Bridging the Gap between Pre-Training and Fine-Tuning for Commonsense Generation *

Haoran Yang[1], Yan Wang[2], Piji Li[2], Wei Bi[2], Wai Lam[1], Chen Xu[3]
[1]The Chinese University of Hong Kong
[2]Tencent AI Lab
[3]Beijing University of Technology
hryang@se.cuhk.edu.hk

Abstract

Commonsense generation aims to generate a plausible sentence containing all given unordered concept words. Previous methods focusing on this task usually directly concatenate these words as the input of a pre-trained language model (PLM). However, in PLMs' pre-training process, the inputs are often corrupted sentences with correct word order. This input distribution discrepancy between pre-training and fine-tuning makes the model difficult to fully utilize the knowledge of PLMs. In this paper, we propose a two-stage framework to alleviate this issue. Firstly, in pre-training stage, we design a new format of input to endow PLMs the ability to deal with masked sentences with incorrect word order. Secondly, during fine-tuning, we insert the special token [MASK] between two consecutive concept words to make the input distribution more similar to the input distribution in pre-training. We conduct extensive experiments and provide a thorough analysis to demonstrate the effectiveness of our proposed method. The code is available at https://github.com/LHRYANG/CommonGen.

1 Introduction

To investigate machines' ability of generating logical sentences, Lin et al. (2020) propose the Commonsense Generation task. Given a set of concept words, this task is designed to generate a sentence which not only contains the given concepts but also can correctly describe the relations between concepts. An example is shown in Table 1.

Existing methods employ the Pre-trained Languege Models (PLMs) such as BART (Lewis et al., 2020), GPT-2 (Radford et al., 2019) as the backbone to solve this problem. They (Liu et al., 2021; Fan et al., 2020; Wang et al., 2021; Li et al., 2021) usually take the concatenated concepts words as the inputs. However, such processing of inputs causes a huge gap between pre-training and fine-tuning. Specifically, these concept words are unordered which means the order of the input words is inconsistent with the order of these words in the references. It seems incompatible to PLMs pre-trained with ordered words (For BART (Lewis et al., 2020), sentence permutation is adopted, nevertheless, the word order within a sentence remains correct.). As studied by Zhao et al. (2022) and Ou et al. (2022), the word order of inputs can hinder the exploitation of knowledge existing in PLMs. Moreover, even if the word order of inputs is correct, for some LMs (e.g., BART (Lewis et al., 2020), T5 (Raffel et al., 2020)), the inputs are masked sentences during pre-training, while in commonsense generation task, the inputs are unconnected word sequences. This kind of discrepancy also degrades the models' performance.

concept words	*{wear, player, field, jersey}*
references	*The player will wear a jersey while on the field.* *A soccer player wears a jersey on the field.* ...
output of our model	*football player wears a jersey on the field.*

Table 1: An example of Commonsense Generation task

In this paper, we propose a two-stage framework to bridge the gap between pre-training and fine-tuning for this task. Specifically, we firstly propose to introduce a domain-specific pre-training stage using the tasks' training dataset. The pre-training objective is designed to recover original sentences given the *masked* and *shuffled* sentences. Therefore, the PLMs' ability of reasoning out new concepts or relations (mask operation) and processing order-agnostic inputs (shuffle operation) is enhanced. Secondly, in downstream task fine-tuning, we insert the special token [MASK] between two consecutive concept words. This makes the input distribution more similar to the distribution in pre-training. The experimental results shows

*This work was done during an internship at Tencent.

Findings of the Association for Computational Linguistics: EACL 2023, pages 376–383
May 2-6, 2023

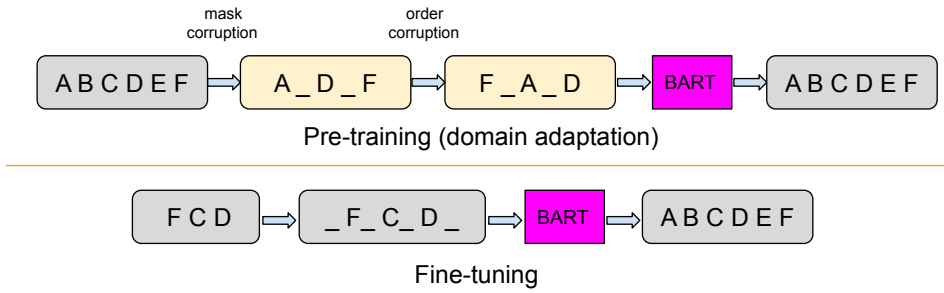

Figure 1: An overview of our model. _ represents the [MASK] token.

that our proposed model can significantly improve the performance of the commonsense generation task. We also conduct experiments to show that our model is superior than baselines in terms of continual learning and few-shot scenarios.

2 Model

We propose a two-stage training framework as shown in Figure 1. We firstly continually pre-train the BART with a newly designed input format. Secondly, we fine-tune the model whose inputs are inserted with the special token [MASK].

Formally, given a concept word set $\mathbf{x} = \{x_1, x_2, \cdots, x_n\} \in X$ (n can be different for different inputs), the task aims to generate a fluent, plausible and grammatically correct sentence $\mathbf{y} = (y_1, y_2, \cdots, y_m) \in Y$ containing all the words in $\mathbf{x}$.

2.1 Domain-Specific Pre-training

Continually pre-training the PLMs on the target domain is beneficial to improving the performance of the target task consistently (Gururangan et al., 2020). We adopt this idea and moreover, we design a new sentence corruption strategy considering that the input words order in target task is shuffled. Below is the procedure for constructing the corrupted inputs for each sentence $\mathbf{y} \in Y$ in training dataset:

1. Randomly select a subset of words in $\mathbf{y}$ and each word is selected with a probability p which is also called the mask probability.
2. Replace the selected words with the special token [MASK]. It should be noted that multiple consecutive [MASK] tokens are merged to one [MASK] token. This allows the PLMs to predict a span (multiple words) based on one [MASK] token, which is more similar to the commonsense generation task as we will see below.
3. The unmasked words are shuffled while the positions of the [MASK] tokens remain unchanged. The corrupted input is denoted by $\tilde{\mathbf{y}}$.

An example of the above process is shown in the upper part of Figure 1. We usually choose a large value for the probability p instead of 15% used by BERT (Devlin et al., 2019). We will study the effect of p (0.5 in our experiment) in Section 3.2. Since a part of concept words and non-concept words are masked, this pre-training process can also enhance PLMs' ability of reasoning out unseen concepts and relations between concepts.

Finally, the pre-training loss function is:

$$\mathcal{L}(\theta) = -\frac{1}{|Y|} \sum_{\mathbf{y} \in Y} \log(\prod_{i=1}^{m} P(y_i | y_{<i}, \tilde{\mathbf{y}}; \theta)) \quad (1)$$

2.2 Fine-tuning

Although the domain-specific pre-training can adapt the PLMs to the target domain and alleviate the problem related to word order, the inputs during fine-tuning are still a list of words while in pre-training for many LMs, the inputs are corrupted sentences with [MASK] tokens. Chada and Natarajan (2021) have shown that aligning the input distribution between pre-training and fine-tuning can boost the few-shot performance on QA tasks. Armed with such finding, we transform the inputs by inserting [MASK] tokens. Formally, given an input $\mathbf{x} = \{x_1, x_2, \cdots, x_n\}$, we transform $\mathbf{x}$ to [1]:

$$\texttt{[MASK]}, x_1, \cdots, \texttt{[MASK]}, x_i, \texttt{[MASK]}, \cdots, x_n, \texttt{[MASK]}$$

Then, we input the transformed $\mathbf{x}$ to the PLM to predict the target $\mathbf{y}$. Through this way, the input distribution is more similar to that in pre-training (especially domain-specific pre-training). This input format is similar to the text infilling task (Donahue et al., 2020), the main differences are that

[1] We send the transformed $\mathbf{x}$ to the tokenizer so that [CLS] and [EOS] will also be added.

Model \ Metrics	ROUGE-2/L		BLEU-3/4		METEOR	CIDEr	SPICE
GPT-2	17.18	39.28	30.70	21.10	26.20	12.15	25.90
UniLM	21.48	43.87	38.30	27.70	29.70	14.85	30.20
T5	22.01	42.97	39.00	28.60	30.10	14.96	31.60
BART	22.23	41.98	36.30	26.30	30.90	13.92	30.60
KG-BART	23.38	44.54	42.10	30.90	32.40	16.83	32.70
NeuroLogic	-	44.70	41.3	30.60	31.00	15.90	31.10
CALM	-	-	-	29.50	31.90	15.61	33.20
EKI-out	24.36	45.42	42.90	32.10	32.00	16.80	32.50
Ours	24.17	44.89	**43.31**	**32.49**	**32.50**	**17.10**	32.81

Table 2: Automatic Evaluation Results.

the words in commonsense generation task are unordered and masked words also account for a large proportion of sentences.

3 Experiments

3.1 Experimental Settings

Dataset We use the CommonGen dataset collected by Lin et al. (2020). The dataset contains 67389/4018/6042 training/development/testing samples with 32651/993/1497 different concept sets (one concept set has multiple references.). For evaluation metrics, we use 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 also report human evaluation score and coverage score. However, due to the space constraint, regarding these two scores, please refer to Appendix **??** for more details.

Baselines We compare our model with several baselines. For PLMs, we choose GPT-2 (Radford et al., 2019), UniLM (Dong et al., 2019), T5 (Raffel et al., 2020), BART (Lewis et al., 2020). We also compare our model with (1) KG-BART (Liu et al., 2021) which incorporates the knowledge graph to BART. (2) NeuroLogic (Lu et al., 2021) which controls the decoding stage to enforce the satisfaction of the given lexical constraints. (3) CALM (Zhou et al., 2021) which designs several self-supervised tasks to obtain a concept-aware language model. (4) EKI-out (Fan et al., 2020) which augments inputs with retrieved sentences from *out-of-domain* corpus. Generally, EKI-out is stronger than other baselines due to the high informativeness of Wikipedia.

Implementation Details We adopt BART-large as the generation model. The max length of $\mathbf{x}$ and $\mathbf{y}$ are set to 48 and 128 respectively. The batch size is set to 32. For Domain-Specific Pre-training, the mask probability p is set to 0.5. The number of training epochs is 10. We use AdamW (Loshchilov and Hutter, 2019) with learning rate 1e-7 to optimize the model. For fine-tuning, the model is optimized using AdamW with an initial learning rate of 2e-5. We also employ linear warmup with steps 10000. We save the model with the highest Rouge-L score on development set for testing.

3.2 Results

Main Results As summarized in Table 2, Ours can generally achieve better performance than all the baselines on BLEU, MENTOR, CIDEr. On ROUGE, Ours outperforms most of the baselines except EKI-out which facilitates the Wikipedia as the external corpus. On SPICE, Ours is superior than most of the baselines except CALM.

Ablative Results We conduct ablation study with three variants. The results are shown in Table 4. We can see that the performance of -mask (Ours without adding `[MASK]` during fine-tuning) and -pretraining (Ours without pretraining) are inferior than Ours. -both (Ours with neither) obtains the worst performance. We can also observe that adding mask and adding pretraining have similar degree of improvement compared to -both. Moreover, since there are numerous ways to insert `[MASK]` to inputs (different positions or different numbers), we compare our model with Random Mask: during pre-training and fine-tuning, *one* mask token is randomly inserted into the corrupted inputs. We can see from Table 4 that Ours outperforms Random Mask. Moreover, we provide some generated examples in Appendix B.

Model	Training on	Evaluation on	ROUGE-L	BLEU-4	METEOR	CIDEr	SPICE
Ours	Size = 3	Size = 3	45.25	18.74	24.06	14.64	34.92
	Size = 4	Size = 3	46.31(+1.06)	19.33(+0.59)	25.76(+1.70)	15.47(+0.83)	36.89(+1.97)
		Size = 4	44.97	31.02	31.25	16.14	31.68
	Size = 5	Size = 3	45.62(-0.69)	19.38(+0.05)	25.50(-0.26)	15.26(-0.21)	36.51(-0.38)
		Size = 4	45.27(+0.30)	32.00(+0.98)	31.63(+0.38)	16.49(+0.35)	31.46(-0.22)
		Size = 5	43.53	30.98	30.93	16.12	31.01
-mask	Size = 3	Size = 3	44.80	17.89	24.24	14.68	34.28
	Size = 4	Size = 3	45.52(+0.72)	17.46(-0.43)	24.71(+0.47)	14.58(-0.10)	35.38(+1.10)
		Size = 4	44.29	31.53	30.98	16.24	32.04
	Size = 5	Size = 3	45.36(-0.16)	17.75(+0.29)	24.77(+0.06)	14.69(+0.11)	35.88(+0.50)
		Size = 4	44.49(+0.20)	30.99(-0.54)	30.88(-0.10)	16.09(-0.15)	31.17(-0.87)
		Size = 5	42.69	29.16	29.63	15.11	30.05

Table 3: Continual Learning Results. The rows with the same color represents the same domain we evaluate the model on. The red number in parentheses is the improvement compared with the previous time step on the same domain. For example, (+1.06) = $46.31 - 45.25$, (-0.69) = $45.62 - 46.31$, (+0.30) = $45.27 - 44.97$.

Model	ROUGE-L	BLEU-4	METEOR	CIDEr	SPICE
Ours	44.89	32.49	32.50	17.10	32.81
-mask	44.67	31.66	32.09	16.51	32.11
-pretraining	44.35	31.60	31.87	16.57	32.33
-both	43.56	29.61	30.87	15.61	30.93
Random Mask	44.43	31.64	32.23	16.69	32.36

Table 4: Variant Analysis Results.

Model	Ours	Ours-w/o-mask	Ours-w/o-pretraining	Ours-w/o-both
human score	4.534	4.367	4.467	4.084
coverage	97.48	96.03	96.07	93.05

Table 5: Human Evaluation Score and Coverage Score

p	ROUGE-L	BLEU-4	METEOR	CIDEr	SPICE
0.2	44.36	31.21	31.16	16.48	32.33
0.4	44.32	30.99	32.05	16.64	32.48
0.5	**44.89**	**32.49**	32.50	**17.10**	**32.81**
0.6	44.37	31.95	**32.64**	16.91	32.72
0.8	44.58	32.25	32.38	16.83	31.90

Table 6: Effects of p.

Effects of Hyperparameter p We investigate the effects of the mask probability p. As presented in Table 6, the performance is the best when p equals 0.5. The reason may be that if p is too large, it is hardly possible to recover corrupted sentences during pre-training. However, if p is too small, most of the masked tokens are not concept words, thus the pre-trained model cannot learn the relations between concepts.

Human Evaluation & Coverage To provide more perspective of the generation quality, we report the human evaluation score and coverage score. For human evaluation, we randomly select 30 sentences and each sentence is given a score ranging from one to five to assess the holis- tic quality. We report the average value of two annotators. The concept coverage score is the average percentage of input concepts that are present in lemmatizatized outputs. The results are shown in Table 5. We can see that Ours achieves the highest human evaluation score and coverage score and Ours-w/o-pretraining achieves a slightly better performance than Ours-w/o-mask, indicating that inserting `[MASK]` to the input is more important than adding the pretraining stage.

3.3 Few-Shot Scenario

We investigate the performance of our model under few-shot scenario. We randomly select $n \in \{16, 32, 64\}$ samples from original training dataset as the new training dataset and the testing dataset remains unchanged. The learning rate is set to 2e-5. Table 7 shows the results. We can see that inserting `[MASK]` to the inputs can significantly boost the performance on all the metrics. Combined with the result in Table 2, we can conclude that inserting `[MASK]` to the inputs is beneficial to the performance on both full-data and few-shot settings.

3.4 Continual Learning Scenario

We investigate the performance of our model under continual learning scenario (Biesialska et al., 2020). We regard concept sets with the same length as a domain. The details of the dataset are described in Appendix A. The model is trained sequentially from the domain with length 3 to the domain with length 5. After the model is trained on a new domain, we also evaluate it on previous domains to measure the backward transfer degree. Backward transfer means that learning a new task may hurt (negative backward transfer) or improve (positive backward transfer) the performance of previously

n	model	ROUGE-L	BLEU-4	METEOR	CIDEr	SPICE
16	Ours	35.04	16.22	21.98	9.02	21.93
	-pretraining	33.49	12.25	19.29	7.46	20.42
	-mask	32.33	7.5	19.23	6.16	17.24
	-both	31.74	6.75	19.83	6.18	16.30
32	Ours	35.66	19.72	23.20	10.00	21.43
	-pretraining	35.92	16.21	21.35	8.88	20.78
	-mask	33.98	15.17	21.34	8.63	18.36
	-both	33.15	11.73	19.36	7.51	19.13
64	Ours	38.94	22.15	25.96	12.31	26.64
	-pretraining	38.17	21.17	25.48	11.6	26.27
	-mask	35.75	18.79	24.73	10.73	24.00
	-both	35.04	15.43	22.58	9.18	21.23

Table 7: Few-Shot Setting.

learned tasks (Lopez-Paz and Ranzato, 2017). The results are shown in Table 3. We can see that Ours generally obtains better performance than -mask. Also, we can see that our model achieves a larger positive backward transfer and a smaller negative backward transfer (forget less) than -mask. For example, ROUGE-L of the domain with concept set size 3 is changed from 45.25 to 46.31 (improved by 1.06) after the model is trained on the domain with concept size 4 for our model. While for -mask, the improvement is only 0.72. Therefore, we can conclude that bridging such a gap is effective under continual learning setting.

4 Conclusion

We study the gap issue between pre-training and fine-tuning for commonsense generation task. We propose a two-stage training framework which is composed of a domain-specific pre-training stage and a fine-tuning stage. Pre-training stage aims to recover the masked and shuffled sentences which could enhance the models' ability of processing unordered inputs and reasoning out the relations and concepts. Inserting `[MASK]` to the inputs during fine-tuning have also been demonstrated very useful. Experimental results show that our model is superior than many baselines, especially under few-shot setting.

Acknowledgement

The work described in this paper is supported by Tencent AI Lab Rhino-Bird Gift Fund (YD4200722).

Limitations

In this work, we study the gap between pre-training and fine-tuning for commonsense generation task. Despite the promising experimental results, there are still several limitations of our work:

1. The order issue is still not fully solved since the original pre-training stage uses ordered sentences. Our proposed domain-specific training stage can only alleviate this issue instead of completely solving it.

2. During fine-tuning, the optimal positions and an optimal number of the `[MASK]` tokens are not well solved.

References

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *ECCV*.

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.

Magdalena Biesialska, Katarzyna Biesialska, and Marta R. Costa-jussà. 2020. Continual lifelong learning in natural language processing: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6523–6541, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Rakesh Chada and Pradeep Natarajan. 2021. FewshotQA: A simple framework for few-shot learning of question answering tasks using pre-trained text-to-text models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6081–6090, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Chris Donahue, Mina Lee, and Percy Liang. 2020. Enabling language models to fill in the blanks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2492–2501, Online. Association for Computational Linguistics.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*.

Zhihao Fan, Yeyun Gong, Zhongyu Wei, Siyuan Wang, Yameng Huang, Jian Jiao, Xuanjing 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, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Haonan Li, Yeyun Gong, Jian Jiao, Ruofei Zhang, Timothy Baldwin, and Nan Duan. 2021. KFCNet: Knowledge filtering and contrastive learning for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2918–2928, Punta Cana, Dominican Republic. 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.

Ye Liu, Yao Wan, Lifang He, Hao Peng, and Philip S. Yu. 2021. Kg-bart: Knowledge graph-augmented bart for generative commonsense reasoning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(7):6418–6425.

David Lopez-Paz and Marc' Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

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, Online. Association for Computational Linguistics.

Zebin Ou, Meishan Zhang, and Yue Zhang. 2022. On the role of pre-trained language models in word ordering: A case study with bart.

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.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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

Han Wang, Yang Liu, Chenguang Zhu, Linjun Shou, Ming Gong, Yichong Xu, and Michael Zeng. 2021. Retrieval enhanced model for commonsense generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3056–3062, Online. Association for Computational Linguistics.

Chao Zhao, Faeze Brahman, Tenghao Huang, and Snigdha Chaturvedi. 2022. Revisiting generative commonsense reasoning: A pre-ordering approach.

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*.

A Continual Learning

We introduce how to construct the dataset used for continual learning. Table 8 shows the distribution of the original dataset. Since there is no testing instances whose concept set size is 3. We randomly sample a number of instances with concept size 3 from the training dataset. Also, since the dataset is unbalanced (the number of instances belonging to the domain with concept size 3 is far larger than that in other domains.) We re-sample the instances to make the dataset more balanced. The statistic of the continual learning setting dataset is shown in Table 9.

Statistics	**Train**	**Dev**	**Test**
Sentences	67,389	4,018	6,042
Concept-Sets	32,651	993	1,497
-Size = 3	25,020	493	-
-Size = 4	4,240	250	747
-Size = 5	3,391	250	750

Table 8: Statistics of Original Dataset.

Statistics	**Train**	**Dev**	**Test**
-Size = 3	5,867	1,819	2,170
-Size = 4	5,352	1,137	2,993
-Size = 5	3,436	1,062	3,049

Table 9: Statistics of Continual Learning Dataset.

B Generated Examples

We list some examples generated by our proposed model and ablative models, which are shown in Table 10.

concept words	*{sheep, wool, shave, hold}*
Ours	*A man is holding a sheep and shaving its wool.*
Ours-w/o-pretraining	*A woman holds a sheep and shaves its wool.*
Our-w/o-mask	*A man is holding a sheep and shaving it with wool.*
Our-w/o-both	*sheep holding their wool in their beaks as they shave.*
concept words	*{stand, fence, feed, goat}*
Ours	*A goat stands at the fence to be fed.*
Ours-w/o-pretraining	*goats standing next to a fence to feed.*
Our-w/o-mask	*A goat standing next to a fence to feed.*
Our-w/o-both	*A goat stands at the fence to feed a goat.*
concept words	*{hold, bag, popsicle, eat, chip}*
Ours	*A boy is eating a popsicle while holding a bag of chips.*
Ours-w/o-pretraining	*A girl is holding a bag of chips and eating a popsicle.*
Our-w/o-mask	*A man holding a bag of chips and a popsicle to eat.*
Our-w/o-both	*A man holding a bag of chips and a popsicle eats a chip.*

Table 10: Generated Examples