P-Tuning v2: Prompt Tuning Can Be Comparable to Finetuning Universally Across Scales and Tasks

Xiao Liu[1,2*], **Kaixuan Ji**[1*], **Yicheng Fu**[1*], **Zhengxiao Du**[1,2], **Zhilin Yang**[1,2†], **Jie Tang**[1,2†]
[1]Tsinghua University, Beijing, China
[2]Beijing Academy of Artificial Intelligence (BAAI), Beijing, China
liuxiao21@mails.tsinghua.edu.cn

Abstract

Prompt tuning, which only tunes continuous prompts with a frozen language model, substantially reduces per-task storage and memory usage at training. However, in the context of NLU, prior work and our results reveal that existing methods of prompt tuning do not perform well for normal-sized pretrained models and for hard sequence tasks, indicating lack of universality. We present a novel empirical finding that properly-optimized prompt tuning can be universally effective across a wide range of model scales and NLU tasks, where it matches the performance of finetuning while having only 0.1%-3% tuned parameters. Our method P-Tuning v2 is not a new method but a version of prefix-tuning (Li and Liang, 2021) optimized and adapted for NLU. Given the universality and simplicity of P-Tuning v2, we believe it can serve as an alternative for finetuning and a strong baseline for future research.[1]

1 Introduction

Pretrained language models improve performance on a wide range of natural language understanding (NLU) tasks such as question answering and textual entailment. A widely-used method, **finetuning**, updates the entire set of model parameters on a target task. While finetuning obtains good performance, it is memory expensive during training because gradients and optimizer states should be stored for all parameters. Moreover, finetuning requires storing a copy of model parameters for each task, which is inconvenient since pretrained models are usually large.

Prompting, on the other hand, freezes all parameters of a pretrained model and uses a natural language prompt to query a language model (Brown et al., 2020). For example, for sentiment analysis, we can concatenate a sample with a prompt "This movie is [MASK]" and ask the pretrained language model to predict the masked token. We can then use the conditional probabilities of "good" and "bad" being the masked token to predict the label of the sample. Prompting requires no training at all and stores one single copy of model parameters. However, prompting was shown to lead to suboptimal performance in many cases compared to finetuning (Liu et al., 2021b; Lester et al., 2021).

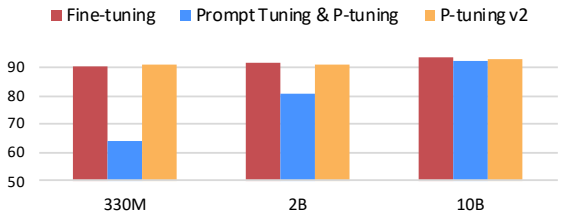


Figure 1: Average scores on RTE, BoolQ and CB of SuperGlue dev. With 0.1% task-specific parameters, P-tuning v2 can be comparable to fine-tuning across different scales of pre-trained models, while prompt tuning & P-tuning can only make it at over 10B scales.

Prompt tuning[2] is an idea of tuning only the continuous prompts. Specifically, Liu et al. (2021b); Lester et al. (2021) proposed to add trainable continuous embeddings to the original sequence of input word embeddings. These continuous embeddings (also called continuous prompts) are analogous to discrete manually-designed prompts as in prompting. Only the continuous prompts are updated during training. While prompt tuning improves over prompting on many tasks (Liu et al., 2021b; Lester et al., 2021), it still underperforms finetuning when the model size is less than 10 billion parameters (Lester et al., 2021). Moreover, as we will show in our experiments, prompt tuning performs poorly compared to finetuning on several hard sequence tasks such as ex-

[1]Codes will be at https://github.com/THUDM/P-tuning-v2
[†] corresponding to: Zhilin Yang (kimiyang@rcrai.com) and Jie Tang (jietang@tsinghua.edu.cn)
[*] indicates equal contribution.

[2]We use "prompt tuning" to refer to a class of methods rather than a particular method.

tractive question answering and sequence tagging.

Our main contribution in this paper is a novel empirical finding that properly-optimized prompt tuning can be comparable to finetuning universally across various model scales and NLU tasks. In contrast to observations in prior work, our finding reveals the universality and massive potential of prompt tuning for NLU.

Technically, we do not introduce any new method in this work. Our approach P-Tuning v2 can be viewed as optimizing and adapting prefix-tuning (Li and Liang, 2021), a method designed for generation, to NLU. The most significant improvement originates from using **deep prompt tuning**, which is to apply continuous prompts for every layer of the pretrained model (Li and Liang, 2021; Qin and Eisner, 2021). Deep prompt tuning increases the capacity of continuous prompts and closes the gap to finetuning across various settings, especially for small models and hard tasks. Moreover, we present a few optimization and implementation details for further enhancement of the results.

Experimental results show that P-Tuning v2 matches the performance of finetuning at different model scales ranging from 300M to 10B parameters, and on various hard NLU tasks such as question answering and sequence tagging. P-Tuning v2 has 0.1% to 3% trainable parameters per task compared to finetuning, which reduces training time memory cost and per-task storage cost substantially.

2 P-Tuning v2

Prompt tuning (Lester et al., 2021), or P-tuning (Liu et al., 2021b), introduces trainable continuous prompts as a substitution to natural language prompts for NLU when backbone language models' parameters are frozen. For example, let $\mathcal{V}$ refers to the vocabulary of a language model $\mathcal{M}$ and $\mathbf{e}$ serves as the embedding function for $\mathcal{M}$.

As shown in Figure 2, to classify a film review $\mathbf{x}$ ="Amazing movie!" as positive or negative, it is natural to think of appending a prompt "It is [MASK]" to the review and generating the conditional probabilities of the mask token being predicted as "good" or "bad" as the classification. In this case, prompt tokens {"It", "is", "[MASK]"} belong to model's vocabulary $\mathcal{V}$, and the input embedding sequence would be

$$[\mathbf{e}(\mathbf{x}), \mathbf{e}(\text{"It"}), \mathbf{e}(\text{"is"}), \mathbf{e}(\text{"[MASK]"})] \quad (1)$$

However, since the model $\mathcal{M}$ is intrinsically continuous, from the perspective of optimization, one can never achieve the optimum with discrete natural prompts. P-tuning, instead, proposes to replace prompt tokens to trainable continuous embeddings $[h_0, ..., h_i]$ and turn the input sequence into

$$[\mathbf{e}(\mathbf{x}), h_0, ..., h_i, \mathbf{e}(\text{"[MASK]"})] \quad (2)$$

and therefore can be differentially optimized. Prompt tuning and P-tuning have been proved quite effective in many NLU tasks, including knowledge probing, sentimental analysis (Li et al., 2021), relation extraction (Han et al., 2021), entity typing (Ding et al., 2021) and so on.

2.1 Lack of Universality

Nevertheless, P-tuning is not yet a comprehensive alternative to fine-tuning considering the following lack of universality:

Fail to work well on small models. (Lester et al., 2021) shows that prompt tuning can be comparable to fine-tuning when model scales to over 10 billion parameters. But for those smaller models (from 100M to 1B), there is a significant discrepancy between performances of prompt tuning and fine-tuning.

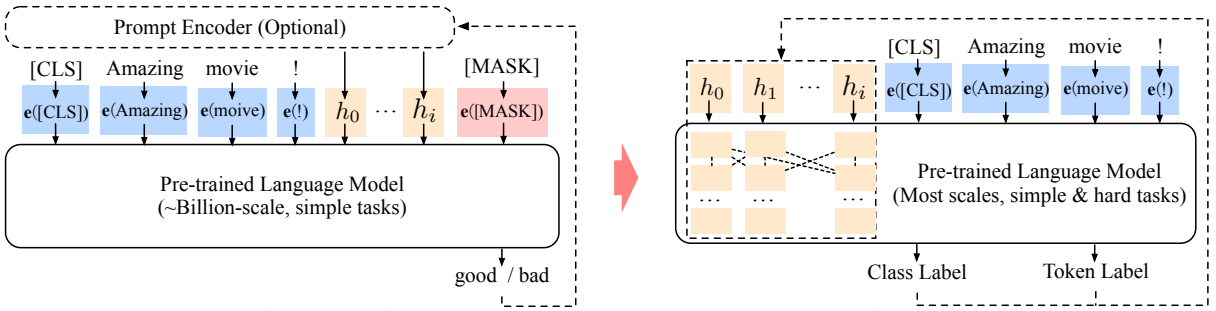
Fail to work well on hard NLU tasks. Though prompt tuning and P-tuning have shown superiority on NLU benchmarks such as GLUE and SuperGLUE, their effectiveness on another large family of hard NLU problems—sequence tagging—is not verified. In our experiment (Cf. Section 3.3 and Table 3), we show that prompt tuning performs poorly on typical sequence tagging tasks compared to fine-tuning, even for models at 10B scale.

Considering these challenges, we propose P-tuning v2, which adapts prefix-tuning as a universal solution across scales and NLU tasks.

2.2 Deep Prompt Tuning

Prefix-tuning (Li and Liang, 2021) was originally proposed for natural language generation (NLG) tasks, but we find it very effective for NLU as well. We describe a version of prefix-tuning adapted to NLU.

The main improvement of P-tuning v2 over P-tuning and Google prompt tuning comes from using multi-layer prompts as in prefix-tuning (Cf. Figure 2 (b)), which results in a larger number of tunable task-specific parameters (from 0.01% to



(a) Prompt tuning & P-tuning (b) P-tuning v2

Figure 2: From prompt tuning (P-tuning) to P-tuning v2. Given a context (e.g. "Amazing movie!"), prompt tuning can only deal with simple NLU tasks with the help of verbalizers (e.g. good/bad); P-tuning v2 does not rely on verbalizers and can extend to hard NLU tasks (e.g. extractive question answering and sequence tagging). Most importantly, while prompt tuning can only match fine-tuning at around 10B model scale, P-tuning v2 can match fine-tuning across most scales.

0.1%-3%) to allow for more per-task capacity (but still much smaller than full model).

2.3 Optimization and Implementation

There are also a few useful optimization and implementation details:

Optimization: Remove reparamerization. Previous methods leverage reparamerization function to increase training speed and robustness (e.g., MLP for prefix-tuning and LSTM for P-tuning). However, in P-tuning v2 we find reparamerization has minimal improvement (especially for smaller models), and it usually yields a second-best performance compared to directly updating the continuous prompts. Thus we remove it.

Optimization: Multi-task learning. Multi-task learning is optional for our method, but could be quite helpful. On one hand, the random initialization of continuous prompts brings in difficulties for optimization, which can be alleviated with more training data or task-related unsupervised pre-training (Gu et al., 2021); on the other hand, continuous prompts serve as perfect carriers of task-specific knowledge across tasks and datasets. Experiment shows that multi-task P-tuning v2 (MPT-2) can be a useful complement in some hard NLU tasks (Cf. Table 2,3,4).

Implementation: [CLS] and token classification, rather than verbalizers. Verbalizer (Schick and Schütze, 2020) has been a central component of prompt tuning, which turns one-hot class labels into meaningful words to make use of the pre-trained language model head. Despite its potential necessity in few-shot setting, verbalizer hinders the application of prompt tuning to scenarios where we need no-actual-meaning labels and sentence embeddings. Therefore, P-tuning v2 returns to the conventional [CLS] label and token label classification (Cf. Figure 2) paradigm with MLP heads.

3 Experiments

3.1 Setup

We conduct extensive experiment over different commonly-used pre-trained models and NLU tasks to verify the effectiveness of P-tuning v2.

NLU Tasks. First, we include part of datasets from GLUE (Wang et al., 2018) and SuperGLUE (Wang et al., 2019) benchmarks to test P-tuning v2's general NLU ability including SST-2, MNLI-m, RTE, BoolQ and CB. More importantly, we introduce a suite of tasks in the form of sequence tagging, which require language model to predict the class of every token in the input sequence, including Named Entity Recognition (CoNLL03 (Sang and De Meulder, 2003), OntoNotes 5.0 (Weischedel et al., 2013) and CoNLL04 (Carreras and Màrquez, 2004)), Extractive Question Answering (SQuAD 1.1 and SQuAD 2.0 (Rajpurkar et al., 2016)) and Semantic Role Labelling (CoNLL05 (Carreras and Màrquez, 2005) and CoNLL12 (Pradhan et al., 2012)).

Pre-trained Models. We include BERT-large (Devlin et al., 2018), RoBERTa-large (Liu et al., 2019), DeBERTa-xlarge (He et al., 2020), GLM-xlarge/xxlarge (Du et al., 2021) for evaluation. They are all bidirectional models designed for NLU purposes, covering a wide range of sizes from about 300M to 10B.

	#Size	GLUE dev						SuperGLUE dev								
		SST-2			MNLI-m			RTE			BoolQ			CB		
		FT	PT	PT-2	FT	PT	PT-2	FT	PT	PT-2	FT	PT	PT-2	FT	PT	PT-2
$\text{BERT}_{\text{large}}$	335M	93.2	92.4	**93.6** (+0.4)	**86.6**	75.6	85.8	70.4	53.5	**78.3** (+7.9)	**77.7**	67.2	75.8	**94.6**	80.4	**94.6** (+0.0)
$\text{RoBERTa}_{\text{large}}$	355M	**96.4**	95.3	96.3	90.2	83.8	**90.4** (+0.1)	86.6	58.8	**88.4** (+1.9)	**86.9**	62.3	84.8	98.2	71.4	**100** (+1.8)
$\text{GLM}_{\text{xlarge}}$	2B	-	-	-	-	-	-	**90.3**	85.6	**90.3** (+0.0)	**88.3**	79.7	87.0	**96.4**	76.4	**96.4** (+0.0)
$\text{GLM}_{\text{xxlarge}}$	10B	-	-	-	-	-	-	**93.1**	89.9	**93.1** (+0.0)	88.7	**88.8**	**88.8** (+0.1)	**98.7**	98.2	96.4

Table 1: Results on part of GLUE and SuperGLUE development set (all metrics are Accuracy). P-tuning v2 significantly surpasses P-tuning & prompt tuning on models smaller than 10B and matches the performance of fine-tuning. (FT: fine-tuning; PT: P-tuning & prompt tuning; PT-2: P-tuning v2).

	#Size	CoNLL03				OntoNotes 5.0				CoNLL04			
		FT	PT	PT-2	MPT-2	FT	PT	PT-2	MPT-2	FT	PT	PT-2	MPT-2
$\text{BERT}_{\text{large}}$	335M	**92.8**	81.9	90.2	91.0	**89.2**	74.6	86.4	86.3	85.6	73.6	84.5	**86.6** (+1.0)
$\text{RoBERTa}_{\text{large}}$	355M	92.6	86.1	92.4	**92.8** (+0.2)	**89.8**	80.8	89.4	**89.8** (+0.0)	88.8	76.2	87.3	**90.6** (+0.8)
$\text{DeBERTa}_{\text{xlarge}}$	750M	93.1	90.2	93.1 (+0.0)	**93.1** (+0.0)	90.4	85.1	90.4 (+0.0)	**90.5** (+0.1)	89.1	82.4	86.5	**90.1** (+1.0)

Table 2: Results on Named Entity Recognition (NER) test set (all metrics are micro-f1 score). P-tuning v2 is generally comparable to fine-tuning, and multitask P-tuning v2 can bring in a further improvement. (FT: fine-tuning; PT: P-tuning & prompt tuning; PT-2: P-tuning v2; MPT-2: Multi-task P-tuning v2)

Comparison Methods. We compare our P-tuning v2 (PT-2) with vanilla fine-tuning (FT), P-tuning & prompt tuning (PT). Additionally, for hard tasks regarding the sequence tagging, we present our results on multi-task P-tuning v2 (MPT-2) with more details presented in Section 3.3.

3.2 P-tuning v2: Across Scales

Table 1 presents P-tuning v2's performances across different model scales. For simple NLU tasks such as SST-2 (single sentence classification), prompt tuning & P-tuning do not show a significant disadvantage at smaller scale. But when it comes complicated challenges from natural language inference and question answering, their performance can be very poor. On the contrary, P-tuning v2 matches the performance of fine-tuning in all the tasks at smaller scale. To our surprise, P-tuning v2 significantly outperforms fine-tuning in RTE, especially for BERT.

In terms of larger scales (2B to 10B) with GLM (Du et al., 2021), the gap between P-tuning & prompt tuning and fine-tuning is gradually narrowed down. At 10B scale, we have a similar observation as is reported in (Lester et al., 2021), that prompt tuning becomes competitive to fine-tuning. However, P-tuning v2 is always comparable to fine-tuning at all scales we examine but with only 0.1% task-specific parameters needed compared to fine-tuning.

Additionally, we observe that in some datasets RoBERTa-large has a poorer performance than BERT-large. Part of the reason is that we empirically find prompt tuning can be quite sensitive to hyper-parameters, and sometimes the tuning just get trapped. P-tuning v2 can be more stable and robust during the tuning. For more details about hyper-parameters, please refer to our code repository.

3.3 P-tuning v2: Across Tasks

In Section 3.2, we discuss P-tuning v2's consistent comparable performance to fine-tuning whatever the scales. However, most tasks on GLUE and SuperGLUE are comparatively simple NLU problems. Another important family of hard NLU challenges lies in sequence tagging, which relates to some more high-level NLP applications including open information extraction, reading comprehension and so on.

To evaluate P-tuning v2's ability on these hard NLU challenges, we select three typical sequence tagging tasks: Name Entity Recognition, Extractive Question Answering (QA) and Semantic Role

	#Size	SQuAD 1.1 dev								SQuAD 2.0 dev							
		FT		PT		PT-2		MPT-2		FT		PT		PT-2		MPT-2	
		EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
$\text{BERT}_{\text{large}}$	335M	**84.2**	**91.1**	1.0	8.5	77.8	86.0	82.3	89.6	**78.7**	**81.9**	50.2	50.2	69.7	73.5	72.7	75.9
$\text{RoBERTa}_{\text{large}}$	355M	**88.9**	**94.6**	1.2	12.0	88.1	94.1	88.0	94.1	**86.5**	**89.4**	50.2	50.2	82.1	85.5	83.4	86.7
$\text{DeBERTa}_{\text{xlarge}}$	750M	90.1	95.5	2.4	19.0	**90.4** (+0.3)	**95.7** (+0.2)	89.6	95.4	88.3	91.1	50.2	50.2	**88.4** (+0.1)	**91.1** (+0.0)	88.1	90.8

Table 3: Results on Question Answering (Extractive QA). Prompt tuning & P-tuning performs extremely poor on question answering, while P-tuning v2's performance is generally reasonable, and can be better than fine-tuning with DeBERTa-xlarge. (FT: fine-tuning; PT: P-tuning & prompt tuning; PT-2: P-tuning v2; MPT-2: Multi-task P-tuning v2)

	#Size	CoNLL12				CoNLL05 WSJ				CoNLL05 Brown			
		FT	PT	PT-2	MPT-2	FT	PT	PT-2	MPT-2	FT	PT	PT-2	MPT-2
$\text{BERT}_{\text{large}}$	335M	84.9	64.5.	82.5	**85.1** (+0.2)	**88.5**	76.0	86.3	**88.5** (+0.0)	82.7	70.0	80.7	**83.1** (+0.4)
$\text{RoBERTa}_{\text{large}}$	355M	**86.5**	67.2	83.8	86.2	**90.2**	76.8	88.3	90.0	85.6	70.7	84.2	**85.7** (+0.1)
$\text{DeBERTa}_{\text{xlarge}}$	750M	86.5	74.1	83.9	**87.1** (+0.6)	**91.2**	82.3	90.6	91.1	86.9	77.7	86.3	**87.3** (+0.4)

Table 4: Results on Semantic Role Labelling (SRL). P-tuning v2 shows a consistent improvement over prompt tuning & P-tuning on SRL. (FT: fine-tuning; PT: P-tuning & prompt tuning; PT-2: P-tuning v2; MPT-2: Multi-task P-tuning v2)

Labelling (SRL), altogether eight datasets.

Name entity recognition (NER). NER aims to give all the spans of word that represent the given kinds of entity given a sentence. We adopted CoNLL03 (Sang and De Meulder, 2003), OntoNotes 5.0 (Weischedel et al., 2013) and CoNLL04 (Carreras and Màrquez, 2004). For CoNLL03 and CoNLL04, we trained our model on the standard train-develop-test split. For OntoNotes 5.0, we use the same train, develop, test split as (Xu et al., 2021). All the datasets are labeled in IOB2 format. We use sequence tagging to solve NER tasks by assigning labels marking the beginning and inside some kinds of entity. The language models generate representation of each tokens and we use a classifier to predict the labels. We use the official scripts to evaluate the results. For multi-task setting, we combine the training set of the three datasets for pre-training. We use different linear classifiers for each dataset while sharing the continuous prompts.

(Extractive) Question Answering (QA). Extractive QA is designed to extract the answer from the context given the context and a question. We use SQuAD (Rajpurkar et al., 2016) 1.1 and 2.0, in which each answer is within a continuous span of the context. Following tradition, we formulate the problem as sequence tagging by assigning one of the two labels: 'start' or 'end' to each token and at last selecting the span of the most possible start-end pair as the extracted answer. If the probability of the most possible pair is lower than a threshold, the model will decide the question to be unanswerable. For multi-task setting, the pre-train training set is a combination of the training set of SQuAD 1.1 and 2.0. When pre-training, we assume that all the questions, regardless of its origin, is probably unanswerable.

Semantic Role Labelling (SRL). SRL assigns labels to words or phrases in a sentence that indicates their semantic role in the sentence. We evaluate P-tuning v2 on CoNLL05 (Carreras and Màrquez, 2005) and CoNLL12 (Pradhan et al., 2012). Since a sentence can have multiple verbs, we add the target verb token to the end of each sentence to help recognize which verb is used for prediction. We classify each word using a linear classifier based on the corresponding semantic role representation. For multi-task setting, the pre-train training set is a combination of the training set of CoNLL05 (Carreras and Màrquez, 2005), CoNLL12 (Pradhan et al., 2012) and propbank-release (a common extend data used for training SRL). The multi-task training strategy is similar to NER.

Results. From Table 2,3,4, we observe that P-tuning v2 can be generally comparable to fine-

tuning on all tasks. P-tuning & prompt tuning show much poorer performance, especially on QA, which might be the most difficult challenge of three tasks. We also notice that there are some abnormal results presented in SQuAD 2.0 (BERT/RoBERTa/DeBERTa show the same performance using prompt tuning). This is probably because compared to SQuAD 1.1, SQuAD 2.0 contains unanswerable questions and the prompt tuning could possibly get the trivial solution.

Multi-task P-tuning v2 generally brings in great improvement over all tasks except for QA (which might still be the consequence of mixing all-answerable SQuAD 1.1 and not-answerable SQuAD 2.0), which implies that randomly initialized prompts' potential is under-explored.

4 Related Work

Pre-trained Language Models Self-supervised (Liu et al., 2020) pre-trained language models has become the backbone of natural language processing. From early stage when GPT (Radford et al., 2019), BERT (Devlin et al., 2018), XLNet (Yang et al., 2019), RoBERTa (Liu et al., 2019) has limited amount of parameters (less than 350M), the advent of T5 (Raffel et al., 2019) and GPT-3 (Brown et al., 2020) boosts the development of giant language models with billion and even trillions of parameters.

Prompting. Prompting (Liu et al., 2021a) refers to leverage special template in the input context to aid the language model prediction with respect to both understanding and generation. Recently, thanks to the success of GPT-3 (Brown et al., 2020), various prompting strategies including discrete natural language prompt (Shin et al., 2020; Gao et al., 2020), continuous prompts (Liu et al., 2021b; Li and Liang, 2021; Lester et al., 2021; Qin and Eisner, 2021), tuning bias (Logan IV et al., 2021) and many other prompting strategies have appeared. In this work, we are especially interested in scaling prompting methods to smaller models and hard NLU tasks.

5 Conclusion

We present P-tuning v2, a prompting method that can be comparable to fine-tuning universally across scales and tasks. P-tuning v2 is not a conceptually new approach, but a optimized and adapted one of prefix-tuning and deep prompt tuning. P-tuning v2 shows consistent improvement for models ranging from 330M to 10B, and outperforms prompt tuning & P-tuning on hard NLU tasks such as sequence tagging by a large margin. P-tuning v2 could be an comprehensive alternative for fine-tuning and strong baseline for future work.

References

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

Xavier Carreras and Lluís Màrquez. 2004. Introduction to the CoNLL-2004 shared task: Semantic role labeling. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 89–97, Boston, Massachusetts, USA. Association for Computational Linguistics.

Xavier Carreras and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 152–164, Ann Arbor, Michigan. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv e-prints*.

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

Ning Ding, Yulin Chen, Xu Han, Guangwei Xu, Pengjun Xie, Hai-Tao Zheng, Zhiyuan Liu, Juanzi Li, and Hong-Gee Kim. 2021. Prompt-learning for fine-grained entity typing. *arXiv preprint arXiv:2108.10604*.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2021. All nlp tasks are generation tasks: A general pretraining framework. *arXiv preprint arXiv:2103.10360*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2021. Ppt: Pre-trained prompt tuning for few-shot learning. *arXiv preprint arXiv:2109.04332*.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. Ptr: Prompt tuning with rules for text classification. *arXiv preprint arXiv:2105.11259*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Chengxi Li, Feiyu Gao, Jiajun Bu, Lu Xu, Xiang Chen, Yu Gu, Zirui Shao, Qi Zheng, Ningyu Zhang, Yongpan Wang, et al. 2021. Sentiprompt: Sentiment knowledge enhanced prompt-tuning for aspect-based sentiment analysis. *arXiv preprint arXiv:2109.08306*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Xiao Liu, Fanjin Zhang, Zhenyu Hou, Zhaoyu Wang, Li Mian, Jing Zhang, and Jie Tang. 2020. Self-supervised learning: Generative or contrastive. *arXiv preprint arXiv:2006.08218*, 1(2).

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv e-prints*.

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

Robert L Logan IV, Ivana Balažević, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel. 2021. Cutting down on prompts and parameters: Simple few-shot learning with language models. *arXiv preprint arXiv:2106.13353*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying lms with mixtures of soft prompts. *arXiv preprint arXiv:2104.06599*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Timo Schick and Hinrich Schütze. 2020. It's not just size that matters: Small language models are also few-shot learners. *arXiv preprint arXiv:2009.07118*.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv e-prints*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv e-prints*.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 23.

Lu Xu, Zhanming Jie, Wei Lu, and Lidong Bing. 2021. Better feature integration for named entity recognition. *arXiv preprint arXiv:2104.05316*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.