8.45)	91.41(+0.53)	59.46(-16.70)	71.05(-165.03)
GPT-4o	98.55(+1.85)	55.95(+2.61)	56.78(+1.62)	23.09(+3.14)	53.50(+2.56)	92.76(-0.15)	83.03(-3.48)	52.32(-5.83)
Gemini-Flash	94.40(+2.61)	20.61(-12.68)	21.84(-14.44)	35.37(+11.34)	62.74(+6.58)	91.80(+0.42)	57.77(-18.16)	55.39(-37.37)
Gemini-Pro	93.23(+4.08)	35.52(+2.84)	38.10(+1.44)	20.20(**+1.53**)	52.79(+1.91)	92.14(**+0.64**)	84.23(-0.25)	62.57(-54.52)

Table 4: Evaluation results for the generated sentences of each LLM with the **ideal example** one-shot setting. The average scores for Ordered CommonGen, where the order of concepts is specified, are reported across all six templates as the main results, with performance differences from the zero-shot Ordered CommonGen, as reported in Table 1, shown in parentheses. Bold scores indicate the LLM with the best performance for each metric, while underlined scores represent the second-best. Higher scores signify better performance for Concepts Coverage and Corpus-level Diversity, whereas lower scores indicate better performance for Sentence-level Similarity and Perplexity. Note that all metrics except perplexity are expressed as percentages. The values in parentheses indicate relative relationships between scores, showing increases or decreases as well as improvements or deteriorations based on the score differences. In other words, they are calculated using simple subtraction.

5.3 One-shot Example as Priming

Motivation. Humans are known to produce responses influenced by recently encountered information, a phenomenon referred to as priming (McNamara, 2005; Bargh and Chartrand, 2000; Zorzi et al., 2004; Lee et al., 2023; Sharma et al., 2024). For instance, if the preceding input includes the concept set *(apple place tree pick)* and the response begins with *"My favorite words are apple, place, tree, and pick"*, subsequent responses are likely to follow a similar pattern. This tendency is influenced by meta-information, such as word order or response format, rather than the semantic meanings of the words, leading to sentences that start

with *"My favorite words are . . . "*. To simulate this, we evaluated the model's performance using a one-shot example where the input was a concept set *(apple place tree pick)*, and the generated sentence was *"My favorite words are apple, place, tree, and pick."*. The evaluation aimed to prime LLMs to produce monotonic outputs by following the template: *"My favorite words are A, B, C, and D"*, where *A*, *B*, *C*, and *D* represent concepts in the order they appear in the set. This priming approach enhances the model's ability to follow instructions more effectively by leveraging a universal template applicable to any arbitrary set of four concepts. Since this template can generate sentences for any concept set, it serves as a universal **"ideal example"** shot to improve instruction adherence.

Results. Table 4 compares the one-shot setting using the ideal example with the zero-shot results of Ordered CommonGen, which assesses the pure inductive ability of LLMs. In Concepts Coverage (w/ order), Table 4 highlights a significant priming effect for Llama3.3-70B and Llama3.1-405B, while other models show a considerable drop in scores. In contrast, Concepts Coverage (w/o order) improves for most LLMs, likely due to the influence of the ideal example's output pattern, which induces reordering. Focusing on the Diverse Rate, we observe a substantial decrease in scores, while similarity metrics worsen across all models. Additionally, the drop in Distinct scores for most models indicates increased monotonicity in the generated sentences. These findings demonstrate that one-shot examples induce monotonic outputs but reduce the diversity of the generated sentences. Furthermore, we frequently observed generated sentences deviating from the ideal example patterns and instead producing natural sentences. This suggests that LLMs are not solely primed by the one-shot example but are also influenced by patterns learned from their training data, prioritizing natural text generation.

Summary. In conclusion, well-instructed LLMs such as Llama3.3-70B and Llama3.1-405B can simulate priming effects for reasoning. However, most LLMs demonstrate stronger capabilities in generating natural sentences based on patterns learned from their training data. Even for Llama3.1-405B, adherence to all concept sets is not guaranteed, highlighting a persistent challenge. Addressing this issue could enable LLMs to achieve more human-like cognitive text composition, unlocking potential applications such as simulations and beyond.

6 Conclusion

In this study, we proposed Ordered CommonGen, a benchmark framework designed to evaluate the instruction-following and compositional generalization capabilities of LLMs. Ordered CommonGen generates permutations of unique concept sets consisting of four concepts and appends the phrase **"in the specified order"** to the instruction templates to test whether LLMs can compose sentences that include the concepts in the specified order across all permutations. The results revealed several key findings: *LLMs understand the intent of instruction prompts; they tend to generate natural sentences while attempting to follow instructions; yet, they often struggle to precisely follow the instructions.*

Furthermore, LLMs produce more diverse outputs when considering concept order but sometimes generate identical sentences even when the concepts are shuffled. Through an analysis of POS patterns and concept sets, we identified specific conditions under which LLMs face challenges in sentence composition. Finally, inspired by human perception, we conducted an experiment in Section 5.3 using an **"ideal example"** that demonstrates how a sentence can be composed regardless of the given concepts. The results suggest that well-instructed models have the potential for more human-like, cognitively grounded text generation.

In conclusion, while LLMs demonstrate some degree of compositional generalization capability across arbitrary concept orders, they do not fully exhibit compositionality. In contrast, humans can compose sentences that adhere strictly to specified orders, indicating the potential to create counterexamples for LLMs. Addressing this gap between human and LLM performance remains an important direction for future work and its applications.

Future extensions to other tasks. The core idea of our evaluation framework can also be applied to other tasks. For example, it may be extended to poem or story composition based on the order of sentences or scene fragments. Moreover, in the vision domain, e.g., manga or anime, one possible application is composing intermediate images that follow a specified sequence of cells or frames, ensuring the intended development of the story, as well as video tasks. We believe that our evaluation framework is highly versatile and holds great potential for a wide range of creative applications.

7 Limitations

Language and Dataset. In this study, we focused on CommonGen (Lin et al., 2020) and conducted evaluations in English, which limits the generalizability of our findings to other languages, such as Korean (Seo et al., 2022). To ensure simplicity and consistency in our analysis, we targeted concept sets consisting of four concepts within the CommonGen-lite subset of the CommonGen test data. While increasing the number of concepts could provide more insights (Madaan et al., 2023), the associated exponential growth in permutations makes such evaluations highly challenging. Addressing these limitations and exploring tasks with larger concept sets is left for future work.

LLMs. Since it is infeasible to evaluate all LLMs, this study focuses on 36 well-known LLMs, a number comparable to other benchmarking studies (Koto et al., 2023, 2024; Li et al., 2024b; Poh et al., 2024; Liu et al., 2024b). As our research emphasizes instruction-following ability, we conducted evaluations using instruction-tuned LLMs. However, it may be necessary to compare pretrained base LLMs without instruction-tuning in future studies. Nevertheless, a pilot study with base LLMs showed that these models entirely failed to follow instructions, frequently producing nonsensical sentences, making meaningful evaluation impossible. Therefore, we chose to report results only for models that have undergone at least supervised fine-tuning or further instruction-tuning, ensuring a baseline level of instruction adherence. Moreover, although variations in tuning (Ismayilzada et al., 2025) and architecture (Aljaafari et al., 2024, 2025) would ideally be explored, this paper focuses on the evaluation framework and therefore does not include such analyses. Nevertheless, we hope that our framework will contribute to future discussions and improvements in these directions.

Decoding and Prompting. Techniques such as prompting (Zhang et al., 2024; Cui et al., 2024), using external resources (Liu et al., 2023; Yu et al., 2022; Hwang et al., 2023), minimum Bayes risk decoding (Deguchi et al., 2024a,b; Suzgun et al., 2023; Jinnai et al., 2024), or other constrained decoding methods (Sha, 2020; Stowe et al., 2022; Willard and Louf, 2023) might enable the generation of diverse outputs or compositions that adhere to constraints, potentially allowing sentences to be composed from the given concept sets. However, such approaches rely on generation techniques rather than the inductive abilities of the model itself, which is beyond the scope of this study. Moreover, we used only the 1-shot setting in Section 5.3 to capture the model's intent, leaving multi-shot settings unexplored. We focus on the inductive generative commonsense reasoning ability of LLMs, considering their instruction-following capability. This focus aligns with how general-purpose LLMs are typically used by everyday users, ensuring that our evaluation reflects practical usage scenarios.

Evaluation. In this study, we did not use traditional reference-based metrics such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), METEOR (Banerjee and Lavie, 2005), CIDEr (Vedantam et al., 2015), and SPICE (Anderson et al., 2016), which rely on comparisons with human-annotated gold labels, as employed in original CommonGen paper (Lin et al., 2020). Instead, we focused on evaluation methods that do not require labeled data, such as coverage and self-correlation metrics. We made this choice for several reasons. First, at the time of submission, the test data for CommonGen was no longer publicly available, and we chose to respect the owner's policy[7]. Second, comparisons with human-annotated labels in generation tasks have inherent limitations (Reiter, 2018; Schmidtova et al., 2024). Third, human labels for CommonGen are unavailable for all permutations of concept sets, making alignment with our task difficult. Furthermore, the high cost of annotation and scalability challenges led us to avoid reference-based metrics. Additionally, we did not adopt human evaluation, nor did we use LLM-as-a-judge approaches. These methods are known to have certain issues, are not suited for evaluating large volumes of text, have no established standard method (Wang et al., 2023b; Gu et al., 2024), and often lead to ambiguous assessment (Clark et al., 2021). Considering future research potential, we chose not to employ them. As the primary focus of our study is on instruction-following capability, such evaluations fall outside the scope of this work and are left for future investigation. Regarding the philosophy behind our framework design, we believe that adopting a low-cost evaluation method is preferable to encourage broader use. Overall, we believe that our current evaluation framework is sufficient for examining the effectiveness of our approach.

[7] https://github.com/INK-USC/CommonGen and https://hf.co/datasets/allenai/commongen_lite

Ethical Considerations

We used FLAN templates, released under the Apache License 2.0, and concept sets from CommonGen, provided under the MIT License. Both the templates and the concept sets were modified and extended to suit our research needs. Since these licenses permit such adaptations, our work complies with all licensing requirements. Additionally, our study does not involve potentially harmful content, and all outputs successfully passed the safety moderation filters of the LLMs. Consequently, this research is free from harmful content.

Acknowledgements

We thank the anonymous reviewers, area chair, and senior area chair for their valuable comments and suggestions. Their feedback has strengthened this work, encouraged its publication, inspired new research directions, and motivated us.

At the time of preparing this camera-ready version, my (first author) grandmother passed away. I am writing this final revision while on my way to her funeral. Amid the heated academic competition, this moment reminded me of the unwavering support of my family, which had slowly faded from my awareness. I am deeply grateful for their support and I hope to carry her spirit with me as I continue my academic journey. Thank you, and may you rest in eternal peace.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. Preprint, arXiv:2404.14219.

Nura Aljaafari, Danilo S. Carvalho, and André Freitas. 2024. Interpreting token compositionality in llms: A robustness analysis. Preprint, arXiv:2410.12924.

Nura Aljaafari, Danilo S. Carvalho, and André Freitas. 2025. Carma: Enhanced compositionality in llms via advanced regularisation and mutual information alignment. Preprint, arXiv:2502.11066.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In Computer Vision – ECCV 2016, pages 382–398, Cham. Springer International Publishing.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

John A. Bargh and Tanya L. Chartrand. 2000. The mind in the middle: A practical guide to priming and automaticity research. In Heinrich T. Reis and Chris M. Judd, editors, Handbook of research methods in social and personality psychology, pages 253–285. New York: Cambridge.

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 5185–5198, Online. Association for Computational Linguistics.

Thomas G. Bever. 2013. The cognitive basis for linguistic structures. In Language Down the Garden Path: The Cognitive and Biological Basis for Linguistic Structures. Oxford University Press.

Joan Bresnan. 2001. Lexical-Functional Syntax. Blackwell Publishers, Oxford, UK.

Natalie Grace Brigham, Chongjiu Gao, Tadayoshi Kohno, Franziska Roesner, and Niloofar Mireshghallah. 2024. Developing story: Case studies of generative ai's use in journalism. Preprint, arXiv:2406.13706.

Joan Bybee. 2006. From usage to grammar: The mind's response to repetition, volume 82 of Language. De Gruyter Mouton, Berlin, Boston. 2010.

Joan Bybee. 2010. Language, Usage and Cognition. Cambridge University Press.

Qi Cao, Takeshi Kojima, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Unnatural error correction: GPT-4 can almost perfectly handle unnatural scrambled text. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 8898–8913, Singapore. Association for Computational Linguistics.

Xiuying Chen, Mingzhe Li, Shen Gao, Zhangming Chan, Dongyan Zhao, Xin Gao, Xiangliang Zhang, and Rui Yan. 2023. Follow the timeline! generating an abstractive and extractive timeline summary in chronological order. ACM Trans. Inf. Syst., 41(1).

Noam Chomsky. 1965. Aspects of the Theory of Syntax. The MIT Press, Cambridge.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Tao He, Haotian Wang, Weihua Peng, Ming Liu, Bing Qin, and Ting Liu. 2024. Navigate through enigmatic labyrinth a survey of chain

of thought reasoning: Advances, frontiers and future. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1173–1203, Bangkok, Thailand. Association for Computational Linguistics.

Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A. Smith. 2021. All that's 'human' is not gold: Evaluating human evaluation of generated text. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 7282–7296, Online. Association for Computational Linguistics.

Wanqing Cui, Keping Bi, Jiafeng Guo, and Xueqi Cheng. 2024. MORE: Multi-mOdal REtrieval augmented generative commonsense reasoning. In Findings of the Association for Computational Linguistics: ACL 2024, pages 1178–1192, Bangkok, Thailand. Association for Computational Linguistics.

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. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. Preprint, arXiv:2501.12948.

Hiroyuki Deguchi, Yusuke Sakai, Hidetaka Kamigaito, and Taro Watanabe. 2024a. mbrs: A library for minimum Bayes risk decoding. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 351–362, Miami, Florida, USA. Association for Computational Linguistics.

Hiroyuki Deguchi, Yusuke Sakai, Hidetaka Kamigaito, Taro Watanabe, Hideki Tanaka, and Masao Utiyama. 2024b. Centroid-based efficient minimum Bayes risk decoding. In Findings of the Association for Computational Linguistics: ACL 2024, pages 11009–11018, Bangkok, Thailand. Association for Computational Linguistics.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Llm.int8(): 8-bit matrix multiplication for transformers at scale. In Advances in Neural Information Processing Systems.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. In Thirty-seventh Conference on Neural Information Processing Systems.

Tim Dettmers and Luke Zettlemoyer. 2023. The case for 4-bit precision: k-bit inference scaling laws. In Proceedings of the 40th International Conference on Machine Learning, ICML'23. JMLR.org.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 516 others. 2024. The llama 3 herd of models. Preprint, arXiv:2407.21783.

Haoshen Fan, Jie Wang, Bojin Zhuang, Shaojun Wang, and Jing Xiao. 2019. A hierarchical attention based seq2seq model for chinese lyrics generation. In PRICAI 2019: Trends in Artificial Intelligence: 16th Pacific Rim International Conference on Artificial Intelligence, Cuvu, Yanuca Island, Fiji, August 26-30, 2019, Proceedings, Part III, page 279–288, Berlin, Heidelberg. Springer-Verlag.

Fernanda Ferreira and Paul E. Engelhardt. 2006. Chapter 3 - syntax and production. In Matthew J. Traxler and Morton A. Gernsbacher, editors, Handbook of Psycholinguistics (Second Edition), second edition edition, pages 61–91. Academic Press, London.

Deepanway Ghosal, Yew Ken Chia, Navonil Majumder, and Soujanya Poria. 2023. Flacuna: Unleashing the problem solving power of vicuna using flan finetuning. Preprint, arXiv:2307.02053.

A.E. Goldberg. 1995. Constructions: A Construction Grammar Approach to Argument Structure. Cognitive Theory of Language and Culture Series. University of Chicago Press.

Takumi Goto, Yusuke Sakai, and Taro Watanabe. 2025a. gec-metrics: A unified library for grammatical error correction evaluation. Preprint, arXiv:2505.19388.

Takumi Goto, Yusuke Sakai, and Taro Watanabe. 2025b. Rethinking evaluation metrics for grammatical error correction: Why use a different evaluation process than human? Preprint, arXiv:2502.09416.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Yuanzhuo Wang, and Jian Guo. 2024. A survey on llm-as-a-judge. Preprint, arXiv:2411.15594.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Jinyi Hu and Maosong Sun. 2020. Generating major types of Chinese classical poetry in a uniformed framework. In Proceedings of the Twelfth Language Resources and Evaluation Conference, pages 4658–4663, Marseille, France. European Language Resources Association.

EunJeong Hwang, Veronika Thost, Vered Shwartz, and Tengfei Ma. 2023. Knowledge graph compression enhances diverse commonsense generation. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 558–572, Singapore. Association for Computational Linguistics.

Yusuke Ide, Yuto Nishida, Justin Vasselli, Miyu Oba, Yusuke Sakai, Hidetaka Kamigaito, and Taro Watanabe. 2025. How to make the most of LLMs' grammatical knowledge for acceptability judgments. In Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 7416–7432, Albuquerque, New Mexico. Association for Computational Linguistics.

Mete Ismayilzada, Antonio Laverghetta Jr., Simone A. Luchini, Reet Patel, Antoine Bosselut, Lonneke van der Plas, and Roger Beaty. 2025. Creative preference optimization. Preprint, arXiv:2505.14442.

Ray Jackendoff. 2002. Foundations of Language: Brain, Meaning, Grammar, Evolution. Oxford University Press.

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. 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, and 7 others. 2024. Mixtral of experts. Preprint, arXiv:2401.04088.

Yuu Jinnai, Ukyo Honda, Tetsuro Morimura, and Peinan Zhang. 2024. Generating diverse and high-quality texts by minimum Bayes risk decoding. In Findings of the Association for Computational Linguistics: ACL 2024, pages 8494–8525, Bangkok, Thailand. Association for Computational Linguistics.

Jongeun Kim, MinChung Kim, and Taehwan Kim. 2023. Effective slogan generation with noise perturbation. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM '23, page 3998–4002, New York, NY, USA. Association for Computing Machinery.

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

Fajri Koto, Haonan Li, Sara Shatnawi, Jad Doughman, Abdelrahman Sadallah, Aisha Alraeesi, Khalid Almubarak, Zaid Alyafeai, Neha Sengupta, Shady Shehata, Nizar Habash, Preslav Nakov, and Timothy Baldwin. 2024. ArabicMMLU: Assessing massive multitask language understanding in Arabic. In Findings of the Association for Computational Linguistics: ACL 2024, pages 5622–5640, Bangkok, Thailand. Association for Computational Linguistics.

Brenden M. Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. Preprint, arXiv:1711.00350.

Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. 2017. Building machines that learn and think like people. Behavioral and Brain Sciences, 40:e253.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, and 4 others. 2024. Tülu 3: Pushing frontiers in open language model post-training. Preprint, arXiv:2411.15124.

Kyungjun Lee, Abhinav Shrivastava, and Hernisa Kacorri. 2023. Leveraging hand-object interactions in assistive egocentric vision. IEEE Trans. Pattern Anal. Mach. Intell., 45(6):6820–6831.

Zeyang Lei, Chao Zhang, Xinchao Xu, Wenquan Wu, Zheng-yu Niu, Hua Wu, Haifeng Wang, Yi Yang, and Shuanglong Li. 2022. PLATO-ad: A unified advertisement text generation framework with multi-task prompt learning. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 512–520, Abu Dhabi, UAE. Association for Computational Linguistics.

Willem J. M. Levelt. 1989. Speaking: From Intention to Articulation. The MIT Press.

Hao Li, Chenghao Yang, An Zhang, Yang Deng, Xiang Wang, and Tat-Seng Chua. 2024a. Hello again! llm-powered personalized agent for long-term dialogue. Preprint, arXiv:2406.05925.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2024b. 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.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 110–119, San Diego, California. Association for Computational Linguistics.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 1823–1840, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In Text Summarization Branches Out, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Fangru Lin, Emanuele La Malfa, Valentin Hofmann, Elle Michelle Yang, Anthony Cohn, and Janet B. Pierrehumbert. 2024. Graph-enhanced large language models in asynchronous plan reasoning. Preprint, arXiv:2402.02805.

Chenzhengyi Liu, Jie Huang, Kerui Zhu, and Kevin Chen-Chuan Chang. 2023. DimonGen: Diversified generative commonsense reasoning for explaining concept relationships. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4719–4731, Toronto, Canada. Association for Computational Linguistics.

Peiyu Liu, Zikang Liu, Ze-Feng Gao, Dawei Gao, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2024a. Do emergent abilities exist in quantized large language models: An empirical study. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pages 5174–5190, Torino, Italia. ELRA and ICCL.

Yixin Liu, Kejian Shi, Alexander R. Fabbri, Yilun Zhao, Peifeng Wang, Chien-Sheng Wu, Shafiq Joty, and Arman Cohan. 2024b. Reife: Re-evaluating instruction-following evaluation. Preprint, arXiv:2410.07069.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning. Preprint, arXiv:2301.13688.

Zhicong Lu, Li Jin, Guangluan Xu, Linmei Hu, Nayu Liu, Xiaoyu Li, Xian Sun, Zequn Zhang, and Kaiwen Wei. 2023. Narrative order aware story generation via bidirectional pretraining model with optimal transport reward. In Findings of the Association for Computational Linguistics: EMNLP 2023, pages 6274–6287, Singapore. Association for Computational Linguistics.

Gary Lupyan and Andy Clark. 2015. Words and the world: Predictive coding and the language-perception-cognition interface. Current Directions in Psychological Science, 24(4):279–284.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In Thirty-seventh Conference on Neural Information Processing Systems.

Koki Maeda, Masahiro Kaneko, and Naoaki Okazaki. 2022. IMPARA: Impact-based metric for GEC using parallel data. In Proceedings of the 29th International Conference on Computational Linguistics, pages 3578–3588, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Mana Makinae, Yusuke Sakai, Hidetaka Kamigaito, and Taro Watanabe. 2024. Simul-MuST-C: Simultaneous multilingual speech translation corpus using large language model. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 22185–22205, Miami, Florida, USA. Association for Computational Linguistics.

Timothy P. McNamara. 2005. Semantic priming: Perspectives from memory and word recognition.

Masato Mita, Soichiro Murakami, Akihiko Kato, and Peinan Zhang. 2024. Striking gold in advertising: Standardization and exploration of ad text generation. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 955–972, Bangkok, Thailand. Association for Computational Linguistics.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 839–849, San Diego, California. Association for Computational Linguistics.

Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irwanto, Jessica Pan, Faiaz Rahman, Ahmad Zaidi, Mutethia Mutuma, Yasin Tarabar, Ankit Gupta, Tao Yu, and 5 others. 2021. DART: Open-domain structured data record to text generation. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 432–447, Online. Association for Computational Linguistics.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert,

Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, and 21 others. 2025. 2 olmo 2 furious. Preprint, arXiv:2501.00656.

OpenAI. 2024. Openai o1-mini: Advancing cost-efficient reasoning. https://openai.com/index/openai-o1-mini-advancing-cost-efficient-reasoning/. Accessed: 2024-12-15.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. Preprint, arXiv:2303.08774.

Zebin Ou, Meishan Zhang, and Yue Zhang. 2022. On the role of pre-trained language models in word ordering: A case study with BART. In Proceedings of the 29th International Conference on Computational Linguistics, pages 6516–6529, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In Advances in Neural Information Processing Systems, volume 35, pages 27730–27744. Curran Associates, Inc.

Thomas Palmeira Ferraz, Kartik Mehta, Yu-Hsiang Lin, Haw-Shiuan Chang, Shereen Oraby, Sijia Liu, Vivek Subramanian, Tagyoung Chung, Mohit Bansal, and Nanyun Peng. 2024. LLM self-correction with DeCRIM: Decompose, critique, and refine for enhanced following of instructions with multiple constraints. In Findings of the Association for Computational Linguistics: EMNLP 2024, pages 7773–7812, Miami, Florida, USA. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Adam Pawłowski and Tomasz Walkowiak. 2024. NLP for digital humanities: Processing chronological text corpora. In Proceedings of the 4th International Conference on Natural Language Processing for Digital Humanities, pages 105–112, Miami, USA. Association for Computational Linguistics.

Soon Chang Poh, Sze Jue Yang, Jeraelyn Ming Li Tan, Lawrence Leroy Tze Yao Chieng, Jia Xuan Tan, Zhenyu Yu, Foong Chee Mun, and Chee Seng Chan. 2024. MalayMMLU: A multitask benchmark for the low-resource Malay language. In Findings of the Association for Computational Linguistics: EMNLP 2024, pages 650–669, Miami, Florida, USA. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In Proceedings of the Third Conference on Machine Translation: Research Papers, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Cheng Qian, Bingxiang He, Zhong Zhuang, Jia Deng, Yujia Qin, Xin Cong, Zhong Zhang, Jie Zhou, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. Tell me more! towards implicit user intention understanding of language model driven agents. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1088–1113, Bangkok, Thailand. Association for Computational Linguistics.

Tao Qian, Fan Lou, Jiatong Shi, Yuning Wu, Shuai Guo, Xiang Yin, and Qin Jin. 2023. UniLG: A unified structure-aware framework for lyrics generation. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 983–1001, Toronto, Canada. Association for Computational Linguistics.

Qwen, :, 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, and 24 others. 2024. Qwen2.5 technical report. Preprint, arXiv:2412.15115.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9.

Vikas Raunak and Arul Menezes. 2022. Finding memo: Extractive memorization in constrained sequence generation tasks. In Findings of the Association for Computational Linguistics: EMNLP 2022, pages 5153–5162, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ehud Reiter. 2018. A structured review of the validity of BLEU. Computational Linguistics, 44(3):393–401.

Yusuke Sakai, Takumi Goto, and Taro Watanabe. 2025. Impara-ged: Grammatical error detection is boosting reference-free grammatical error quality estimator. Preprint, arXiv:2506.02899.

Yusuke Sakai, Hidetaka Kamigaito, and Taro Watanabe. 2024a. mCSQA: Multilingual commonsense reasoning dataset with unified creation strategy by language models and humans. In Findings of the Association for Computational Linguistics: ACL 2024, pages 14182–14214, Bangkok, Thailand. Association for Computational Linguistics.

Yusuke Sakai, Mana Makinae, Hidetaka Kamigaito, and Taro Watanabe. 2024b. Simultaneous interpretation corpus construction by large language models in distant language pair. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 22375–22398, Miami, Florida, USA. Association for Computational Linguistics.

Yusuke Sakai, Adam Nohejl, Jiangnan Hang, Hidetaka Kamigaito, and Taro Watanabe. 2024c. Toward the evaluation of large language models considering score variance across instruction templates. In Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP, pages 499–529, Miami, Florida, US. Association for Computational Linguistics.

Md Sadman Sakib and Yu Sun. 2023. From cooking recipes to robot task trees – improving planning correctness and task efficiency by leveraging llms with a knowledge network. Preprint, arXiv:2309.09181.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, and 21 others. 2022. Multitask prompted training enables zero-shot task generalization. In International Conference on Learning Representations.

Patricia Schmidtova, Saad Mahamood, Simone Balloccu, Ondrej Dusek, Albert Gatt, Dimitra Gkatzia, David M. Howcroft, Ondrej Platek, and Adarsa Sivaprasad. 2024. Automatic metrics in natural language generation: A survey of current evaluation practices. In Proceedings of the 17th International Natural Language Generation Conference, pages 557–583, Tokyo, Japan. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7881–7892, Online. Association for Computational Linguistics.

Jaehyung Seo, Seounghoon Lee, Chanjun Park, Yoonna Jang, Hyeonseok Moon, Sugyeong Eo, Seonmin Koo, and Heuiseok Lim. 2022. A dog is passing over the jet? a text-generation dataset for Korean commonsense reasoning and evaluation. In Findings of the Association for Computational Linguistics: NAACL 2022, pages 2233–2249, Seattle, United States. Association for Computational Linguistics.

Lei Sha. 2020. Gradient-guided unsupervised lexically constrained text generation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 8692–8703, Online. Association for Computational Linguistics.

Mandar Sharma, Rutuja Taware, Pravesh Koirala, Nikhil Muralidhar, and Naren Ramakrishnan. 2024. Laying anchors: Semantically priming numerals in language modeling. In Findings of the Association for Computational Linguistics: NAACL 2024, pages 2653–2660, Mexico City, Mexico. Association for Computational Linguistics.

Tianxiao Shen, Myle Ott, Michael Auli, and Marc'Aurelio Ranzato. 2019. Mixture models for diverse machine translation: Tricks of the trade. In Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pages 5719–5728. PMLR.

Harmanpreet Singh, Nikhil Verma, Yixiao Wang, Manasa Bharadwaj, Homa Fashandi, Kevin Ferreira, and Chul Lee. 2024. Personal large language model agents: A case study on tailored travel planning. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 486–514, Miami, Florida, US. Association for Computational Linguistics.

Mark Steedman. 2000. The Syntactic Process. MIT Press.

Kevin Stowe, Debanjan Ghosh, and Mengxuan Zhao. 2022. Controlled language generation for language learning items. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 294–305, Abu Dhabi, UAE. Association for Computational Linguistics.

Mirac Suzgun, Luke Melas-Kyriazi, and Dan Jurafsky. 2023. Follow the wisdom of the crowd: Effective text generation via minimum Bayes risk decoding. In Findings of the Association for Computational Linguistics: ACL 2023, pages 4265–4293, Toronto, Canada. Association for Computational Linguistics.

Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024. Large language models for data annotation and synthesis: A survey. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 930–957, Miami, Florida, USA. Association for Computational Linguistics.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, Xinyang Geng, Fred Alcober, Roy Frostig, Mark Omernick, Lexi Walker, Cosmin Paduraru, Christina Sorokin, and 1116 others. 2024a. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. Preprint, arXiv:2403.05530.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak

Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, and 179 others. 2024b. Gemma 2: Improving open language models at a practical size. Preprint, arXiv:2408.00118.

Lindia Tjuatja, Graham Neubig, Tal Linzen, and Sophie Hao. 2025. What goes into a LM acceptability judgment? rethinking the impact of frequency and length. In Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 2173–2186, Albuquerque, New Mexico. Association for Computational Linguistics.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In CVPR, pages 4566–4575. IEEE Computer Society.

Hongru Wang, Lingzhi Wang, Yiming Du, Liang Chen, Jingyan Zhou, Yufei Wang, and Kam-Fai Wong. 2023a. A survey of the evolution of language model-based dialogue systems. Preprint, arXiv:2311.16789.

Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023b. Is ChatGPT a good NLG evaluator? a preliminary study. In Proceedings of the 4th New Frontiers in Summarization Workshop, pages 1–11, Singapore. Association for Computational Linguistics.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023c. Self-instruct: Aligning language models with self-generated instructions. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. Transactions of the Association for Computational Linguistics, 7:625–641.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022a. Finetuned language models are zero-shot learners. In International Conference on Learning Representations.

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 Advances in Neural Information Processing Systems.

Brandon T. Willard and Rémi Louf. 2023. Efficient guided generation for large language models. Preprint, arXiv:2307.09702.

Genta Indra Winata, Hanyang Zhao, Anirban Das, Wenpin Tang, David D. Yao, Shi-Xiong Zhang, and Sambit Sahu. 2024. Preference tuning with human feedback on language, speech, and vision tasks: A survey. Preprint, arXiv:2409.11564.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 38–45, Online. Association for Computational Linguistics.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 technical report. Preprint, arXiv:2407.10671.

Haoran Yang, Yan Wang, Piji Li, Wei Bi, Wai Lam, and Chen Xu. 2023. Bridging the gap between pre-training and fine-tuning for commonsense generation. In Findings of the Association for Computational Linguistics: EACL 2023, pages 376–383, Dubrovnik, Croatia. Association for Computational Linguistics.

Zihao Yi, Jiarui Ouyang, Yuwen Liu, Tianhao Liao, Zhe Xu, and Ying Shen. 2024. A survey on recent advances in llm-based multi-turn dialogue systems. Preprint, arXiv:2402.18013.

Ryoma Yoshimura, Masahiro Kaneko, Tomoyuki Kajiwara, and Mamoru Komachi. 2020. SOME: Reference-less sub-metrics optimized for manual evaluations of grammatical error correction. In Proceedings of the 28th International Conference on Computational Linguistics, pages 6516–6522, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Chengyue Yu, Lei Zang, Jiaotuan Wang, Chenyi Zhuang, and Jinjie Gu. 2024. CharPoet: A Chinese classical poetry generation system based on token-free LLM. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), pages 315–325, Bangkok, Thailand. Association for Computational Linguistics.

Wenhao Yu, Chenguang Zhu, Lianhui Qin, Zhihan Zhang, Tong Zhao, and Meng Jiang. 2022. Diversifying content generation for commonsense reasoning with mixture of knowledge graph experts. In Findings of the Association for Computational Linguistics: ACL 2022, pages 1896–1906, Dublin, Ireland. Association for Computational Linguistics.

Ran Zhang and Steffen Eger. 2024. Llm-based multi-agent poetry generation in non-cooperative environments. Preprint, arXiv:2409.03659.

Tianhui Zhang, Danushka Bollegala, and Bei Peng. 2023. Learning to predict concept ordering for common sense generation. In Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 2: Short Papers), pages 10–19, Nusa Dua, Bali. Association for Computational Linguistics.

Tianhui Zhang, Bei Peng, and Danushka Bollegala. 2024. Improving diversity of commonsense generation by large language models via in-context learning. In Findings of the Association for Computational Linguistics: EMNLP 2024, pages 9226–9242, Miami, Florida, USA. Association for Computational Linguistics.

Chao Zhao, Faeze Brahman, Tenghao Huang, and Snigdha Chaturvedi. 2022. Revisiting generative commonsense reasoning: A pre-ordering approach. In Findings of the Association for Computational Linguistics: NAACL 2022, pages 1709–1718, Seattle, United States. Association for Computational Linguistics.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024a. Wildchat: 1m chatGPT interaction logs in the wild. In The Twelfth International Conference on Learning Representations.

Yu Zhao, Huifeng Yin, Bo Zeng, Hao Wang, Tianqi Shi, Chenyang Lyu, Longyue Wang, Weihua Luo, and Kaifu Zhang. 2024b. Marco-o1: Towards open reasoning models for open-ended solutions. Preprint, arXiv:2411.14405.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. Fine-tuning language models from human preferences. Preprint, arXiv:1909.08593.

Marco Zorzi, Ivilin Peev Stoianov, and Carlo Umilta. 2004. Computational modeling of numerical cognition. The Handbook of Mathematical Cognition.

A Detailed LLMs Information

Table 5 shows the source information of each LLM. We used the Transformers (Wolf et al., 2020) and bitsandbytes (Dettmers et al., 2022) libraries for the inference of open LLMs. We used a single A6000 48GB GPU for most open LLMs, while Mistral-large was run on two A6000 48GB GPUs, and Llama3.1-405B was run on eight A6000 48GB GPUs. Note that we used only instruction-tuned LLMs because base models do not follow instructions precisely, and this aligns with our research purpose, which is to evaluate instruction-following capabilities. For instruction-tuned LLMs, we conducted inference by applying their specific chat templates. As a system prompt, we included the phrase: *"You are a helpful assistant. Please generate only an answer."*. Upon randomly sampling 50 outputs per model and manually inspecting them, we observed no notable generation of supplementary information or flavor text. Therefore, the generated text was directly used for evaluation. The same procedure was applied to proprietary models.

LLMs	HuggingFace ID / API Name
Llama3.2-1B	meta-llama/Llama-3.2-1B-Instruct
Llama3.2-3B	meta-llama/Llama-3.2-3B-Instruct
Llama3.1-8B	meta-llama/Meta-Llama-3.1-8B-Instruct
Llama3.1-70B	meta-llama/Meta-Llama-3.1-70B-Instruct
Llama3.3-70B	meta-llama/Llama-3.3-70B-Instruct
Llama3.1-405B	meta-llama/Meta-Llama-3.1-405B-Instruct
Qwen2-0.5B	Qwen/Qwen2-0.5B-Instruct
Qwen2-1.5B	Qwen/Qwen2-1.5B-Instruct
Qwen2-7B	Qwen/Qwen2-7B-Instruct
Qwen2-72B	Qwen/Qwen2-72B-Instruct
Gemma-2-2B	google/gemma-2-2b-it
Gemma-2-9B	google/gemma-2-9b-it
Gemma-2-27B	google/gemma-2-27b-it
Phi-3-mini	microsoft/Phi-3-mini-4k-instruct
Phi-3-small	microsoft/Phi-3-small-8k-instruct
Phi-3-medium	microsoft/Phi-3-medium-4k-instruct
Mistral-7B	mistralai/Mistral-7B-Instruct-v0.1
Mistral-small	mistralai/Mistral-Small-Instruct-2409
Mistral-large	mistralai/Mistral-Large-Instruct-2407
Mixtral-8x7B	mistralai/Mixtral-8x7B-Instruct-v0.1
Mixtral-8x22B	mistralai/Mixtral-8x22B-Instruct-v0.1
Qwen2.5-0.5B	Qwen/Qwen2.5-0.5B-Instruct
Qwen2.5-1.5B	Qwen/Qwen2.5-1.5B-Instruct
Qwen2.5-3B	Qwen/Qwen2.5-3B-Instruct
Qwen2.5-7B	Qwen/Qwen2.5-7B-Instruct
Qwen2.5-14B	Qwen/Qwen2.5-14B-Instruct
Qwen2.5-32B	Qwen/Qwen2.5-32B-Instruct
Qwen2.5-72B	Qwen/Qwen2.5-72B-Instruct
Tülu3-8B	allenai/Llama-3.1-Tulu-3-8B
Tülu3-70B	allenai/Llama-3.1-Tulu-3-70B
OMLo2-7B	allenai/OLMo-2-1124-7B
OMLo2-13B	allenai/OLMo-2-1124-13B
GPT-3.5	OpenAI/gpt-3.5-turbo-0125
GPT-4o	OpenAI/gpt-4o-2024-05-13
Gemini-Flash	Gemini/gemini-1.5-flash-001
Gemini-Pro	Gemini/gemini-1.5-pro-001

Table 5: Details of the LLMs for the experiments.

B Examples of Instruction Template

Table 6 shows the instruction template used in the Ordered CommonGen experiment framework. Each concept from the concepts set is assigned to

ID	Instruction Template
0	Concepts: {concepts} Write a sentence that includes all these words **in the specified order**. Answer:
1	Keywords: {concepts} What is a sentence that includes all these keywords **in the specified order**? Answer:
2	Here are some concepts: {concepts} What is a sentence about these concepts **in the specified order**? Answer:
3	Produce a sentence which mentions all of these concepts **in the specified order**: {concepts} Answer:
4	Write a sentence about the following things **in the specified order**: {concepts} Answer:
5	Generate a sentence that includes all the following words **in the specified order**: {concepts} Answer:

Table 6: The instruction templates for our experiments. We created it based on the FLAN template and inserted it **"in the specified order"** in the bold one. Each concept from the concepts set is assigned to the curly brackets as space-separated values.

the curly brackets **{concepts}** as space-separated values. Following the findings of Zhang et al. (2023), which demonstrated slightly better performance with space-separated values and were corroborated by our pilot study, we opted for space separation. This separation method is also reasonable from a mechanistic perspective: when considering the ID token sequence converted by tokenization of LLMs, separating concepts with commas or other delimiters introduces additional tokens between concepts, which could affect performance.

C Additional Discussions

C.1 Performance of Reasoning Models

Reasoning using Chain of Thought (CoT) (Wei et al., 2022b; Chu et al., 2024) prompts enhances the reasoning capabilities of LLMs. Recently, reasoning models (DeepSeek-AI et al., 2025; OpenAI, 2024) have emerged that achieve comparable capabilities through training rather than prompt-based control. Due to variations in CoT reasoning, unified evaluation is challenging; therefore, we evaluated using reasoning models. Table 7 compares the evaluation results for Qwen2-7B (Yang et al., 2024)[8] and its reasoning model Marco-o1 (Zhao et al., 2024b)[9], as well as GPT-4o-mini (OpenAI et al., 2024)[10] and its reasoning model o1-mini (OpenAI, 2024)[11][12]. As shown in Table 7, reasoning models, particularly o1-mini, demonstrate significant improvements in Ordered Rate compared to their base models. While Marco-o1 shows a decrease in scores like w/ order, its Ordered Rate improves, reflecting enhanced inference capabilities through effective reasoning. These results indicate that reasoning is effective; however, even these models do not guarantee 100% adherence. Thus, while reasoning enhances inductive generative commonsense reasoning in LLMs, a noticeable gap remains between these models and human capabilities in composing sentences with arbitrary orders.

	w/o order	w/ order	Ordered Rate
Qwen2-7B	**89.28**$_{(+5.16)}$	**34.22**$_{(+7.96)}$	38.33$_{(+7.11)}$
Macro-o1	79.88$_{(+6.88)}$	33.27$_{(+6.90)}$	**41.65**$_{(+5.53)}$
GPT-4o-mini	87.72$_{(+4.99)}$	43.27$_{(+16.58)}$	49.33$_{(+17.07)}$
o1-mini	**99.09**$_{(+2.80)}$	**84.70**$_{(+70.38)}$	**85.48**$_{(+70.60)}$

Table 7: Comparison of evaluation results for the generated sentences between each base model and reasoning model. We use the single template with ID 0 from Table 6 in Appendix B. We report the Concepts Coverage metrics, where higher scores indicate better performance, with performance differences from the original CommonGen, which does not consider concept order, shown in parentheses. The upper row of each section represents the base model, while the lower row represents its corresponding reasoning model.

C.2 Future Evaluation Direction

In this evaluation framework, as the primary focus of our study is on instruction-following capability, we prioritized self-evaluation-based methods. This

[8] https://hf.co/Qwen/Qwen2-7B-Instruct

[9] https://hf.co/AIDC-AI/Marco-o1

[10] gpt-4o-mini-2024-07-18

[11] o1-mini-2024-09-12

[12] We attempted experiments with o1 (o1-2024-09-12) but could not complete them due to financial cost constraints. However, we observed instances where the model generated outputs that did not follow the specified order. This suggests that the same limitations apply to o1, indicating that it does not yet guarantee 100% adherence to instructions.

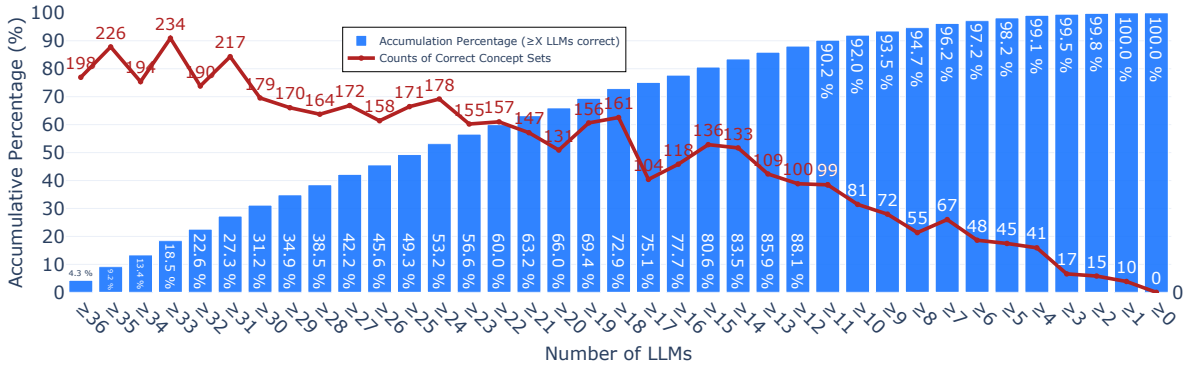

Figure 3: The distribution of all 4,608 concept sets successfully composed by all 36 LLMs using any of the six templates. The x-axis represents the number of LLMs that successfully composed given concept sets, while the y-axis shows the accumulative percentage. The line graph indicates the number of concept sets in each bin.

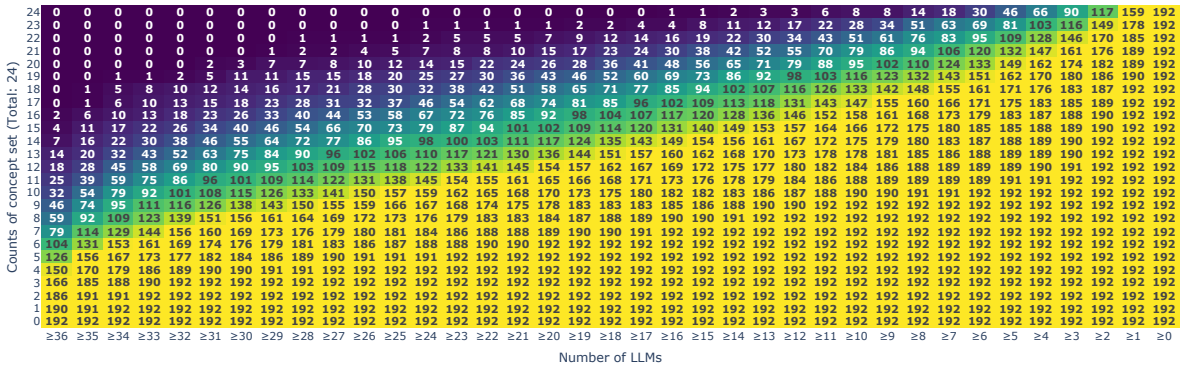

Figure 4: The distribution of successfully composed permutations of concept sets by all 36 LLMs using any of the six templates. The x-axis represents the number of LLMs that successfully composed each permuted concept set, while the y-axis shows the counts of concept sets. A total of 192 unique concept sets is evaluated, with a maximum of 24 permutations per set ($4! = 24$). Black numbers represent the majority of concept sets, while white numbers indicate the minority.

allowed us to effectively assess model performance even in open-ended generation tasks. However, if the goal shifts toward generating more semantically and syntactically natural sentences, additional evaluation methods may be needed. For example, it will be important to estimate grammatical quality by detecting grammatical errors (Sakai et al., 2025; Goto et al., 2025a,b; Maeda et al., 2022; Yoshimura et al., 2020), or to assess linguistic acceptabilities (Warstadt et al., 2019; Tjuatja et al., 2025; Ide et al., 2025). We share these as future directions and challenges toward advancing reference-free evaluation of text generation, as well as potential extensions of our Ordered CommonGen task.

C.3 Which Concept Sets Are Difficult for LLMs to Compose?

Figure 3 shows the distribution of the number and proportion of LLMs that successfully composed sentences from each of the 4,608 concept sets in Ordered CommonGen. As shown in Figure 3, every concept set was composed by at least one LLM. These results reveal that while at least one LLM can compose sentences using the given concepts in the specified order, compositional generalization abilities vary significantly across models, indicating instability in this capability. Figure 4 shows which permutations of concept sets are the most difficult for LLMs. The results indicate that at least one LLM was able to compose sentences for all permutations of 159 concept sets. However, for 33 concept sets, no LLM could successfully compose sentences for any of their permutations. Notably, if a success rate of 75% (18 out of 24 permutations) is acceptable, at least one LLM could successfully compose sentences for all concept sets. These findings suggest that while LLMs can compose sentences for a wide range of concept order permutations, they do not consistently adhere to instructions across all permutations, further highlighting the overall instability and limitations in compositional generalization.