Learning to Predict Concept Ordering for Common Sense Generation

Tianhui Zhang **Danushka Bollegala** **Bei Peng**
University of Liverpool, Amazon
{tianhui.zhang, danushka, bei.peng}@liverpool.ac.uk

Abstract

Prior work has shown that the ordering in which concepts are shown to a commonsense generator plays an important role, affecting the quality of the generated sentence. However, it remains a challenge to determine the optimal ordering of a given set of concepts such that a natural sentence covering all the concepts could be generated from a pretrained generator. To understand the relationship between the ordering of the input concepts and the quality of the generated sentences, we conduct a systematic study considering multiple language models (LMs) and concept ordering strategies. We find that `BART-large` model consistently outperforms all other LMs considered in this study when fine-tuned using the ordering of concepts as they appear in CommonGen training data as measured using multiple evaluation metrics. Moreover, the larger GPT3-based large language models (LLMs) variants do not necessarily outperform much smaller LMs on this task, even when fine-tuned on task-specific training data. Interestingly, human annotators significantly reorder input concept sets when manually writing sentences covering those concepts, and this ordering provides the best sentence generations independently of the LM used for the generation, outperforming a probabilistic concept ordering baseline.[1]

1 Introduction

In Generative Commonsense Reasoning (GCR) (Lin et al., 2020), the goal is to generate a natural commonsense adhering sentence that covers all of the input concepts. GCR is a challenging natural language generation (NLG) task because it requires (a) relational reasoning using background commonsense knowledge associated with the input concepts, and (b) compositional generalisation ability to work on unseen concept combinations. CommonGen dataset (Lin et al., 2020) was specifically developed to evaluate the ability of a generative model to produce sentences covering a given set of concepts. For example, given the set of concepts {*ball*, *batter*, *pitcher*, *throw*}, a human annotator would come up with a sentence such as *The **pitcher throw**s the **ball**, and the **batter** hits a home run!* The input concepts (shown in boldface fonts) are ordered in a specific manner in the produced sentence such as to create a commonsense bearing sentence.

Although neural generation systems produce fluent texts when compared to template-based methods, they fall short in fluency and faithfulness to the input and do not allow control over the output structure (Puzikov and Gurevych, 2018). Similar observations are made by prior work on GCR, where the input ordering of concepts has been reported to influence the quality of the generated sentences (Zhao et al., 2022; Yang et al., 2023). However, the relationship between (a) the ordering of the concepts given as the input to a neural text generation model and (b) the architecture of the underlying LM used in the generator remains unexplored, which is the focus of our study in this paper. Although we consider the concept ordering problem in the context of GCR, we note that it has broader relevance to other tasks in NLP such as multi-document summarisation (Bollegala et al., 2005, 2006, 2012) and textual coherence modelling (Barzilay and Lapata, 2005)

We evaluate the quality of the sentences generated by five commonsense generation LMs: `BERT-gen` (Bao et al., 2020), `BART-base` (Lewis et al., 2019), `T5-base` (Raffel et al., 2019), `BART-large` (Lewis et al., 2019) and `T5-large` (Raffel et al., 2019), where the input concepts are ordered following three different strategies as described in § 3.2. We use seven evaluation metrics to compare the generated sentences against human-written sentences in CommonGen. Specifically, we fine-tune each LM on the train

[1]Code: https://github.com/TianhuiZhang/concept_ordering

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
November 1–4, 2023. ©2023 Association for Computational Linguistics

sentences in CommonGen dataset using next token prediction as the training objective.

We find `BART-large` to consistently outperform the other LMs across several evaluation metrics when fine-tuned using the ordering of concepts as they appear in CommonGen training data. To further compare the concept ordering in the model-generated sentence against the human-generated sentence for the same set of concepts, we use the Kendall rank correlation coefficient (i.e., Kendall's τ) as an evaluation metric. We find that there exists only a weak correlation ($\tau = 0.328$) between the ordering of concepts shown to the human annotators and the ordering in which those concepts appear in the sentences written by those annotators. This indicates that human annotators significantly reorder the input concepts when writing sentences that convey commonsense relationships among given concepts. Interestingly, all the generator models we compare in this paper are able to produce better quality sentences when they are presented with the input concepts in the same ordering as ordered in the sentences.

2 Related Work

Concept Ordering. Concept Ordering aims to reorder the given concepts in a sequence according to their importance and inner relevance. Recent work by Ou et al. (2022) highlighted the effectiveness of pre-trained language models, such as BART (Lewis et al., 2019), in tasks related to concept ordering. Huang et al. (2023) fine-tune the LM to find the most complementary concepts to the given one. Hoyle et al. (2021) and (Zhao et al., 2022) show that suitable concept ordering could increase the quality of the generated sentences. In this work, we aim to explore the relation between the concept ordering and different LMs, assessing their performance in the GCR task.

Generative Commonsense Reasoning. Recently, a series of works have been proposed to evaluate the commonsense reasoning quality of the model's generation. One strand of research leverages these generations as the external commonsense explanation (Chen et al., 2023) or chain-of-thought (Zhang et al., 2023), aiding models in various tasks, including question answering. Another approach compares the generation with the references, examining if the model could write as natural as human, such as CommonGen (Lin et al., 2020) and ROCStories (Mostafazadeh et al., 2016). Liu et al. (2021) and Liu et al. (2022) incorporate external knowledge into pre-trained LMs to enrich the generation information.

Text Generation. Prior work on text generation from structured data such as RDF has shown that the ordering in which a set of entities shown to neural text generation models significantly influences the quality of the generated text (Moryossef et al., 2019; Zhao et al., 2020). Unlike template-based generation methods (Reiter and Dale, 2000; Gatt and Krahmer, 2018), neural text generation methods such as Seq2Seq (Sutskever et al., 2014) models perform *text planning* (how to order the inputs) as well as *plan realisation* (how to verbalise the plan) as a single end-to-end task (Gardent et al., 2017).

3 Methods

3.1 Task definition

Given a set $\mathcal{X} = \{x_1, x_2, \ldots, x_m\}$ of lemmatised tokens representing concepts x_i, the goal of the GCR task is to generate a natural and grammatically correct sentence $\boldsymbol{y} = y_1, y_2, \ldots, y_n$, with tokens (or sub-tokens) y_j. Note that $\mathcal{X}$ is an *unordered* set of concepts, while the tokens in $\boldsymbol{y}$ are ordered. We must use *all* concepts in $\mathcal{X}$ when generating $\boldsymbol{y}$. However, we are allowed to use different morphological forms (inflections) of the concepts for this purpose. Typically in CommonGen dataset ca. $m = 5$, whereas n is determined by the generator model used to produce $\boldsymbol{y}$. Given a set of concepts $\mathcal{X}$, there exists $m!$ number of possible ordering of concepts.

Let us denote one such ordering by $\boldsymbol{x} = x_1, x_2, \ldots, x_m$. Given $\boldsymbol{x}$, a generator parameterised by $\boldsymbol{\theta}$ produces the output sequence $\boldsymbol{y} = y_1, y_2, \ldots y_n$ according to the generation probabilities given by (1).

$$p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}) = \prod_{i=2}^{n} p(y_i|y_{1:i-1}, \boldsymbol{x};\boldsymbol{\theta}) \qquad (1)$$

We generate sentences $\boldsymbol{y}$ using different pre-trained generators for the same set of concepts $\mathcal{X}$, ordered using different strategies as described in § 3.2.

3.2 Concept Ordering Strategies

We name the ordering of input concepts in a train/test instance in CommonGen as the **Original** ordering. We propose three different strategies to re-order the Original ordering for the purpose of

fine-tuning LMs using the CommonGen training instances. In **Random** ordering, the concepts in the input set are randomly ordered. The **Example** ordering considers the ordering of concepts as they appear in a human-written example sentence in the CommonGen train dataset.[2] However, this concept ordering information is not available at test time, we thus use Random Ordering of input concept sets at test time even for the LMs fine-tuned using the Example Ordering.

We induce a **Probabilistic** ordering among the concepts in a given set using transition probabilities $p(x_j|x_i)$ for ordering the concept x_j after x_i because we want to find a method that could be used both at training and test times and do not extract the concept ordering from pre-trained models' outputs. Therefore, inspired by Glove word embeddings (Pennington et al., 2014), which use the co-occurrence information between words, we use the co-occurrence frequency of each concept pair inside the paths of the ConceptNet to determine the ordering in a given concept set. Specifically, we first perform random walks over the ConceptNet graph starting from vertices that appear in the CommonGen train sentences, limiting to a maximum path length of five concepts. Next, we count the number of paths, $\#(x_i \rightarrow x_j)$, where x_i appears before x_j. The transition probabilities are estimated from the path counts as in (2).

$$p(x_j|x_i) = \frac{\#(x_i \rightarrow x_j)}{\#(x_i \rightarrow x_j) + \#(x_j \rightarrow x_i)} \quad (2)$$

Finally, the probability of generating an ordering $\boldsymbol{x} = x_1, x_2, \ldots, x_m$ is computed assuming a first-order Markov chain such that $p(\boldsymbol{x}) = \prod_{i=2}^{m} p(x_i|x_{i-1})$. The **Probabilistic** ordering strategy enables us to incorporate external knowledge from ConceptNet for the purpose of determining the ordering of input concepts. Moreover, unlike the **Example** ordering, **Probabilistic** ordering can be used both at training and test times.

4 Experiments

4.1 Experiments setting

Dataset: Our experiments are conducted on the CommonGen dataset (Lin et al., 2020), which contains 3.5K distinct concept sets (32651/993/1497) with 67389/4018/6042 human written sentences in the training/development/test splits. Each instance contains ca. 3-5 input concepts with multiple human-written reference sentences.

Evaluation Metrics: We conduct two types of evaluations. First, to evaluate the quality of the generated sentences, we employ the following generation evaluation metrics: BLEU (Papineni et al., 2002), , ROUGE (Lin, 2004), METEOR (Banerjee and Lavie, 2005), CIDEr (Vedantam et al., 2015) and SPICE (Anderson et al., 2016). We report the Coverage of concepts (Lin et al., 2020), which is defined as the average percentage of input concepts that are present in the generated sentences. Second, we evaluate the ordering of the concepts produced by different LMs (in the order that they appear in the generated sentences) by comparing that to the ordering in human-written test example sentences using the Kendall rank correlation coefficient ($\tau \in [-1, 1]$). A higher τ indicates a closer correlation between a concept ordering and their order observed in reference sentences. If there are multiple human-written sentences with different orderings for the same input concept set, we take the highest τ over all of the orderings.

4.2 Results

Main Results: We use each concept ordering method to fine-tune five LMs: BERT-Gen (Bao et al., 2020), T5-base/large (Raffel et al., 2019), and BART-base/large (Lewis et al., 2019). Implementation details and hyperparameters can be found in Appendix C. To assess the effectiveness of the concept set ordering strategies, we compare their performance against two types of prior methods. The first uses the **Original** ordering specified in CommonGen (Lin et al., 2020) to fine-tune LMs. The second set includes models that incorporate external knowledge, such as KG-BART (Liu et al., 2021) and EKI-BARTD_{out} (Fan et al., 2020), as well as models that improve performance through pre-training tasks, such as CALM (Zhou et al., 2021), NeuroLogic (Lu et al., 2021), and the [MASK] (Yang et al., 2023).

As shown in Table 1, both **Probabilistic** and **Example** ordering outperform the **Original** Ordering presented in CommonGen (Lin et al., 2020). This shows that refining the ordering strategy for input concepts could enhance an LM's performance in commonsense generation. Moreover, **Example**

[2]As detailed in Appendix A, in our preliminary experiments, we evaluated three different input formats for a concept ordering: (a) space delimited, (b) comma delimited, and (c) space delimited with a special end-of-ordering token (i.e., [ORDERING]). We found the input format method (c) to perform slightly better than the other two, albeit no significant improvements were observed.

	Model	BLEU3	BLEU4	ROUGE-2	ROUGE-L	METEOR	CIDEr	SPICE	Coverage
Original Ordering (Lin et al., 2020)	BERT-Gen	30.4	21.1	18.1	40.5	27.3	12.5	27.3	86.1
	T5-base	26.0	16.4	14.6	34.6	23.0	9.2	22.0	76.7
	BART-large	36.3	26.3	*22.2*	42.0	*30.9*	13.9	30.6	*97.4*
	T5-large	*39.0*	*28.6*	22.0	*43.0*	30.1	*15.0*	*31.6*	95.3
Random ordering	BERT-Gen	30.5	21.1	18.0	40.1	27.4	12.6	27.4	86.5
	BART-base	36.0	26.6	22.0	43.3	28.8	13.9	28.5	92.0
	T5-base	37.7	26.9	20.9	43.2	30.2	15.1	31.2	94.2
	BART-large	*42.4*	*31.8*	*23.8*	44.7	*32.8*	*16.6*	*32.3*	*98.8*
	T5-large	40.6	30.0	23.1	*44.9*	31.5	15.9	32.1	97.6
Probabilistic ordering	BERT-Gen	29.1	19.9	16.9	38.6	26.7	12.2	26.7	85.7
	BART-base	33.5	23.8	20.4	41.2	28.0	13.4	27.8	92.6
	T5-base	37.6	26.8	20.5	42.4	30.6	15.1	31.1	96.4
	BART-large	38.5	28.0	21.8	42.4	*31.7*	15.4	32.2	*98.8*
	T5-large	*40.8*	*29.8*	*22.2*	*43.9*	31.6	*16.2*	*32.4*	95.7
Example ordering	BERT-Gen	33.0	23.6	19.3	41.8	28.9	13.7	28.8	93.3
	BART-base	36.0	26.8	22.7	44.4	29.3	14.7	29.1	97.5
	T5-base	40.0	30.0	22.7	45.0	31.3	16.0	32.3	97.6
	BART-large	***44.3***	***33.8***	***24.5***	***45.6***	***32.8***	***17.2***	33.1	***99.1***
	T5-large	43.4	32.7	23.8	45.6	32.3	16.9	***33.5***	98.0
Other Prior Methods	KG-BART	42.1	30.9	23.4	44.5	32.9	17.5	32.7	98.7
	EKI-BARTD_{out}	42.9	26.3	24.4	45.4	32.0	16.8	32.5	-
	CALM	-	29.5	-	-	31.9	15.6	33.2	-
	NeuroLogic	41.3	36	-	44.7	31.0	15.9	31.1	-
	[MASK]	43.3	32.5	24.2	44.9	32.5	17.1	32.8	-

Table 1: Evaluating the quality of the generated sentences on CommonGen test instances. The input concept sets are ordered using the three strategies described in § 3.2 during fine-tuning different LMs. Best performing LM fine-tuned with each ordering strategy is shown in italics, while the overall best is shown in bold.

ordering reports the best performance for all five LMs, even outperforming prior methods that use pre-training tasks or external resources.

Ordering	τ	BLEU4	ROUGE-L	METEOR	CIDEr	SPICE
Original	0.328	19.0	36.4	29.0	12.9	27.7
Random	0.327	18.9	36.2	29.0	12.7	27.7
Probabilistic	0.402	20.9	37.9	29.6	13.4	29.1
BERT-Gen	0.595	29.0	43.1	31.4	16.0	31.3
BART-base	0.648	31.5	44.2	31.9	16.6	32.1
T5-base	0.627	30.3	43.9	31.9	16.4	32.0
BART-large	**0.697**	**33.4**	**45.5**	**32.7**	**17.3**	**33.0**
T5-large	0.696	32.9	45.2	32.6	17.0	32.7
Reference	1	35.3	57.4	33.3	18.1	33.5

Table 2: Kendall's τ between the reference ordering and the concept orderings produced using different methods: Lines 2-4 show Original, Random and Probabilistic orderings (§ 3.2), Lines 5-9 show the orderings of concepts as they appear in the sentences generated using LMs fine-tuned on CommonGen, and Line 10 is the Reference ordering (i.e., ordering of the concepts in human-written CommonGen test sentences).

Effect of concept ordering: The ordering of concepts as they appear in the human-written CommonGen test sentences (here onwards referred to as the **Reference** ordering) can be considered as a gold standard for concept ordering. To evaluate the level of re-ordering of input concepts conducted by an LM, we compare the ordering of concepts in a sentence generated using an LM against their **Reference** ordering using the Kendall's τ. Specifically, we first fine-tune each LM using the concept sets as ordered in the CommonGen train instances (i.e., **Original** ordering). Next, we present each fine-tuned LM the concept sets in CommonGen test instances in their **Original** ordering and generate a sentence containing all of the input concepts. Finally, we compare the ordering of the concepts in the generated sentence against their **Reference** ordering using τ. The reference sentences in the CommonGen dataset are written by human annotators given the shuffled concepts. Therefore, we think it could be considered as a "golden" ordering of input concepts. To further evaluate the correlation between the ordering of input concepts and the quality of the sentences generated by an LM, we use `BART-large` fine-tuned with **Example** ordering as a sentence generator, where we provide it a set of concepts ordered using different methods.

As seen from Table 2, among the three concept ordering strategies described in § 3.2, the **Probabilistic** ordering reports the highest τ value. In particular, **Original** has a low τ value, comparable to **Random** ordering, indicating that human annotators had to significantly re-order the concept sets shown to them when writing natural sentences.

concept words	throw daughter stream rock daddy
Reference	Daddy and daughter throwing rocks into stream
Turbo(zero shot)	As we hiked beside the stream, my daughter asked if she could throw a rock in the water, but I reminded her that it was too dangerous and daddy
Babbage(Original ordering)	little girl throwing a tantrum in the park with her father and sister by the stream and crying because she did not get a toy in the stream.
Curie(Original ordering)	mother throwing her daughter in the stream near the rocks with daddy
Babbage(Example ordering)	A father and daughter are throwing rocks into a stream.
Curie(Example ordering)	daddy and daughter throwing rocks into a stream
concept words	dog throw frisbee catch
Reference	A man throws away his dog's favorite frisbee expecting him to catch it in the air.
Turbo(zero shot)	The dog loves to play fetch and is always ready to catch the frisbee when it's thrown.
Babbage(Original ordering)	A dog is throwing a frisbee.
Curie(Original ordering)	A dog is throwing a frisbee and catching it.
Babbage(Example ordering)	A dog catching a frisbee and throwing it.
Curie(Example ordering)	A dog catching a frisbee thrown by a man.
concept words	hang squeeze shut head eye
Reference	Hanging her head, someone squeezes her eyes shut.
Turbo(zero shot)	I tried to hang the picture but couldn't do it with just one hand, so I had to squeeze my eyes shut and use both hands to get it done
Babbage(Original ordering)	Someone grabs someone by the eyes and squeezes them shut, then drags him away.
Curie(Original ordering)	A woman is hanging upside down and squeezing her legs shut, then she opens her eyes.
Babbage(Example ordering)	Someone squeezes his eyes shut and hangs his head.
Curie(Example ordering)	Someone squeezes his eyes shut and hangs his head.
concept words	mirror gear picture hold take
Reference	The man holds the gear and uses the picture taken by the mirror.
Turbo(zero shot)	She carefully held the picture up to the mirror to take a closer look at the intricate gear design.
Babbage(Original ordering)	A picture of a man in a space suit, holding a mirror, and taking off his gear.
Curie(Original ordering)	The mirror has gears in it and a picture of a train engine held up.
Babbage(Example ordering)	A man is holding on to a railing to take a picture of his gear and mirror.
Curie(Example ordering)	A man is holding a camera and taking a picture of himself in the mirror.

Table 3: Generated examples by different LLMs

Interestingly, all five LMs fine-tuned with the **Example** ordering strategy outperform those three strategies, suggesting that the sentences carry useful contextual clues for determining the ordering of concepts that are exploited by the LMs. We find that the extracted concept ordering generated by BART-large model outperforms that generated by other pre-trained models. It is consistent with the conclusions of Ou et al. (2022) that a large amount of dependency structure knowledge exists in BART. Moreover, we see that τ values closely aligns with other automatic evaluation metrics, indicating that concept ordering with a higher τ results in high quality sentence generations. Overall, these results suggest that if the concept orderings more closely align with the orderings found in the dataset reference sentences, the Example Ordering trained model could generate superior sentences, leveraging the inherent commonsense knowledge embedded within the pre-trained models.

4.3 Large Language Model Scenario

Large Language Models (LLMs) have shown impressive performance in complex reasoning compared to the smaller size models (Brown et al., 2020; Thoppilan et al., 2022). To investigate their effectiveness for concept ordering, we use the GPT-3 and GPT-3.5 (Brown et al., 2020) as LLMs and use a prompt-based instruction (see Appendix B for the details of the prompts) to induce an ordering among a given set of concepts. Due to the space limitations we show the results in the Appendix Table 4. We find that the Example Ordering outperforms the unordered inputs, indicating that concept ordering is also important to the LLMs. Moreover, from the generated sentences (shown in Table 3) we find that the Example ordering improves their quality as well as the concept coverage. Interestingly however, the best results obtained using LLMs with prompts are worse than that compared to the fine-tuned `BART-large`, which indicates importance of fine-tuning on CommonGen.

5 Conclusion

We examined the impact of concept ordering on the quality of generated sentences in GCR using multiple LMs. We find that ordering the input concepts can improve performance, and all fine-tuned LMs generate better quality sentences when the input concepts were presented in an order consistent with that found in human-written sentences.

6 Limitations

In this work, we limited our investigation to the generation of English sentences and to a finite set of pre-trained language models (PLMs). This limitation was largely due to the nature of the CommonGen dataset, the only publicly available dataset we found for concept-to-sentence tasks, which is

primarily English-centric. Therefore, our evaluation of the generation quality was limited to English, which is a morphologically limited language. Different languages have different grammars and sentence structures, but the automatic evaluation metrics such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004) and CIDEr (Vedantam et al., 2015) could also be used for other languages. Therefore, we consider it to be an important next step to evaluate the concept ordering strategies described in this work on languages other than English.

Furthermore, we focus only on the experiments around concept ordering in the LLM scenario. However, as reported in much prior work, the commonsense knowledge inside the LLMs could be efficiently exploited by intermediate reasoning steps and designed prompts (Wei et al., 2022; Zhang et al., 2023). We utilised identical prompts for each backbone LLM and did not investigate the potential influence of prompt variations. we observed that some sentences generated by LLMs contradict commonsense. Therefore, generative commonsense reasoning tasks remain a challenge for LLMs. Given the scope of the current paper, research on prompt design and encoding concept ordering into intermediate steps will be pursued in future work.

Given a set of concepts, it is possible to write multiple natural sentences that arrange the input concepts in different orders. However, in the CommonGen dataset only a small number of human-written sentences are available for a given concept set, which does not cover all possible orderings. Therefore, the Kendall's τ values reported in this paper must be taken as a lower bound on the agreement with a human determining the ordering of concepts in a sentence. Moreover, we it is important to consider other types of rank evaluation metrics in addition to Kendall's τ for future evaluations.

7 Ethical Considerations

While we conducted our research primarily on the CommonGen dataset, which to the best of our knowledge does not present any explicit ethical issues, it is essential to acknowledge the potential for social biases in the LMs (Blodgett et al., 2021). One of the pre-trained language models we used in our experiments, BART are significantly prone to prediction errors related to gender bias (Sharma et al., 2021) and we are not evaluating for the biases in the generated sentences here which should be done before LMs are deployed in the downstream NLP applications. Given that we are fine-tuning LMs on the CommonGen dataset, some social biases could get amplified during this fine-tuning process. The predicted sentences are possibly influenced by such biases. It is still an open question for how to effectively mitigate these biases, particularly in the context of generative commonsense reasoning tasks.

References

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*. Springer, pages 382–398.

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.

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, et al. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *International conference on machine learning*. PMLR, pages 642–652.

Regina Barzilay and Mirella Lapata. 2005. Modeling local coherence: An entity-based approach. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. Association for Computational Linguistics, Ann Arbor, Michigan, pages 141–148. https://doi.org/10.3115/1219840.1219858.

Su Lin Blodgett, Gilsinia Lopez, Alexandra Olteanu, Robert Sim, and Hanna Wallach. 2021. Stereotyping norwegian salmon: An inventory of pitfalls in fairness benchmark datasets. 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 1004–1015.

Danushka Bollegala, Naoaki Okazaki, and Mitsuru Ishizuka. 2005. A machine learning approach to sentence ordering for multidocument summarization and its evaluation. In *Second International Joint Conference on Natural Language Processing: Full Papers*. https://doi.org/10.1007/11562214_55.

Danushka Bollegala, Naoaki Okazaki, and Mitsuru Ishizuka. 2006. A bottom-up approach to sentence ordering for multi-document summarization.

In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Sydney, Australia, pages 385–392. https://doi.org/10.3115/1220175.1220224.

Danushka Bollegala, Naoaki Okazaki, and Mitsuru Ishizuka. 2012. A preference learning approach to sentence ordering for multi-document summarization. *Information Sciences* 217:78 – 95.

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.

Qianglong Chen, Guohai Xu, Ming Yan, Ji Zhang, Fei Huang, Luo Si, and Yin Zhang. 2023. Distinguish before answer: Generating contrastive explanation as knowledge for commonsense question answering. In *Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics, Toronto, Canada, pages 13207–13224. https://doi.org/10.18653/v1/2023.findings-acl.835.

Zhihao Fan, Yeyun Gong, Zhongyu Wei, Siyuan Wang, Yameng Huang, Jian Jiao, Xuan-Jing Huang, Nan Duan, and Ruofei Zhang. 2020. An enhanced knowledge injection model for commonsense generation. In *Proceedings of the 28th International Conference on Computational Linguistics*. pages 2014–2025.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The WebNLG challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation*. Association for Computational Linguistics, Santiago de Compostela, Spain, pages 124–133.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *J. Artif. Int. Res.* 61(1):65–170.

Bernal Jiménez Gutiérrez, Nikolas McNeal, Clay Washington, You Chen, Lang Li, Huan Sun, and Yu Su. 2022. Thinking about gpt-3 in-context learning for biomedical ie? think again. *arXiv preprint arXiv:2203.08410* .

Alexander Miserlis Hoyle, Ana Marasović, and Noah A Smith. 2021. Promoting graph awareness in linearized graph-to-text generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. pages 944–956.

Jie Huang, Yifan Gao, Zheng Li, Jingfeng Yang, Yangqiu Song, Chao Zhang, Zining Zhu, Haoming Jiang, Kevin Chen-Chuan Chang, and Bing Yin. 2023. Ccgen: Explainable complementary concept generation in e-commerce. *arXiv preprint arXiv:2305.11480* .

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Michael Lewis et al. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *meeting of the 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*. Association for Computational Linguistics, Online, pages 1823–1840.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. pages 74–81.

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

Ye Liu, Yao Wan, Lifang He, Hao Peng, and Philip S Yu. 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.

Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Neurologic decoding:(un) supervised neural text generation with predicate logic constraints. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 4288–4299.

Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019. Step-by-step: Separating planning from realization in neural data-to-text generation. 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)*. Association for Computational Linguistics, Minneapolis, Minnesota, pages 2267–2277.

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.

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.

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.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. pages 1532–1543.

Yevgeniy Puzikov and Iryna Gurevych. 2018. E2E NLG challenge: Neural models vs. templates. In *Proceedings of the 11th International Conference on Natural Language Generation*. Association for Computational Linguistics, Tilburg University, The Netherlands, pages 463–471.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* .

Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press.

Shanya Sharma, Manan Dey, and Koustuv Sinha. 2021. Evaluating gender bias in natural language inference. https://openreview.net/forum?id=bnuU0PzXl0-.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NIPS'14*.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239* .

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.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed H Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, pages 38–45. https://www.aclweb.org/anthology/2020.emnlp-demos.6.

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.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. 2023. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923* .

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*. Association for Computational Linguistics, Seattle, United States, pages 1709–1718. https://doi.org/10.18653/v1/2022.findings-naacl.129.

Chao Zhao, Marilyn Walker, and Snigdha Chaturvedi. 2020. Bridging the structural gap between encoding and decoding for data-to-text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, pages 2481–2491.

Wangchunshu Zhou, Dong-Ho Lee, Ravi Kiran Selvam, Seyeon Lee, and Xiang Ren. 2021. Pre-training text-to-text transformers for concept-centric common sense. In *International Conference on Learning Representations*. https://openreview.net/forum?id=3k20LAiHYL2.

Supplementary Appendix

A Different Input Formats for Example Ordering

We conducted an evaluation to examine the impact of input formats on the quality of generation. We designed three types of input formats, given a concept set: 1) the ordered concepts are concatenated together, 2) the ordered concepts are concatenated together and separated (delimitted) by commas, and 3) the unordered concepts are concatenated with the ordered concepts and denoted by a special "*[ORDERING]*" tokens. An example using the concept set *ski*, *mountain*, *skier*, is provided in Table 5. As shown in Table 6, the performance of the generated sentences does not change significantly

	Parameters	BLEU4	ROUGE-L	METEOR	CIDEr	SPICE
Turbo(zero shot)	135B	13.9	33.5	27.2	8.7	24.0
Babbage (Original)	6.7B	19.7	37.7	27.1	12.0	26.7
Curie(Original)	13B	13.5	32.6	26	10.0	24.6
Babbage (Example)	6.7B	23.6	41.0	29.2	13.6	29.2
Curie (Example)	13B	25.5	41.9	30.0	14.4	30.1

Table 4: Evaluation of the effect of different concept ordering strategies on the LLMs. The results show that a good concept ordering strategy could also help the quality of large model generation.

with the different input formatting methods. However, the best performance among the three input formatting methods is achieved when the concepts are concatenated with the ordered concepts around tokens. This may be attributed to the fact that, with the tokens and ordered concepts, a model might better comprehend the ordering task and use the syntactic knowledge embedded in the pre-trained models (Ou et al., 2022).

Format	Example
w/o Comma	skier ski mountain
Comma	skier, ski, mountain
Token	ski mountain skier `[ORDERING]`
	skier ski mountain `[ORDERING]`

Table 5: Three different input formats given a concept set {*ski*, *mountain*, *skier*}

Concept set	BLEU4	ROUGE-L	METEOR	CIDEr	SPICE
w/o Comma	33.4	45.5	32.6	17.1	32.8
Comma	32.5	45	32.4	16.8	32.6
Token	33.8	45.6	32.8	17.2	33.1

Table 6: The generated evaluation of different inputs formats.

B Concept Ordering on LLMs

We also conducted concept ordering experiments using the CommonGen dataset on LLMs containing more than one billion parameters. Specifically, we selected GPT3 models (Babbage and Curie) and GPT3.5 Turbo from OpenAI as our backbone models. Since the Turbo model does not support fine-tuning, it was evaluated solely in a zero-shot setting. We evaluated the quality of the generated outputs using both the Original ordering and Example ordering strategies on the Babbage and Curie models.

As shown in Table 4, the Example Ordering strategy also enhanced the performance of LLMs on the CommonGen dataset task. However, we observed that the three LLMs underperform compared against the smaller fine-tuned LMs. We notice that this problem also exists in other NLP tasks (Brown et al., 2020; Gutiérrez et al., 2022). Therefore, we examined some sentences generated by these LLMs as presented in Table 3. Under the zero-shot setting, Turbo was capable of generating more complex sentences than those provided in the references, and these sentences were consistent with commonsense. When we try to use Example Ordering Strategy for the Turbo model with the prompt *Given a concept list: [concepts], please generate a sentence that aligns the ordering of the concepts:*, we find that the model could not always follow the concept ordering given even under the few-shot setting.

For the Babbage and Curie models, the Original Ordering strategy often resulted in generated sentences that did not encompass all concepts from the given input set. The Example Ordering strategy ameliorated this issue; however, even when the grammar was correct, some sentences were inconsistent with commonsense (e.g., *A dog catching a Frisbee and throwing it*). We surmise that these issues are the primary reasons why the LLMs did not perform well. As is suggested by OpenAI website, the performance of the model is based on the description given and external content would improve the performance. However, the input of CommonGen only has a single concept set. Potential improvements for the performance of LLMs on such tasks could involve providing more content around the concept set such as description about the relations among the concepts.

C Training Details

We utilised the BERT, BART, and T5 models as implemented in the Transformers library (Wolf et al., 2020). Detailed hyper-parameters are provided in the accompanying bash scripts (submitted as supplementary materials). The primary hyper-parameters were initialised in alignment with the standards set out in the CommonGen paper (Lin et al., 2020). We fine-tuned each model using the ADAM optimiser (Kingma and Ba, 2015). To prevent over-fitting, we adopted an early stopping strategy based on the development set's loss. We train the model with one NVIDIA A6000 GPU and one V100 GPU. Each model could be fine-tuned in less than 6 hours.

The training of the LLMs utilised OpenAI's API.[3] Training a Babbage model incurred a cost of $3 for the CommonGen dataset, while training a Curie model incurred a cost of $14. During model training, each instance would start with the prompt "*Generate a sentence containing all the concepts in the concept set:*" for each concept set and we added a separator " ->" at the end of the prompt. We also use an ending token "/n" at the end of the target sentence.

Models	**Random**	**Probabilistic**	**Example**
BERT-Gen	32*2, 5e-5	32*2, 5e-5	32*2, 5e-5
BART-base	64*4, 3e-5	64*3, 5e-5	64*6, 3e-5
T5-base	64*4, 5e-5	64*3, 5e-5	64*4, 5e-5
BART-large	96*4, 3e-5	64*4, 2e-5	64*4, 3e-5
T5-large	32*4, 2e-5	64*4, 2e-5	64*4, 3e-5
GPT3-babbage	-	-	128, -
GPT3-curie	-	-	128, -

Table 7: Key parameters in experiments. In each cell, the left is the batch size and the right is the learning rate.

[3] https://platform.openai.com/docs/guides