|  |  |  |
| --- | --- | --- |
| **Journal Pre-proof**  GPT understands, too  Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, Jie Tang | |  |
| PII: | S2666-6510(23)00014-1 |
| DOI: | <https://doi.org/10.1016/j.aiopen.2023.08.012> |
| Reference: | AIOPEN 62 |
| To appear in: | *AI Open* |

Received date : 19 August 2023   
Accepted date : 22 August 2023

Please cite this article as: X. Liu, Y. Zheng, Z. Du et al., GPT understands, too. *AI Open* (2023), doi: <https://doi.org/10.1016/j.aiopen.2023.08.012>.

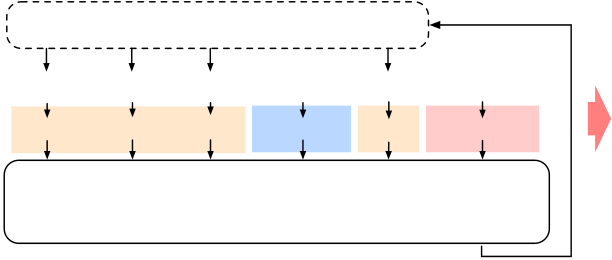
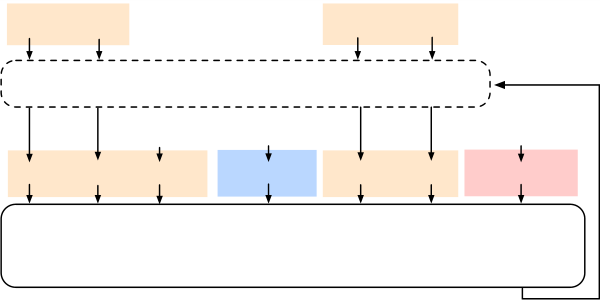
This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2023 The Authors. Publishing services by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an open access article under the CC BY license ([http://creativecommons.org/licenses/ by/4.0/](http://creativecommons.org/licenses/by/4.0/)).

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  | | --- | | **Journal Pre-proof** | | | | | | | | |  |
| Manuscript | Click here to view linked References | | | | | | | |  |
| 1 | **GPT Understands, Too** | | | | | | | |
| 2 | **Xiao Liu**1*∗***, Yanan Zheng**1*∗***, Zhengxiao Du**1**, Ming Ding**1**, Yujie Qian**2**, Zhilin Yang**1*†***, Jie Tang**1*†* | | | | | | | |
| 3 |
| 4 |
| 5 |
| 6 |
| 1Tsinghua University | | 2Massachusetts Institute of Technology | | | | | |
| 7 |
| 8 | **Abstract** | | | Prompt | | | P@1 | P@1 |
| 9 |
| 10 |
| 11 |
|  |
| 12 |  |
| 13 |
| w/o PT | w/ PT |
|  |
|  | Prompting a pretrained language model with | | | [X] is located in [Y]. *(original)* | | | 31.3 | 57.8 |  |
| 15 |
| natural language patterns has been proved effec- | | | | | [X] is located in which country or state? [Y]. 19.8 | | 57.8 |
| 16 |
|  | tive for natural language understanding (NLU). | | | | | [X] is located in which country? [Y]. | 31.4 | 58.1 |  |
| 17 |
|  | [X] is located in which country? In [Y]. | 51.1 | 58.1 |  |
| 18 | However, our preliminary study reveals that | | | | |
| 19 | manual discrete prompts often lead to unsta-ble performance—e.g., changing a single word in the prompt might result in substantial per-formance drop. We propose a novel method P-Tuning that employs trainable continuous prompt embeddings in concatenation with dis-crete prompts. Empirically, P-Tuning not only | | | Table 1: Discrete prompts suffer from instability (high variance), while P-Tuning stabilizes and improves per-formance. Results are precision@1 on LAMA-TREx P17 with BERT-base-cased. “PT” refers to P-Tuning, which trains additional continuous prompts in concate-nation with discrete prompts. | | | | |
| 20 |
| 21 |
| 22 |
| 23 |
| 24 |
| 25 |
| 26 | stabilizes training by minimizing the gap be- | | | | | | | |
| 27 |
| tween various discrete prompts, but also im-proves performance by a sizeable margin on | | | However, we observe that manual discrete | | | | |
| 28 |
| 29 |
| a wide range of NLU tasks including LAMA and SuperGLUE. P-Tuning is generally effec-tive for both frozen and tuned language models, under both the fully-supervised and few-shot settings. | | | prompts suffer from a large degree of instability. As shown in Table 1, with a frozen language model, changing a single word in the prompt might result in substantial performance drop. As we will show in Section 3, when the language model is tuned, | | | | |
| 30 |
| 31 |
| 32 |
| 33 |
| 34 |
| 35 |
| **1** | **Introduction** | | the instability problem is alleviated but the perfor- | | | | |
| 36 |
| mance difference between different prompts is still | | | | |
| 37 |
| Pretrained language models (PLMs; Brown et al., | | | sizeable, especially in the few-shot setting. Such | | | | |
| 38 |
| 39 | 2020) have significantly advanced the performance | | | an instability issue of discrete prompts poses a crit- | | | | |
| 40 | of natural language understanding (NLU). PLMs | | | ical challenge in practice. Recent approaches of | | | | |
| 41 |
| are trained with different pretraining objectives, | | | automatic prompting have attempted to search for a | | | | |
| 42 |
| such as masked language modeling (Devlin et al., | | | better-performing prompt given a task (Shin et al., | | | | |
| 43 |
| 44 | 2018), autoregressive language modeling (Radford | | | 2020; Gao et al., 2020; Jiang et al., 2020b), but | | | | |
| 45 | et al., 2019), seq2seq (Raffel et al., 2019), and per- | | | | these methods do not change the unstable nature of | | | |
| 46 | mutation language modeling (Yang et al., 2019). | | | | discrete prompts. | | | |
| 47 |
| PLMs can be further enhanced with prompting | | | To reduce the instability of discrete prompts, | | | | |
| 48 |
| (Brown et al., 2020; Schick and Schütze, 2020), | | | we propose a novel method P-Tuning that em- | | | | |
| 49 |
| 50 | which employs manually written prompt patterns as | | | ploys trainable continuous prompt embeddings in | | | | |
| 51 | additional input to a language model. With prompt- | | | | concatenation with discrete prompts. Specifically, | | | |
| 52 |
| ing while PLMs are either finetuned on a small la- | | | | given a discrete prompt as the input, P-Tuning con- | | | |
| 53 |
| beled dataset or frozen for direct inference on down- | | | | catenates continuous prompt embeddings with the | | | |
| 54  55 |
| stream tasks. Prompting has significantly improved | | | discrete prompt tokens and feeds them as the input | | | | |
| the performance of many NLU tasks (Brown et al., | | | to the language model. The continuous prompts are | | | | |
| 56 |
| 57 | 2020; Schick and Schütze, 2020). | | | updated by backpropagation to optimize the task | | | | |
| 58 |
| *†* corresponding to: Zhilin Yang (zhiliny@tsinghua.edu.cn) and Jie Tang (jietang@tsinghua.edu.cn) *∗* indicates equal contribution. | | | objective. The intuition is that continuous prompts incorporate a certain degree of learnability into the input, which may learn to offset the effects of mi- | | | | |
| 59 |
| 60 |
| 61 |
| 62 |
| 63 |
| 64  65 |

|  |  |
| --- | --- |
| **Journal Pre-proof** | |
| nor changes in discrete prompts to improve training | the instability issue, searching in the discrete space |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | stability. To further improve performance, we em- | | might not be able to fully leverage the gradients | | | |
| 2 | ploy a prompt encoder using LSTMs or MLPs to | | from backpropagation, which will potentially result | | | |
| 3 |
| model the dependency between continuous prompt | | in suboptimal solutions. To this end, we explore | | | |
| 4 |
| embeddings. | | the possibility of training continuous prompts to | | | |
| 5 |
| We experiment with two NLU benchmarks: the | | stabilize and improve the performance of language | | | |
| 6 |
| 7 | LAMA (Petroni et al., 2019) knowledge probing | | model adaptation. | | | |
| 8 | and SuperGLUE (Wang et al., 2019a). On LAMA, | | **2.2** | **P-Tuning** | | |
| 9 |
| with the language model frozen, P-Tuning out- | |
| 10 | with a hidden size of *h* and a vocabulary size of Formally, let *M* be a pretrained language model | | | |
| performs manual discrete prompts and searched | |
| 11 |
| 12 | prompts by 20+ points and 9 points respectively | |
| 13 | with the same pretrained models. On SuperGLUE, | | *|V|*. Let *{*(**x***i,* **y***i*))*}i* be a labeled dataset for an NLU task, where **x**0:*n* = *{x*0*, x*1*, ..., xn}* is an input consisting of a sequence of discrete tokens, | | | |
| 14 | with the language model finetuned, P-Tuning out- | |
| 15 |
| performs PET (Schick and Schütze, 2020) with | |
| 16 |
| the best discrete prompts under both the fully- | | and **y** *∈ Y* is a label. Our goal is to estimate the conditional probability for classification *fM*(*x*) = ˆ*p*(*y|x*) with parameters of *M* either finetuned or frozen. | | | |
| 17 |
| supervised and few-shot settings. In addition to im- | |
| 18 |
| 19 | proving performance, our results show that across | |
| 20 | a wide range of tasks and settings, P-Tuning sub- | |
| 21 | stantially reduces the performance gap between dif- | | Prompting was proposed in the format of | | | |
| 22 |
| ferent discrete prompts, which results in improved | | discrete tokens (Schick and Schütze, 2020). | | | |
| 23 |
| stability for language model adaptation. | | Let [D*i*] be a discrete prompt token. | | Each | |
| 24 |
| prompt can be described as a template *T* | | | = |
| 25 |
| **2** | **Method** |
| 26 | *{*[D0:*i*]*,* **x***,* [D(*i*+1):*j*]*,* **y***,* [D(*j*+1):*k*]*}*, which could organize the labeled data (including the inputs **x** | | | |
| 27 |
| **2.1** | **Issues with Discrete Prompts** |
| 28 |
| and the label **y**) into a sequence of text tokens, such | | | |
| 29 |
| Prompting employs natural language patterns as | | that the task could be reformulated as filling in the | | | |
| 30 |
| additional inputs to pretrained language models for | | blanks of the input text. For example, for the task of | | | |
| 31 |
| 32 | adaptation to downstream tasks (Brown et al., 2020; | | predicting a country’s capital (LAMA-TREx P36), | | | |
| 33 | Schick and Schütze, 2020). Prior work (Zheng | | a prompt could be “The capital of [INPUT] is [LA- | | | |
| 34 | et al., 2021) has pointed out that prompting has | | BEL].” With a piece of labeled data “(Britain, Lon- | | | |
| 35 |
| achieved consistent and substantial improvements | | don)”, the reformulated text would be “The capital | | | |
| 36 |
| on a number of NLP tasks. However, it still re- | | of Britain is [MASK].”, where “[MASK]" should | | | |
| 37 |
| 38 | mains a challenging problem of how to write high- | | predict the given label “London”. Both discrete | | | |
| 39 | performing discrete prompts. | | prompts and discrete data are together mapped into | | | |
| 40 | We performed preliminary experiments using | | input embeddings: | | | |
| 41 |
| different manual prompts on the LAMA knowledge | | *{***e**(*D*0)*...***e**(*Di*)*,* **e**(*x*0)*, ...,* **e**(*xn*)*, ...,* **e**(*Dk*)*}* | | | |
| 42 |
| probing task (Petroni et al., 2019), which aims to | |
| 43 |
| extract triplet knowledge from a language model | | through the pretrained embedding layer, where **e** *∈*R*|V|×d*. | | | |
| 44 |
| 45 | by predicting the tail entities. Results in Table 1 | |
| 46 | show that manual discrete prompts lead to unstable | | However, as is discussed in Section 2.1, such | | | |
| 47 |
| performance. For example, if we compare the last | | discrete prompts tend to be extremely unstable | | | |
| 48 |
| two prompts in the table, changing a single word | | and might not be optimal with back-propagation. | | | |
| 49 |
| in prompt causes a drastic decrease of 20 points in | | Therefore, we propose P-Tuning that uses contin- | | | |
| 50 |
| 51 | performance. | | uous prompt embeddings to improve and stabilize | | | |
| 52 | In light of the challenge, recent works propose to | | prompting. Let [P*i*] be the *i*thcontinuous prompt | | | |
| 53 | embedding. The prompt template for P-Tuning is | | | |
| automate the search procedure of discrete prompts | |
| 54 55 |
| by mining the training corpus (Jiang et al., 2020b), | | as follows: | | | |
| gradient-based searching (Shin et al., 2020), and us- | | *T* = *{*[P0:*i*]*,* **x***,* [P(*i*+1):*j*]*,* **y***,* [P(*j*+1):*k*]*}*  P-Tuning leverages an extra embedding function | | | |
| 56 |
| 57 | ing pretrained generative models (Gao et al., 2020). | |
| 58 |
| However, these works aim at searching for better- | |
| 59 | *f* : [P*i*] *→ hi* to map the template to | | | |
| performing prompts but do not change the nature | |
| 60 |
| of instability for discrete prompts. In addition to | | *{h*0*, ..., hi,* **e**(**x**)*, hi*+1*, ..., hj,* **e**(**y**)*, hj*+1*, ..., hk}* | | | |
| 61 |
| 62 |
| 63 |
| 64 65 |



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Journal Pre-proof** | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1   2   3   4   5   6   7   8   9  10  11  12  13  14  15 | Prompt Generator | | | | | | | Discrete rewards | | Pseudo Prompts | | | | | [P0] | … | [P*i*] | | | Prompt Encoder | | | [P*i*+1] | ]… | [P*m*] | | Back |
| Propagation |
| The | | | capital | of | Britain | is | | [MASK] | capital | | | | | | | | | | | Britain | | | [MASK] | | | |
| Input embedding | | **e**(T he) | **e**(capital) | **e**(of) | **e**(B ritain) | **e**(is) | | **e**([M A SK ]) | Input embedding | | | | | *h*0 | … | *hi* | **e**(capital) | | | **e**(B ritain) | | *hi*+1 | … | *hm* | **e**([M A SK ]) | |
| Pre-trained Language Model | | | | | | | | | Pre-trained Language Model | | | | | | | | | | | | | | | | | |
| (GPT, BERT, …� | | | | | | | | | (GPT, BERT, …� | | | | | | | | | | | | | | | | | |
| (a) Discrete Prompt Search | | | | | | | | | (b) P-tuning | | | | | | | | | | | | | | | | | |
| Figure 1: An example of prompt search for “The capital of Britain is [MASK]”. Given the context (blue zone,“Britain”) and target (red zone, “[MASK]”), the orange zone refer to the prompt. In (a), the prompt generator only receives discrete rewards; on the contrary, in (b) the continuous prompt embeddings and prompt encoder can be optimized in a differentiable way. | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Finally, we update the embeddings *{Pi}k* timize a task loss function. *i*=1to op- | | | | | | | | | | | Improved | | | | | | | LAMA | | | Full SG | | Few SG | | | |
| 16 |
| 17  18 | frozen | | | tuned | | tuned | | | |
| It is noteworthy that we can also concatenate | | | | | | | | | | |
| 19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65 | ✓ | | | ✓ | | ✓ | | | |
| discrete prompts with continuous prompts, which | | | | | | | | | | |
| Stabilized | | | | | | | ✓ | | | ✗ | | ✓ | | | |
| performs better and is adopted throughout our ex- | | | | | | | | | | |
| periments. P-Tuning is applicable to both frozen and finetuned language models. | | | | | | | | | Table 2: Task settings and summary of results in our experiments. P-tuning shows improvement over base- | | | | | | | | | | | | | | | | | |
| **2.3** | **Prompt Encoder** | | | | | | | | lines on all task settings, and can stabilize performance | | | | | | | | | | | | | | | | | |
| on LAMA and Few SG. For Full SG, the gap between | | | | | | | | | | | | | | | | | |
| In the aforementioned framework, we employ a mapping function *f* to map trainable embeddings *{Pi}* to model inputs *{hi}*. The intuition is that by using a mapping function, it is more conve-nient to model the dependency between different prompt embeddings, compared to using indepen- | | | | | | | | | | discrete prompts is not large and training is stable even without P-Tuning. (Full SG: fully-supervised learn-ing on SuperGLUE; Few SG: few-shot SuperGLUE; Improved: overall performance improved; Stabilized: training stabilized by minimizing difference between discrete prompts). | | | | | | | | | | | | | | | | |
| dent learnable embeddings. In our implementation, we use a lightweight neural network to formulate the function *f*. Specifically, we experiment with using long short-term memory (LSTM) networks, | | | | | | | | | with both tuned and frozen language models.   The overall task setup and a summary of results are shown in Table 2. | | | | | | | | | | | | | | | | | |
| multi-layer perceptrons (MLPs), and the identity mapping function in Section 3. | | | | | | | | | **3.1** | | | **Knowledge Probing** | | | | | | | | | | | | | | |
| **3.1.1** | | | | **Setup** | | | | | | | | | | | | | |
| **3** | **Experiments** | | | | | | | |
| Knowledge probing, or referred to as fact retrieval, | | | | | | | | | | | | | | | | | |
| We include two NLU benchmarks: LAMA (Petroni et al., 2019) for knowledge probing (§ 3.1) and Su-perGLUE (Wang et al., 2019a) for general natural language understanding. On SuperGLUE, we con-sider both the fully-supervised learning (§ 3.2) and | | | | | | | | | | evaluates how much real-world knowledge has language models gained from pre-training. The LAMA (Petroni et al., 2019) dataset evaluates it with cloze tests created from triples selected in the knowledge bases. | | | | | | | | | | | | | | | | |
| few-shot learning (§ 3.3) settings. | | | | | | | | | **Datasets and vocabulary.** LAMA enforces all | | | | | | | | | | | | | | | | | |
| On LAMA, following Shin et al. (2020); Jiang | | | | | | | | | answers in single-token format. We first adopt | | | | | | | | | | | | | | | | | |
| et al. (2020b), language models are frozen and only | | | | | | | | | the original LAMA-TREx dataset, consisting of | | | | | | | | | | | | | | | | | |
| the discrete or continious prompts are tuned. For | | | | | | | | | 41 Wikidata relations and altogether 34,039 test- | | | | | | | | | | | | | | | | | |
| SuperGLUE, following Schick and Schütze (2020); | | | | | | | | | ing triples (namely LAMA-34k, which covers all | | | | | | | | | | | | | | | | | |
| Zheng et al. (2021), language models are tuned. In | | | | | | | | | BERT vocabularies). Since different pretrained | | | | | | | | | | | | | | | | | |
| our setting, we jointly optimize the language model | | | | | | | | | models share distinct vocabularies, to allow direct | | | | | | | | | | | | | | | | | |
| parameters and the continuous prompts. This setup | | | | | | | | | comparison, we follow previous work (Shin et al., | | | | | | | | | | | | | | | | | |
| not only follows the common, standard settings in | | | | | | | | | 2020) to adopt a subset that covers the intersection | | | | | | | | | | | | | | | | | |
| prior work, but also allows evaluating P-Tuning | | | | | | | | | of GPT’s and BERT’s vocabularies. This is caled | | | | | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Journal Pre-proof** | | | | | | | | | | |
|  |
| Prompt type | | Model | P@1 | | Model | | | | MP | P-tuning |
| Original | |  | 31.1 | | BERT-base (109M) | | | | 31.7 | 52.3 (+20.6) |
| 1   2   3   4 | BERT-base |
| BERT-large | 32.3 | | -AutoPrompt (Shin et al., 2020) | | | | - | 45.2 |
| (MP) | | BERT-large (335M) | | | | 33.5 | 54.6 (+21.1) |
| E-BERT | 36.2 | |
|  | Discrete | | LPAQA (BERT-base) | 34.1 | RoBERTa-base (125M) | | | | | 18.4 | 49.3 (+30.9) |
| 5   6   7 |
| -AutoPrompt (Shin et al., 2020) | | | | | - | 40.0 |
| LPAQA (BERT-large) | 39.4 |
| RoBERTa-large (355M) | | | | | 22.1 | 53.5 (+31.4) |
| AutoPrompt (BERT-base) | 43.3 |
| 8 | GPT2-medium (345M) | | | | | 20.3 | 46.5 (+26.2) |
| 9  10 | P-tuning | | BERT-base | 48.3 |
| GPT2-xl (1.5B) | | | | | 22.8 | 54.4 (+31.6) |
| BERT-large | **50.6** | MegatronLM (11B) | | | | | 23.1 | **64.2** (+41.1) |
| Table 3: Knowledge probing Precision@1 on LAMA-34k (left) and LAMA-29k (right). P-tuning outperforms all the discrete prompt searching baselines. (MP: Manual prompt; PT: P-tuning). | | | | | | | | | | |
| 11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65 |
| LAMA-29k. We again follow Shin et al. (2020) to | | | | | **3.2** | | **Fully-supervised Learning** | | | |
| construct the training, development, and test data | | | | | **3.2.1** | | **Setup** | | | |
| to allow for fair comparison. | | | | |
| **Setup.** LAMA has provided a handcraft prompt for each relation, as shown in Table 1, which are effective but likely sub-optimal. For bidirectional masked language models, we only need to replace“[X]” with the subject entity and “[Y]” with the [MASK] token; for unidirectional language models such as GPT, following LAMA’s original setting on Transformer-XL (Dai et al., 2019), we use the network output just before the target position. | | | | | **Dataset.** To evaluate P-tuning on fully-supervised learning tasks, we adopt the SuperGLUE bench-mark (Wang et al., 2019b), consisting of 8 challeng-ing natural language understanding (NLU) tasks. We focus on 7 of them since the ReCoRD (Zhang et al., 2018) task adopts no discrete prompts, thus P-tuning is not directly applicable. The tasks in-clude question answering (BoolQ (Clark et al., 2019a) & MultiRC (Khashabi et al., 2018)), tex-tual entailment (CB (De Marneffe et al., 2019) & | | | | | |
| The number of prompt tokens and positions are | | | | | RTE (Dagan et al., 2005)), co-reference resolution | | | | | |
| selected based on the development sets, and for | | | | | (WiC (Pilehvar and Camacho-Collados, 2018)), | | | | | |
| simplicity we choose the (3, sub, org\_prompt, 3, | | | | | causal reasoning (COPA (Roemmele et al., 2011)), | | | | | |
| obj, 3) template for bidirectional models and (3, | | | | | and word sense disambiguation (WSC (Levesque | | | | | |
| sub, org\_prompt, 3, obj) for unidirectional models | | | | | et al., 2012)). | | | | | |
| as this configuration performs well for most rela-tions (where the number indicates the number of continuous prompt tokens). Continuous prompts are concatenated with original discrete prompts. During the prompt training, we set the learning rate to 1e-5 and use the Adam optimizer. | | | | | | **Comparison methods.** We experiment with P-tuning on both unidirectional and bidirectional pretrained models, i.e., GPT and BERT. We include four variants BERT-Base, BERT-Large, GPT2-Base, and GPT-medium. For each model, we compare standard classification finetuning, | | | | |
| PET (Schick and Schütze, 2020) (a typical fine- | | | | | | | | | | |
| **3.1.2** | **Main results** | | tuning method based on manual discrete prompts) | | | | | | | |
| and our P-tuning. | | | | | | | |
| The results are presented in Table 3. P-tuning sig- | | | | | | **Configuration.** | | We use the same metrics as | | |
| nificantly improves the best results of knowledge | | | | | in (Wang et al., 2019b). For fully-supervised learn- | | | | | |
| probing from 43.3% to 50.6% on LAMA-34k and | | | | | ing, we use a large training set to finetune pre- | | | | | |
| from 45.2% to 64.2% on LAMA-29k. Moreover, | | | | | trained models and use a development set for hyper- | | | | | |
| P-tuning outperforms previous discrete prompt | | | | | parameter and model selection. Specifically, the | | | | | |
| searching approaches such as AutoPrompt (Shin | | | | | AdamW optimizer with a linearly decayed learn- | | | | | |
| et al., 2020) and LPAQA (Jiang et al., 2020b) on | | | | | ing rate is used for training. We use a learning | | | | | |
| the same-size models. This confirms our intuition in Section 2 that discrete prompts might not be optimal. | | | | | rate of *{*1*e −* 5*,* 2*e −* 5*,* 3*e −* 5*}*, a batch size of *{*16*,* 32*}*, and a warm-up ratio of *{*0*.*0*,* 0*.*05*,* 0*.*1*}*. For small datasets (i.e., COPA, WSC, CB, RTE), | | | | | |

|  |
| --- |
| **Journal Pre-proof** |

(a) Fully-supervised performance with base-scale models.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 2 | Method | | | BoolQ | CB | | | (F1) | | WiC | | RTE | MultiRC | | | WSC | COPA | | Avg. |
| (Acc.) | (Acc.) | | | (Acc.) | | (Acc.) | (EM) | (F1a) | | (Acc.) | (Acc.) | |
| 3 4 5 | BERT-Base (109M) | | CLS-FT | 72.9 | 85.1 | 73.9 | | | | 71.1 | | 68.4 | 16.2 | 66.3 | | 63.5 | 67.0 | | 66.2 |
| PET-FT | 73.7 | 87.5 | 90.8 | | | | 67.9 | | 70.4 | 13.7 | 62.5 | | 60.6 | 70.0 | | 67.1 |
| P-tuning | 73.9 | 89.2 | 92.1 | | | | 68.8 | | 71.1 | 14.8 | 63.3 | | 63.5 | 72.0 | | 68.4 |
| 6 7 8 | GPT2-Base (117M) | | CLS-FT | 71.2 | 78.6 | 55.8 | | | | 65.5 | | 67.8 | 17.4 | 65.8 | | 63.0 | 64.4 | | 63.0 |
| PET-FT | 74.8 | 87.5 | 88.1 | | | | 68.0 | | 70.0 | 23.5 | 69.7 | | 66.3 | 78.0 | | 70.2 |
| P-tuning | 75.0 | 91.1 | 93.2 | | | | 68.3 | | 70.8 | 23.5 | 69.8 | | 63.5 | 76.0 | | 70.4 |
| 9 10 11 | (b) Fully-supervised performance with large-scale models. | | | | | | | | | | | | | | | | | | |
| 12 13 | Method | | | BoolQ | CB | | | | (F1) | WiC | | RTE | MultiRC | | | WSC | | COPA | Avg. |
| (Acc.) | (Acc.) | | | | (Acc.) | | (Acc.) | (EM) | | (F1a) | (Acc.) | | (Acc.) |
| 14 15 16 | BERT-Large | | CLS-FT1 | 77.7 | 94.6 | | 93.7 | | | 74.9 | | 75.8 | 24.7 | 70.5 | | 68.3 | 69.0 | | 72.5 |
| PET-FT | 77.2 | 91.1 | | 93.5 | | | 70.5 | | 73.6 | 17.7 | 67.0 | | 80.8 | 75.0 | | 73.1 |
| (335M) | |
| P-tuning | 77.8 | 96.4 | | 97.4 | | | 72.7 | | 75.5 | 17.1 | 65.6 | | 81.7 | 76.0 | | 74.6 |
| 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 | GPT2-Med. (345M) | | CLS-FT | 71.0 | 73.2 | | 51.2 | | | 65.2 | | 72.2 | 19.2 | 65.8 | | 62.5 | 66.0 | | 63.1 |
| PET-FT | 78.3 | 96.4 | | 97.4 | | | 70.4 | | 72.6 | 32.1 | 74.4 | | 73.0 | 80.0 | | 74.9 |
| P-tuning | 78.9 | 98.2 | | 98.7 | | | 69.4 | | 75.5 | 29.3 | 74.2 | | 74.0 | 81.0 | | 75.6 |
| 1 We report the same results taken from SuperGLUE (Wang et al., 2019a).   Table 4: Fully-supervised performance on SuperGLUE development set. | | | | | | | | | | | | | | | | | | |
| we fine-tune pretrained models for 20 epochs. For | | | | | | | | | the few-shot learning performance of pretrained | | | | | | | | | |
| larger datasets (i.e., WiC, BoolQ, MultiRC), we | | | | | | | | | models on challenging tasks. | | | | | | | | | |
| reduce the number of training epochs to be 10 as | | | | | | | | | **3.3.1** | | **Setup** | | | | | | | |
| the model converges earlier. Early stopping is used | | | | | | | | |
| to avoid over-fitting the training data. | | | | | | | | | **Few-shot Evaluation.** The few-shot performance | | | | | | | | | |
| **3.2.2** | **Main Results** | | | | | | | | | is sensitive to lots of factors (e.g., the order of train-ing examples, random seed, and prompt patterns), and thus suffers from high variance (Zhao et al., 2021a; Lu et al., 2021; Zhang et al., 2020). There-fore, the few-shot evaluation strategy should make sure that the improvements are indeed from an im-proved method instead of variance. To this end, we follow the FewNLU evaluation procedure (Zheng et al., 2021) that has addressed and handled the issue. Specifically, we use random data splits to perform model selection only on a small labeled set to prevent overfitting a large dev set.  **Dataset.** We use the few-shot SuperGLUE (also known as FewGLUE) benchmark (Schick and Schütze, 2020) and follow the setting in prior work (Zheng et al., 2021) in terms of data split construc-tion. | | | | | | | | |
| The main results of fully-supervised learning are shown in Table 4. We observe that P-tuning can improve fully-supervised learning performance on both BERTs and GPTs. (1) Specifically, on the BERT-Base model, P-tuning achieves best perfor-mance on 5/7 tasks, while with BERT-Large, P-tuning outperforms other methods on 4/7 tasks. The exceptions are WiC and MultiRC, both of which have relatively large training sets. We find that P-tuning might not have large gains over CLS-FT on such high-resource tasks, while benefits more on low-resource tasks. On average, P-tuning improves over the considered baselines. (2) On GPT2-Base and GPT2-Medium models, P-tuning consistently achieves the best performance on all tasks. | | | | | | | | | |
| **3.3** | **Few-Shot Learning** | | **Baseline and Hyper-parameter.** In few-shot learn- | | | | | | | | | | | | | | | |
| ing, we again compare P-tuning with PET (Schick | | | | | | | | | | | | | | | |
| While GPT-3 has shown decent few-shot learning | | | | | | | | | and Schütze, 2020), which was shown to out- | | | | | | | | | |
| potential with handcrafted prompts, it still struggles | | | | | | | | | perform GPT-3 on some of the tasks. | | | | | | | | | Similar |
| on some of the challenging tasks (e.g., natural lan- | | | | | | | | | | to (Schick and Schütze, 2020), we use ALBERT- | | | | | | | | |
| guage inference) (Brown et al., 2020). We are mo- | | | | | | | | | | xxLarge as the base model. For hyper-parameters | | | | | | | | |
| tivated to study whether P-tuning can also improve | | | | | | | | | that are shared by PET and P-tuning (e.g., learn- | | | | | | | | | |

|  |  |
| --- | --- |
| **Journal Pre-proof** | |
| ing rate, maximum training step, evaluation fre- | that it would be better if we insert continuous |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | quency), we use the same search space for fair | | prompt tokens at the location where it does not | |
| 2 | comparison. Specifically, we search the learning | | segment the sentences. For example, in case#1, | |
| 3 |
| rate in *{*1*e −* 5*,* 2*e −* 5*}*, the maximum training step in *{*250*,* 500*}*, and the evaluation frequency in *{*0*.*02*,* 0*.*04*}*.  **Construction of Prompt Patterns.** For PET, we | | “[P]” breaks the completeness of sentence “[Hy- | |
| 4 |
| pothesis]?” while in case#3, “[P]” is located | |
| 5 |
| between sentences. | |
| 6 |
| 7 | 2. By comparing #2 (or #3) with #4, we find that | |
| 8 |
| use the same manual prompts reported by Schick | | there’s no special preference for placing on the | |
| 9 |
| and Schütze (2020). When constructing prompt | | edge or in the middle of the inputs. | |
| 10 |
| 11 | patterns for P-tuning, based on the same manual | | 3. It is suggested to write a number of pattern can- | |
| 12 |
| prompts as PET, we insert different numbers of | | didates and then search over them for the best | |
| 13 |
| continuous prompt tokens into different positions, | | for each task. | |
| 14 |
| thus formulating a number of pattern candidates. | |
| 15 |
| **Number of Prompt Tokens** We also study the in- | |
| 16 | We then select the best pattern for P-tuning using | |
| 17 | the validation strategy of FewNLU (Zheng et al., | | fluence of the number of prompt tokens and show | |
| 18 | 2021). We also conduct further analysis of the num- | | the results in Table 7. By comparing #3, #6, #7, | |
| 19 |
| ber and the position of continuous prompt tokens | | and #8, we can conclude that the number of prompt | |
| 20 |
| in §3.3.3. | | tokens has a great impact on the few-shot perfor- | |
| 21 |
| **3.3.2** | **Main Results** | mance. However, it is not that a larger number of | |
| 22 |
| 23 | prompt tokens would always be better. We conjec- | |
| 24 | **Few-Shot Performance.** Table 5 shows the main | |
| ture that it could be that due to the limited training | |
| 25 |
| data, it becomes difficult to learn the parameters | |
| 26 | results of few-shot learning. We find that, on AL- | |
| when excessively increasing the number of contin- | |
| 27 | BERT, P-tuning consistently outperform PET on | |
| 28 | uous prompt tokens. In practice, it is suggested | |
| average by more than 1 points. It outperforms | |
| 29 | to search for the best number of prompt tokens | |
| PromptTuning by more than 13 points. It proves | |
| 30 |
| through model selection. | |
| that by automatically learning continuous prompt | |
| 31 |
| 32 | tokens, the pretrained models can achieve better | |
| 33 | few-shot performance on NLU tasks. | | **3.3.4** | **Comparison with Discrete Prompt** |
| 34 |
| **3.3.3** | **Ablation Study** | **Search** | |
| 35 |
| 36 | Prior work (Gao et al., 2020) proposed to automati- | |
| 37 | **Type of Prompt Encoder** Prior work (Shin et al., | |
| 38 | cally search discrete prompts and achieved better | |
| 2020) proposes to simply use an MLP as the prompt | |
| 39 | results than those of manual prompts. We now | |
| encoder, we perform further ablation analysis for | |
| 40 |
| proceed to compare P-Tuning with auto-searched | |
| prompt encoder selection, and results are shown | |
| 41 |
| discrete prompts. For fair comparison, we follow | |
| 42 | in Table 8. We consider LSTM, MLP, and EMB | |
| the setting of LM-BFF (Gao et al., 2020) to also | |
| 43 | (i.e., we directly optimize the word embeddings | |
| 44 | conduct experiments on some of the GLUE tasks | |
| without using additional parameters). From the | |
| 45 | (Wang et al., 2018) with RoBERTa-Large model | |
| results, we can see that LSTM, MLP, and EMB | |
| 46 |
| (Liu et al., 2019). Since the the evaluation proto- | |
| all work as a prompt encoder. Results show that | |
| 47 |
| cols have large impacts on few-shot performance, | |
| both LSTM and MLP generally work well on these | |
| 48 |
| we use the top-3 discrete prompts searched by LM- | |
| 49 | tasks, while EMB is unstable and can substantially | |
| 50 | BFF and experiment with using only the discrete | |
| under-perform the other two on some tasks (e.g,. | |
| 51 | prompts and additionally applying P-Tuning. For | |
| WiC and CB). To sum up, both LSTM and MLP | |
| 52 |
| P-Tuning, the prompt patterns are constructed by | |
| could be taken into account when working on new | |
| 53 |
| concatenating the same discrete prompts as well as | |
| tasks. | |
| 54 55 |
| continuous prompts. Results in Table 9 show that | |
| **Location of Prompt Tokens** To study at which | |
| additionally incorporating continuous prompts can | |
| 56 |
| location to insert continuous prompt tokens, we | | further improve few-shot performance. P-Tuning is | |
| 57 |
| 58 | perform experiments as Table 7 shows. From the | |
| easy to be combined with existing discrete prompts, | |
| 59 | results, we have the following findings. | |
| while further improving stability as discussed in | |
| 60 |
| 1. By comparing #1 (or #2) with #3 (or #4), we find | | Section 3.4. | |
| 61 |
| 62 |
| 63 |
| 64 65 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  | | --- | | **Journal Pre-proof** | | | | | | | | | | | | | | | | | | | | |
| 1 | Method | | BoolQ | RTE | | WiC | | (Acc.) | CB | | (F1.) | MultiRC | | | | | | WSC | COPA | Avg |
| (Acc.) | (Acc.) | | (Acc.) | | (F1a.) | (EM.) | | | | | (Acc.) | (Acc.) |
| 2   3   4 | Prompt Tuning | | 58.47 *±*1.00 76.70 *±*1.85 76.55 *±*2.68 | 54.42 *±*3.05 72.83 *±*1.30 63.27 *±*3.63 | | 52.74 *±*2.36 53.87 *±*4.47 55.49 *±*1.21 | | 75.45 *±*2.25 84.38 *±*4.47 88.39 *±*3.72 | | 67.73 *±*5.70 62.56 *±*7.66 84.24 *±*5.15 | | 59.28 *±*4.73 76.51 *±*1.52 75.91 *±*1.74 | | 15.03 *±*4.11 36.46 *±*2.13 38.01 *±*0.78 | | | 74.04 *±*2.99 80.05 *±*2.53 78.85 *±*1.76 | | 61.50 *±*4.36 81.75 *±*4.03 85.25 *±*3.30 | 58.56 |
| PET-FT | | 70.74 |
| P-tuning | | 71.81 |
| 5   6   7   8   9  10 |  |
| Table 5: The few-shot performance of PET (Schick and Schütze, 2020), Prompt Tuning (Lester et al., 2021) and our P-tuning over seven tasks based on ALBERT. Each result is averaged over 4 runs with different data splits. Results show that P-tuning consistently improves average few-shot performance by more than 1 point compared to PET and by more than 13 points compared to Prompt Tuning. | | | | | | | | | | | | | | | | | | | |
| 11 |
| Method | | | | | | P#0 | P#1 | | P#2 | | P#3 | P#4 | | P#5 | | | STD | | |
| 12  13  14  15  16  17  18 | FSL | | | | PET-FT | | 77.10 | 67.96 | | 74.14 | | 72.48 | 71.77 | | | 60.86 | | 5.68 | | |
| *±*2.21 75.41 | *±*2.69 75.11 | | *±*1.38 73.43 | | *±*4.31 71.35 | *±*2.56 71.31 | | | *±*3.99 65.86 | |
| (BoolQ) | | | | P-tuning | | 3.52 | | |
| *±*3.09 | *±*1.61 | | *±*2.60 | | *±*4.57 | *±*8.58 | | | *±*3.80 | |
| LAMA | | | | MP | | 31.3 | 19.8 | | 31.4 | | 51.1 | 34.0 | | 32.7 | | | 10.1 | | |
| (P17) | | | | P-tuning | | 57.8 | 57.8 | | 58.1 | | 58.1 | 58.9 | | 58.7 | | | 0.46 | | |
| 19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65 | Table 6: Upper table: Few-shot learning (FSL) of PET and P-tuning in terms of each pattern on SuperGLUE with ALBERT; Lower table: Manual prompt (MP) and P-tuning performance on LAMA-P17 with BERT-base-cased. For each column, P-tuning and compared methods share the same manual prompts, while P-tuning additionally concatenates continuous prompt tokens. We report the standard deviation over multiple results of different patterns. Results show that P-tuning achieves smaller standard deviation, proving that P-tuning can improve stability w.r.t. the choice of discrete patterns. | | | | | | | | | | | | | | | | | | | |
| **3.4** | **Stabilizing Language Model Adaptation** | | | | | | | | 2021; Zhao et al., 2021b) as a way of prompting to | | | | | | | | | | |
| In the above sections, we have shown that P-Tuning improves over performance across multiple set-tings. Now we present results to demonstrate that P-Tuning also stabilizes language model adapta-tion; i.e., reducing the differences between differ-ent prompts. As we have shown in Table 1, manual prompts have a large impact on the performance. When it comes to few-shot learning, the perfor-mance gap of different prompts is prominent due to the sensitivity of few-shot learning (Zheng et al., 2021). Results in Table 6 show that P-tuning im-proves the performance of the worst-performing patterns (e.g., P#5), and achieves a smaller stan-dard deviation over multiple patterns. Compared to PET-FT, P-tuning increases the stability w.r.t. the choice of patterns.  On LAMA, we observe similar a phenomenon that while manual prompts often yield quite volatile results, appending trainable continuous prompts on top of the manual prompts can stabilize their per-formances, reducing the standard deviation from | | | | | | | | | transfer knowledge from pretraining to downstream tasks. Schick and Schütze (2020) proposed to use cloze patterns, which removes the constraint that the masked token is the last token of the sentence. This further minimizes the gap between pretrain-ing and downstream tasks. To improve prompting for NLU, recent works have proposed methods to automatically search for high-performing prompts by mining the training corpus (Jiang et al., 2020b), gradient-based search (Shin et al., 2020), or using pretrained generative models (Gao et al., 2020). Our approach is different from these prior works in that we resort to using continuous prompt em-beddings, which are found to be complementary to discrete prompts in our experiments.  Recently, some concurrent works also proposed the use of continuous prompts. Prefix-tuning (Li and Liang, 2021) adds continuous prompts at the beginning of the sequence for each layer. In con-trast to our work, prefix-tuning targets natural lan-guage generation tasks. | | | | | | | | | | |
| 10.1 to 0.46. | | In the area of NLU, a few concurrent methods | | | | | | | | | | | | | | | | | |
| **4** | **Related work** | | | were proposed based on continuous prompts, fo- | | | | | | | | | | | | | | | |
| cusing on improving knowledge probing (Qin and | | | | | | | | | | | | | | | |
| Eisner, 2021; Zhong et al., 2021). Lester et al. | | | | | | | | | | | | | | | | | | | |
| **Language Model Prompting.** | | | | | GPT-3 (Brown | | | | (2021) showed that with large pretrained models, | | | | | | | | | | |
| et al., 2020) uses in-context examples (Liu et al., | | | | | | | | | only tuning continuous prompts with a frozen lan- | | | | | | | | | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Journal Pre-proof** |  |  |  |  |  |
| ID | Prompt Patterns of P-tuning | Seg. | Pos. | #[P] | Acc. | F1. | Avg. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 2 3 4 5 6 7 8 | 1 | [Premise] Question: [Hypothesis] [P] ? Answer: [M]. | | | | | | | | | | | Yes | Mid | | 1 | 87.95 | 76.70 | 82.33 |
| 2 | [Premise] Question [P]: [Hypothesis] ? Answer: [M]. | | | | | | | | | | | Yes | Mid | | 1 | 88.39 | 78.57 | 83.48 |
| 3 | [Premise] Question: [Hypothesis] ? [P] Answer: [M]. | | | | | | | | | | | No | Mid | | 1 | 89.29 | 79.86 | 84.58 |
| 4 | [Premise] [P] Question: [Hypothesis] ? Answer: [M]. | | | | | | | | | | | No | Miid | | 1 | 89.73 | 82.15 | 85.94 |
| 5 | [Premise] Question: [Hypothesis] ? Answer: [M]. [P] | | | | | | | | | | | No | Edge | | 1 | 87.50 | 83.39 | 85.45 |
| 6 | [Premise] Question: [Hypothesis] ? [P][P] Answer: [M]. | | | | | | | | | | | No | Mid | | 2 | 88.39 | 84.74 | 86.57 |
| 7 | [Premise] Question: [Hypothesis] ? [P][P][P][P] Answer: [M]. | | | | | | | | | | | No | Mid | | 4 | 88.39 | 85.14 | 86.76 |
| 8 | [Premise] Question: [Hypothesis] ? [P][P][P][P][P][P][P][P] Answer: [M]. | | | | | | | | | | | No | Mid | | 8 | 83.48 | 73.32 | 78.40 |
| 9 10 11 12 13 14 | Table 7: The few-shot performance of P-tuning on the CB task on ALBERT with different prompt patterns. “Seg.”means whether the inserted prompt tokens segment complete sentences. “Pos.” indicates inserting the prompt tokens at the edge or in the middle of the inputs. “[P]” is continuous prompt token. “[M]” is the mask token. | | | | | | | | | | | | | | | | | | |
| Task | | | LSTM | | MLP | | EMB | | prompt encoder. | | | | | | | | | |
| 15 |
| 16 17 | WiC-ACC | | | 56.27 *±*1.54 | | 55.25 *±*3.09 | | 53.96 *±*3.23 | | **Knowledge** | | | **in** | **Language** | | | **Models.** | | Self- |
| 18 19 | CB-ACC. | | | 81.70 *±*7.49 77.41 *±*9.15 | | 88.39 *±*3.72 84.24*±*5.15 | | 82.59*±*3.69 67.27*±*6.78 | |
| supervised (Liu et al., 2020) pre-trained language | | | | | | | | | |
| CB-F1. | | |
| models (Han et al., 2021) including GPT (Rad- | | | | | | | | | |
| 20 21 22 23 24 25 26 27 28 29 30 31 | BoolQ-ACC. | | | 75.41*±*3.09 | | 76.46*±*2.84 | | 76.87*±*1.69 | |
| ford et al., 2019), BERT (Devlin et al., 2018), XL- | | | | | | | | | |
| Table 8: The few-shot performance on WiC, CB and | | | | | | | | | Net (Yang et al., 2019), RoBERTa (Liu et al., 2019) | | | | | | | | | |
| BoolQ tasks with ALBERT using different prompt en-coders. Results show that both LSTM and MLP gener-ally work well on these tasks, while EMB is unstable and can substantially under-perform the other two on some tasks (e.g,. WiC and CB). “EMB” means using an identity mapping for the prompt encoder. | | | | | | | | | | have been observed to learn not only contextual-ized text representations but also linguistic and | | | | | | | | |
| world knowledge. | | | | (Hewitt and Manning, 2019) | | | | |
| demonstrates that contextualized representations produced by language models can form a parse tree in the embedding space. (Vig, 2019; Clark et al., | | | | | | | | |
| Task | | | | LM-BFF (Auto) | | P-Tuning | | 2019b) look into the multi-head attention patterns | | | | | | | | | | |
| within transformers and discover that certain atten- | | | | | | | | | | |
| 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 | SST-2 | | | | 92.89 | | 92.78 | |
| tion heads may correspond to some grammatical | | | | | | | | | | |
| MNLI | | | | 57.53 | | 58.70 | |
| functions, including co-reference and noun modi- | | | | | | | | | | |
| MRPC | | | | 68.26 | | 69.49 | |
| fiers. LAMA (Petroni et al., 2019, 2020) propose | | | | | | | | | | |
| Table 9: | | Few-shot performance of automatically | | | | | | | | the LAMA task that leverages cloze tests to pre-dict the fact triples of knowledge bases to examine | | | | | | | | |
| searched prompts and P-Tuning. We evaluated LM- | | | | | | | | | |
| BFF (Auto) using the reported top-3 searched patterns | | | | | | | | | language model’s ability of memorizing facts with | | | | | | | | | |
| under our evaluation procedure. P-Tuning also uses the same discrete prompts, in concatenation with con-tinuous prompts. Results show that P-Tuning can be effectively combined with existing discrete patterns and achieve further performance improvement. | | | | | | | | | | answers in the single-token format. In (Wang et al., 2020), the authors investigate the attention matrices to find evidence about knowledge triples contained | | | | | | | | |
| in the context. | | | (Jiang et al., 2020a) develops a | | | | | |
| multi-token fact retrieval dataset based on LAMA. | | | | | | | | |
| guage model achieves comparable performance to | | | | | | | | | | | | | | | | | | |
| full-model tuning. | | | | | | | | | **5** | | **Conclusions** | | | | | | | |
| Compared to these concurrent works on NLU, | | | | | | | | |
| P-Tuning reaches a unique conclusion that contin- | | | | | | | | | | | | | | | | | | |
| uous prompts improve performance and stabilize | | | | | | | | | In this paper, we present a method P-Tuning that | | | | | | | | | |
| training with either frozen or tuned models under | | | | | | | | | uses continuous prompts in concatenation with dis- | | | | | | | | | |
| both the few-shot and fully-supervised settings. For | | | | | | | | | crete prompts. P-Tuning improves performance | | | | | | | | | |
| example, no concurrent works have shown that | | | | | | | | | and stabilizes training for pretrained language | | | | | | | | | |
| continuous prompts can improve performance with | | | | | | | | | model adaptation. P-Tuning is effective with both | | | | | | | | | |
| a tuned language model. Technically, P-Tuning | | | | | | | | | tuned and frozen language models under both the | | | | | | | | | |
| also has a few unique designs such as using hy- | | | | | | | | | | few-shot and fully-supervised setings. | | | | | | | | |
| brid continuous-discrete prompts and employing a | | | | | | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Journal Pre-proof** | | | | | | | |
| 1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65 | **References**  Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda | | Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020b. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438. | | | | | |
| Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*. | | Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking | | | | | |
| Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019a. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chap-ter of the Association for Computational Linguistics:* | | | | beyond the surface: A challenge set for reading com-prehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chap-ter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Pa-pers)*, pages 252–262. | | | |
| *Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936. | | Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt | | | | | |
| Kevin Clark, Urvashi Khandelwal, Omer Levy, and | | tuning. *ArXiv*, abs/2104.08691. | | | | | |
| Christopher D Manning. 2019b. What does bert look at? an analysis of bert’s attention. *arXiv preprint arXiv:1906.04341*. | | Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thir-teenth International Conference on the Principles of* | | | | | |
| Ido Dagan, Oren Glickman, and Bernardo Magnini. | | | | *Knowledge Representation and Reasoning*. Citeseer. | | | |
| 2005. The pascal recognising textual entailment chal-lenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.  Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Car- | | | Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*. | | | | |
| bonell, Quoc V Le, and Ruslan Salakhutdinov. | | | Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, | | | | |
| 2019. | Transformer-xl: Attentive language mod- | | | Lawrence Carin, and Weizhu Chen. 2021. | | | What |
| els beyond a fixed-length context. *arXiv preprint* | | makes good in-context examples for gpt-3? *arXiv* | | | | | |
| *arXiv:1901.02860*. | | *preprint arXiv:2101.06804*. | | | | | |
| Marie-Catherine De Marneffe, Mandy Simons, and Ju-dith Tonhauser. 2019. The commitmentbank: Inves-tigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23, pages 107–124. | | | 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). | | | | |
| Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understand-ing. *arXiv preprint arXiv:1810.04805*. | | | Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Man-dar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining ap-proach. *arXiv preprint arXiv:1907.11692*. | | | | |
| Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*. | | | Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Over- | | | | |
| Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Liang Zhang, Wentao | | coming few-shot prompt order sensitivity. *CoRR*, abs/2104.08786. | | | | | |
| Han, Minlie Huang, et al. 2021. Pre-trained models: Past, present and future. *AI Open*. | | Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H Miller, and | | | | | |
| John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representa-tions. In *North American Chapter of the Association* | | | | Sebastian Riedel. 2020. How context affects lan-guage models’ factual predictions. *arXiv preprint arXiv:2005.04611*. | | | |
| *for Computational Linguistics: Human Language Technologies (NAACL)*. Association for Computa-tional Linguistics.  Zhengbao Jiang, Antonios Anastasopoulos, Jun Araki, Haibo Ding, and Graham Neubig. 2020a. X-factr: | | | Fabio Petroni, Tim Rocktäschel, Patrick Lewis, An-ton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowl-edge bases? *arXiv preprint arXiv:1909.01066*. | | | | |
| Multilingual factual knowledge retrieval from pre- | | | Mohammad Taher Pilehvar and José Camacho-Collados. | | | | |
| trained language models. In *Proceedings of the 2020* | | 2018. | | | Wic: | 10, 000 example pairs for eval- | |
| *Conference on Empirical Methods in Natural Lan-* | | | | uating context-sensitive representations. | | | *CoRR*, |
| *guage Processing (EMNLP)*, pages 5943–5959. | | abs/1808.09121. | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Journal Pre-proof** | | | | | | | |
| 1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65 | Guanghui Qin and J. Eisner. 2021. Learning how to ask: | | | | | Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng | | |
| Querying lms with mixtures of soft prompts. *ArXiv*, | | | | | Gao, Kevin Duh, and Benjamin Van Durme. 2018. | | |
| abs/2104.06599. | | | | | Record: Bridging the gap between human and ma- | | |
| chine commonsense reading comprehension. *arXiv* | | | | | | | |
| 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. | | | | | *preprint arXiv:1810.12885*.  Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q. Wein-berger, and Yoav Artzi. 2020. Revisiting few-sample BERT fine-tuning. *CoRR*, abs/2006.05987. | | |
| 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 trans-former. *arXiv preprint arXiv:1910.10683*. | | | | | | Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and | |
| Sameer Singh. 2021a. | Calibrate before use: Im- |
| proving few-shot performance of language models. *CoRR*, abs/2102.09690. | |
| Melissa Roemmele, Cosmin Adrian Bejan, and An- | | | | | | Tony Z Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021b. Calibrate before use: Improv-ing few-shot performance of language models. *arXiv preprint arXiv:2102.09690*. | |
| drew S Gordon. 2011. | | Choice of plausible alter- | | | |
| natives: An evaluation of commonsense causal rea-soning. In *AAAI Spring Symposium: Logical Formal-* | | | | | |
| *izations of Commonsense Reasoning*, pages 90–95. | | | | | Yanan Zheng, Jing Zhou, Yujie Qian, Ming Ding, Jian | | |
| Timo Schick and Hinrich Schütze. 2020. It’s not just size that matters: Small language models are also few-shot learners. *Computing Research Repository*, arXiv:2009.07118. | | | | | Li, Ruslan Salakhutdinov, Jie Tang, Sebastian Ruder, and Zhilin Yang. 2021. Fewnlu: Benchmarking state-of-the-art methods for few-shot natural language un-derstanding. | | |
| Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with | | | | | Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual probing is [mask]: Learning vs. learning to recall. *ArXiv*, abs/2104.05240. | | |
| automatically generated prompts. | | | *arXiv preprint* | | | | |
| *arXiv:2010.15980*. | | | | | | | |
| Jesse Vig. 2019. | A multiscale visualization of at- | | | | | | |
| tention in the transformer model. | | | *arXiv preprint* | | | | |
| *arXiv:1906.05714*.  Alex Wang, Yada Pruksachatkun, Nikita Nangia, Aman- preet Singh, Julian Michael, Felix Hill, Omer Levy,  and Samuel R Bowman. 2019a. Superglue: A stickier  benchmark for general-purpose language understand- ing systems. *arXiv preprint arXiv:1905.00537*.  Alex Wang, Yada Pruksachatkun, Nikita Nangia, Aman- preet Singh, Julian Michael, Felix Hill, Omer Levy,  and Samuel R. Bowman. 2019b. SuperGLUE: A  Stickier Benchmark for General-Purpose Language  Understanding Systems. In *NeurIPS 2019*, pages  3261–3275.  Alex Wang, Amanpreet Singh, Julian Michael, Felix  Hill, Omer Levy, and Samuel R. Bowman. 2018.  Glue: A multi-task benchmark and analysis plat- | | | | | | | |
| form for natural language understanding. | | | | *ArXiv*, | | | |
| abs/1804.07461.  Chenguang Wang, Xiao Liu, and Dawn Song. 2020.  Language models are open knowledge graphs. *arXiv*  *preprint arXiv:2010.11967*.  Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Car- bonell, Ruslan Salakhutdinov, and Quoc V Le.  2019. Xlnet: Generalized autoregressive pretrain- | | | | | | | |
| ing for language understanding. | | | *arXiv preprint* | | | | |
| *arXiv:1906.08237*. | | | | | | | |

|  |
| --- |
| **Journal Pre-proof** |

**Declaratio if ioterettt**

☒The authors declare that they have no known competng fnancial interests or personal relatonships

that could have appeared to infuence the work reported in this paper.

☐The authors declare the following fnancial interestsppersonal relatonships which may be considered

as potental competng interests:

|  |
| --- |
|  |