The Mystery of Compositional Generalization in Graph-based Generative Commonsense Reasoning

Xiyan Fu
Dept. of Computational Linguistics
Heidelberg University
fu@cl.uni-heidelberg.de

Anette Frank
Dept. of Computational Linguistics
Heidelberg University
frank@cl.uni-heidelberg.de

Abstract

While LLMs have emerged as performant architectures for reasoning tasks, their compositional generalization capabilities have been questioned. In this work, we introduce a Compositional Generalization Challenge for Graph-based Commonsense Reasoning (CGGC) that goes beyond previous evaluations that are based on sequences or tree structures – and instead involves a reasoning graph: It requires models to generate a natural sentence based on given concepts and a corresponding reasoning graph, where the presented graph involves a previously unseen combination of relation types. To master this challenge, models need to learn how to reason over relation tuples within the graph, and how to compose them when conceptualizing a verbalization. We evaluate seven well-known LLMs using in-context learning and find that performant LLMs still struggle in compositional generalization. We investigate potential causes of this gap by analyzing the structures of reasoning graphs, and find that different structures present varying levels of difficulty for compositional generalization. Arranging the order of demonstrations according to the structures' difficulty shows that organizing samples in an easy-to-hard schema enhances the compositional generalization ability of LLMs. [1]

1 Introduction

Reasoning (Brachman and Levesque, 2004) has been widely explored and extended to a wide variety of situations involving logical or commonsense reasoning (Rashkin et al., 2018; Talmor et al., 2019; Bisk et al., 2020). Recently, LLMs such as GPT-3 (Brown et al., 2020) and Llama2 (Touvron et al., 2023) have demonstrated astonishing performance on reasoning tasks (Lourie et al., 2021).

However, existing works found that LLMs are limited in scenarios that require generalization abilities, such as out-of-domain (Shen and Kejriwal, 2021; Wang et al., 2021), low-resource (Klein and Nabi, 2021) and complex compositional (Dziri et al., 2024) tasks. Hupkes et al. (2023) concluded that inferior performance of models in such cases originates from a lack of compositional generalization ability – the ability to infer, from known components, a potentially infinite number of novel combinations suitable to solve the given task. With this ability, LLMs are expected to generalize to unseen and more complex reasoning scenarios without relying on large amounts of training instances.

To explore the compositional generalization abilities of LLMs in reasoning, existing works introduce benchmarks across various domains involving different data representations, such as natural language (Liu et al., 2022a; Yanaka et al., 2020; Fu and Frank, 2023) and tree-based structures (Saparov et al., 2023; Fu and Frank, 2024a; Kudo et al., 2023). These works facilitate the compositional generalization exploration in reasoning and have shown that LLMs are able to generalize to some extent, while being limited in specific circumstances. However, to date, we note a gap regarding the evaluation of compositional generalization abilities in the context of graph-based reasoning. Graphs, as commonly used in real-world applications, offer flexible and diverse reasoning paths. Recent evidence (Besta et al., 2024) suggests that graphs enhance LLM reasoning by enabling the use and combination of diverse reasoning paths.

Our work fills this gap by proposing CGGC, a Compositional Generalization Challenge for Graph-based Commonsense Reasoning. CGGC builds on the generative commonsense reasoning task CommonGen (Lin et al., 2020), which tasked models to generate a coherent natural language sentence from a *set of given concepts*. CGGC extends this task by requiring models to reason over a set of concepts presented in a connected graph structure. Fig.1.b shows an example where a model is expected to generate a sentence such as 'He puts a

[1] Code & data: https://github.com/Heidelberg-NLP/CGGC

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 8376–8394

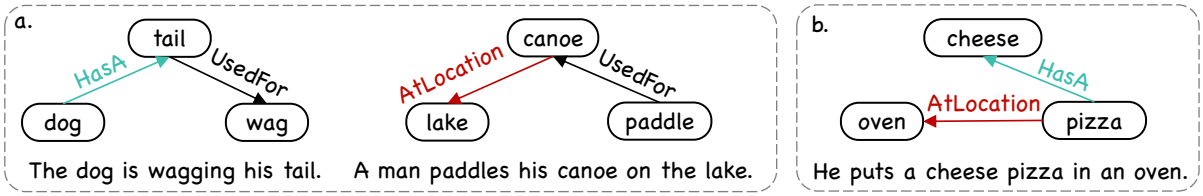

Figure 1: An instance of Compositional Generalization in Graph-based Commonsense Reasoning (CGGC). A model is expected to solve a test sample (b, *composition*) that presents an input graph with an *unseen* combination of relation types (here: HasA&AtLocation). The ICL demonstrations of the task in (a), by contrast, show each relation primitive in combination with other relation types, here: HasA&UsedFor and AtLocation&UsedFor.

cheese pizza in an oven' from a knowledge graph that contains the set of target concepts {cheese, oven, pizza}. For this task, we design the compositional generalization test CGGC: The core idea of the CGGC challenge is to probe models on specific relationship compositions that have not previously been seen in the learning sets. Fig. 1 illustrates how an unseen combination of relations types must be jointly verbalized in a sentence, whereas each of the primitives has been seen in combination with other relation types. In the example, the model is required to reason over an unseen relation combination, here, AtLocation&HasA – while each of these primitive relations has been seen in combination with other relation types, here: AtLocation in AtLocation&UsedFor, and HasA in HasA&UsedFor.

With the CGGC challenge, we systematically measure a model's compositional generalization ability in graph-based commonsense reasoning in an in-context learning (ICL) regime (Brown et al., 2020). Empirical results for seven well-known LLMs reveal challenges in compositionally generalizing to novel subgraph configurations. We analyze the factors that impact compositional generalization in such cases by *examining error trends* that change as a function of: i) *compositions*. Focusing on the *structure* of compositional reasoning graphs, we identify different schemas that result from composing primitive relations. Experiments show varying performance across different graph structures. E.g., relation compositions with uniform directionality seem more straightforward compared to compositions that end in a common target node, or that start from a common node but end in distinct nodes (Fig. 3). ii) *primitives*. We analyze performances based on the distribution of primitives according to different relation types. We find that LLMs more easily generalize to compositions involving common and frequent primitive relations.

Given the observed performance variations, we arrange the order of presentation of graph structures in ICL demonstrations according to their degree of difficulty. Results indicate that ordering ICL demonstrations in an easy-first manner enhances the models' compositional generalization ability.

2 Related Work

Analyzing Commonsense Reasoning Existing analyses of commonsense reasoning focus on representation (Zhou et al., 2020; Su et al., 2022), interpretability (Rajani et al., 2019), bias (Sotnikova et al., 2021; An et al., 2023) and consistency (Maler, 2023). Further, Davison et al. (2019); Petroni et al. (2019); Singh et al. (2023) probed language models for commonsense knowledge. Others construct complex reasoning scenarios such as logical queries on commonsense knowledge graphs (Fang et al., 2024) and geometric knowledge reasoning (Ding et al., 2024). Commonsense reasoning has also been analyzed with downstream tasks, such as machine translation (He et al., 2020; Liu et al., 2023c), temporal question answering (Jain et al., 2023; Wenzel and Jatowt, 2023), etc.

Beyond these perspectives, generalization is another important research direction. Existing works have shown that language models suffer from overfitting and are limited in generalization to out-of-domain examples (Sen and Saffari, 2020; Kejriwal and Shen, 2020), novel answers (Ma et al., 2021), and various tasks (Zhang et al., 2023b). In addition, Shwartz and Choi (2020) found that LLMs tend to overestimate and amplify biases in training data. Our work extends the research and analyses of generalization in commonsense reasoning, from the perspective of compositional generalization.

Compositional Generalization Despite the success of LLMs on downstream tasks, their compositional generalization abilities are poorly under-

Task & Works	Examples	Rep
Question Answering Liu et al. (2022a)	train: Cow is a national animal of which country? When did pandas come to USA? test: Panda is a national animal of which country?	natural language
NLI Yanaka et al. (2020) Fu and Frank (2023)	train: He realizes a woman is smiling → A woman is smiling A woman is smiling ↛ A man is smiling test: He realizes a woman is smiling ↛ A man is smiling.	natural language
Deductive Reasoning Fu and Frank (2024a) Saparov et al. (2024)	train: Alex is a dog. All dogs are mammals. → Alex is a mammal. Fae is a cat. Fae is soft. → Fae is soft and a cat. test: Alex is a dog. All dogs are mammals. Alex is not mean. → Alex is a mammal and not mean.	tree
Commonsense Reasoning (ours)	train: (dog, tail, HasA), (tail, wag, UsedFor) → The dog is wagging his tail. (paddle, lake, AtLoc), (paddle, canoe, UsedFor) → A man paddles his canoe on the lake. test: (cheese, pizza, AtLoc), (pizza, cheese, HasA) → He puts a cheese pizza in an oven.	graph

Table 1: Comparison of tasks exploring compositionality in reasoning. 'Rep' shows the format of compositions.

stood (Fodor and Pylyshyn, 1988; Lake et al., 2017; Hupkes et al., 2020). Prior works have evaluated aspects of compositionality in PLMs in semantic parsing (Lake and Baroni, 2018; Kim and Linzen, 2020; Qiu et al., 2022b), machine translation (Li et al., 2021; Dankers et al., 2022), image caption generation (Nikolaus et al., 2019; Bogin et al., 2021), etc., concluding that state-of-the-art PLMs are still not able to perform compositional generalization. To solve the issue, many approaches have been proposed, including data augmentation (Qiu et al., 2022a; Levy et al., 2023), specialized architectures (Zheng and Lapata, 2021; Herzig and Berant, 2021; Fu and Frank, 2024a), meta-learning (Conklin et al., 2021; Lake and Baroni, 2023), etc.

Recently, the exploration of compositional generalization in *reasoning* has attracted increasing attention. Existing works measure the compositional generalization abilities of models on reasoning tasks such as question answering (Liu et al., 2022a), deductive reasoning (Saparov et al., 2023), natural language inference (Yanaka et al., 2020; Fu and Frank, 2023), and arithmetic reasoning (Kudo et al., 2023). Our study differs from prior work in terms of representation types. We focus on the compositionality of reasoning on graph-based representations, which could facilitate complex reasoning by offering diverse reasoning paths (Besta et al., 2024). Table 1 shows an overview of tasks with their underlying representations. Our work is related to Xu et al. (2023), who propose to cluster predicates for compositional data-to-text generation, and who further test on compositions with *more* predicates in novel domains. In contrast, we focus on *novel* compositions by *recombining* known relations, serving as acomplementary work.

Unlike prior work we conduct detailed analyses of compositional graph structures, hence our findings can facilitate future reasoning tasks.

Generative Commonsense Reasoning The CommonGen task proposed by Lin et al. (2021), aims to advance machine commonsense towards generative reasoning abilities. Based on this benchmark, prior works have improved generation quality by incorporating explicit knowledge (Liu et al., 2021, 2022b) and visualizing relational scenes (Wang et al., 2022). More recent work focused on enhancing diversity (Liu et al., 2023a; Zhang et al., 2024; Jinnai et al., 2024) and robustness (Neerudu et al., 2023). The task has also been extended to other domains, such as testing negative knowledge (Chen et al., 2023) and visual commonsense generation (Tang et al., 2023; Cui et al., 2024). We complement these studies by providing a new perspective on the generalization ability of LLMs.

CommonGen and CGGC are grounded in the same dataset, but differ in motivation and research fields: CommonGen tasks a model to generate a natural language sentence based on a set of given concepts. It advances machine commonsense toward generative reasoning abilities. We extended this dataset with reasoning graphs, and split it elaborately to test for the compositional generalization abilities of models, in a generative graph-based commonsense reasoning task. Our benchmark CGGC offers the *first graph-based compositional generalization evaluation benchmark*, enriching the compositional generalization research field.

3 Defining a new CGGC Challenge

Our new challenge CGGC, aiming to test for Compositional Generalization abilities in Graph-based generative Commonsense Reasoning tasks, extends the existing CommonGen task of Lin et al. (2021). CGGC asks systems to generate a plausible natural language sentence given *a set of concepts* and

a reasoning graph that connects these concepts, showing how they can relate to each other. The sentence is expected to serve as a description covering commonsense relations between given concepts.

Given the novel task, we aim to explore whether and to what extent LLMs can perform compositional generalization – which consists in understanding and verbalizing novel *compositions* of previously seen *primitive relation types*, presented in a graph-structured representation. In our work we define a relation type in the reasoning graph as a primitive, and a compound that requires several relation types in the reasoning graph as a composition. Using {*oven, pizza, cheese*} in Fig. 1 as example, the relation compound AtLocation&HasA is regarded as a composition; its constituent relation types AtLocation and HasA are seen as primitives.

4 Datasets

In this section we introduce the benchmark for our novel CGGC challenge. It relies on the CommonGen dataset (Lin et al., 2020) and the commonsense resource ConceptNet (Speer et al., 2017) (Section 4.1). With these resources, we extend samples with reasoning graphs (Section 4.2), and further split the constructed data based on the compositional generalization features for evaluation (Section 4.3).

4.1 Dataset Pre-Processing

CommonGen (Lin et al., 2020) [2] tasks models to generate a coherent sentence given a set of common concepts. For example, {*tail, dog, wag*} → *The dog is wagging his tail*. The input is an unordered set of k concepts, denoted as $\mathcal{X} = \{c_1, c_2, ..., c_k\}$. Each concept c_i is a common object (noun) or action (verb), which is guaranteed to appear as a ConceptNet unigram (Speer et al., 2017). The expected output is a coherent sentence $\mathcal{Y}$ that describes a common scenario from daily life, using all given concepts in $\mathcal{X}$. Each reference consists of an average of 11 words and introduces about 3 new meaningful words not included in the given concept set (see the Appendix A.3 for details).

We enrich each concept set with a commonsense reasoning graph that provides related commonsense facts and relations. As knowledge resource, we choose ConceptNet, as it encompasses all candidate concepts. ConceptNet nodes represent concepts, and its edges provide commonsense knowledge relations between them, covering 34 relation types. [3] To facilitate the following compositionality evaluations, we merge some relations with similar meanings (e.g., {/r/InstanceOf, /r/MannerOf} → /r/IsA) and ignore some infrequent relations (e.g., /r/LocatedNear, /r/SymbolOf). We finally select 13 frequent relations following (Becker et al., 2021). More details are given in the Appendix A.1.

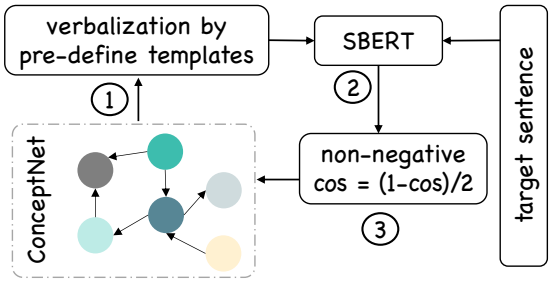

Figure 2: The process of constructing a reasoning graph given ConceptNet and a target sentence.

4.2 Graph Construction

To construct a ConceptNet-based reasoning graph that fits a given concept set and an associated target sentence y, following Plenz et al. (2023), we i) construct a similarity-weighted ConceptNet subgraph, that assigns weights to each triple based on their similarity to the reference sentence y, and ii) apply Dijkstra's algorithm (Dijkstra, 2022) to find weighted shortest paths between all concept pairs.

Fig. 2 illustrates this construction process: We verbalize the triplets in ConceptNet to sentences using pre-defined templates (①) and further encode these sentences using S(entence-)BERT (Reimers and Gurevych, 2019) (②). For instance, the triple (tail, UsedFor, wag) is verbalized as 'Tail is used for wag'. We also encode a target sentence in y using SBERT, to attain its representation. We calculate cosine similarity between all candidate triple representations and the target sentence representation. To provide non-negative weights for Dijkstra's algorithm, we transform the calculated cosine similarity into a non-negative value, signifying semantic dissimilarity between concepts (③). This conversion involves scaling the cosine similarity value to a new weight using the formula $\frac{1-\cos}{2}$. The resulting values serve as weights for all triples.

Hence, for each sample, we assign weights to triples based on the corresponding target sentences.

[2] We use the train and dev data together, and ignore the test data given it is unavailable for reasons of leaderboard testing.

[3] https://github.com/commonsense/conceptnet5/wiki/Relations

Concepts Sets	general			graph		
	#set	#sent	#len$_s$	#nodes	#edges	#rel
all	26819	55269	10.68	9.07	7.73	4.63
-size=3	19684	42747	10.16	8.31	7.01	4.41
-size=4	3947	8548	11.75	10.85	9.43	5.15
-size=5	3188	3972	13.89	13.33	11.81	5.76

Table 2: Data statistics for our novel CGGC benchmark. We provide details about concept sets, reference sentences s, and the original vs. extended reasoning graphs.

		sample	label
	test	{cheese, pizza, oven} → He puts a cheese pizza in an oven.	AtLocation-HasA
demonstrations	id	{cup, tea, table} → He puts a cup of tea on the table.	AtLocation-HasA
	ood	{tail, dog, wag} → The dog is wagging his tail.	UsedFor-HasA
		{lake, paddle, canoe} → A man paddles his canoe on the lake.	AtLocation-UsedFor

Table 3: Example of in-context learning evaluation. We show the test sample and corresponding in-distribution (id) and out-of-distribution (ood) demonstrations.

The ensuing shortest subgraph calculation minimizes the cumulative edge weights, effectively maximizing semantic similarity across concepts and references. Ultimately, each sample will be assigned a reasoning graph [4]. We conduct a human evaluation to assess the quality of the constructed graphs in Appendix B.

4.3 Compositional Generalization Label

To perform compositional generalization evaluation on our dataset, we assign a graph label to each sample. The label indicates the types of relations that will be needed to infer and generate a sentence that entails the given concept set. E.g., the graph label for the sample {*oven, cheese, pizza*} shown in Fig. 1 is AtLocation&HasA.

We define graph labels for all data samples and only select compositional labels which contain at least two distinct relations for compositional generalization tests. After filtering we select 468 compositional graph labels for our experiments. These labels are related to 26,819 examples in total, where each label corresponds to at least five samples. We categorize all selected data instances into three groups, based on the input concept set sizes. Table 2 provides statistics of the data, with further details about the extended reasoning graphs.[5]

5 Experimental Setup

5.1 Evaluated LLMs and Methods

We select 7 open access autoregressive LLMs: i) Llama2, Llama2-chat (Touvron et al., 2023); ii) Mistral-v0.1, Mistral-Instructv0.1 (Jiang et al., 2023); iii) Falcon, Falcon-instruct (Penedo et al., 2023); iv) GPT-J (Wang and Komatsuzaki, 2021). The first three model types include both *vanilla* pre-trained models and *instruct* models fine-tuned on instruction datasets. We will refer to all instruction tuning-based model variants as x-chat, e.g., Llama2-chat. For all LLMs, we choose their 7 billion versions (GTP-J is aligned with 6B) to avoid the model scale effects.

In addition, we evaluate GPT4 (Achiam et al., 2023), to also assess the compositional generalization abilities of a larger-scale LLMs.

In all experiments we use in-context learning with a fixed prompt as task adaptation technique, since existing works have proved its superior effectiveness in compositional generalization (Qiu et al., 2022b; Saparov et al., 2023).[6] We give four demonstrations per prompt [7] in our main experiments.

5.2 Evaluation Data Split

To conduct the compositional generalization evaluations, we control the dataset splits based on primitives and compositions. Specifically, we design two settings: i) **In-Distribution**. Demonstration samples and the evaluated samples come from the same distribution, meaning that all samples share the same reasoning graph label. ii) **Out-of-Distribution** (i.e., compositional generalization). Here, demonstration samples and evaluated samples come from different distributions. We guarantee that the primitive relation types in the evaluated sample are encountered in the demonstration samples. Table 3 shows examples for both settings.

5.3 Evaluation Metrics

Quality Evaluation Following Lin et al. (2020), we use seven evaluation metrics from three categories, focusing on i) *surface similarity* by concentrating on n-gram overlap between generations and references, using BLEU (Papineni et al., 2002),

[4]For a concept set with multiple reference sentences, we construct an individual reasoning graph for each reference.

[5]Note that during graph construction, intermediate concept nodes may be added. This can be seen in column 4 of Table 2.

[6]Details about prompts can be found in Appendix C.

[7]The value is determined by the empirical experiments.

Models	Dis	ROUGE-2/L		BLEU-3/4		METEOR	CIDEr	SPICE	Coverage
Llama2	id	8.57	27.49	14.00	9.00	16.19	7.98	24.30	71.79
	ood	6.27	24.57	10.10	5.80	15.84	6.48	22.80	67.75
	Δ	2.30	2.92	3.90	3.20	0.35	1.50	1.50	4.04
Llama2-chat	id	8.35	26.85	13.60	8.70	16.13	7.50	24.00	70.42
	ood	6.11	23.03	9.50	5.30	14.97	5.09	21.20	65.64
	Δ	2.14	3.82	4.10	3.40	1.16	2.41	2.80	4.78
Mistral	id	9.50	28.79	15.50	10.40	17.02	8.60	24.50	69.92
	ood	6.15	23.69	9.50	5.50	15.49	5.85	21.00	59.88
	Δ	3.35	5.10	6.00	4.90	1.53	2.75	3.50	10.04
Mistral-chat	id	8.37	26.86	13.70	8.70	15.33	7.67	23.50	71.89
	ood	5.56	21.79	9.10	5.10	13.45	5.40	19.10	55.72
	Δ	2.81	5.07	4.60	3.60	1.88	2.27	4.40	16.17
Falcon	id	8.44	26.97	13.80	9.00	15.14	7.59	22.10	64.06
	ood	5.99	22.73	9.80	5.70	13.51	5.57	19.10	55.21
	Δ	2.45	4.24	4.00	3.30	1.63	2.02	3.00	8.85
Falcon-chat	id	7.42	25.48	12.30	7.90	14.31	6.83	20.10	60.22
	ood	5.03	21.24	8.00	4.50	12.89	4.95	17.40	50.55
	Δ	2.39	4.24	4.30	3.40	1.42	1.88	2.70	9.67
GPT-J	id	7.31	24.40	12.10	7.60	13.33	6.37	19.00	52.20
	ood	5.10	20.68	8.00	4.10	12.56	4.86	16.60	43.71
	Δ	2.21	3.72	4.10	3.50	0.77	1.51	2.40	8.49
GPT-4o[5]	id	10.43	29.72	15.70	10.70	17.78	9.75	37.70	97.68
	ood	8.46	27.04	12.40	8.10	16.67	8.63	35.50	95.17
	Δ	1.97	2.68	3.30	2.60	1.11	1.12	2.20	2.51

Table 4: Performance of seven LLMs on the CGGC tasks in two configurations: *in-distribution* (id) and *compositional generalization* (ood). Δ indicates the gap between the setting of id and ood, calculated as Δ = id - ood. We highlight the maximum value of id and ood by ■ and ■ respectively, and the minimum value of Δ in ■. Aggregated results are shown in the Appendix F for a better illustration of comparisons among different models.

ROUGE (Lin, 2004) and METEOR (Banerjee and Lavie, 2005); ii) *concept associations*, assuming system generations and human references use similar concepts and focusing on evaluating the associations between mentioned concepts, using CIDEr (Vedantam et al., 2015) and SPICE (Anderson et al., 2016); iii) *task performance*, by analyzing whether the model completes the given task. Here, Coverage (Lin, 2004) calculates the average percentage of input concepts present in the lemmatized output.

Human Evaluation Following (Lu et al., 2022; Meng et al., 2022; Zhang et al., 2023a), we conduct a human evaluation of the generated sentences y across four dimensions: i) *Quality*: Is the sentence well-formed and fluent? ii) *Plausibility*: Does the sentence describe a plausible situation? iii) *Concepts*: Does the sentence include the given concepts in a meaningfull way? iv) *Overall*: Considering the above three metrics, does the sentence meaningfully combine all given concepts into a well-formed scenario? For each aspect, annotators indicated their agreement with the pre-defined statement using the scale: *Yes* (3 points), *Somewhat* (2 points), and *No* (1 point). [8]

Relation Verification CGGC performs compositional generalization evaluation based on the reasoning relations, assuming that LLMs that understand them can use them for generating sentences. However, existing research indicates that LLMs might produce correct answers without applying the correct reasoning. In our context, this means a fitting sentence could be generated without utilizing the provided relations. To address this issue, we developed a *verification model* to ensure that the generated sentences indeed incorporate the intended reasoning relations.

For this purpose we use Llama2 with a feed-forward classification layer to classify relation types based on two concepts and a target sentence. For example, given the concepts {*oven, pizza*} and the sentence *He puts a cheese pizza in an oven*, the model is expected to predict the relation */r/AtLocation*. To evaluate the model, we compare

[8] Evaluation guidelines are provided in the Appendix E.

the predicted relation with those specified in the reasoning graph. If the prediction matches the provided relation, we consider it to be genuinely used, aligning with our expectations for compositional evaluation.[9] For further analyses, we use the *verification model* to filter results, focusing solely on examples where all composed relations are accurately applied. After filtering, an average of 44.74% and 44.81% of the data is removed for in-distribution and compositional generalization, respectively.

6 Results

6.1 Overall Results

Table 4 shows the performance of all LLMs in two data configurations: *in-distribution* (id) and *compositional generalization* (ood).[10] We provide aggregated results in Appendix F for better overview.

According to eight evaluation metrics, we observe that *Mistral* and *Llama2* generally achieve the best performance under in-distribution and compositional generalization settings, respectively (highlighted in green and yellow in Table 4). Across all evaluated models, including the powerful GPT-4, the gap (Δ) between the two data configurations consistently shows positive values. This suggests that the **evaluated LLMs still lack compositional generalization ability to some extent** when encountering unseen composition instances. In addition, we find that various model groups show variance in absolute performance and the compositional generalization abilities. Llama2 models show high absolute performance in both data configurations and superior compositionality. Mistral models achieve relatively high absolute performance but low compositionality, whereas GPT-J shows the opposite trend (Fig. 7 in the Appendix F shows details).

We also compare the performances of the vanilla vs. chat version for each model type,[11] e.g., Llama2 vs. Llama2-chat. Results do not indicate a consistent trend: For Llama2, the vanilla version shows superior compositionality in 7 metrics but inferior results in the remaining (ROUGE-2) metric, i.e., 7 vs. 1. This trend is not observed with Mistral and Falcon, where the metrics of superior and inferior results between vanilla and chat versions are more balanced: 3 vs. 5 for Mistral and 3 vs. 4 for Falcon.

Models	Dis	Quality	Plausibility	Concepts	Overall
Llama2	id	2.39	2.43	2.29	2.22
	ood	1.96	1.61	2.02	1.95
Llama2-chat	id	2.18	2.41	2.32	2.14
	ood	1.88	1.93	1.87	1.99
Mistral	id	2.20	2.20	2.28	2.10
	ood	2.02	2.00	1.90	1.80
GPT-4o	id	2.54	2.56	2.94	2.62
	ood	2.26	2.24	2.80	2.46

Table 5: Human evaluation results of four models on the CGGC task in two configurations: in-distribution (id) and compositional generalization (ood).

6.2 Human Evaluation

To avoid the limitation of rigid automatic evaluation metrics, we also conduct human evaluations following (Meng et al., 2022; Zhang et al., 2023a). We selected 50 samples from each of four representative models (maximum performance / minimum compositionality gap) under both settings (in-distribution and out-of-distribution), resulting in a total of 400 samples. These samples were mixed and presented to two annotators.

Table 5 shows human evaluation results. Comparing the results between the in-distribution (id) and compositional generalization (ood) settings, the four evaluated models represent higher values in id compared to ood setting. This trend is consistent with the automated metrics, reinforcing that LLMs still face challenges with compositional generalization. Notably, GPT-4 achieves the best performance across both configurations, outperforming other baselines by 0.4 and 0.47 points overall.

7 Error Analysis

In this section we investigate potential causes of limitations in compositional generalization, by analyzing error trends in relation to compositions (Section 7.1) and primitives (Section 7.2). [12]

7.1 Composition Analysis

Graph Structure We examine the *graph structures* that indicate how the composition is structured. Considering the complexity of compositions with multiple primitives, we constrain the experi-

[9] For details on this model, see Appendix D.

[10] Due to the high cost of large-scale LLMs, we randomly sample 100 instances for GPT-4o's evaluation. For fair comparison, results of GPT-4o will not be compared for the following qualitative analysis.

[11] GPT-J does not have a chat version, so we ignore it here.

[12] Results of Llama2 are used for further analysis, given its superior performance. For each analysis, we perform three runs with different seeds.

Type	ΔROUGE-2/L(↓)		ΔBLEU-3/4(↓)		ΔMETEOR(↓)	ΔCIDEr(↓)	ΔSPICE(↓)	ΔCoverage(↓)	Rank
A	2.28	2.91	2.63	1.44	0.07	1.44	0.78	2.96	①
B	2.58	2.94	3.07	1.61	0.09	1.81	1.00	3.06	②
C	2.72	3.50	3.09	2.09	0.11	1.97	1.51	4.35	③

Table 6: The performance **gap** between in-distribution and compositional generalization (Δ = id − ood) for three reasoning graph structures (shown in Fig. 3). *Rank* indicates the difficulty levels, calculated by the performance gap.

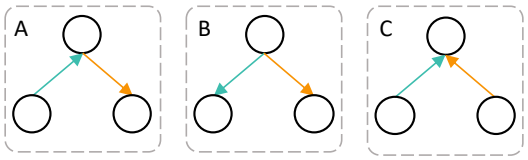

Figure 3: Different types of compositional connection schemas. ■ and ■ arrows (→) denote primitive relations.

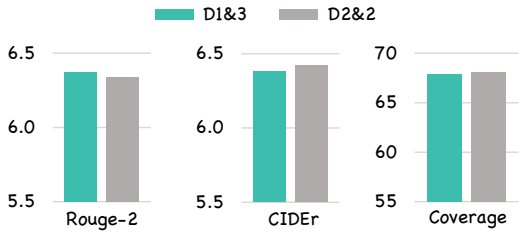

Figure 4: Two ways of providing the required primitive relations in demonstrations for in-context learning.

ment to two primitives. [13] Graphs composed of two primitives (i.e., connecting three nodes) are defined as basic graphs as they are the smallest meaningful graphs that can model composition patterns. They are grouped as: A) *transitive*: directed primitive relations are connected in a uniform direction; B) *common source*: the starting node of two directed primitive relations are shared; C) *common target*: the end nodes of two primitive directed relations are shared. Fig. 3 illustrates these connection schemas.

We categorize the graph structures of test samples along the classes A-C. Table 6 shows the performance gap between in-distribution and compositional generalization for the 3 classes. [14] We observe that **different structures present varying levels of difficulty for compositional generalization**, with A) *transitivity* < B) *common source* < C) *common target*. We further computed the occurrence frequency of subgraphs of the different composition types, with detailed statistics provided in Appendix A.2. It shows that Type A occurs more frequently than Types B and C, while the complexity of the three structures in terms of number of nodes and edges is comparable. We speculate that the difference in difficulty is likely because, compared to a common source or target structure, transitive structures are more common and straightforward to construct into a natural sentence.

Structuring Demonstrations Although we guarantee that primitive relations required to solve unseen composition samples are fully covered by in-context demonstrations, the distribution of these primitive relations in ICL demonstrations is not fully constrained. For example, for a test sample with graph label *Causes & IsA & HasProperty & UsedFor*, the demonstrations could provide primitive relations in the following two ways: i) $D_{1\&3}$. The compositional relations could be separated into one relation and the three remaining ones, such as samples with *Causes* and *HasProperty & IsA & UsedFor*; or ii) $D_{2\&2}$. The compositional relations could be separated into two groups with two primitive relations each, such as *Causes & HasProperty* and *IsA & UsedFor*. Figure 4 illustrates the comparison between these two alternative ways of providing the required primitive relations. [15] We observe that the alternative options in structuring ICL demonstrations in terms of packaging primitives show minor differences. This indicates that alternative options for presenting *the same set of primitives* for a given sample do not affect the performance of compositional reasoning significantly.

7.2 Primitive Analysis

Next, we examine the correlation between primitive relation types and the models' performances in the two data configurations, aiming to determine whether specific primitive relations have a specific impact (i.e., greater or lower difficulty) on a model's compositional generalization abilities. Specifically, we count the occurrence frequency of all primitive relation types in the reasoning graphs,

[13] Experimental data is sampled from graphs of original concept set size 3. For more details see Appendix A.2.

[14] We compare results of each two data structures, the pairwise t-test at 5% significance level.

[15] Given space limits, we select a representative metric from each metric category. Other choices show a similar trend.

	Dimension	ROUGE-2/L		BLEU-3/4		METEOR	CIDEr	SPICE	Coverage
level	easy (typeA)	5.99	23.52	10.50	5.80	13.96	6.10	20.10	66.63
	medium (typeB)	5.69	23.39	9.90	5.40	14.01	6.12	20.60	66.43
	hard (typeC)	5.67	22.98	10.00	5.50	14.13	6.01	21.10	66.58
order	easy-to-hard	**7.11**	**27.77**	**12.00**	**7.10**	**16.76**	**7.65**	**27.20**	**79.98**
	hard-to-easy	7.04	26.60	11.50	6.80	15.61	7.25	24.40	73.06

Table 7: Performance results when controlling the demonstration of graph structures in ICL along two dimensions: *level of difficulty* (level) and *ordering according to difficulty* (order). The pairwise t-test at 5% significance level.

Dis	RBO_{R2}	RBO_{CIDEr}	RBO_{Cov}	Avg
id	0.5445	0.5736	0.5141	0.5441
ood	0.4927	0.5473	0.5119	0.5173
Δ	0.4603	0.3997	0.4997	0.4532

Table 8: Rank Biased Overlap (RBO) compares the rank of relation types according to their frequency and their associated performance results across three data configurations (Dis). Avg indicates the average results.

and rank these relations from most to least frequent. The sorted rank is denoted as R_{freq}. We also examine the difficulty of primitive relation types based on the two data configurations (id vs. ood) from easiest to hardest, and record the gap in performance (Δ) between the two configurations from smallest (best, highest rank) to largest (worst, lowest rank). Each rank for a given relation under a given data configuration is compared with its frequency rank R_{freq}. Table 8 shows Rank Biased Overlap (RBO) (Webber et al., 2010) results for three metrics and their average value.

We find that the average correlations of the three configurations are around 50%, indicating a moderate correlation between performance and the frequency of primitive relation types. Notably, the *id* and *ood* configurations show correlations of 0.5441 and 0.5173, respectively, which are higher than the 0.4532 correlation observed for gap performance. This suggests that primitive relation types have a greater impact on absolute generation quality than on a model's compositional generalization abilities.

8 Difficulty-based Demonstrations

We conclude from the analyses in Sec. 7 (see especially Table 6) that reasoning graph structures are a significant factor affecting compositional generalization. Hence, we aim to investigate whether, and to what extent the compositional generalization ability of evaluated LLMs could be enhanced by controlling the demonstration of graph structures in in-context learning. We group the samples by the ranked difficulty of their graph structures into three groups: *hard*: type C, *medium*: type B, and *easy*: type A (see Fig. 3). We then select and arrange demonstration candidates along two dimensions: i) **level of difficulty**, by selecting demonstrations from a specific graph structure type (A/B/C) and ii) **ordering according to difficulty**, by arranging demonstration types according to a given level of difficulty and following a specific order, from *easy-to-hard* (A→B→C) or *hard-to-easy* (C→B→A).

Table 7 shows results for both dimensions. We find that the evaluated model benefits more when demonstrations are *ordered by difficulty*. That is, combining graph structures of different difficulty levels considerably enhances the model's ability to perform compositional reasoning – compared to relying on a single structure type. This finding aligns with results in Levy et al. (2023) who experimented on tree structures. Furthermore, we observe ordering demonstrations in an *easy-to-hard* manner achieves superior compositionality performance compared to the reverse demonstration order. This result parallels the findings of Fu and Frank (2024b), who show that ordering compositional textual NLI problems in an easy-to-hard manner improves model performance in continual learning.

9 Conclusion

We propose a Compositional Generalization challenge for Graph-based Commonsense Reasoning that extends CommonGen to a compositional *generative commonsense reasoning* task from *graph-structured inputs*. Extensive experiments on seven LLMs using In-Context Learning indicate that they struggle with compositional generalization settings. We investigate potential causes of the limitations, and find that the topology of the graph structures is a significant factor. We show that arranging the order of demonstrations in an easy-to-hard schema enhances the compositional generalization ability.

10 Limitations

We use ConceptNet to enrich each CommonGn sample (a concept set and reference sentence) with a commonsense reasoning graph, which constrains potential relations between concepts to 13 common commonsense relation types. This limitation restricts the variety of composition types compared to real-world applications. However, even on this restricted set of basic relations we were able to establish weaknesses of current LLMs. Additionally, the construction of the data is unsupervised and relies on the quality of ConceptNet and the SBERT model. However, the proposed construction method for CGGC is flexible and can be extended with other high-quality and high-coverage commonsense resources in the future.

For in-context learning, we use a fixed prompt as described in Appendix C, chosen based on recommended best practices and preliminary experiments. We leave the exploration of other prompt constructions, such as incorporating explanations within the prompts, for future work.

11 Acknowledgments

We are grateful to anonymous reviewers for their valuable comments that have helped to improve this paper. We also thank annotators for their valuable work on human evaluations. This work has been supported through a scholarship provided by the Heidelberg Institute for Theoretical Studies gGmbH.

References

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

Haozhe An, Zongxia Li, Jieyu Zhao, and Rachel Rudinger. 2023. SODAPOP: Open-ended discovery of social biases in social commonsense reasoning models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1573–1596, Dubrovnik, Croatia. Association for Computational Linguistics.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, pages 382–398. Springer.

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.

Maria Becker, Katharina Korfhage, Debjit Paul, and Anette Frank. 2021. CO-NNECT: A framework for revealing commonsense knowledge paths as explicitations of implicit knowledge in texts. In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pages 21–32, Groningen, The Netherlands (online). Association for Computational Linguistics.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Ben Bogin, Shivanshu Gupta, Matt Gardner, and Jonathan Berant. 2021. COVR: A test-bed for visually grounded compositional generalization with real images. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9824–9846, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ronald Brachman and Hector Levesque. 2004. *Knowledge representation and reasoning*. Morgan Kaufmann.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jiangjie Chen, Wei Shi, Ziquan Fu, Sijie Cheng, Lei Li, and Yanghua Xiao. 2023. Say what you mean! large language models speak too positively about negative commonsense knowledge. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9890–9908, Toronto, Canada. Association for Computational Linguistics.

Henry Conklin, Bailin Wang, Kenny Smith, and Ivan Titov. 2021. Meta-learning to compositionally generalize. 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 3322–3335, 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 and virtual meeting. Association for Computational Linguistics.

Verna Dankers, Elia Bruni, and Dieuwke Hupkes. 2022. The paradox of the compositionality of natural language: A neural machine translation case study. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4154–4175, Dublin, Ireland. Association for Computational Linguistics.

Joe Davison, Joshua Feldman, and Alexander Rush. 2019. Commonsense knowledge mining from pretrained models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1173–1178, Hong Kong, China. Association for Computational Linguistics.

Edsger W Dijkstra. 2022. A note on two problems in connexion with graphs. In *Edsger Wybe Dijkstra: His Life, Work, and Legacy*, pages 287–290.

Wenxuan Ding, Shangbin Feng, Yuhan Liu, Zhaoxuan Tan, Vidhisha Balachandran, Tianxing He, and Yulia Tsvetkov. 2024. Knowledge crosswords: Geometric knowledge reasoning with large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 2609–2636, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, et al. 2024. Faith and fate: Limits of transformers on compositionality. *Advances in Neural Information Processing Systems*, 36.

Tianqing Fang, Zeming Chen, Yangqiu Song, and Antoine Bosselut. 2024. Complex reasoning over logical queries on commonsense knowledge graphs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11365–11384, Bangkok, Thailand. Association for Computational Linguistics.

Jerry A Fodor and Zenon W Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71.

Xiyan Fu and Anette Frank. 2023. SETI: Systematicity evaluation of textual inference. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4101–4114, Toronto, Canada. Association for Computational Linguistics.

Xiyan Fu and Anette Frank. 2024a. Dynamic modularized reasoning for compositional structured explanation generation. In *Proceedings of the 13th Joint Conference on Lexical and Computational Semantics (*SEM 2024)*. Association for Computational Linguistics.

Xiyan Fu and Anette Frank. 2024b. Exploring continual learning of compositional generalization in nli. *Transactions of the Association for Computational Linguistics*.

Jie He, Tao Wang, Deyi Xiong, and Qun Liu. 2020. The box is in the pen: Evaluating commonsense reasoning in neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3662–3672, Online. Association for Computational Linguistics.

Jonathan Herzig and Jonathan Berant. 2021. Span-based semantic parsing for compositional generalization. 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 908–921, Online. Association for Computational Linguistics.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2020. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795.

Dieuwke Hupkes, Mario Giulianelli, Verna Dankers, Mikel Artetxe, Yanai Elazar, Tiago Pimentel, Christos Christodoulopoulos, Karim Lasri, Naomi Saphra, Arabella Sinclair, et al. 2023. A taxonomy and review of generalization research in nlp. *Nature Machine Intelligence*, 5(10):1161–1174.

Raghav Jain, Daivik Sojitra, Arkadeep Acharya, Sriparna Saha, Adam Jatowt, and Sandipan Dandapat. 2023. Do language models have a common sense regarding time? revisiting temporal commonsense reasoning in the era of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6750–6774, Singapore. Association for Computational Linguistics.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

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 and virtual meeting. Association for Computational Linguistics.

Martin Josifoski, Marija Sakota, Maxime Peyrard, and Robert West. 2023. Exploiting asymmetry for synthetic training data generation: SynthIE and the case of information extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1555–1574, Singapore. Association for Computational Linguistics.

Mayank Kejriwal and Ke Shen. 2020. Do fine-tuned commonsense language models really generalize? *arXiv preprint arXiv:2011.09159*.

Najoung Kim and Tal Linzen. 2020. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105, Online. Association for Computational Linguistics.

Tassilo Klein and Moin Nabi. 2021. Towards zero-shot commonsense reasoning with self-supervised refinement of language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8737–8743, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Keito Kudo, Yoichi Aoki, Tatsuki Kuribayashi, Ana Brassard, Masashi Yoshikawa, Keisuke Sakaguchi, and Kentaro Inui. 2023. Do deep neural networks capture compositionality in arithmetic reasoning? In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1351–1362, Dubrovnik, Croatia. Association for Computational Linguistics.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International conference on machine learning*, pages 2873–2882. PMLR.

Brenden M Lake and Marco Baroni. 2023. Human-like systematic generalization through a meta-learning neural network. *Nature*, pages 1–7.

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.

Itay Levy, Ben Bogin, and Jonathan Berant. 2023. Diverse demonstrations improve in-context compositional generalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1401–1422, Toronto, Canada. Association for Computational Linguistics.

Yafu Li, Yongjing Yin, Yulong Chen, and Yue Zhang. 2021. On compositional generalization of neural machine translation. 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 4767–4780, Online. Association for Computational Linguistics.

Bill Yuchen Lin, Seyeon Lee, Xiaoyang Qiao, and Xiang Ren. 2021. Common sense beyond English: Evaluating and improving multilingual language models for commonsense reasoning. 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 1274–1287, Online. 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.

Chenzhengyi Liu, Jie Huang, Kerui Zhu, and Kevin Chen-Chuan Chang. 2023a. 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.

Linqing Liu, Patrick Lewis, Sebastian Riedel, and Pontus Stenetorp. 2022a. Challenges in generalization in open domain question answering. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2014–2029, Seattle, United States. Association for Computational Linguistics.

Wei Liu, Xiyan Fu, and Michael Strube. 2023b. Modeling structural similarities between documents for coherence assessment with graph convolutional networks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7792–7808, Toronto, Canada. Association for Computational Linguistics.

Xin Liu, Dayiheng Liu, Baosong Yang, Haibo Zhang, Junwei Ding, Wenqing Yao, Weihua Luo, Haiying Zhang, and Jinsong Su. 2022b. Kgr4: Retrieval, retrospect, refine and rethink for commonsense generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11029–11037.

Xuebo Liu, Yutong Wang, Derek F. Wong, Runzhe Zhan, Liangxuan Yu, and Min Zhang. 2023c. Revisiting commonsense reasoning in machine translation: Training, evaluation and challenge. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15536–15550, Toronto, Canada. Association for Computational Linguistics.

Ye Liu, Yao Wan, Lifang He, Hao Peng, and S Yu Philip. 2021. Kg-bart: Knowledge graph-augmented bart for generative commonsense reasoning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 6418–6425.

Nicholas Lourie, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Unicorn on rainbow: A universal commonsense reasoning model on a new multitask benchmark. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13480–13488.

Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, Noah A. Smith, and Yejin Choi. 2022. NeuroLogic a*esque decoding: Constrained text generation with lookahead heuristics. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 780–799, Seattle, United States. Association for Computational Linguistics.

Kaixin Ma, Filip Ilievski, Jonathan Francis, Satoru Ozaki, Eric Nyberg, and Alessandro Oltramari. 2021. Exploring strategies for generalizable commonsense reasoning with pre-trained models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5474–5483, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Adrian Maler. 2023. Evaluating common-sense reasoning in pretrained transformer-based language models using adversarial schemas and consistency metrics.

Tao Meng, Sidi Lu, Nanyun Peng, and Kai-Wei Chang. 2022. Controllable text generation with neurally-decomposed oracle. *Advances in Neural Information Processing Systems*, 35:28125–28139.

Mohsen Mesgar and Michael Strube. 2015. Graph-based coherence modeling for assessing readability. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 309–318, Denver, Colorado. Association for Computational Linguistics.

Pavan Kalyan Reddy Neerudu, Subba Oota, Mounika Marreddy, Venkateswara Kagita, and Manish Gupta. 2023. On robustness of finetuned transformer-based NLP models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7180–7195, Singapore. Association for Computational Linguistics.

Mitja Nikolaus, Mostafa Abdou, Matthew Lamm, Rahul Aralikatte, and Desmond Elliott. 2019. Compositional generalization in image captioning. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 87–98, Hong Kong, China. 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.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Moritz Plenz, Juri Opitz, Philipp Heinisch, Philipp Cimiano, and Anette Frank. 2023. Similarity-weighted construction of contextualized commonsense knowledge graphs for knowledge-intense argumentation tasks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6130–6158, Toronto, Canada. Association for Computational Linguistics.

Linlu Qiu, Peter Shaw, Panupong Pasupat, Pawel Nowak, Tal Linzen, Fei Sha, and Kristina Toutanova. 2022a. Improving compositional generalization with latent structure and data augmentation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4341–4362, Seattle, United States. Association for Computational Linguistics.

Linlu Qiu, Peter Shaw, Panupong Pasupat, Tianze Shi, Jonathan Herzig, Emily Pitler, Fei Sha, and Kristina Toutanova. 2022b. Evaluating the impact of model scale for compositional generalization in semantic parsing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9157–9179, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy. Association for Computational Linguistics.

Hannah Rashkin, Antoine Bosselut, Maarten Sap, Kevin Knight, and Yejin Choi. 2018. Modeling naive psychology of characters in simple commonsense stories.

In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2289–2299, Melbourne, Australia. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Abulhair Saparov, Richard Yuanzhe Pang, Vishakh Padmakumar, Nitish Joshi, Mehran Kazemi, Najoung Kim, and He He. 2023. Testing the general deductive reasoning capacity of large language models using OOD examples. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Abulhair Saparov, Richard Yuanzhe Pang, Vishakh Padmakumar, Nitish Joshi, Mehran Kazemi, Najoung Kim, and He He. 2024. Testing the general deductive reasoning capacity of large language models using ood examples. *Advances in Neural Information Processing Systems*, 36.

Priyanka Sen and Amir Saffari. 2020. What do models learn from question answering datasets? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2429–2438, Online. Association for Computational Linguistics.

Ke Shen and Mayank Kejriwal. 2021. On the generalization abilities of fine-tuned commonsense language representation models. In *Artificial Intelligence XXXVIII: 41st SGAI International Conference on Artificial Intelligence, AI 2021, Cambridge, UK, December 14–16, 2021, Proceedings 41*, pages 3–16. Springer.

Vered Shwartz and Yejin Choi. 2020. Do neural language models overcome reporting bias? In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6863–6870, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Shikhar Singh, Ehsan Qasemi, and Muhao Chen. 2023. VIPHY: Probing "visible" physical commonsense knowledge. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7113–7128, Singapore. Association for Computational Linguistics.

Anna Sotnikova, Yang Trista Cao, Hal Daumé III, and Rachel Rudinger. 2021. Analyzing stereotypes in generative text inference tasks. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4052–4065, Online. Association for Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Ying Su, Zihao Wang, Tianqing Fang, Hongming Zhang, Yangqiu Song, and Tong Zhang. 2022. MICO: A multi-alternative contrastive learning framework for commonsense knowledge representation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1339–1351, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Tianyi Tang, Yushuo Chen, Yifan Du, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Learning to imagine: Visually-augmented natural language generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9468–9481, Toronto, Canada. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Peifeng Wang, Filip Ilievski, Muhao Chen, and Xiang Ren. 2021. Do language models perform generalizable commonsense inference? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3681–3688, Online. Association for Computational Linguistics.

PeiFeng Wang, Jonathan Zamora, Junfeng Liu, Filip Ilievski, Muhao Chen, and Xiang Ren. 2022. Contextualized scene imagination for generative commonsense reasoning. In *International Conference on Learning Representations*.

William Webber, Alistair Moffat, and Justin Zobel. 2010. A similarity measure for indefinite rankings. *ACM*

Transactions on Information Systems (TOIS), 28(4):1–38.

Georg Wenzel and Adam Jatowt. 2023. An overview of temporal commonsense reasoning and acquisition. *arXiv preprint arXiv:2308.00002*.

Xinnuo Xu, Ivan Titov, and Mirella Lapata. 2023. Compositional generalization for data-to-text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9299–9317, Singapore. Association for Computational Linguistics.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, and Kentaro Inui. 2020. Do neural models learn systematicity of monotonicity inference in natural language? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6105–6117, Online. Association for Computational Linguistics.

Honghua Zhang, Meihua Dang, Nanyun Peng, and Guy Van den Broeck. 2023a. Tractable control for autoregressive language generation. In *International Conference on Machine Learning*, pages 40932–40945. PMLR.

Hongming Zhang, Yintong Huo, Yanai Elazar, Yangqiu Song, Yoav Goldberg, and Dan Roth. 2023b. CIKQA: Learning commonsense inference with a unified knowledge-in-the-loop QA paradigm. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 114–124, Dubrovnik, Croatia. 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. *arXiv preprint arXiv:2404.16807*.

Hao Zheng and Mirella Lapata. 2021. Compositional generalization via semantic tagging. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1022–1032, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xuhui Zhou, Yue Zhang, Leyang Cui, and Dandan Huang. 2020. Evaluating commonsense in pre-trained language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9733–9740.

A Data

A.1 ConceptNet Relations

We merge some relations with similar meanings, e.g., {/r/InstanceOf, /r/MannerOf} → /r/IsA. Table 9 shows all instantiations of merged relations. We also ignore some infrequent relations, e.g., /r/LocatedNear, /r/SymbolOf. Finally selected 13 relations are listed in the bottom of Table 9.

	source	target
merge_rel	/r/HasFirstSubevent, /r/HasLastSubevent	/r/HasSubevent
	/r/InstanceOf, /r/MannerOf	/r/IsA
	/r/PartOf	/r/HasA
final_rel	/r/IsA, /r/UsedFor, /r/AtLocation, /r/HasSubevent, /r/HasPrerequisite, /r/CapableOf, /r/CausesDesire, /r/Causes, /r/MotivatedByGoal, /r/HasProperty, /r/ReceivesAction, /r/HasA, /r/Desires	

Table 9: Instantiations of merged relations and finally selected relations from ConceptNet.

A.2 Data Statistics of Graph Types

We examine the graph structures (Mesgar and Strube, 2015; Liu et al., 2023b) that indicate how the composition is structured. Considering the complexity of compositions with multiple primitives, we constrain the experiment to two primitives. Experimental data is sampled from graphs of original concepts set size 3. Selected test samples are categorized given their graph structures, as illustrated in Fig. 3. Table 10 presents the data statistics for the three different types of graph structures. We observe that type A occurs more frequently than the other graph types. However, the graph complexity in terms of node and relation counts is comparable, with Type A showing marginally lower counts.

	general		graph		
	#num	#sent	#nodes	#edges	#rel
Type A	3517	6250	8.49	7.17	4.41
Type B	1037	1697	9.22	7.91	4.72
Type C	1185	1964	9.12	7.86	4.73

Table 10: Data statistics of different graph types of test samples in CGGC. We include the general information of concept sets and extended reasoning graph details.

A.3 Data Statistics of Target Sentences

We segmented the target sentences and counted the involved words. The column #len$_s$ in Table 11 presents the results, showing that each sentence contains an average of 11 words. We further explored the new words required to generate the target sentence. Specifically, we removed stop words in each sentence. New words are defined as follows: i) *w/o graph*, only given concepts are counted as given words; ii) *w/ graph*, concepts contained in the graph are also counted as given words. The column #nw$_{wg}$ in Table 11 indicates that a generated sentence requires roughly 3 new meaningful words. We also analyze the part-of-speech tagging of these novel words. C$_{nw}$@5 shows the top 5 categories of missing words. It shows that the categories of missing words mainly include verbs, nouns, adjectives, and prepositions.

A.4 Data Statistics of Graph Labels

As mentioned in Section 4.3, we ultimately selected 468 compositional graph labels for our experiments. Figure 6 illustrates the sample distribution for the top 40 graph labels. We further count various relation types (primitives), where Figure 5 show the sample distribution for used relations.

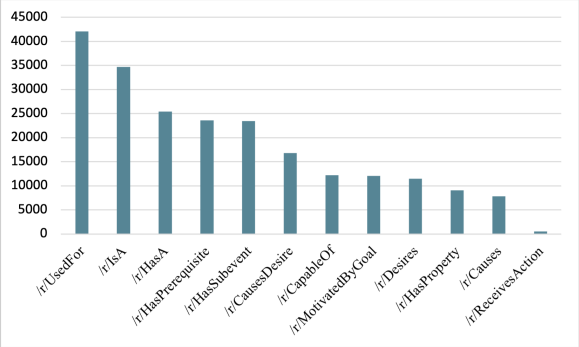

Figure 5: Distribution of primitive relations. The y-axis indicates number of samples.

A.5 Data Splits for *Verification* & *Composition*

As mentioned in Section 5.3, we need to construct a verification model [16] to verify if the generated sentence uses the reasoning relations provided in the graph as we expected. Hence, we split the dataset into two groups: i) *verification*, for evaluating whether the generated sentences do in fact express the target relations. Train and val data are used for verification model training and validation.

[16] For details of the verification model see Appendix D.

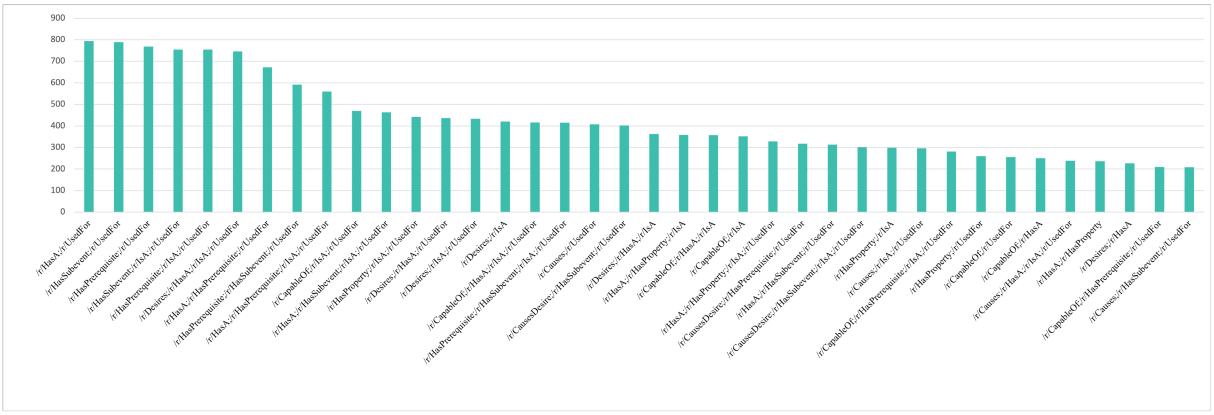

Figure 6: Distribution of compositional *graph labels*. The y-axis indicates number of samples.

	#len$_s$	w graph		w/o graph	
		#nw$_{wg}$	C$_{nw}$@5	#nw$_{wog}$	C$_{nw}$@5
all	10.68	2.82	VBG,CD,NN,JJ,IN	2.87	VBG,CD,NN,JJ,IN
-size=3	10.16	2.81	VBG,CD,NN,JJ,IN	2.86	VBG,CD,NN,JJ,IN
-size=4	11.75	2.83	RB,JJ,IN,NN,NNS	2.88	RB,JJ,IN,NN,NNS
-size=5	13.89	2.98	NN,JJ,NNS,VBG,VBD	3.05	NN,JJ,NNS,VBG,VBD

VBG: Gerund or Present Participle; **CD**: Cardinal Number; **NN**: Noun, Single; **JJ**: Adjective; **IN**: Preposition; **NNS**: Noun, Plural; **RB**: Adverb; **VBD**: Verb, Past Tense

Table 11: Data statistics for the target sentences of the CGGC Benchmark. *#nw* indicates the number of novel words (removing stop words and punctuation) except the given concepts contained in the target sentence. The subscript $_{wg}$ and $_{wog}$ denotes whether concepts contained in the reasoning graph are counted in known concepts. C$_{nw}$@5 denotes the top 5 categories of missing words.

Results are shown in Appendix D; and ii) *composition*, for graph-based compositional generalization tests. Here the train and test data are used for constructing demonstrations and compositional test. Results are shown in Section 6. Table 12 shows the data statistics for the above two groups.

	Verification		Composition	
	train	val	train	test
#set	6375	1125	11591	7728

Table 12: Data statistics for verification and composition. The sample counts are based on the number of concept sets.

B Human Evaluation of Reasoning Graphs

As our reasoning graphs are automatically constructed, we perform a human evaluation to assess whether these graphs are related to the target sentence, following (Josifoski et al., 2023). We randomly selected 50 test samples for this evaluation. We hired two annotators who majored in computational linguistics for annotation. For each sample, annotators were presented with the target sentence and all triples extracted from the corresponding graph. Each triple is comprised by two concepts and their relation. For each triple, annotators were instructed to determine if the relation between the two given concepts could be inferred from the target sentence. We annotate all triples in one sample, and each sample was rated on a scale of 0 (not related), and 1 (related). The percentage agreement between annotators is 96%, with a Cohen's kappa of 81.13%. The results demonstrate a 91% accuracy in the extracted subgraph's relevance to the target sentence, indicating the high quality of the constructed graphs.

C Experiment Details

Prompts To guide the given task, we add an instruction at the beginning of all inputs. We provide a prompt example as follows:

Dis	Llama2	Llama2-chat	Mistral	Mistral-chat	Falcon	Falcon-chat	GPT-J
id	44.64	44.53	45.11	44.80	44.47	44.27	45.33
ood	44.42	44.69	45.25	45.28	44.44	44.18	45.43

Table 13: The filtering rate (%) by the verification model for seven evaluated models under two test configurations in-distribution (id) and compositional generalization (ood).

Quality	Is the sentence well-formed well-formed? *Yes*: The sentence is well-formed and fluent. *Somewhat*: The sentence is understandable but a bit awkward. *No*: The sentence is neither well-formed or fluent.
Plausibility	Does the sentence describe a plausible scenario? *Yes*: The sentence describes a realistic or plausible scenario. *Somewhat*: The sentence describes an acceptable scenario but a bit awkward. *No*: The sentence describes a nonsensical scenario.
Concepts	Does the sentence include the given concepts meaningfully? *Yes*: The sentence meaningfully includes all of the concepts. *Somewhat*: The sentence meaningfully includes some, but not all of the concepts. Or, the sentence includes all concepts but some of them are not meaningful or properly incorporated. *No*: The sentence does not include concepts in a meaningful way.
Overall	Considering your answers to 1), 2) and 3), does the sentence meaningfully combine all of the concepts into a well formed and plausible scenario? *Yes*: The sentence is reasonably understandable, and meaningfully combines all the concepts into a plausible scenario. *Somewhat*: The sentence looks okay in terms of above questions. *No*: The sentence is not well-formed/understandable, or fails to properly combine all the concepts into a plausible scenario.

Table 14: Human evaluation guidelines for evaluating the generated sentences.

Please generate a natural sentence with the provided concepts and their commonsense reasoning graphs.
concepts: oven, cheese, pizza
commonsense reasoning graph: <H> pizza <R> HasA <T> cheese, <H> pizza <R> AtLocation <T> oven
sentence:

Following the requirements of LLMs, we also add some special tokens to the prompts: i) for *Llama2* models, we add '[INST] «SYS»' (detailed templates can be found in the official website [17]); ii) for *Falcon* and *Mistral* models, we add 'User:' and 'Assistant:'.

D Verification Model

We aim to construct a *verification model* to ensure that the generated sentence accurately employs the required reasoning relations. This model is a multi-label classifier based on a LLM. Specifically, we use Llama2 along with a feed-forward classification layer. The model classifies the relation type based on two concepts and a target sentence. For instance, given the concepts {*oven, pizza*} and the target sentence *He puts a cheese pizza in an oven*, we expect the model to predict */r/AtLocation*. To test the model, we compare the predicted relation with the relations specified in the reasoning graph. If the predicted relation matches the provided relation, we consider this relation to have been genuinely used, aligning with our expectations for compositional evaluation.

Table 12 presents the data statistics for the verification model. We use 6735 samples for training, achieving 90.21% accuracy on a validation set of 1125 samples. To ensure data quality for verification, we sampled data for human evaluation. The results in Appendix B confirm that the data is suitable for verification.

We use this verification model to filter the results of the compositional generation test. When doing so, we focus exclusively on examples where all composed relations are accurately applied, i.e., samples that achieved 100% verification accuracy. Table 13 shows the filtering ratio (%) for each model in the two evaluation configurations: in-distribution and compositional generalization.

E Human Evaluation Guidelines

Following (Lu et al., 2022; Meng et al., 2022; Zhang et al., 2023a), we conduct a human evaluation of the generated sentences y across four dimensions. The guidelines of these four aspects for annotators are provided in the Table 14.

[17] https://llama.meta.com/docs/model-cards-and-prompt-formats/meta-llama-2/

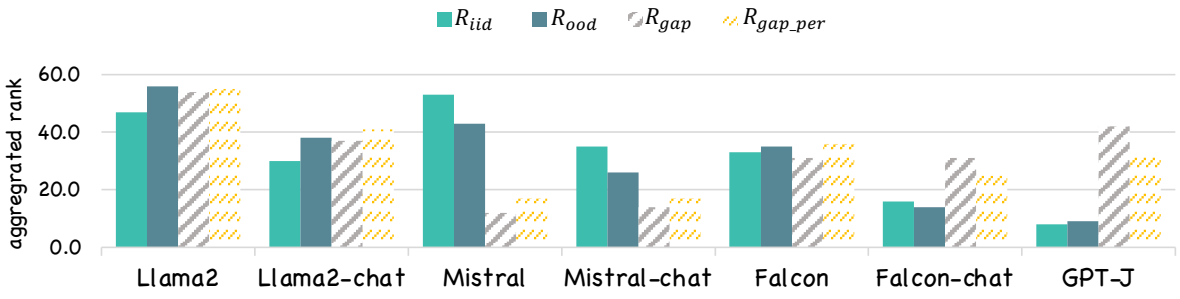

Figure 7: Aggregated results based on eight evaluation metrics for seven evaluated LLMs. Under each evaluation metric, LLMs' results are ranked in an ascending order. The results for *id* and *ood* represent absolute performance, i.e., higher rank is better. Conversely, the *gap* and *gap percentage* show gap performance, i.e., lower rank is better. We then score LLMs based on their ranks. Scores for *id* and *ood* are assigned from 1 (lowest=worst) to 8 (highest=best), while for *gap* and *gap percentage* are assigned from 8 (highest=best) to 1 (lowest=worst). For each model of a specific configuration, we summarize the rank scores across the eight metrics (minimum value: 8*1=8, maximum value: 8*7=56).

F Aggregated Results

Figure 7 aggregates the main results shown in Table 4 for better illustration. Under each evaluation metric, we ranked and scored the LLMs' results for a specific configuration in a specific order. Under each evaluation metric, LLMs' results are ranked in an ascending order. The results for id and ood represent absolute performances, that is, higher rank is better. Conversely, the gap and gap percentage (the percentage drop of the gap value in relation to id results, i.e., smaller is better) represent relative gap performance, that is, lower rank is better. We then score seven LLMs based on their ranks. For absolute performance under id and ood configurations, scores are assigned from 1 to 7. Conversely, scores for gap performance are assigned from 7 to 1. For example, under the Rouge-2 evaluation for the *id* data configuration, the seven LLMs will be ranked and scored as: (7) Mistral > (6) Llama2 > (5) Falcon > (4) Mistral-chat > (3) Llama2-chat > (2) Falcon-chat > (1) GPT-J. For each model of a specific configuration, we summarize the rank scores across the eight metrics. The minimum score is eight metrics with score one, i.e., 8*1 =8, and the maximum value is eight metrics with score seven, i.e., 8*7=56. Aggregated main results are illustrated in Fig. 7.

The aggregated ranks under in-distribution (R_{id}) and compositional generalization (R_{ood}) represent the absolute generation ability, while the remaining two ranks based on the gap performance (R_{gap}) and gap percentage(R_{gap_per}) indicate the models' compositional generalization capability. Given the aggregated scores shown in Fig. 7, we find Mistral achieves the best performance under in-distribution and Llama2 obtains the best results under compositional generalization and smallest gaps. Further, we observe that **various model groups show variance in absolute performance and compositional generalization**. Llama2-based models (Llama2 and Llama2-chat) and Falcon show consistent ability in generation and compositional generalization, while Llama2 represents the top ability. In contrast, the remaining LLMs show a difference between these two abilities. Mistral-based models achieve relatively high absolute performance but low compositional generalization capability, whereas GPT-J shows the opposite trend.