PIP: Parse-Instructed Prefix for Syntactically Controlled Paraphrase Generation

Yixin Wan and **Kuan-Hao Huang** and **Kai-Wei Chang**
Computer Science Department, University of California, Los Angeles
elaine1wan@g.ucla.edu
{khhuang, kwchang}@cs.ucla.edu

Abstract

Syntactically controlled paraphrase generation requires language models to generate paraphrases for sentences according to specific syntactic structures. Existing fine-tuning methods for this task are costly as all the parameters of the model need to be updated during the training process. Inspired by recent studies on parameter-efficient learning, we propose Parse-Instructed Prefix (PIP), a novel adaptation of prefix-tuning to tune large pre-trained language models on syntactically controlled paraphrase generation task in a low-data setting with significantly less training cost. We introduce two methods to instruct a model's encoder prefix to capture syntax-related knowledge: direct initiation (PIP-Direct) and indirect optimization (PIP-Indirect). In contrast to traditional fine-tuning methods for this task, PIP is a compute-efficient alternative with $10\times$ times less learnable parameters. Compared to existing prefix-tuning methods, PIP excels at capturing syntax control information, achieving significantly higher performance at the same level of learnable parameter count.

1 Introduction

Syntactically controlled paraphrase generation (SCPG) has attracted increasing attention as it can diversify the generated paraphrases (Iyyer et al., 2018; Huang and Chang, 2021; Sun et al., 2021). Given an input sentence and a target syntax specification, an SCPG model aims to generate paraphrases that satisfy the specific syntax requirement. Such generation systems are promising in benefiting multiple application areas in natural language processing (NLP), such as text summarization (Fan et al., 2018), dialogue systems (Niu and Bansal, 2018; Gao et al., 2020), diverse question generation (Yu and Jiang, 2021), creative generation (Tian et al., 2021), and improving the robustness of models (Iyyer et al., 2018; Huang and Chang, 2021).

However, prior studies on SCPG mainly explore fine-tuning strategies, which require updating the parameters of the entire language model to adapt to the newly included syntax information. Therefore, many previously proposed methods suffer from tremendous training cost (Lewis et al., 2019; Raffel et al., 2020; Brown et al., 2020). With the recent rise of larger pre-trained language models (PLMs), this problem has become even more imminent. Nevertheless, a lightweight and more resource-efficient tuning method would allow easier application of large PLMs on the SCPG task.

Resource-efficient training methods such as prompt-tuning and prefix-tuning (Li and Liang, 2021; Lester et al., 2021) have proven to be effective in tuning large PLMs on various NLP tasks, such as text classification (Liu et al., 2021a), sequence labeling (Liu et al., 2021a), and summarization (Li and Liang, 2021). Prefix-tuning freezes a PLM's parameters and optimizes a small task-oriented continuous prefix that is prepended to the model's Transformer layers. It is a promising alternative to fine-tuning in a low-data setting with significantly fewer learnable parameters. However, no previous literature has explored the potential of prefix-tuning on the SCPG task.

In light of the lack of previous studies, we are amongst the first to study the application of resource-efficient training methods on the SCPG task. Our work has two main contributions. To begin with, we are among the first to study prefix-tuning's application on the SCPG task as a compute-efficient alternative for fine-tuning. Secondly, we propose *parse-instructed prefix (PIP)*, a novel adaptation of prefix-tuning for enhanced syntax control in paraphrase generation. Similar to prefix-tuning, PIP freezes all parameters of a PLM and only optimizes the prefix parameters, reducing the number of tune-able parameters to almost $10\times$ less than that required for fine-tuning. Prefix-tuning methods initialize the prefix as continuous and completely free parameters. For the SCPG task, this means that the prefix would need to learn the syntax control from scratch, since the PLMs were not pre-trained on any syntax-related task. In contrast, PIP provides syntax-related guidance to the prefix, allowing for better capturing of syntax knowledge. Specifically, we introduce two methods to guide the process of syntax knowledge capturing: direct initiation and indirect optimization. We prove that prefix-tuning-based methods achieve promising performance in a low-data setting with significantly fewer learnable parameters. In addition, our proposed PIP methods outperform prefix-tuning at the same level of training cost.[1]

2 Related Work

Syntactically controlled paraphrase generation. For the SCPG task, given a source sentence and a target syntax structure, a language model is trained to output a paraphrase sentence of the source sentence that (1) is semantically similar to the source sentence, and (2) conforms to the given target syntax structure, or the "syntax control". Prior works mainly adopted encoder-decoder model structures and used sequence-to-sequence training for the SCPG task (Iyyer et al., 2018; Kumar et al., 2020; Huang and Chang, 2021; Sun et al., 2021), while exploring different means to include syntax control signal during training. The first type of approach encodes the source sentence and the target syntactic tree separately, then concatenates them at decoder input (Iyyer et al., 2018; Kumar et al., 2020). The second type of approach concatenates linearized target constituency parse and source sentence at model input (Huang and Chang, 2021; Huang et al., 2022; Sun et al., 2021). However, the aforementioned methods require updating all model parameters during tuning at a high training cost.

Prompt-tuning and prefix-tuning. Prompting (Brown et al., 2020; Sun and Lai, 2020) provides PLMs with a discrete task-specific "prompt" to generate task-related outputs without task-specific fine-tuning. Prompt-tuning-based methods (Liu et al., 2021b; Qin and Eisner, 2021; Lester et al., 2021; Vu et al., 2022; Min et al., 2021), Prefix-Tuning (Li and Liang, 2021) and P-Tuning v2 (Liu et al., 2021a) derived from prompting and propose to only optimize a small sequence of continuous vectors. However, since prefix-tuning learns a prefix that was initiated as a continuous vector with completely free parameters, the prefix would need to learn task information from scratch during training. In addition, the training process for prefix-tuning does not allow for incorporation of any task-specific guidance. In summary, existing prefix-based methods fail to consider both specific task instruction and model-learned knowledge.

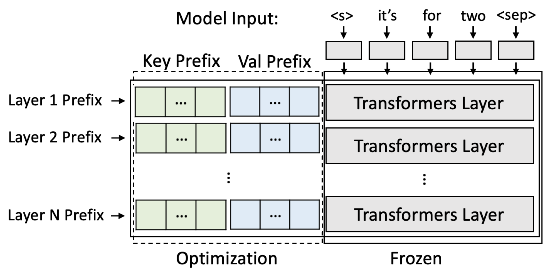


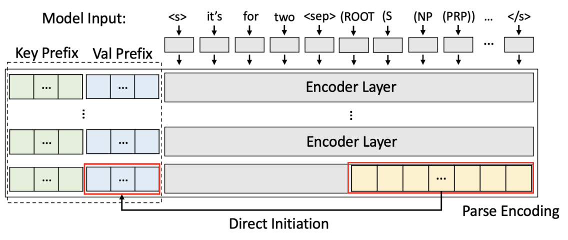
Figure 1: Structure of the prefix-tuning method.

3 Method

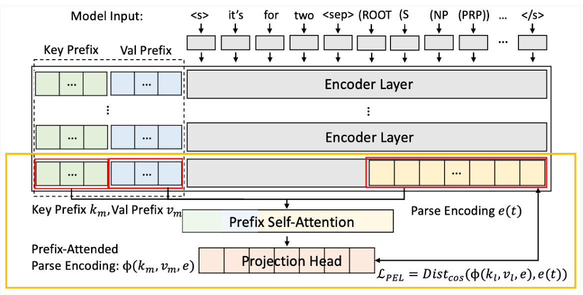
Problem formulation. Following previous studies (Iyyer et al., 2018; Huang and Chang, 2021; Huang et al., 2022), we use an encoder-decoder model structure and utilize the constituency parse as the control signal. Denote the source sentence as s_{src}, the target parse as t, and the target sentence as s_{tgt}. The goal of an SCPG model is to generate the target sentence s_{tgt} that is semantically similar to s_{src} while conforming to the syntax of t. In our study, the model is provided with s_{src} and t at input, and is supervised by the target sentence s_{tgt} at output during training. Notice that previous methods (Sun et al., 2021; Huang et al., 2022) mainly fine-tune all PLM parameters and therefore suffer from high training costs.

Prefix-tuning. We investigate a resource-efficient method for training an SCPG model based on the prefix-tuning method (Li and Liang, 2021; Liu et al., 2021a). Li and Liang (2021) freezes all pre-trained LM parameters and only optimizes a small sequence of continuous prefixes that are then prepended to keys and values of the attention module in the input layer of the model's encoder and decoder. Liu et al. (2021a) further extends this approach and applies prefixes to every layer of the model encoder and decoder. We follow the previous approach (Li and Liang, 2021) and consider prepending additional prefix parameters to the key and value matrices of each Transformer layer in the PLM. Specifically, we establish a prefix p with length $|p|$ for a PLM with l layers and hidden dimension $dim(h)$ and produce a set of key prefixes $K_p = \{k_1, k_2, ..., k_l\}$

[1]Our code for this work is available at `https://github.com/uclanlp/PIP`



(a) Direct Parse-Instructed Prefix (PIP-Direct).


Model Input:
<s> it's for two <sep> (ROOT (S (NP (PRP)) ... </s>
Key Prefix
Val Prefix
Encoder Layer
Encoder Layer
Key Prefix k_m, Val Prefix v_m
Prefix Self-Attention
Parse Encoding $e(t)$
Prefix-Attended Parse Encoding: $\phi(k_m, v_m, e)$
Projection Head
$\mathcal{L}_{PEL} = Dist_{cos}(\phi(k_l, v_l, e), e(t))$

(b) Indirect Parse-Instructed Prefix (PIP-Indirect).

Figure 2: Structure of the proposed PIP-Direct and PIP-Indirect Models. Note that we only visualize the encoder of the BART model. The model decoder follows the regular prefix-tuning setting without modifications. In (a), the value prefix of the last encoder layer is directly initiated by the model encoding of the target parse. In (b), the Parse Encoding Loss (PEL) is calculated between the prefix-attended parse encoding and the model parse encoding.

and a set of value prefixes $V_p = \{v_1, v_2, ..., v_l\}$, where $k_i, v_i \in |p| \times dim(h)$ denotes the key and value prefixes of layer i, respectively. For an encoder-decoder PLM, the key and value prefixes will then influence the model's encoding and decoding process through the attention mechanism, where the prefixes directly attend with the hidden states of the model. Figure 1 visualizes structure of the prefix-tuning method.

3.1 Parse-Instructed Prefix

Intuition. In prefix-tuning, the learned prefix acts as a context that influences the encoding of inputs through extracting task-related information from the PLM (Li and Liang, 2021). However, as prefix-tuning optimizes a prefix with completely free parameters, the prefix is learned from scratch and is unable to receive task-specific guidance during tuning. Since we use the constituency parse of target paraphrase as the control signal, which the PLM has never seen during pre-training, it will take a long time for the prefix to adapt to and learn the encoding for the control syntax. Specifically, for the SCPG task, the prefix will need to learn to: (1) capture semantic and syntax information from model input, and (2) combine the extracted semantic and syntax knowledge to produce an encoding for paraphrase generation under target syntax control. Since the prefix is implemented as pre-pended parameters for keys and values in Transformer layers, it first retrieves semantic and syntax information by attending to the source sentence and target parse in model input. Ideally, the prefix will then combine both the retrieved semantic and syntax information by influencing the output encoding. The combined information will then be captured by the decoder to output a paraphrase that conforms to the syntactic control. Therefore, guiding the prefix at encoder output to capture and leverage syntax knowledge will enhance the syntax control signal for improved performance on the SCPG task.

Therefore, we propose parse-instructed prefix (PIP) at the model's last encoder layer to "instruct" and augment the prefix's ability to capture task-specific syntax control for paraphrase generation. Specifically, we introduce two PIP-based methods for better capturing of syntax control information: Direct Parse-Instructed Prefix (PIP-Direct) and Indirect Parse-Instructed Prefix (PIP-Indirect). Different from prefix-tuning, where all prefix parameters are learned from scratch, PIP instructs the value prefix v_m of the last encoder layer with task-specific information. Through the attention mechanism, the instructed value prefix will help to better capture syntax information in the model's encoding output.

Direct parse-instructed prefix. We propose Direct Parse-Instructed Prefix (PIP-Direct) as an intuitive way to enhance knowledge on syntax control at model encoding. PIP-Direct directly updates the parameters of the value prefix at the last encoder layer with the model's encoding of the target parse. That is, for an input with target syntax t and an LM encoder with m layers, we first retrieve the model's encoding output of the target parse, which we denote as $e(t)$. Then, for the set of model encoder's value prefixes $V_p = \{v_1, v_2, ..., v_m\}$, we replace the value prefix of the last encoder layer with the parse encoding $e(t)$. The final value prefix prepended to the LM value state is then:

$$v_i^* = \begin{cases} e(t), & \text{if } i = m \\ v_i, & \text{otherwise} \end{cases}$$

This method directly augments syntax-related information at the last model encoder layer, which

Dataset	Model	# Params	BLEU ↑	ROUGE-1↑	ROUGE-2↑	ROUGE-L↑	TMA↑	TED-3↓
ParaNMT	Seq2seq	139.42M	57.17	79.20	64.31	78.81	89.94	0.59
	Prefix-Tuning	15.83M	47.75	73.85	55.76	73.17	82.75	1.12
	PIP-Direct	15.83M	49.61	74.84	57.39	74.26	85.81	0.90
	PIP-Indirect	18.78M	**50.82**	**75.58**	**58.46**	**75.00**	**86.20**	**0.87**
Pan	Seq2seq	139.42M	42.68	68.50	47.59	67.45	76.61	1.54
	Prefix-Tuning	15.83M	38.11	66.29	43.66	64.47	66.75	2.41
	PIP-Direct	15.83M	39.72	67.29	45.24	65.73	72.75	**1.95**
	PIP-Indirect	18.78M	**40.44**	**67.38**	**45.48**	**65.91**	**72.95**	1.98
MRPC	Seq2seq	139.42M	48.17	71.95	53.06	70.15	87.24	1.33
	Prefix-Tuning	15.83M	43.21	69.63	49.00	66.90	77.34	2.42
	PIP-Direct	15.83M	45.00	70.13	50.66	68.04	**83.65**	**1.88**
	PIP-Indirect	18.78M	**45.33**	**70.47**	**50.71**	**68.15**	83.49	1.89
Quora	Seq2seq	139.42M	49.93	78.45	58.29	77.54	79.64	0.84
	Prefix-Tuning	15.83M	42.4	74.94	51.21	73.49	70.69	1.40
	PIP-Direct	15.83M	**46.33**	**77.20**	**55.55**	**76.00**	76.00	1.07
	PIP-Indirect	18.78M	45.78	76.68	54.36	75.39	**77.34**	**1.04**

Table 1: Experiment results. "# Params" denotes the number of learnable parameters for each method. The PIP methods achieve highest performance amongst the three prefix-based methods on all valid and test datasets.

enables the key prefix of the same layer to capture the augmented syntax knowledge through attention. Structure of the direct initiation PIP is demonstrated on the left of Figure 2a.

Indirect parse-instructed prefix. We propose Indirect Parse-Instructed Prefix (PIP-Indirect) as an alternative way to guide the capturing of target syntax knowledge at the last encoder layer. Instead of directly replacing the prefix parameters, we utilize the Parse Encoding Loss (PEL) to indirectly augment the prefix's ability to capture syntax knowledge. Given a parse input, the prefix will influence the original model encoding by attending to the parse input, resulting in a modified encoding of the parse information. We can therefore augment the prefix's syntax information capturing by improving the ability to reconstruct the original parse encoding from the prefix-modified encoding. For a target parse t with encoding $e(t)$, we can obtain its prefix-modified encoding through an additional prefix self-attention layer $\mathcal{A}(\cdot)$, in which the prefix directly attends to the parse encoding $e(t)$. The prefix attention layer has the same structure as a prefix Transformer layer in the model, with the key prefix k_m and value prefix v_m of the last encoder layer m prepended to the attention key and value. We denote the output encoding of this prefix self-attention layer as $\mathcal{A}(k_m, v_m, e(t))$. To examine the ability to reconstruct the original parse encoding $e(t)$ from the prefix-modified encoding $\mathcal{A}(k_m, v_m, e(t))$, we leverage a learnable projection head, denoted by $\mathcal{H}(\cdot) : dim(h) \rightarrow dim(h)$, to approximate the process of reconstruction. We denote the output of the projection head as: $\phi(k_m, v_m, e(t)) = \mathcal{H}(\mathcal{A}(k_m, v_m, e(t)))$. Then, we establish the PEL loss by measuring the projected output's cosine distance, or the reconstructed parse encoding, with the original model encoding of target parse, $e(t)$. The PEL loss is mathematically formulated as:

$$\mathcal{L}_{PEL} = Dist_{cos}(\phi(k_m, v_m, e(t)), e(t)),$$

where $Dist_{cos}$ denotes the cosine distance. By integrating the PEL loss with the LM output loss during optimization, we can indirectly guide the prefix to better capture syntax-related information in model encoding. The structure of PIP-Indirect is demonstrated on the right of Figure 2b.

4 Experiments

We conducted experiments on our proposed PIP-Direct and PIP-Indirect methods, as well as two baseline methods for comparison. All four training methods are implemented on the BART-base model (Lewis et al., 2019). For all models, we concatenate the source sentence and target parse as input, and train the models to output a paraphrase that follows the given target syntax.

Dataset. We use ParaNMT (Chen et al., 2019) as the training and validation dataset for all models. Specifically, we sample 30,000 and 6,400 data entries from ParaNMT as our training set and dev set, respectively. To test the models' abilities to generate paraphrases with syntax control in unseen domains, we follow previous work (Huang and Chang, 2021; Huang et al., 2022) and apply the trained models on three mainstream test datasets: Quora (Iyer et al., 2017), MRPC (Dolan et al., 2004), and PAN (Madnani et al., 2012).

Evaluation metrics. Conforming to prior works (Huang et al., 2022; Sun et al., 2021), we evaluate generations of models on both alignment-based and syntactic conformation metrics. Alignment-based metrics measure the similarity between target paraphrases and model-generated paraphrases. We consider 4 alignment-based evaluation metrics: BLEU (Papineni et al., 2002), ROUGE-1, ROUGE-2, and ROUGE-L (Lin, 2004). Syntactic conformation metrics measure the quality of syntactic control in generated paraphrases. We consider 2 syntactic conformation evaluation metrics: Template Matching accuracy (TMA) and Tree-Edit Distances score (Zhang and Shasha, 1989) at height three (TED-3).

Baselines. We establish 2 baseline training methods for our study. The first baseline is the vanilla fine-tuning method that updates all parameters in the PLM, which we denote as **Seq2Seq**, In addition, we consider **prefix-tuning**, which freezes the PLM and learns a small set of parameters as the prefix, as the second baseline. Experiments on prefix-tuning in this study are based on our implementation of Li and Liang and Liu et al.'s work.

Implementation details. We train all baselines and our proposed PIP methods for 10 epochs with batch size 64. At decoding stage, we use beam search with beam size 4. We use the AdamW optimizer (Loshchilov and Hutter, 2017) and apply gradient clipping for training all models. For fine-tuning, we set learning rate to 10^{-5} with a linear scheduler. For prefix-tuning and PIP methods, we set learning rate to 3×10^{-4}.

Results. Results of experiments show that our proposed PIP methods outperform prefix-tuning on the validation dataset and all test datasets by a significant margin in a low-data setting and at the same level of training cost. Note that we separate results of the "Seq2Seq" model just for reference. We observe that PIP-Indirect achieves highest performance across all metrics among the 3 prefix-tuning-based approaches on the validation set of ParaNMT, as well as on testsets Pan and MRPC. PIP-Direct outperforms other prefix-tuning-based methods for the Quora dataset.

Model	BLEU ↑	TMA↑	TED-3↓
PIP-Direct	49.61	85.81	0.90
PIP-Indirect	**50.82**	**86.20**	**0.87**
Prefix-Tuning	47.75	82.75	1.12
Prefix-Tuning-Large	47.67	82.94	1.10

Table 2: Ablation experiment results.

Analysis. We conduct additional ablation experiments to further validate experimental results. Specifically, we examine if PIP-Indirect's performance gain is due to the effectiveness of design or the slightly higher parameter count compared to prefix-tuning. We experiment on prefix-tuning with an additional linear layer during prefix construction, denoted as Prefix-Tuning-Large. Prefix-Tuning-Large has 31.56M learnable parameters, 12.78M more than the PIP-Indirect method.

Table 2 demonstrates results of the ablation experiment on ParaNMT's validation dataset. We observe that although having more parameters, Prefix-Tuning-Large fails to outperform the PIP methods. In addition, Prefix-Tuning-Large even fails to outperform the original Prefix-Tuning method, which only has 15.83M parameters. This provides further insights that 1) a larger number of parameters in prefix-tuning-based methods does not guarantee performance gain on downstream tasks, and 2) outstanding performance of the PIP methods on SCPG task is due to the effectiveness of method design.

5 Conclusion

This research is amongst the first to study resource-efficient training methods for syntactically controlled paraphrase generation tasks. In this work, we proposed Parse-Instructed Prefix (PIP), a compute-efficient method that only requires 10× less learnable parameters than traditional fine-tuning methods. We introduce Direct and Indirect PIP methods to further improve prefix-tuning's performance by providing task-specific guidance and augmenting task-related knowledge at the fine-tuning stage. Through extensive experiments, we find out that both PIP-Direct and PIP-Indirect outperform prefix-tuning in a low-data setting at the same level of parameter count, and are promising as a resource-efficient alternative to fine-tuning. With ablation studies, we further validate that the performance gain of the proposed PIP methods is due to the effectiveness of the design.

Acknowledgments

We thank anonymous reviewers for their helpful feedback. We thank the UCLA-NLP group for their valuable discussions and comments. This work is supported in part by National Science Foundation (NSF) via Award No. 2200274 and a CISCO research grant.

Limitations

We identify some limitations of our study. First, due to a lack of computing resources, we were not able to experiment with even larger pre-trained language models such as GPT-2 and GPT-3. In future explorations, we would like to seek the opportunity to investigate the potential of instructed prefix-tuning on even larger-scale language models across a variety of generation tasks. Second, since our work are amongst the first to explore the application of prefix-tuning on the task of syntactically-controlled paraphrase generation, we were not able to identify state-of-the-art prior works on the same subject in the field to establish comparison with. We believe, however, that with our demonstration of promising application of prefix-tuning for SCPG, researchers will soon propose new ideas to utilize prefix for tuning large PLMs on this task at even lower training costs.

Ethics Considerations

The experimented and proposed model in this study is based on large-scale Pre-trained Language Models (PLM). Recent studies have revealed that large PLMs that are trained on large textual corpora might learn or even amplify the bias existing in its training dataset. Therefore, since our method is established on top of a large PLM that is potentially at risk of demonstrating or amplifying bias, we recommend that potential harm and biases be evaluated before deploying our method in real-world situations.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019. A multi-task approach for disentangling syntax and semantics in sentence representations. 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 2453–2464, Minneapolis, Minnesota. Association for Computational Linguistics.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 350–356, Geneva, Switzerland. COLING.

Angela Fan, David Grangier, and Michael Auli. 2018. Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54, Melbourne, Australia. Association for Computational Linguistics.

Silin Gao, Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020. Paraphrase augmented task-oriented dialog generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 639–649, Online. Association for Computational Linguistics.

Kuan-Hao Huang and Kai-Wei Chang. 2021. Generating syntactically controlled paraphrases without using annotated parallel pairs. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Kuan-Hao Huang, Varun Iyer, Anoop Kumar, Sriram Venkatapathy, Kai-Wei Chang, and A. G. Galstyan. 2022. Unsupervised syntactically controlled paraphrase generation with abstract meaning representations. In *Findings of the Association for Computational Linguistics: EMNLP 2022 (EMNLP-Findings)*.

Shankar Iyer, Nikhil Dandekar, Kornél Csernai, et al. 2017. First quora dataset release: Question pairs. *data. quora. com*.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *North American Chapter of the Association for Computational Linguistics*.

Ashutosh Kumar, Kabir Ahuja, Raghuram Vadapalli, and Partha Talukdar. 2020. Syntax-guided controlled generation of paraphrases. *Transactions of the Association for Computational Linguistics*, 8:329–345.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. 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 4582–4597, Online. Association for Computational Linguistics.

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

Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021a. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *CoRR*, abs/2110.07602.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. Gpt understands, too. *arXiv:2103.10385*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Nitin Madnani, Joel Tetreault, and Martin Chodorow. 2012. Re-examining machine translation metrics for paraphrase identification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 182–190, Montréal, Canada. Association for Computational Linguistics.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2021. Noisy channel language model prompting for few-shot text classification. *arXiv preprint*.

Tong Niu and Mohit Bansal. 2018. Polite dialogue generation without parallel data. *Transactions of the Association for Computational Linguistics*, 6:373–389.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics*.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5203–5212, Online. Association for Computational Linguistics.

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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Fan-Keng Sun and Cheng-I Lai. 2020. Conditioned natural language generation using only unconditioned language model: An exploration. *ArXiv*, abs/2011.07347.

Jiao Sun, Xuezhe Ma, and Nanyun Peng. 2021. Aesop: Paraphrase generation with adaptive syntactic control. In *The 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yufei Tian, Arvind krishna Sridhar, and Nanyun Peng. 2021. HypoGen: Hyperbole generation with commonsense and counterfactual knowledge. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1583–1593, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou', and Daniel Cer. 2022. SPoT: Better frozen model adaptation through soft prompt transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5039–5059, Dublin, Ireland. Association for Computational Linguistics.

Xiaojing Yu and Anxiao Jiang. 2021. Expanding, retrieving and infilling: Diversifying cross-domain question generation with flexible templates. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3202–3212, Online. Association for Computational Linguistics.

Kaizhong Zhang and Dennis Shasha. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal on Computing*, 18(6):1245–1262.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
6

☑ A2. Did you discuss any potential risks of your work?
7

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☐ A4. Have you used AI writing assistants when working on this paper?
Not applicable. Left blank.

B ☑ Did you use or create scientific artifacts?

4

☑ B1. Did you cite the creators of artifacts you used?
4

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

4

☐ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
No response.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☐ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
No response.

☐ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
No response.

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
No response.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?
Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.