Instance-aware Prompt Learning for Language Understanding and Generation

Feihu Jin\textsuperscript{1,2}, Jinliang Lu\textsuperscript{1,2}, Jiajun Zhang\textsuperscript{1,2} \* and Chengqing Zong\textsuperscript{1,2,3}

\textsuperscript{1}National Laboratory of Pattern Recognition, Institute of Automation, CAS
\textsuperscript{2}School of Artificial Intelligence, University of Chinese Academy of Sciences
\textsuperscript{3}CAS Center for Excellence in Brain Science and Intelligence Technology
jinfeihu2020@ia.ac.cn, \{jinliang.lu, jjzhang, cqzong\}@nlpr.ia.ac.cn

Abstract

Recently, prompt learning has become a new paradigm to utilize pre-trained language models (PLMs) and achieves promising results in downstream tasks with a negligible increase of parameters. The current usage of discrete and continuous prompts assumes that the prompt is fixed for a specific task and all samples in the task share the same prompt. However, a task may contain quite diverse samples in which some are easy and others are difficult, and diverse prompts are desirable. In this paper, we propose an instance-aware prompt learning method that learns a different prompt for each instance. Specifically, we suppose that each learnable prompt token has a different contribution to different instances, and we learn the contribution by calculating the relevance score between an instance and each prompt token. The contribution weighted prompt would be instance aware. We apply our method to both unidirectional and bidirectional PLMs on both language understanding and generation tasks. Extensive experiments demonstrate that our method obtains considerable improvements compared to strong baselines. Especially, our method achieves the state-of-the-art on the SuperGLUE few-shot learning benchmark.\textsuperscript{1}

1 Introduction

Prompt learning aims to design or learn appropriate prompts which can induce the capacity from pre-trained language models (PLMs) to perform specific tasks, and it becomes a new paradigm to use PLMs due to its flexibility and fewer extra parameters. There are typically two kinds of prompts, namely discrete prompt and continuous prompt. Discrete prompt such as GPT-3 [Brown et al., 2020] uses the task instructions and task-related instances as prompt for zero-shot and few-shot learning respectively. PET/iPET [Schick and Schütze, 2021a; Schick and Schütze, 2021b] utilizes the manually-designed prompts to reformulate many tasks as cloze questions (e.g., by appending phrases such as “Similar sense of two sentences?”) and performs gradient-based fine-tuning with smaller PLMs. To simplify PET/iPET, ADAPET [Tam et al., 2021] decouples the losses for the label tokens and a label-conditioned masked language modeling objective over the full original input. Considering that manually designing the discrete prompts is time-consuming and labor-intensive, several efforts focus on searching proper discrete prompts automatically [Shin et al., 2020; Gao et al., 2021; Zhong et al., 2021].

Although discrete prompts can reflect rationality from the perspective of humans, it may not be necessarily suitable for PLMs. To tackle this problem, a lot of studies begin to focus on continuous prompts. Continuous prompts can be thought of as special tokens. [Lester et al., 2021] proposes prompt tuning and concatenates the continuous prompts with the embedding layer of PLMs. When using small PLMs, the performance of prompt tuning has a clear gap with fine-tuning. [Li and Liang, 2021] proposes prefix tuning and shows comparable results with fine-tuning on generation tasks. Prefix tuning concatenates continuous prompts with each layer in the decoder and only optimizes 0.1\% of the model parameters. However, the current usage of discrete and continuous prompts assumes that all samples in one task share the same prompt, and does not consider the diversity of the instances in which some are easy and others are difficult. Therefore, it is desirable to learn a special prompt for each instance.

In this paper, we propose an Instance-aware Prompt Learning method (abbreviated as IPL) which learns a unique prompt for each instance. As shown in Figure 1, we use two different manually-designed patterns to formalize the in-
stances into cloze-style questions and feed them into the pre-trained language model (PLM). As we can see, for each instance, using different prompts can get different answers. Pattern 1 is suitable for instance 1, while pattern 2 fits instance 2, which means that every instance needs a specific prompt for itself. However, it is difficult to dynamically find an appropriately discrete prompt for each instance. Therefore, we consider to utilize a look-up module and obtain a dynamic continuous prompt for each instance. Specifically, we take each learnable prompt token as a query and calculate its contribution to each instance through the look-up module. After doing this, each learnable prompt token has a different contribution to the instance and the contribution weighted continuous prompts are then utilized to guide the PLMs to perform the downstream task more instance-aware.

We evaluate our approach on natural language understanding (NLU) and generation (NLG) tasks. For NLU tasks, we conduct experiments on SuperGLUE [Wang et al., 2019] with both GPT2 [Radford et al., 2019] and RoBERTa [Delobelle et al., 2020]. For NLG tasks, we conduct experiments on table-to-text generation and summarization using GPT2. Experimental results on various tasks demonstrate that our method obtains considerable improvements compared to strong baselines. Especially, our method achieves the new state-of-the-art on the SuperGLUE few-shot learning benchmark. In summary, our key contributions can be listed as follows:

- We propose an instance-aware prompt learning method that can learn a unique prompt for each instance.
- Extensive experiments on both language understanding and generation tasks under both unidirectional and bidirectional PLMs verify the effectiveness of our method.
- Detailed analyses verify that IPL can indeed dynamically learn appropriate continuous prompts for each instance.

2 Approach

In this section, we present the details of our model IPL. Previous studies demonstrate that prompt learning is promising for downstream tasks. However, using fixed prompts (e.g., discrete prompts like “convert the table into a sentence” or continuous prompts after optimization) for diverse instances in one task ignores the peculiarity of different instances. To address this problem, our instance-aware prompt learning method IPL can learn a special prompt for each specific instance. Next, we introduce prompt learning first and then present our IPL model.

2.1 Prompt Learning

For the standard paradigm of pre-training and fine-tuning, there is a gap (e.g., inconsistent objective function) between the pre-training stage and the fine-tuning stage. Fortunately, prompt learning bridges this gap by formalizing the downstream tasks into the form of a conditional language model or masked language model. Discrete prompt is an important method in prompt learning, for instance, given a masked language model $\mathcal{M}$, we first use the prompt to formulate a question and answer instance $x$ (e.g., [passage]). Can you have too much oxygen in your body? as:

$$\hat{x} = x \text{ the answer is [MASK]}.$$ 

Then $\hat{x}$ is fed into $\mathcal{M}$, and let $\mathcal{M}$ determine whether “Yes” or “No” is more appropriate to replace [MASK] [Gao et al., 2021].

Continuous prompt is an alternative approach in prompt learning. Suppose we get the embedding sequence $\{e_1, e_2, \cdots, e_n\}$ of the instance $x$, and we concatenate the continuous prompt (e.g., as shown in the red dotted box in Figure 2) $\{p_1, p_2, \cdots, p_l\}$ with the embedding sequence, which can be formalized as follows:

$$\hat{x} = \{p_1, p_2, \cdots, p_l; e_1, e_2, \cdots, e_n\}$$

Then, $\hat{x}$ is fed into $\mathcal{M}$ to generate the target labels.

In this paper, we combine the advantages of continuous prompts with discrete prompts and propose an instance-aware prompt learning method which learns a unique prompt for each instance. Next, we detail our proposed IPL model.

2.2 Instance-aware Prompt Learning

We denote $\mathcal{T}$ as an instance in a form of a sequence consisting of $n$ tokens $\mathcal{T} = \{T_1, T_2, \cdots, T_n\}$. We follow Li and Liang, 2021; Lester et al., 2021, and use learnable prompts of special tokens $P = \{P_1, P_2, \cdots, P_l\}$ and update the embeddings of these prompt tokens during prompt learning. As shown in Figure 2, we first use the pre-trained language model to obtain the embedding sequence $\{e_1, e_2, \cdots, e_n\}$ of $\mathcal{T}$, and get a matrix $X \in \mathbb{R}^{n \times d_x}$, where $n$ is the length of instance $\mathcal{T}$, $d_x$ is the dimension of the embedding space. Then we create a learnable matrix $P = \{p_1, p_2, \cdots, p_l\}$ of $P$, where $l$ is the length of the prompt, and $P \in \mathbb{R}^{l \times d_x}$. After we get the prompt matrix $P$ and embedding matrix $X$, we use the projection matrices $W^M \in \mathbb{R}^{d_x \times d_h}$ and $W^N \in \mathbb{R}^{d_x \times d_h}$ to map $P$ and $X$ into matrix $M \in \mathbb{R}^{l \times d_h}$ and $N \in \mathbb{R}^{n \times d_h}$ respectively, where $d_h$ is the dimension of projection space.

$$M = PW^M$$

$$N = XW^N$$

(1)
We suppose that each learnable prompt token has a different contribution to different instances and we learn the contribution scores by calculating the relevance score between matrix $M = \{p'_1, p'_2, \ldots, p'_l\}$ and $N = \{e'_1, e'_2, \ldots, e'_n\}$. After we get the relevance score, we pass the score to the look-up module. In the look-up module, we adopt a method of mean operation and apply a sigmoid function $\sigma$ to obtain how much does each learnable prompt token contributes to the instance $T$. The detailed calculation is as follows:

$$s_j = \sigma\left(\frac{1}{n}\sum_{i=1}^{n} p'_j \cdot e'_n^T\right)$$  \hspace{1cm} (2)$$

$$\hat{p}_j = s_j \cdot p_j$$  \hspace{1cm} (3)$$

where $e'_n^T$ is the transpose of $e_n$. $p'_j$ is the contribution of the $j$-th prompt token to the instance $T$. $s_j$ is the contribution score of the $j$-th prompt token after applying a sigmoid function $\sigma$, and $\hat{p}_j$ is the $j$-th weighted representation for the instance. After doing such a calculation for all prompt tokens, we get the weighted prompt as $\hat{P} = \{\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_l\}$. Then we concatenate weighted continuous prompt with the embedded instance as a new matrix $[\hat{P}, \mathbf{X}] \in \mathbb{R}^{(l+n) \times d_e}$, and feed it into the pre-trained language model.

During training, we optimize the parameters of prompt module and PLMs. We do not freeze the model parameters as [Lester et al., 2021] does, because their results demonstrates that the gap between prompt tuning and fine-tuning disappear only when the model size increases to 10 billion parameters.

### 3 Experiments

To evaluate our method IPL, we conduct experiments on both NLU tasks and NLG tasks. For NLU tasks, we evaluate IPL on SuperGLUE$^2$ [Wang et al., 2019]. And we evaluate IPL for few-shot learning by using 32 labeled examples per task from FewGLUE$^3$ [Schick and Schütze, 2021b]. For NLG tasks, we select 3 standard table-to-text generation tasks: E2E [Novikova et al., 2017], WebNLG [Gardent et al., 2017], DART [Radev et al., 2021] and a dialogue summarization task: SamSum [Gliwa et al., 2019].

#### 3.1 Experiments on NLU Tasks

Our code is implemented based on PET$^4$ using HuggingFace [Wolf et al., 2020]. The experiment results include the few-shot learning results and fully-supervised learning results.

### Few-shot Learning Results

For a fair comparison, we choose ALBERT-xxlarge-v2 [Lan et al., 2020] for experiments and use the same data split as [Schick and Schütze, 2021b], which consists of 32 labeled examples for each task. We use a default setting training for 20 epochs, using a learning rate of $1e^{-5}$, a batch size of 8, and a prompt length of 16.

Our main results on the validation and test sets on SuperGLUE are shown in Table 1 and Table 2. We compare against GPT-3, PET/iPET and ADAPET. Initially, ADAPET does not use the unlabeled data and achieves the state-of-the-art on

\[\text{https://supergluebenchmark.com/}\]

\[\text{https://github.com/timoschick/fewglue}\]

\[\text{https://github.com/timoschick/pet}\]
SuperGLUE few-shot learning tasks compared to PET/iPET which uses the unlabeled data. And for IPL, we train IPL with a single pattern and do not use the unlabeled data.

As can be seen from Table 1, on average, IPL outperforms GPT-3 by 6 points; outperforms PET’s iterative variant, iPET, by 2.5 points, and even outperforms the previous state-of-the-art model ADAPET by 2 points on the dev set. Specifically, compared with iPET and GPT-3, IPL achieves improvements on 5 out of the 8 tasks and 6 out of the 8 tasks respectively, demonstrating the effectiveness of our method in few-shot NLU tasks. We will conduct detailed analysis in 4.2.

We also report the test set on SuperGLUE in Table 2. IPL outperforms all other models including ADAPET, which is the previous state-of-the-art model, and obtains the new state-of-the-art for few-shot learning on SuperGLUE.

### Fully-supervised Learning Results

To verify IPL on various PLMs, we perform experiments on 5 out of the 8 tasks of SuperGLUE benchmark including BoolQ, MultiRC, RTE, CB, and WiC, where we choose both unidirectional PLMs GPT-2 and bidirectional PLMs RoBERTa. We report the performance of fine-tuning with PET [Schick and Schütze, 2021a], prompt tuning [Lester et al., 2021], and our method IPL. We use a default setting training for 20 epochs, using a learning rate of 2e-5, a batch size of 32, and a prompt length of 16.

Table 3 and Table 4 show our main results on GPT-2 and RoBERTa. On unidirectional PLMs like GPT2-base and GPT2-large, IPL outperforms PET fine-tuning and prompt tuning on all 5 tasks with GPT2-base and 4 out of 5 tasks on GPT2-large. On bidirectional PLMs like RoBERTa-base and RoBERTa-large, IPL outperforms all other RoBERTa-based models on all 5 tasks. Based on the experiment results, we demonstrate that IPL can achieve great results on both GPT-2 and RoBERTa models.

### 3.2 Experiment on NLG Tasks

For NLG tasks, we compare IPL on GPT2-base and GPT2-large with two baseline methods: the standard fine-tuning, and prompt tuning, where we do not freeze the model parameters as IPL does. The experiment results are illustrated in Table 5 and Table 6. We choose three table-to-text tasks and a summarization task. For table-to-text tasks, on E2E, we use the official evaluation script, which reports BLUE [Papineni et al., 2002], NIST [Belz and Reiter, 2006], ROUGE-L [Lin, 2004], and CIDEr [Vedantam et al., 2015]. On WebNLG, we use the official evaluation script, which reports BLEU, METEOR, and TER [Snover et al., 2004], and CIDEr [Vedantam et al., 2015]. On DART, we use the official evaluation script, which reports BLEU, METEOR, TER, and BERTScore [Zhang et al., 2020].

As for the summarization task: SamSum, we report ROUGE-1, ROUGE-2, and ROUGE-L. The hyperparameters we tune include the number of epochs, batch size, learning rate, and prefix length. For table-to-text tasks, we set batch size as 32, prefix length as 10, the number of epochs as 10 for both GPT2-base and GPT2-large, in addition to the learning rate as 5e-5 for GPT-base, 5e-6 for GPT-large. For the summarization task, we set the prefix length as 100 and other hyperparameters keep consistent with table-to-text tasks.

#### Results on NLG Tasks

As shown in Table 5, on GPT2-base, IPL performs better than fine-tuning and prompt tuning on E2E and WebNLG, while on DART, which is an open domain table-to-text dataset, IPL slightly underperforms prompt tuning. On GPT2-large, IPL outperforms fine-tuning and can be comparable or better than prompt tuning. Additionally, IPL obtains better performance on WebNLG unseen domains suggesting that IPL can generalize to other domains better. For the summarization task, the results in Table 6 show IPL performs better than fine-tuning and prompt tuning on both GPT2-base and GPT2-large models, suggesting it has the potential to scale to even larger models. Above all, the results demonstrate that IPL can achieve
We visualize the relationship between the performance and different prompt lengths (other settings are fixed). For NLU tasks, we conduct experiments on three tasks of SuperGLUE including CB, WSC, WiC using ALBERT-xxlarge-v2. Figure 3(a) shows that performance increases as the prompt length increases up to a threshold (16 for CB and WiC, 20 for WSC), and then the performance slightly drops. Figure 3(b) shows the effect of prompt length on the performance of different model sizes on SamSum. We can see that the performance consistently increases until the prompt length is up to 50. Continuing to increase the prompt length cannot yield significant improvements.

4.2 Visualization of Instance-aware Prompt
We choose similar instances and dissimilar instances from WSC [Levesque et al., 2012] for analysis. Figure 4 shows the analysis results for IPL on similar instances and dissimilar instances. We visualize the attention matrix of the instance and prompt. As shown in 4(a) and 4(b), the attention matrices between 4(a) and 4(b) are similar, which means IPL can produce similar prompts for similar instances. Comparing 4(a) and 4(c) or 4(b) and 4(c), we find that the attention matrices are not similar, suggesting that IPL can produce different prompts for dissimilar instances. Consequently, our approach learns a special prompt for each instance and can be aware of the important information of the instance.

4.3 Case study
As shown in Figure 5, for indistinguishable instances, PET utilizes a fixed discrete prompt and makes a wrong judgment on the meaning of the word ‘put’ and ‘department’. Prompt tuning preends the fixed continuous prompt with the two instances based on the pattern of PET and also gives wrong answers. In contrast, our method IPL learns a unique prompt for each instance and contains much information of the instance yielding the correct answer.

5 Related Work
GPT-3 [Brown et al., 2020], which uses the task description and several typical examples as prompt to guide the generation, indicates the language models are few-shot learners and leads to the waves of prompt learning. Recently,
Table 6: Results for summarization on SamSum using GPT2 models. The FT refers to fine-tuning. PT refers to prompt tuning, and the best score is in bold.

| Method   | R-1   | R-2   | R-L   | R-1   | R-2   | R-L   |
|----------|-------|-------|-------|-------|-------|-------|
| GPT2-base| 49.7  | 24.8  | 45.0  | 47.2  | 22.2  | 42.8  |
| FT       | 46.5  | 21.4  | 41.8  | 49.3  | 24.5  | 44.8  |
| PT       | 46.6  | 21.7  | 42.0  | 49.7  | 24.8  | 45.0  |

Figure 5: The instances are chosen from WiC dataset in SuperGLUE, which is shown on the top. We use the manually-designed pattern from PET. PT refers to prompt tuning. Our method is shown on the bottom, and the color words mean our approach can be aware of the important words in the instance.

PET/iPET [Schick and Schütze, 2021b] utilizes the manually-designed prompts to reformulate natural language understanding tasks as cloze-style questions with gradient-based fine-tuning. There are also a lot of studies that utilize the manually-designed prompt to mine the knowledge from the PLMs [Jiang et al., 2020; Trinh and Le, 2018]. Since manual-designed prompt is time-consuming and the search space is huge, researches focus on automatic prompt search [Gao et al., 2021; Shin et al., 2020; Zhong et al., 2021].

However, the handcrafted prompt can only reflect rationality from the perspective of humans, which midwifery the exploration in continuous prompts. [Li and Liang, 2021] proposes prefix tuning and concatenates learnable prompt at each layer of transformer while only optimizing the prefix parameters. In contrast, prompt tuning [Lester et al., 2021] concatenates learnable prompt only in the embedding layer and only optimizes the prompt parameters in the embedding layer. Although [Lester et al., 2021] demonstrates the effectiveness of light-weight prompt-tuning, the gap with full parameter fine-tuning still exists especially when the PLM is small.

There are also a lot of works that interleave the prompt throughout the input layer. [Hambardzumyan et al., 2021] proposes WARP, initializing the prompt parameters either with word embeddings of [MASK] or similar to the vectors from the word embedding layer. Their work is based on a series of masked language models [Delobelle et al., 2020; Lan et al., 2020] and uses a learnable output layer to project the mask to class logits, which restricts the model and only produces a single output. [Liu et al., 2021] proposes P-tuning, using the patterns based on human design and putting the continuous prompts interleave throughout the embedded input. When optimizing the model, P-tuning jointly updates both the prompt and model parameters to perform on SuperGLUE. Similarly, we borrow the idea of human designed patterns to convert different tasks into the form of conditional language model or masked language model, and also apply our method on GPT-2 and RoBERTa.

However, the above usage of the discrete and continuous prompts assumes that the prompt is fixed for a specific task and all samples in the task share the same prompt. Different from the above methods, our proposed IPL dynamically learns a special prompt for each instance and obtains considerable improvements compared to strong baselines.

Very recently, a contemporaneous work also presents another instance dependent prompt generation approach [Anonymous, 2021], which studies only masked language model on only NLU tasks. In contrast, our IPL model is simple and effective for both unidirectional and bidirectional PLMs on both NLU and NLG tasks.

6 Conclusion and Future Work

In this paper, we propose an instance-aware prompt learning method named IPL, which learns a unique prompt for each instance. We find that IPL has the potential to be applied to both unidirectional and bidirectional PLMs on both language understanding and generation tasks. In the few-shot learning SuperGLUE benchmark, IPL outperforms all other models and obtains the new state-of-the-art. Detailed analysis demonstrates that our IPL model can indeed dynamically learn appropriate prompts for various instances.

In the future, we would explore how to generate better instance-aware prompts, and apply the instance-aware method to parameter-efficient tuning for more natural language processing tasks.

References

[Anonymous, 2021] Anonymous. Idpg: An instance-dependent prompt generation method. In Openreview for ACL Rolling Review - November Submission, 2021.

[Belz and Reiter, 2006] Anja Belz and Ehud Reiter. Comparing automatic and human evaluation of nlg systems. In EACL, 2006.

[Brown et al., 2020] 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. Language models are few-shot learners. In NeurIPS, 2020.
[Delobelle et al., 2020] Pieter Delobelle, Thomas Winters, and Bettina Berendt. RobBERT: a Dutch RoBERTa-based Language Model. In *Findings of EMNLP*, 2020.

[Gao et al., 2021] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In *ACL*, 2021.

[Gardent et al., 2017] Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. The WebNLG challenge: Generating text from RDF data. In *ICNLG*, 2017.

[Gliwa et al., 2019] Bogdan Gliwa, Iwona Mochol, Maciej Biese, and Aleksander Wawer. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, 2019.

[Hambardzumyan et al., 2021] Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. WARP: Word-level Adversarial ReProgramming. In *ACL*, 2021.

[Jiang et al., 2020] Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *TACL*, 2020.

[Lan et al., 2020] Zhennzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *ICLR*, 2020.

[Lavie and Agarwal, 2007] Alon Lavie and Abhaya Agarwal. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, 2007.

[Lester et al., 2021] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *EMNLP*, 2021.

[Levesque et al., 2012] Hector Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *International Conference on the Principles of Knowledge Representation and Reasoning*, 2012.

[Li and Liang, 2021] Xiang Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL*, 2021.

[Lin, 2004] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *ACL*, 2004.

[Liu et al., 2021] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands too. *ArXiv*, abs/2103.10385, 2021.

[Novikova et al., 2017] Jekaterina Novikova, Ondrej Dusek, and Verena Rieser. The e2e dataset: New challenges for end-to-end generation. In *SIGDIAL*, 2017.

[Papineni et al., 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

[Radev et al., 2021] Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chia-Hsuan Hsieh, Nazneen Rajani, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irwanto, Jessica Pan, Faiaz Rahman, Ahmad Zaidi, Murori Mutuma, Yasin Tarabar, Ankit Gupta, Tao Yu, Yi Chern Tan, Xi Victoria Lin, Caiming Xiong, and Richard Socher. Dart: Open-domain structured data record to text generation. In *NAACL-HLT*, 2021.

[Radford et al., 2019] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8), 2019.

[Schick and Schütze, 2021a] Timo Schick and Hinrich Schütze. Exploiting cloze-questions for few-shot text classification and natural language inference. In *EACL*, 2021.

[Schick and Schütze, 2021b] Timo Schick and Hinrich Schütze. It’s not just size that matters: Small language models are also few-shot learners. In *NAACL-HLT*, 2021.

[Shin et al., 2020] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *EMNLP*, 2020.

[Snover et al., 2006] Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, 2006.

[Tam et al., 2021] Derek Tam, Rakesh R. Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. Improving and simplifying pattern exploiting training. In *EMNLP*, 2021.

[Trinh and Le, 2018] Trieu H. Trinh and Quoc V. Le. A simple method for commonsense reasoning. *ArXiv*, abs/1806.02847, 2018.

[Vedantam et al., 2015] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015.

[Wang et al., 2019] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In *NeurIPS*, 2019.

[Wolf et al., 2020] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Transformers: State-of-the-art natural language processing. In *EMNLP*, 2020.

[Zhang et al., 2020] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *ICLR*, 2020.

[Zhong et al., 2021] Zexuan Zhong, Dan Friedman, and Danqi Chen. Factual probing is [MASK]: Learning vs. learning to recall. In *NAACL-HLT*, 2021.