Evaluating and Comparing Soft-prompt Tuning Methods

Haoming(Hammond) Liu
NYU Shanghai
hl3797@nyu.edu

Xiaochen(Nigel) Lu
NYU Shanghai
xl3139@nyu.edu

Wenbin(Jim) Qi
NYU Shanghai
wq372@nyu.edu

1 Motivation & Objective

Pre-trained language models have achieved great success in processing natural languages, and there have been numerous works exploring how to adapt these general-purpose models to downstream tasks effectively and efficiently. **Fine-tuning**, though found effective by some dominent works like GPT (Radford et al., 2018) and BERT (Devlin et al., 2018), needs to store a model copy for each downstream task and update all the parameters for adaption, which is not efficient in terms of storage and computational resources.

To alleviate this issue, Brown et al. proposes to condition the behavior of a frozen GPT-3 model via hand-crafted descriptions and examples, termed **prompts**. This results in a single generalist model serving for various downstream tasks simultaneously. However, such a design is still inevitably error-prone and requires human involvement.

Accordingly, some recent works adopt learnable **soft-prompts** to further facilitate and automate the prompt design procedure (Lester et al., 2021; Liu et al., 2021a,b; Li and Liang, 2021). Among these works, however, limited comparisons have been made against one another since they are generally using different pre-trained models and targeting various tasks. Hence, this project aims to build a unified framework that fairly evaluates and compares different soft-prompt tuning methods. On basis of that, we can categorize previous works, re-evaluate their performance under the same settings, and conclude some effective strategies for soft-prompt tuning. Due to time concerns, this project will primarily evaluate and compare their performance on NLU tasks.

2 Project Timeline

The project will be split into four stages: literature review, cosebase setup, methods integration, comparison & write-up. The following sub-sections discuss the major tasks and products of each stage.

2.1 Literature Review (By Apr. 6th)

This stage gathers and selects soft-prompt tuning methods. We'll start from the works suggested by Jason Phang (Lester et al., 2021; Liu et al., 2021a,b; Li and Liang, 2021) and skim through the works they cite or cite them. To ensure the consistency and credibility, the selected works are ideally:

1. Targeting general NLU tasks (e.g., sentence- or sentence-pair classification, Q&A, etc.);
2. Evaluated on NLU benchmarks (e.g. GLUE, SuperGLUE, etc.) (Wang et al., 2018, 2019);
3. Published on top NLP conferences; or preprint paper with a considerable amount of citations;
4. With public available implementation that can be used under the license conditions.

Notably, works that are not targeting NLU tasks may still be included in this project if the proposed method is novel or effective enough.[1] The choices of methods (to be evaluated and compared) will be listed in the partial draft.

2.2 Codebase Setup (By Apr. 13th)

The codebase are expected to include baseline models with unified and hand-crafted prompts (i.e., not applying soft-prompt tuning), under a same set of pre-trained language models, hyper-parameter search configurations, and evaluation benchmark.

Pre-trained language models. The major concerns here would be the variety and scale of the pre-trained models. For this project, we plan to start with T5-(Small/Base/Large) to perform the

[1] This may not be a fair comparison but undoubtedly fits the objective of this project. Potential biases will be disclosed in the analysis section of the final paper.

evaluation at different scales (Raffel et al., 2020). Also, to alleviate the variety concern, we may further expand to RoBERTa and DeBERTa models (of similar scales) if time permits (Liu et al., 2019; He et al., 2021).

Hyper-parameter Search Configurations. In the unified framework, the searching configurations will be a union of the hyper-parameters from all the methods, and the configurations of range or values will be rigorously determined based on the original settings. Taking the work of Lester et al. as an example, the searching configurations include: learning rate, batch size, number of epochs/steps, warmup epochs/steps, decay factor, epochs/steps per decay, prompt length, and so on.

Evaluation Benchmark. As far as the literature review goes, the evaluation of most soft-prompt tuning works (targeting NLU tasks) overlaps on the SuperGLUE benchmark (Wang et al., 2019). More details about the SuperGLUE benchmark are discussed in Section 3. By default, we'll adopt the SuperGLUE benchmark for evaluation and comparison, unless a better alternative appears in the literature review stage.

The link to our Github codebase will be included in the partial draft.

2.3 Methods Integration (By Apr. 20th)

This stage mainly focuses on integrating the implementation of the soft-prompt tuning methods. As we've unifed the tuning settings, this stage may not take too long, considering that the implementations of the methods are publicly available as well. The integrated implementations will be available on our Github codebase.

2.4 Comparison & Write-up (By May 13th)

After collecting the results under different configurations, we can analyze the results and conclude some effective strategies for soft-prompt tuning. The statistics and findings will be included in the final paper. Note that more time is reserved for this stage to expand pre-trained model choices and accommodate unexpected circumstances.

Note: This may sound like a crazy plan, but we still choose this topic because it's an academically interesting question, and we are confident with our knowledge, experience, and coding skills.

3 Data & Tools

SuperGLUE. SuperGLUE is a benchmark that targets at various NLU tasks. Compared to its predecessor (i.e., GLUE benchmark), SuperGLUE incorporates more challenging, diverse tasks and comprehensive human baselines (Wang et al., 2018, 2019). SuperGLUE also offers a suite of tools for training, evaluating, and analyzing NLU systems.

Hugging Face. We will use the pre-trained language models (e.g., T5, RoBERTa, DeBERTa) through Hugging Face packages (Raffel et al., 2020; Liu et al., 2019; He et al., 2021).

GitHub. The codebase will be maintained as a public repository on GitHub.

High Performance Computer. All the experiments will be conducted on NYU Greene and GCP.

Collaboration Statement. All group members participated and contributed equally at this stage.

Acknowledgements. Special thanks to Jason Phang for the project idea and the start up papers. Moreover, Jason will continuously follow up and advise on this project.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. 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. *CoRR*, abs/2104.08691.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *CoRR*, abs/2101.00190.

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

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. GPT understands, too. *CoRR*, abs/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.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

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. *CoRR*, abs/1905.00537.

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. *CoRR*, abs/1804.07461.