Pre-training Text-to-Text Transformers to Write and Reason with Concepts

Wangchunshu Zhou*
Beihang University
zhouwangchunshu@buaa.edu.cn

Dong-Ho Lee*
University of Southern California (USC)
dongho.lee@usc.edu

Ravi Kiran Selvam
USC
rselvam@usc.edu

Seyeon Lee
USC
seyeonle@usc.edu

Bill Yuchen Lin
USC
yuchen.lin@usc.edu

Xiang Ren
USC
xiangren@usc.edu

Abstract

Pretrained language models (PTLM) have achieved impressive results in natural language understanding (NLU) and generation (NLG) tasks. However, current pre-training objectives do not explicitly model the relational and compositional commonsense knowledge about everyday concepts, which is crucial to many downstream tasks requiring commonsense reasoning. To augment PTLMs with common sense, we propose generative and contrastive objectives as intermediate self-supervised pre-training tasks between general pre-training and downstream task-specific fine-tuning. We also propose a joint training framework to unify generative and contrastive objectives so that these objectives can be more effective. Our proposed objectives can pack more commonsense knowledge into the parameters of a pre-trained text-to-text transformer without relying on external knowledge bases, yielding better performance on both NLU and NLG tasks. We apply our method on a pre-trained T5 model in an intermediate task transfer learning fashion to train a *concept-aware language model* (**CALM**) [2], and experiment with five commonsense benchmarks (four NLU tasks and one NLG task).

1 Introduction

Pre-trained language models (PLTMs) such as BERT [1] and T5 [2] have revolutionized the field of NLP, yielding impressive performance on various conventional natural language understanding (NLU) and generation (NLG) tasks. BERT and its novel variants such as RoBERTa [3] and ALBERT [4] capture syntactical and semantic knowledge mainly from the pre-training task of *masked language modeling*, while T5-style models such as BART [5] instead focus on *masked span infilling* tasks. Though yielding better performance on downstream tasks, these pre-training objectives, however, do not explicitly guide the models to reason with concept-centric commonsense knowledge from language. This leaves room for equipping current PTLMs with richer commonsense reasoning ability.

Model mistakes in commonsense reasoning become a bottleneck for current PTLMs [6]. Towards augmenting PTLMs with more knowledge, prior works mainly focus on training larger models [7], adding knowledge-specific architectures [8], or incorporating knowledge bases for pre-training [9]. In this paper, we instead look to explicitly *teach* pre-trained models to write and reason with common concepts through novel pre-training strategies.

*Equal Contribution. The work has done when Wangchunshu was visiting USC.

[2]Code and data have been uploaded and will be published: https://anonymous.4open.science/repository/6fdeed55-ec2c-4ffa-aee8-0cc3b7f5ade5

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

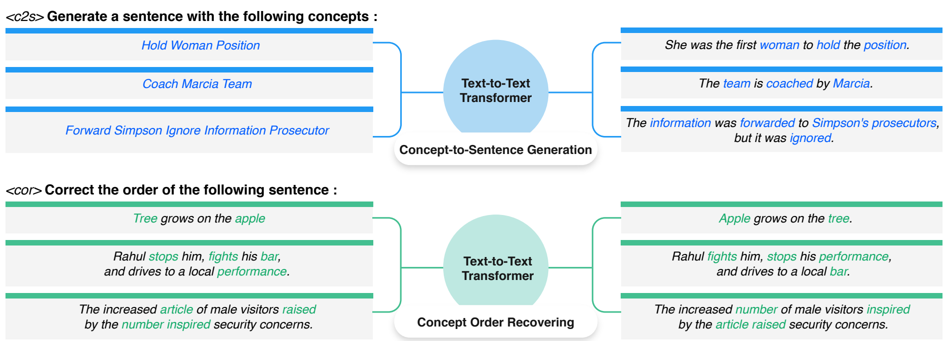

Figure 1: **Overview of Generative self-supervised pre-training objectives.**

We present two kinds of self-supervised pre-training tasks: **concept-to-sentence generation (C2S)** and **concept order recovering (COR)**. C2S trains the pre-trained model to compose ("write") sentences given a set of concepts, and expects the generated sentences to be fluent and plausible in terms of commonsense. COR aims to teach models to detect and revise a corrupted sentence with incorrect ordering of concepts. As illustrated in Figure 1, both tasks require a pre-trained model to recall relevant commonsense facts about the concepts and to understand the underlying commonsense relations between them. These pre-training objectives can together explicitly encourage the model to capture the relational concept-centric commonsense knowledge and perform compositional reasoning.

We evaluate our method in an intermediate-task transfer learning setting [10] based on the pre-trained T5-base model to train a *Concept-Aware Language Model* (**CALM**). CALM consistently outperforms T5-base on four commonsense-NLU datasets (i.e., CommonsenseQA, OpenBookQA, PIQA, and aNLI) and CommonGEN, a NLG dataset. Our results and careful ablation studies demonstrate the potential of our method to serve as a "plug-and-play" method for any pre-trained text-to-text transformer before fine-tuning on commonsense-related tasks. To the best of our knowledge, our work is the first to investigate concept-centric self-supervised objectives that improve both generative and discriminative commonsense reasoning ability of a pre-trained language model.

2 Self-supervised Objectives for Concept-centric Learning

2.1 Generative Objectives

Similar to many other pre-training tasks such as masked language modeling, we aim to teach models to recover original sentences from corrupted inputs, which is often regarded as a *denoising* process. We propose two generative self-supervised pre-training objectives: concept-to-sentence generation (C2S) and concept order recovering (COR).

Concept Extraction. Given an input $\mathbf{x} = [x_1, x_2, \ldots, x_n]$, we first conduct part-of-speech tagging for the sentence and extract *Verb*, *Noun*, and *Proper Nouns* from the sentence to use as concepts[3]. Next, we form concept-sets $\mathcal{C} = [v_1, v_2, \ldots, v_p; n_1, n_2 \ldots, n_q]$ where v_i and n_i denotes the i-th verb or noun/proper noun concept (token) in $\mathbf{x}$. We denote $\mathcal{C}_v$ and $\mathcal{C}_n$ as the set of verb and noun/proper noun concepts respectively in $\mathcal{C}$. (i.e. $\mathcal{C}_v = [v_1, v_2, \ldots, v_p]$ and $\mathcal{C}_n = [n_1, n_2, \ldots, n_q]$.)

Concept-to-Sentence Generation (C2S). The concept-to-sentence generation (C2S) objective requires the text-to-text transformer to recover the original sentence given only a few unordered keywords of the sentence. Specifically, given a sentence, we shuffle the extracted concept-set $\mathcal{C}$ to create the perturbed source sequence and train the model to generate the original sentence with a prefix (denoted as `<c2s>`) as described in Fig. 1. Formally, the C2S objective can be formulated as:

$$L_{c2s} = \mathbb{E}\Big(\sum_{i=1}^{n} -\log p(x_i | \texttt{<c2s>}; \textsc{permute}(\mathcal{C}); x_{1:i-1})\Big) \quad (1)$$

where the PERMUTE() function randomly shuffle the concepts in the concept-set. This objective requires the model to construct an acceptable commonsense sentence by adhering to and reasoning over the commonsense relations between the given concepts. Therefore, relational commonsense knowledge is implicitly injected into the parameters of the model, motivated by the task [11].

[3]We split the concepts with multiple tokens (under Spacy tokenization) into single token to ensure the concepts discussed afterwards all contain a single token.

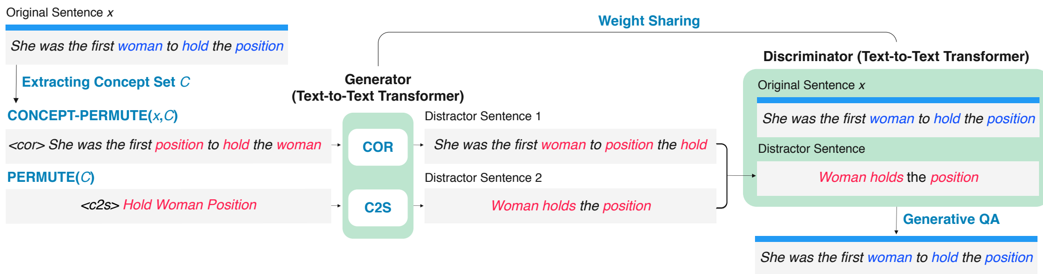

Figure 2: **Proposed Joint Training Framework.** Given an input sentence x, we extract concept-set $\mathcal{C}$. Given x and $\mathcal{C}$, we produce corrupted source sequence x' either for C2S and COR. The generator trained with the corresponding objective recovers sentences as distractors x'' to the discriminator. The discriminator is trained to distinguish truth sentences from randomly selected distractor objectives.

Concept Order Recovering (COR). As for the concept order recovering (COR) objective, we shuffle the order of concept in a sentence and train the model to recover the original sentence, as illustrated in Figure 1. The noise introduced by concept shuffling is different from that by traditional self-supervised objectives like mask language modeling and mask span prediction because the corrupted source sentences are in general complete and grammatically correct, while not acceptable in terms of commonsense because the order and relation between concepts are shuffled. By training the model to detect and correct the disorder of concepts in a sentence, the model is expected to acquire some relational commonsense knowledge like "apple generally grows on a tree" instead of "tree grows on an apple." Formally, the COR objective can be formulated as:

$$L_{cor} = \mathbb{E}\Big(\sum_{i=1}^{n} -\log p(x_i|\texttt{<cor>}; \text{CONCEPT-PERMUTE}(\mathbf{x}, \mathcal{C}); x_{1:i-1})\Big), \tag{2}$$

where <cor> is the prefix for the COR objective illustrated in Figure 1. The function CONCEPT-PERMUTE$(\mathbf{x}, \mathcal{C})$ permutes the order between concepts in the same category (i.e. noun or verb) in the sentence, which can be formally defined as:

$$\text{CONCEPT-PERMUTE}(\mathbf{x}, \mathcal{C}) = [x'_1, x'_2, \ldots, x'_n] \text{ where } x'_i = \begin{cases} x_i & x_i \notin \mathcal{C} \\ \text{PERMUTE}(\mathcal{C}_v)[j] & x_i = v_j \\ \text{PERMUTE}(\mathcal{C}_n)[j] & x_i = n_j \end{cases} \tag{3}$$

The key difference between our proposed objectives and others is that our objectives requires the model to capture the relational commonsense knowledge and perform compositional commonsense reasoning in order to successfully reconstruct the original sentence; while traditional pre-training objectives mainly focuses on general token-level patterns.

2.2 Contrastive Objective

The contrastive objective encourages the pre-trained model to distinguish the real sentence from a distractor sentence: a sentence that is similar to the real sentence, generally grammatically correct, but may not follow commonsense. We expect it to improve the pre-trained model's discriminative commonsense reasoning ability so that the model's performance on commonsense-reasoning-discriminative tasks, like CommonsenseQA, can be improved. We formulate the contrastive objective as a **Generative QA** task. Detailed objective is described in the appendix A.1.

3 Joint Training with Generative and Contrastive Objectives

We argue that these two objectives can mutually reinforce each other: the generated sentences from the generative objective can help the contrastive module learn to distinguish commonsense sentences from less plausible ones. Therefore, we propose a joint training framework to unify generative objectives and contrastive objectives by using the generator to produce *distractors* for learning towards contrastive objective. Detailed illustration is in the appendix A.2.

4 Experiments

In this section, motivated by the observation of [10] that tasks requiring commonsense reasoning ability generally serve as good intermediate task, we test our method in the intermediate task transfer

Methods	CSQA	OBQA	PIQA	aNLI	CommonGEN			
	Accuracy				BLEU-4	METEOR	CIDEr	SPICE
T5-base	61.88(±0.08)	58.20(±1.0)	68.14(±0.73)	61.10(±0.38)	24.90	31.20	12.99	32.40
T5-base w/ additional epochs	61.92(±0.45)	58.10(±0.9)	68.19(±0.77)	61.15(±0.52)	25.10	31.00	13.12	32.40
T5-base + SSM	62.08(±0.41)	58.30(±0.8)	68.27(±0.71)	61.25(±0.51)	25.20	31.20	13.28	32.40
CALM (Generative-Only)	62.28(±0.36)	58.90(±0.4)	68.91(±0.88)	60.95(±0.46)	25.80	31.20	13.81	32.60
CALM (Contrastive-Only)	62.73(±0.41)	59.30(±0.3)	<u>70.67</u>(±0.98)	61.35(±0.06)	25.50	31.20	13.58	32.60
CALM (Mix-only)	<u>63.02</u>(±0.47)	<u>60.40</u>(±0.4)	70.07(±0.98)	<u>62.79</u>(±0.55)	<u>26.00</u>	31.20	<u>13.82</u>	<u>32.80</u>
CALM (w/o Mix warmup)	62.18(±0.48)	59.00(±0.5)	69.21(±0.57)	61.25(±0.55)	25.80	31.20	13.77	32.60
CALM	**63.32(±0.35)**	**60.90(±0.4)**	**71.01(±0.61)**	**63.20(±0.52)**	**26.40**	**31.40**	**13.88**	**33.00**

Table 1: **Experimental results.** The first group of models are baselines. The models in the middle group are trained with the proposed objectives independently and the last group of models are trained by joint training. Best models are bold and second best ones are <u>underlined</u> within each metric.

Methods	CSQA	OBQA	PIQA	aNLI	CommonGEN			
	Accuracy (official dev)				BLEU-4	METEOR	CIDEr	SPICE
BERT-large	57.06(±0.12)	60.40(±0.6)	67.08(±0.61)	66.75(±0.61)	-	-	-	-
T5-large	69.81(±1.02)	61.40(±1.0)	72.19(±1.09)	75.54(±1.22)	28.60	30.10	14.96	31.60
CALM-large (Mix-only)	70.26(±0.23)	62.50(±1.0)	73.70(±1.09)	75.99(±1.26)	<u>29.20</u>	31.30	<u>15.24</u>	<u>33.10</u>
CALM-large	<u>71.31(±0.04)</u>	**66.00(±1.0)**	<u>75.11(±1.65)</u>	<u>77.12(±0.34)</u>	**29.50**	**31.90**	**15.61**	**33.20**
RoBERTa-large[4]	**71.81(±0.25)**	<u>63.90(±0.8)</u>	**76.90(±0.62)**	**82.35(±0.54)**	-	-	-	-

Table 2: **Experimental results on large model.** Comparison between large models of other PTLMs and CALM. Best models are bold and second best ones are <u>underlined</u> within each metric.

setting. Specifically, we initialize our model with T5-base, a pre-trained text-to-text transformer model, and training the model with our proposed method as intermediate task before fine-tuning and target downstream tasks. Another reason for adopting this setting is because we expect our method to serve as a "plug-and-play" method that can be applied to any pre-trained text-to-text transformer by simply continually training for a few steps.

Details for Pre-training and Fine-tuning CALM is continually pre-trained with our proposed self-supervised objectives as intermediate tasks based on the pre-trained T5-base model following the setting in [10]. We randomly sample 500K sentences from the English Wikipedia corpus, which is used for pre-training BERT and its variants, as the source dataset for our proposed self-supervised objectives which serve as intermediate tasks. We then fine-tune the CALM on each downstream task individually and report the average performance of three runs with different random seeds for fine-tuning on each dataset since the performance is sensitive to different random seeds. Training details and hyper-parameter settings are presented in Appendix A.3 and A.4.

Datasets We categorize commonsense benchmark datasets into discriminative and generative tasks. Discriminative tasks are classification tasks while generative tasks are text generation tasks. We consider four datasets for discriminative task: **CommonsenseQA** [12], **OpenbookQA** [13], **PIQA** [14], **aNLI** [15] and one dataset for generative task: **CommonGEN** [11]. Details are in Appendix A.5.

Compared Methods We compare our model with following models continually trained with different intermediate tasks based on the pre-trained T5-base model. Details are described in Appendix A.6.

Evaluation Metrics For discriminative tasks, we choose accuracy as our metric following other conventional question answering tasks. For generative tasks, we report automated metrics including BLEU [16], METEOR [17], CIDEr [18], and SPICE [19] following the leaderboard of COMMONGEN [11]. Results for COMMONGEN are on the test set and others are on the development set.

4.1 Experimental Results

The result is presented in Table 1. First, we can see that our CALM model consistently outperforms the backbone T5-base model on all five datasets by a margin range from 1.5 to 2.9 accuracy on discriminative tasks and 1.5/0.6 BLEU/SPICE score on CommonGEN. This is an impressive result since we are only performing intermediate training on a relatively small dataset for only around 20k updates. It demonstrates the potential of our method for serving as a "plug-and-play" method for packing more commonsense knowledge into a pre-trained text-to-text transformer.

In addition, we can observe that both the proposed generative and contrastive objective outperforms the backbone T5-base model, as well as its variants that continually pre-trained with the original masked span prediction objective and the concept-specific salient span masking scheme, when applied independently. Note that we find the variant of salient span masking that focuses on concept is not very effective. We suspect this is because the resulting training data would be somewhat similar to the original text infilling objective because concepts are very common in the corpus and we only train for a few steps. The combination of the generative and contrastive objectives (i.e., CALM(Mix-only)) yields further improvement upon the model trained independently with either generative or contrastive objectives. Also, we find that the CALM model consistently outperforms CALM(Mix), demonstrating the effectiveness of the proposed joint training framework. Applying joint training directly on top of a pre-trained model (i.e., CALM(w/o Mix warmup)) does not work very well, demonstrating the necessity of applying mixed training to initialize the model before starting joint training.

Methods	Params	CommonGEN			
		BLEU-4	METEOR	CIDEr	SPICE
GPT-2 [20]	774M	21.10	26.20	12.15	25.90
UniLM [21]	340M	27.70	29.70	14.85	30.20
BART [22]	406M	26.30	30.90	13.92	30.60
T5-base [2]	220M	16.40	23.00	9.16	22.00
T5-large[5] [2]	770M	28.60	30.10	14.96	31.60
KG-BART[6] [23]	406M	**30.90**	**32.40**	**16.83**	32.70
T5-base (our implementation)	220M	24.90	31.20	12.99	32.40
CALM	220M	26.40	31.40	13.88	**33.00**

Table 3: Comparison between PTLMs on CommonGEN. Above baselines are reported number in the leaderboard. T5-base(our implementation) uses different hyperparmeter setting than that reported in the leaderboard.

To further confirm the effectiveness of our approach, we also apply our method to continually pre-train T5-large with the same data and number of training steps. We then compare the performance of the resulting model with that of the original T5-large model in Table 10. We find that both the proposed training objectives and the joint training framework consistently improve upon the original T5-large, showing our approach is effective for models with different sizes. Our model also outperforms BERT-large by a large margin. However, our model performs slightly worse compared to RoBERTa-large. We suspect this is because RoBERTa-large is optimized for more steps than T5-large and our CALM-large. This is also observed in many other tasks and datasets. Furthermore, We discuessed various ablation studies for performance analysis in Appendix A.7.

5 Conclusion

We propose novel self-supervised strategies that encourage the model to focus on concept-centric information that is related to commonsense understanding and reasoning instead of simple word co-ocurrence patterns so that the commonsense learning capability of pre-trained text-to-text transformers can be improved. Despite merely continually pre-trained on a small dataset with only around 20k steps, our CALM model consistently outperforms the T5-base model on all commonsense-related datasets, and even yields better performance compared with some larger size PTLMs on the CommonGEN dataset. The performance gain is larger when we use fewer examples for fine-tuning on different downstream tasks, indicating that CALM effectively encodes more commonsense knowledge and rely less on fitting superficial patterns of datasets compared to traditional pre-trained language models.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[2] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

[3] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[4] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[5] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[6] Ernest Davis and Gary Marcus. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM*, 58:92–103, 08 2015.

[7] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[8] Matthew E Peters, Mark Neumann, Robert L Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. Knowledge enhanced contextual word representations. *arXiv preprint arXiv:1909.04164*, 2019.

[9] Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. In *International Conference on Learning Representations*, 2020.

[10] Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R Bowman. Intermediate-task transfer learning with pretrained models for natural language understanding: When and why does it work? *arXiv preprint arXiv:2005.00628*, 2020.

[11] Bill Yuchen Lin, Wangchunshu Zhou Ming Shen, , Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. Commongen: A constrained text generation challenge for generative commonsense reasoning. *CoRR*, abs/1911.03705, 2019.

[12] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*, 2018.

[13] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

[14] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

[15] Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Scott Wen-tau Yih, and Yejin Choi. Abductive commonsense reasoning. *arXiv preprint arXiv:1908.05739*, 2019.

[16] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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, July 2002. Association for Computational Linguistics.

[17] Satanjeev Banerjee and Alon Lavie. 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, June 2005. Association for Computational Linguistics.

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

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

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

[21] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 13063–13075. 2019.

[22] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 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*, Online, July 2020. Association for Computational Linguistics.

[23] Ye Liu, Yao Wan, Lifang He, Hao Peng, and Philip S. Yu. Kg-bart: Knowledge graph-augmented bart for generative commonsense reasoning, 2020.

[24] WA Falcon. Pytorch lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning*, 3, 2019.

[25] 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. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.

[26] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*, 2020.

[27] Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model? *arXiv preprint arXiv:2002.08910*, 2020.

[28] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 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, November 2019. Association for Computational Linguistics.

[29] Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vassilis Plachouras, Tim Rocktäschel, et al. Kilt: a benchmark for knowledge intensive language tasks. *arXiv preprint arXiv:2009.02252*, 2020.

[30] Robyn Speer and Catherine Havasi. Representing general relational knowledge in ConceptNet 5. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3679–3686, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA).

[31] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[32] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.

[33] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003.

[34] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305, 2017.

[35] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[36] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

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

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[39] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020.

[40] Corby Rosset, Chenyan Xiong, Minh Phan, Xia Song, Paul Bennett, and Saurabh Tiwary. Knowledge-aware language model pretraining. *arXiv preprint arXiv:2007.00655*, 2020.

[41] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*, 2019.

[42] Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Cuihong Cao, Daxin Jiang, Ming Zhou, et al. K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*, 2020.

A Appendix

A.1 Detailed Illustration of Contrastive Objective

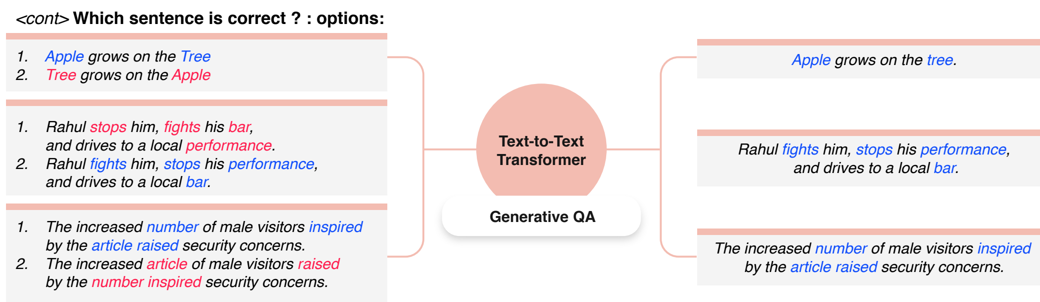

Figure 3: **Overview of Contrastive self-supervised pre-training objectives.** Generative QA style contrastive objective requires model to distinguish truth sentences from less plausible ones.

we take the concatenation of a prefix <cont> (question / context), the real sentence x (answer), and the distractor x' (distractor) as the input and train the model to output the real sentence x. Formally, we have the loss function of the contrastive objective defined as:

$$L_{cont} = \mathbb{E}\big(-\log p(x|\texttt{<cont>}; \text{PERMUTE}(x; x'))\big), \tag{4}$$

where the <cont> is described in Figure 3. The distractor x' is either constructed by concept shuffling as described previously (i.e. $x' = \text{CONCEPT-PERMUTE}(\mathbf{x}, \mathcal{C})$) when used independently, or generated by a generator trained with the aforementioned generative objectives when used in the joint training framework, which will be described in the next section.

A.2 Detailed Illustration of Joint Learning

Specifically, we have a **generator** G_θ (trained with the generative objectives) and a **discriminator** D_ϕ (trained with the contrastive objective). Given an input sentence $\mathbf{x}$, we first use the method for either C2S or COR to produce the corrupted source sequence $\mathbf{x}'$. Then, we use the generator G_θ trained with the corresponding objective to generate the recovered sentence $\mathbf{x}'' = G_\theta(x')$. We then take $\mathbf{x}''$ as the distractor to train the discriminator D_ϕ with the contrastive objective. The loss function of the proposed joint training framework consists of two parts: the first part is the loss of generative objectives, which is identical to the loss described in Eq.(1) and Eq.(2) and is used to update the generator G_θ. The second part is the loss of the contrastive objective as described in Eq.(4), which can be formulated as:

$$L_{cont\_joint\_c2s} = \mathbb{E}\Big(-\log \mathrm{D}_\phi(y|\texttt{<cont>}; x; \mathrm{G}_\theta(\texttt{<c2s>}; \text{PERMUTE}(\mathcal{C}))\Big) \tag{5}$$

$$L_{cont\_joint\_cor} = \mathbb{E}\Big(-\log \mathrm{D}_\phi(y|\texttt{<cont>}; x; \mathrm{G}_\theta(\texttt{<cor>}; \text{CONCEPT-PERMUTE}(\mathbf{x}, \mathcal{C}))\Big) \tag{6}$$

where $L_{cont\_joint\_c2s}$ and $L_{cont\_joint\_cor}$ is the contrastive loss with the distractor generated with either the C2S or the COR objective. We then have the overall objective for the joint training framework defined as :

$$L_{joint} = (L_{c2s} + L_{cor}) + \beta(L_{cont\_joint\_c2s} + L_{cont\_joint\_cor}). \tag{7}$$

L_{c2s} and L_{cor} are defined in Eq.(1) and Eq.(2) respectively and β is a hyperparameter controlling the relative weight between the generative and contrastive objectives. Note that since we would like to inject both generative and discriminative commonsense reasoning ability into the parameters of a single text-to-text transformer, we share the parameters between the generator G_θ and the discriminator D_ϕ.

Finally, we describe the overall procedure to apply the proposed self-supervised objectives and the joint training framework on a pre-trained text-to-text transformer. We apply a two-stage training strategy. During the first stage, we apply our proposed generative and contrastive objectives individually on the model in a multi-task learning fashion with different prefixes. This provides a good starting point for the second stage where the joint training framework is applied. We summarize the workflow of our method in Algorithm 1.

Algorithm 1: Pre-training Concept-Aware Language Model (CALM).

Input: Text-to-Text Transformer T_θ, Text corpus X=[x_1, x_2,..., x_n].
repeat
  for *each* $x_i \in X$ **do**
    Extract the concept-set $\mathcal{C}_i$;
    Construct the distractor sentence $x' = \text{CONCEPT-PERMUTE}(\mathbf{x}_i, \mathcal{C}_i)$;
    Update T_θ with Eq.(1, 2, 4);
until *maximum iterations reached*;
repeat
  for *each* $x_i \in X$ **do**
    Update T_θ with Eq.(7)
until *maximum iterations reached*;

A.3 Pre-Training Details

The following details apply to both base architecture and joint-training architecture. We implement our pre-train models using Pytorch-lightning [24] and Hugginface's Pytorch Transformers [25]. For pre-training phase, we use the Adam optimizer with maximum sequence length 256, train batch size 8, gradient accumulation 8, warmup steps 10000, weight decay 0.01 and adam epsilon 1e-6. We train the models with 8 V100 GPUs and FP32 precision for 17 hours. The model is pre-trained for at most 3 epochs to prevent overfitting. We searched for the best learning rate for our model out of [1e-4, 2e-5, 2e-6, 5e-7].

A.4 Fine-Tuning Details

For fine-tuning, we use 4 V100 GPUs and use FP32. For all discriminative tasks, we use the Adam optimizer with maximum sequence length 256, batch size 4 and gradient accumulation 16. For generative task, we use the Adam optimizer with maximum source length 32, maximum target length 32, batch size 8, gradient accumulation 16. For all tasks, we use warmup fraction 0.01. Learning rates and train epochs are listed in Table 4.

Hyperparameter	CommonsenseQA	OpenbookQA	PIQA	aNLI	CommonGEN
Learning rate	[1e-4, 2e-4, 3e-4]	[5e-5, 1e-4, 2e-4, 3e-4]	[1e-4, 2e-4, 3e-4]	[2e-5, 3e-5]	[2e-5]
Train Epochs	20	20	20	10	20

Table 4: **Fine-tuning hyperparameters.**

A.5 Dataset Properties

- **CommonsenseQA** [12] is a multiple-choice question answering task, which picks the most appropriate answer on general commonsense questions.
- **OpenbookQA** [13] is a multiple-choice question answering task, which is modeled after open book exams on elementary-level core science questions. The task requires open book fact and additional commonsense which is not contained in the book. To test the commonsense reasoning ability, we do not use open book fact.
- **PIQA** [14] is multiple-choice question answering task, which chooses the most appropriate solution for physical commonsense questions.
- **aNLI** [15] is a binary-classification task, which picks the most plausible explanatory hypothesis given two observations from narrative contexts.
- **CommonGEN** [11] is a constrained text generation task, which generates a coherent sentence describing an everyday scenario using common concepts.

Dataset	Train	Development	Test	Source Example	Target Example
CommonsenseQA	9,741	1,221	1,140	context: *What home entertainment equipment requires cable?* options: 1: *radio shack* 2: *substation* 3: *cabinet* 4: *television* 5: *desk*	4
OpenbookQA	4,957	500	500	context: *You can make a telescope with* options: 1: *straw* 2: *glass* 3: *candle* 4: *mailing tube*	2
PIQA	16,113	1,838	3,084	context: *When boiling butter, when it's ready, you can* options: 1: *Pour it onto a plate* 2: *Pour it into a jar*	2
aNLI	169,654	1,532	3,040	context: *It was my birthday. When I got home the party was set up for my brother.* options: 1: *I was so excited.* 2: *I was so mad.*	2
CommonGEN	67,389	4,018	6,042	generate a sentence with these concepts: *Apple Grow Tree*	*Apple grows* on the *tree*

Table 5: **Properties of Commonsense benchmark datasets.**

A.6 Baseline Details

(1) **T5-base** is the pre-trained T5-base model without continually training on any intermediate task. (2) **T5-base w/ additional epochs** is continually pre-trained using the original pre-training objective of T5 with additional training steps. The total number of additional training steps is equal to that of our final model. (3) **T5-base + SSM** is continual pre-trained with a variant of the *salient span masking* objective [26, 27] objective that masks text spans of concepts extracted with POS tagging instead of named entities extracted by a pre-trained NER model, which makes it more focused on concepts. (4) **CALM(Generative-Only)** is continually pre-trained with the proposed generative objectives including concept-to-sentence generation(C2S) and concept order recovering(COR) as intermediate tasks. (5) **CALM(Contrastive-Only)** is continually pre-trained with the proposed contrastive objective as described in section 2.2 using the distractor generated by concept shuffling. (6) **CALM(Mix-only)** is continually pre-trained with both the generative objectives and the contrastive objective, combined with a multi-task learning fashion[7] with identical weights for each objective as the intermediate task. (7) **CALM (w/o Mix warmup)** is continually pre-trained with the joint training objective described in Eq (7) directly from the pre-trained T5-base model. (8) **CALM** is our main model trained as described in Algorithm 1. The difference between CALM and CALM (Joint) is that the former is initialized by the CALM(Mix).

A.7 Performance Analysis

Performance with fewer training examples To investigate the effectiveness of our objective in the low-resource setting, we explore the performance of our model and baselines fine-tuning with different fractions of the training data. From Figure 4, we can see that the performance improvement yielded by our models upon the T5-base model is more significant in the low-resource regime. This shows that CALM may already pack some commonsense knowledge in its parameters so that it does not require much data for fine-tuning before obtaining a good performance. In contrast, the original T5 model requires much data for fine-tuning, which suggests it may fail to encode much commonsense knowledge and must fit the correlation patterns in the downstream datasets to get a good performance.

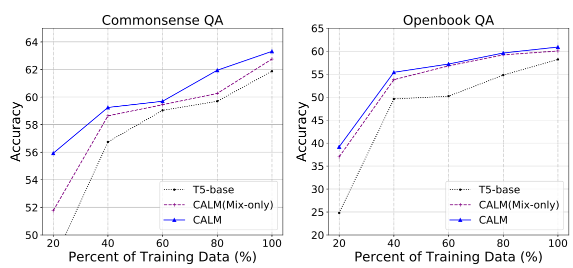

Figure 4: Performance of compared models fine-tuned with different fraction of the datasets.

Analysis on Generative objective To investigate the contribution of each generative objective, we conduct an ablation study by continually pre-training three models from the same T5-base model with C2S, COR, and text infilling, which is the original objective for pre-training T5, as the objective for the intermediate task. We continually pre-train these models for the same number of steps and then evaluate their performance by fine-tuning on different target tasks. The result is shown in Table 6. We can see that both C2S and COR works better than the original masked span infilling objective

[7] We tried combining them by training with the generative and contrastive objectives sequentially but the preliminary results are much worse than directly combining them with multi-task learning.

Methods	CSQA	PIQA	CommonGEN			
	Accuracy		BLEU-4	METEOR	CIDEr	SPICE
T5 - Text Infilling	61.92	68.19	25.10	31.00	13.13	32.40
CALM - COR	**62.36**	**68.77**	25.70	31.20	13.65	32.60
CALM - C2S	62.24	68.75	**25.90**	**31.40**	**13.94**	**32.80**

(a) Generative objectives

Methods	CSQA	PIQA	CommonGEN			
	Accuracy		BLEU-4	METEOR	CIDEr	SPICE
Multi-choice QA	62.21	68.82	25.00	31.20	13.28	32.60
True/False	62.24	67.81	25.10	31.20	13.41	32.60
Generative QA	**62.73**	**70.67**	**25.50**	31.20	**13.58**	32.60

(b) Contrastive objectives

Table 6: **Analysis on Contrastive and Generative objectives.** Left table shows the performance on downstream tasks by pre-training with different generative objective (COR, C2S, and original objective for pre-training T5). Right table shows the performance on downstream tasks by pre-training with different task formats of contrastive objective.

on itself. This confirms the effectiveness of our proposed generative objectives on improving the commonsense reasoning ability of pre-trained text-to-text transformers.

Task Formulation of the Contrastive objectives For contrastive objectives, we test three different task formats: Multi-choice QA, Generative QA, and True/False. Multi-choice QA and Generative QA takes the concatenation of the real sentence and the distractor. Then, Multi-choice QA output the index of the real sentence following other conventional Multi-choice QA tasks, and Generative QA output the real sentence respectively. True/False takes either the real sentence or the distractor and train the model to perform a binary classification problem of whether the input sentence makes sense. The result is shown in Table 6. We could find that the format of Generative QA performs the best. We suspect this is because the Generative QA format is closer to the format used during the original pre-training stage of the T5 model and the format used for fine-tuning.

Comparison of Generated Data Table 7 shows the comparison of generated examples for the COMMONGEN test set between T5-base and CALM. We can see that the sentences generated by CALM are generally more acceptable in terms of commonsense plausibility while T5-base sometimes generates sentences that do not make sense.

Concept-set	T5-base	CALM
Grass, Dog, Ball, Chase	a dog is chased by a ball on the grass.	dog chasing a ball in the grass.
Net, Cast, Boat, Water	fishing boat casts a net in the water.	fisherman casts a net into the water from a fishing boat.
Hole, Tree, Plant, Dig	a man digs a hole in a tree to plant a new tree . he digs the	man digging a hole to plant a tree.
Ingredient, Add, Pan, Fry	a pan filled with ingredients adds a touch of spice to the fry .	add the ingredients to a pan and fry.
Water, Hold, Hand, Walk	A man holding a hand and walking in the water. A man is holding water.	man holding a bottle of water in his hand as he walks down the street.
Place, Use, Metal tool	A man uses a metal tool to make a piece of metal.	woman uses a metal tool to make a piece of jewelry.
Hair, Wax, Apply, Remove	remove the wax from your hair and apply it to your hair .	woman applies wax to her hair and then removes it with a comb.
Sidewalk, Dog, Walk, Leash	A dog walking on a leash on the sidewalk.	dog walking on a sidewalk with a leash.

Table 7: **Comparison of generated sentences with same concept-set.** For same concept-set which is from CommonGEN test set, we compare generated sentences between T5-base and CALM.

Knowledge Probing To investigate how much concept-centric knowledge our model pack, we conducted two probing methods with our model : Language Model Analysis (LAMA) probe [28], Knowledge Intensive Language Task (KILT) [29]. We summarize the results on Table 8. LAMA

Methods	MRR	Precision@50	Precision@10	Precision@1
T5-Base	11.53	38.52	21.60	5.93
CALM (Mix-only)	11.77	38.93	21.92	6.10
CALM	**12.09**	**39.69**	**22.53**	**6.46**

(a) LAMA probe

Methods	FEVER	AY2
T5-base	76.65	74.97
CALM (Mix-only)	77.05	76.27
CALM	**77.44**	**77.24**

(b) KILT task

Table 8: **Experimental results on Knowledge Probing.** Left table shows the mean precision on LAMA probing task of ConceptNET. Right table shows the performance on Fact checking and Entity linking, which are from KILT task.

probe is consisting of a set of knowledge sources, each comprised of a set of fact. It defines that a pre-trained language model knows a fact (subject, relation, object) such as (Bird, CapableOf, Fly) if it can predict masked objects in cloze statement such as "Birds can [MASK]". For evaluation, we first filtered out examples that mask label is not in vocabulary list of T5. Then, we evaluate the model based on how highly it ranks the ground truth token against every other word in a fixed vocabulary list of T5, and get mean precision at k to check whether the object is ranked among the top k results. We summarize the results of ConceptNet [30] in Table 8. Unlike other language models which are optimised to masked word anywhere in a given sequence, T5 is trained with different denoising method. It might cause low performance on such slot filling task, but compared to T5, our model shows better performance compared to base model.

KILT task is a benchmark for assessing models that need to access specific knowledge in a defined snapshot of Wikipedia to solve tasks spanning five domains. The goal is to analyze the model whether it has task-agnostic representations of knowledge. We test our model on domain of fact checking, entity linking. Fact checking verifies textual claims against textual sources. For this task, we use FEVER [31] which is a large dataset for claim veracity that requires evidence from multiple Wikipedia pages to determine whether the claim is supported or refuted. Entity Linking assigns Wikipedia page to entities mentioned in text. We use AIDA CoNLL-YAGO (AY2) [32] which supplements the CoNLL 2003 [33] with Wikipedia URL annotations for all entities. We could find that our model outperforms the baseline.

A.8 Experiments with BART as Backbone

To show that our approach is versatile to different pre-trained models, we conduct experiments with BART as the backbone model. We can see that our approach consistently and significantly (with p-value < 0.01) improves BART-base on all datasets. This result shows that our method is versatile to different pre-trained models.

Methods	CSQA	OBQA	PIQA	aNLI	CommonGEN			
	Accuracy (official dev)				BLEU-4	METEOR	CIDEr	SPICE
BART-base (Mix-only)	56.31(±0.28)	58.30(±1.1)	67.53(±1.01)	59.85(±1.14)	25.10	29.50	13.16	30.20
CALM (BART-base)	**58.22(±0.21)**	**59.10(±1.0)**	**69.40(±1.23)**	**61.28(±0.30)**	**26.40**	**29.90**	**13.71**	**31.10**

Table 9: **Experimental results with BART as backbone model.** Best models are bold.

A.9 Experiments with Noun/Verb as Concepts

We also conducted an ablation study about the choice of using either nouns or verbs as concepts. We can see that using either nouns-only or verbs-only as concepts for our approach leads to substantial performance drop. This supports our choice about using both nouns and verbs as concepts.

Methods	CSQA	OBQA	PIQA	aNLI	CommonGEN			
	Accuracy (official dev)				BLEU-4	METEOR	CIDEr	SPICE
CALM	**63.32(±0.35)**	**60.90(±0.4)**	**71.01(±0.61)**	**63.20(±0.52)**	**26.40**	**31.40**	**13.88**	**33.00**
CALM-nouns	62.45(±0.42)	59.40(±0.5)	69.05(±0.70)	61.55(±0.58)	25.70	31.20	13.17	32.60
CALM-verbs	62.51(±0.47)	59.10(±0.7)	69.24(±0.65)	61.40(±0.51)	25.60	31.20	13.24	32.60

Table 10: **Experimental results with Noun/Verb as Concepts.** Best models are bold.

A.10 Related Works

Self-Supervised Language Representation Pre-Training. Motivated by the fact that words can have different meanings in different contexts, contextual language representation methods [34, 35] have been developed and shown superior performance on downstream tasks compared with static word embeddings [36, 37]. More recently, large scale language models based on transformer architecture [38] pre-trained with either mask language modeling objective [1, 3, 4] or mask span infilling objective [5, 2] have been explored further advanced the state-of-the-art on multiple NLU and NLG tasks. Our method is based on these techniques and we focus on improving the commonsense reasoning ability of pre-trained text-to-text transformers. More recently, [39] propose a new self-supervised pre-training objective called Replaced Token Detection (RTD). RTD uses a mask language model like BERT to fill in the mask and train a discriminator to predict whether a token is generated or real. This pre-training paradigm is related to our proposed joint training framework. Some major differences include that (1) Our method employs sentence-level distractors that are in general grammatically correct but not in line with commonsense, thus require the model to perform relational commonsense reasoning while RTD is a token-level discrimination task and can often be solved with syntactic and shallow semantic knowledge [40]; (2) Our method unifies generative and contrastive objectives with one model, which can be applied to both NLU and NLG downstream tasks; and (3) The discriminator in our framework is "contrastive", takes both the real sentence and the distractor as input simultaneously.

Knowledge-augmented PTLMs. As standard pre-trained language models usually do not explicitly model knowledge, a number of works have examined the problem of incorporating world knowledge with the PTLMs. Recent work [41, 8, 42, 23] utilizes an external knowledge base to incorporate entity knowledge with PTLMs; however, these approaches require specialized resources like knowledge bases, which limits the domain they can be applied to. [9] proposes WikiLM that encodes world knowledge into the parameters of a BERT[1]-like pre-trained model with a novel entity replacement detection objective that incorporates Wikipedia to form distractors. Their approach differs from ours because it requires an external knowledge base (i.e., Wikipedia) which limits the domain it can be applied, is limited to discriminative pre-training objectives and downstream tasks, and focuses on world knowledge instead of relational commonsense knowledge. More recently, [40] propose KALM, an entity-aware language model with more world knowledge packed into its parameters. Their method is restricted to the training of language models instead of masked language models or text-to-text transformers which can be used for more downstream tasks. Also, all the aforementioned work mainly focuses on world knowledge of named entities. In contrast, our work mainly focuses on commonsense knowledge about quotidian concepts.