Complex Probes are Favored: A Revisit of Probe Complexity

Yilei Tu[1], **Jiaoda Li**[1], **Ryan Cotterell**[1]
[1] Department of Computer Science, ETH Zurich
yiletu@ethz.ch, {jiaoda.li, ryan.cotterell}@inf.ethz.ch

Abstract

Probes are widely used to discern linguistic information embedded in pretrained representations. A common preference is towards simple probes, arguing that complex probes, endowed with high learning capabilities, blur the distinction whether the high performance is due to inherent linguistic knowledge in the representations or the probe's learning on the task. This argument builds upon an implicit assumption: a probe with higher complexity can learn any task better. We investigate the performance of probes with varying complexities on three different types of representation in a practical setting with a gradient-descent based optimization algorithm and a training set of fixed size. Contrary to a common belief, we find that more complex probes perform worse, especially when linguistic information is not present in the representations. Yet on pretrained representations that are believed to contain linguistic information, the performance of probes exhibits robustness, much less affected by complexity. Our results challenge the common wisdom of favoring low-complexity probes.[1]

1 Introduction

Transformer-based (Vaswani et al., 2017) pretrained language models (PLMs), such as series of GPT (Radford et al., 2018, 2019; Brown et al., 2020), have become a pillar in the realm of natural language processing (NLP) and exhibited commendable performance across a myriad of NLP tasks (Rajpurkar et al., 2016; Wang et al., 2018, 2019). Consequently, understanding the internal mechanisms of PLMs has emerged as a focal area of research, with a proliferation of literature in recent years dedicated to exploring this subject (Alishahi et al., 2019; Gubelmann and Handschuh, 2022; Cui et al., 2022; Wu et al., 2023). One promising tool for understanding and interpreting

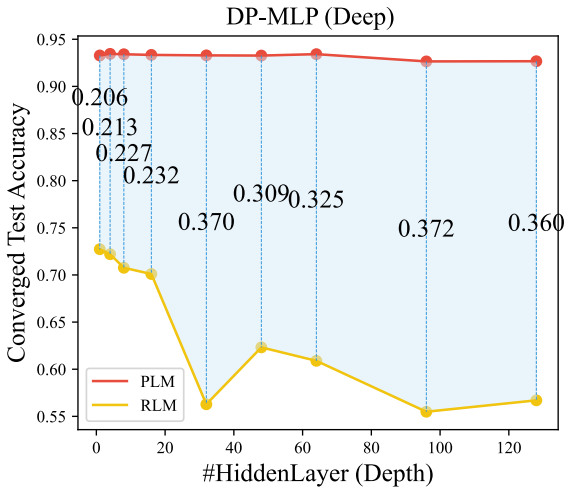

Figure 1: The probing accuracy of MLP on PLM remain stable regardless of the increase in complexity, while that of MLP on RLM exhibit a declining trend.

PLMs in NLP is *probing*, which trains a supervised classifier atop frozen PLMs on linguistic tasks, intended to unveil linguistic knowledge embedded in the model's representations (Adi et al., 2017; Conneau et al., 2018; inter alia).

Probes of different architecture have been proposed. As a linear prediction head is always employed at the end of neural networks, Alain and Bengio (2017) attempts to uncover at which layer information emerges that is readily useful for a linear classifier. Subsequently, more complex models are used to probe information that is not necessarily linearly separable. Meanwhile, there are concerns regarding the use of complex probes raised by Hewitt and Liang (2019); Maudslay et al. (2020), who argue for the use of simple probes since an adequately expressive probe can learn any task. Pimentel et al. (2020) introduces Pareto probing to strike a balance between the complexity and accuracy of probes. Cao et al. (2021); Li et al. (2022) propose to use techniques such as pruning and

[1]Our code is available here: https://github.com/yileitu/complex_probe

prompting to create more *selective* probes, which by definition are probes that are expressive enough to extract information from the model representations while learning less information on their own.

An implicit *assumption* made by most previous works is *the more complex a probe is, the better it would learn any given task*. This is apparently true, *if* the representation is lossless, and the model is properly optimized (Hewitt and Liang, 2019). However, the representations of a probed model could very well be lossy, and in practice, the optimization of a model depends on many factors — the optimization algorithms, hyperparameters, amount of training data, to name a few.

In this paper, we examine this assumption under a realistic condition, where a heuristic optimization method (Adam; Kingma and Ba, 2015) is used, and the size of the training set is fixed and does not scale with complexity. We train a sequence of probes of varying complexities on a randomly-initialized language model (RLM), and discover that counter-intuitively, the probing performance does not improve as the complexity of the probe increases — an almost opposite trend is observed: A 2-layer multilayer perceptron (MLP; Haykin, 1994) achieves an accuracy of 72.73% on a probing task (named entity labeling, NEL), whereas an MLP with 128 hidden layers yields merely 56.70% (Fig. 1 and Tab. 1). On the other hand, probes on a pretrained language model (PLM) demonstrate remarkable *robustness* — they consistently achieve a high accuracy of around 93% regardless of the probe's complexity. Our results imply that when useful information is non-existent in a model (RLM), with higher complexity comes greater training difficulty; but when linguistic information is present (PLM), the difficulty can be easily overcome. As a consequence, following the rationale that a probe should learn less itself (Hewitt and Liang, 2019; Maudslay et al., 2020), an over-parametrized probe should be favored.

2 Method

2.1 Edge Probing

The Edge Probing framework (Tenney et al., 2019a,b) transforms a number of structured linguistic task (OntoNotes 5.0; Weischedel et al., 2013)[2] into multi-class classification tasks of a unified format. Under this framework, a probe is trained to predict the labels of spans of interest. We primarily conduct experiments on named entity labeling (NEL), which is solely in English.

Given a language model (LM) with L layers, and an input sentence with n tokens, we denote the internal representations as $\mathbf{H}^{(0)}, \cdots, \mathbf{H}^{(L)}$ where $\mathbf{H}^{(\ell)} = \left[\mathbf{h}_0^{(\ell)}, \cdots, \mathbf{h}_n^{(\ell)}\right], \forall \ell \in [0, \cdots, L]$. Scalar-mixed representations $\mathbf{H}^{\text{mix}} = \left[\mathbf{h}_0^{\text{mix}}, \cdots, \mathbf{h}_n^{\text{mix}}\right]$ are computed as the weighted average of layer representations:

$$\mathbf{h}_i^{\text{mix}} = \sum_{\ell=1}^{L} w^{(\ell)} \cdot \mathbf{h}_i^{(\ell)} \tag{1}$$

where weights $w^{(\ell)}$ are learned during training. A contextual representation of a span is then obtained by pooling the per-token representations of each token within the span (Lee et al., 2017), which is fed into the probes as input.

2.2 Diagnostic Probe

The main object of study is diagnostic probing (DP), as termed in Lasri et al. (2022); Li et al. (2022), which is the most popular method of probing (Hupkes and Zuidema, 2018; Hewitt and Liang, 2019; Belinkov, 2022). It trains a classifier atop frozen representations of a PLM to perform a task related to a certain linguistic property. The prediction accuracy is then used as an indication of whether information about the linguistic property of interest is embedded within the representations.

We choose MLP as the architecture for probes as it is amenable to complexity alterations — We control its complexity by varying their number of hidden layers (depths) the number of neurons in each layer (width). An MLP is trained on top of the contextual representations to perform the probing task. A non-linearity ($\tanh$) and a layer normalization (LN; Ba et al., 2016) are inserted between layers.

We compare probes of different complexities. Note that the complexity here roughly corresponds to the number of parameters in a network, but not necessarily to its expressivity (depending on the definition of expressivity). Theoretically, a two-layer MLP is a universal approximator (Cybenko, 1989; Hornik et al., 1989), so increasing its depth does not increase its approximation power.

We also conduct experiments with diagnostic probes based on Multinomial Logistic Regression (LR) and probing via prompting (PP; Li et al.,

[2]LDC liscence.

#HiddenLayer	**Majority**	**MLP**		
		PLM	RLM	OH
1		0.9328 (10)	0.7273 (80)	0.8877 (1)
4		0.9345 (10)	0.7220 (141)	0.8819 (1)
8		0.9341 (10)	0.7076 (88)	0.8808 (1)
16		0.9334 (10)	0.7010 (145)	0.8825 (3)
32	0.1591	0.9329 (10)	0.6756 (66)	0.8851 (3)
48		0.9327 (10)	0.6232 (125)	0.8761 (4)
64		0.9342 (10)	0.6089 (102)	0.8880 (14)
96		0.9265 (10)	0.5549 (83)	0.8674 (13)
128		0.9267 (10)	0.5670 (100)	0.8674 (15)

Table 1: Test accuracy of MLP probes on PLM, RLM and OH with varying number of hidden layers. Width is fixed at 512. Integers within parentheses indicate the number of training epochs. For PLM, the accuracy maintains at a high value (around 93%) regardless of depths, whereas for RLM, accuracy tends to decrease as the complexity increases. For OH, MLP probes consistently yields an accuracy of 88%.

2022). LR can only capture information that is linearly separable and PP is computationally expensive to train. Nevertheless, similar patterns are observed in our preliminary experiments. We refer interested readers to Apps. A.3 and A.4 for more details.

3 Experiment Setup

3.1 Representations

In this work, we investigate GPT-2 (Radford et al., 2019) with 124M parameters. Three representations are probed:

PLM The pretrained weights are loaded from Wolf et al. (2020). The existence of NEL information in it has been verified by previous works, e.g. Li et al. (2022).

RLM The weights are randomly-reset (see App. A.1 for details). It serves as a baseline to gauge a probe's ability to learn a probing task from random representations.

One-Hot (OH) One-hot representations retain the identities of different tokens losslessly. However, it lacks the contextual information, so a probe on it can at best learn a mapping from spans to named entities, regardless of a span's context in the input sentence.

3.2 Probes

We vary the complexity of MLP probes by changing their depths, where the number of hidden layers increased while keeping the dimension of each hidden layer (width) constant. We also conduct experiments on varying widths (see App. A.5).

In addition to the above-mentioned diagnostic probes, we also introduce a **majority** baseline that always predicts the most frequent category.

3.3 Training Details

The dataset is split into three partitions: `train` 128,738 data points, `dev` 20,354 data points, and `test` 12,586 data points. We train probes on RLM for 256 epochs on `val` and select learning rate η that yielded the highest accuracy. All the other hyperparameters are set to their default values in `HuggingFace`'s `transformers` (Wolf et al., 2020) or `PyTorch` (Paszke et al., 2019) libraries, as detailed in App. A.2.

On RLM and OH, we train the probes with optimal η for 256 epochs, early stopped (Girosi et al., 1995; Prechelt, 2012) with a *patience* of 10. Accuracy on `test` of the model with the highest development accuracy is then reported.

On PLM, all probes are trained for only 10 epochs, since their performances are insensitive to number of training epochs and hardly improve beyond 10 epochs (see §4.2).

The majority class is decided based on `val` and the proportion it takes up in `test` is reported.

4 Results and Analyses

The test accuracy of probes of different complexities on PLM, RLM and OH are presented in Tab. 1. The number in parentheses indicates the required training epochs.

4.1 RLM

We observe that the probing accuracy of MLP on RLM displays a decreasing tendency as the complexity increases. When depth is less than or equal to 16, the accuracy remains above 70%; however, when it exceeds 16, the accuracy drops to and stays at around 56%. This is contrary to the common belief that a more complex model should learn any task on its own. In fact, it is the primary argument for the use of simple probes (Hewitt and Liang, 2019; Maudslay et al., 2020). We conjecture that the reason for this rather peculiar behavior is that when the depth increases, an increased training difficulty is introduced. A model with higher complexity typically requires larger amount of training data, a more sophisticated optimization algorithm, and a more nuanced selection of hyperparameters, which are not always met in practice.

The training time required for probes on RLM is relatively long, ranging from 60 to 150 epochs. The initial learning rate has a greater impact on its performance.

4.2 PLM

On the other hand, on PLM, the accuracy stays at around 93% despite the increasing complexity. Additionally, we find performance of probes on PLM is *robust*, as shown in Fig. 2: (1) the development accuracy converges rapidly (within 10 epochs) to a high level, and (2) its converged accuracy is insensitive to the choice of hyperparameters.

4.3 OH

The probe on OH consistently yields an accuracy of 88%. It is higher than that on RLM, which validates that the great amount of noise in the randomized representations impedes learning. And it is escalated when the complexity of the probe is higher. Furthermore, it still underperforms probes on PLM, which shows the contextual information useful for NEL is indeed encoded in the pretrained representations.

4.4 Summary

It is believed that an ideal probe should be able to extract information encoded in the representations while learning less on its own (Cao et al., 2021; Li et al., 2022), which translates to high accuracy on PLM and low accuracy on RLM. By this criterion, a probe that is more complex than enough is favored over simple probes.

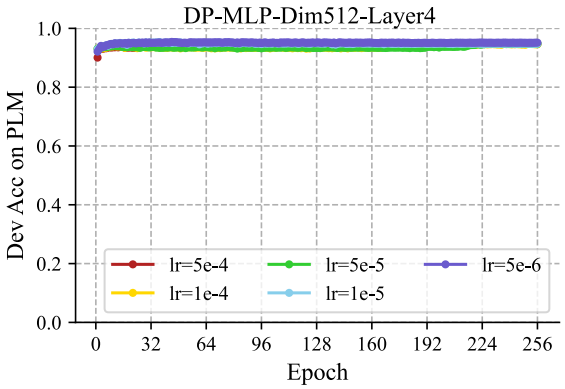

Figure 2: The development accuracy of MLP probes on PLM with varying initial learning rates. Its depth is fixed at 4, and width is fixed at 512. The accuracy swiftly converges to around 94% within a few epochs and is virtually unaffected of different learning rates.

5 Conclusion

In this study, we investigate the probing accuracy of MLP probes across varying complexities. The empirical results unveil several insightful findings and shed light on the selection of probing models.

For RLM, our findings reveal a counter-intuitive trend that increased complexity leads to lower accuracy, implying that the more complex the probe, the poorer it learns the tasks. This challenges the common belief that we bear a higher risk of the probe learning the task by itself when we choose a complex probe. Probing accuracy on RLM is lower than that on OH, substantiating that the noise in randomized representations hampers learning, and it is exacerbated with increasing probe complexity.

For PLM, a high probing accuracy is observed across varying complexities, indicating that pretrained representations can reliably provide linguistic information to the probe. This robustness also manifests in insensitivity to hyperparameter variations. Probes on PLM outperforms those on OH, underscoring the pivotal role of contextual information encoded in pretrained representations.

6 Limitations

We investigate only one language (English) and one task (NEL) due to limited computational resources. Experiments are run on a single RTX3090. Although we try to obtain the best performance in each setting, the possibility that better performances can be achieved through more hyperparameter tuning cannot be ruled out.

Ethical Considerations

We foresee no ethical issues.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *5th International Conference on Learning Representations, ICLR, Conference Track Proceedings*.

Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes. In *5th International Conference on Learning Representations, ICLR, Workshop Track Proceedings*.

Afra Alishahi, Grzegorz Chrupała, and Tal Linzen. 2019. Analyzing and interpreting neural networks for nlp: A report on the first blackboxnlp workshop. *Natural Language Engineering*, 25(4):543–557.

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *Computing Research Repository*, arXiv:1607.06450. Version 1.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

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

Steven Cao, Victor Sanh, and Alexander Rush. 2021. Low-complexity probing via finding subnetworks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 960–966, Online. Association for Computational Linguistics.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Yiming Cui, Wei-Nan Zhang, Wanxiang Che, Ting Liu, Zhigang Chen, and Shijin Wang. 2022. Multilingual multi-aspect explainability analyses on machine reading comprehension models. *iScience*, 25(5):104176.

G. Cybenko. 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, 2(4):303–314.

Federico Girosi, Michael Jones, and Tomaso Poggio. 1995. Regularization theory and neural networks architectures. *Neural Computation*, 7(2):219–269.

Reto Gubelmann and Siegfried Handschuh. 2022. Context matters: A pragmatic study of PLMs' negation understanding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4602–4621, Dublin, Ireland. Association for Computational Linguistics.

Simon Haykin. 1994. *Neural networks: a comprehensive foundation*. Prentice Hall PTR, Hoboken, NJ.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.

K. Hornik, M. Stinchcombe, and H. White. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366.

Dieuwke Hupkes and Willem Zuidema. 2018. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure (extended abstract). In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 5617–5621. International Joint Conferences on Artificial Intelligence Organization.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR, Conference Track Proceedings*.

Prasanth Kolachina, Nicola Cancedda, Marc Dymetman, and Sriram Venkatapathy. 2012. Prediction of learning curves in machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22–30, Jeju Island, Korea. Association for Computational Linguistics.

Karim Lasri, Tiago Pimentel, Alessandro Lenci, Thierry Poibeau, and Ryan Cotterell. 2022. Probing for the usage of grammatical number. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8818–8831, Dublin, Ireland. Association for Computational Linguistics.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Jiaoda Li, Ryan Cotterell, and Mrinmaya Sachan. 2022. Probing via prompting. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1144–1157, Seattle, United States. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR, Conference Track Proceedings*.

Rowan Hall Maudslay, Josef Valvoda, Tiago Pimentel, Adina Williams, and Ryan Cotterell. 2020. A tale of a probe and a parser. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7389–7395, Online. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Tiago Pimentel, Naomi Saphra, Adina Williams, and Ryan Cotterell. 2020. Pareto probing: Trading off accuracy for complexity. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3138–3153, Online. Association for Computational Linguistics.

Lutz Prechelt. 2012. *Early Stopping — But When?*, pages 53–67. Springer Berlin Heidelberg, Berlin, Heidelberg.

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

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019b. What do you learn from context? probing for sentence structure in contextualized word representations. In *7th International Conference on Learning Representations, ICLR, Conference Track Proceedings*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2013. OntoNotes Release 5.0.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame,

Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Weiqi Wu, Chengyue Jiang, Yong Jiang, Pengjun Xie, and Kewei Tu. 2023. Do PLMs know and understand ontological knowledge? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3080–3101, Toronto, Canada. Association for Computational Linguistics.

A Appendix

A.1 Parameter Initialization of RLM

All weights of GPT-2 except for LN (Ba et al., 2016) are randomly generated from a normal distribution with a mean of 0.0 and a standard deviation of 0.02. All biases are set to 0.0. All biases in the Layer Norm are set to 0.0, and weights are set to 1.0. These settings are the default configurations in the source code of `HuggingFace`'s GPT-2 API[3].

A.2 Hyperparameters

In our preliminary experiments, we observed that the warmup and scheduler have a little impact on accuracy. We set the scheduler to linear decay without warmup, tune the initial learning rate, and compare the accuracy of different probes in the case of their respective *optimal* learning rates. All other hyperparameters are default values in `HuggingFace`'s `transformers` (Wolf et al., 2020) or `PyTorch` (Paszke et al., 2019) libraries, for example, weight decay (Loshchilov and Hutter, 2019) and dropout (Srivastava et al., 2014) are set to 0.0.

Given that η is continuous, it is infeasible to exhaustively explore all possible values. Consequently, we adopt series of η of the form 1×10^n and 5×10^n for training. Through a sequence of preliminary experiments, we have progressively narrowed the range of optimal values for each type of probe based on the final validation accuracy:

- DP-MLP (Deep and Wide): $5 \times 10^{-4}, 1 \times 10^{-4}, 5 \times 10^{-5}, 1 \times 10^{-5}, 5 \times 10^{-6}$
- DP-LR: $1 \times 10^{-4}, 1 \times 10^{-5}, 1 \times 10^{-6}$
- PP: $5 \times 10^{-5}, 1 \times 10^{-5}, 5 \times 10^{-6}$

A.3 DP-Logistic Regression

From Tab. 2, we observe that the probing accuracy of LR on RLM also decreases as the complexity increases. As the depth increases from 1 to 48, the accuracy declines from 42.94% to the majority ratio (15.91%), at which it remains when the depth goes up till 128.

However, different from MLP, the robustness of LR does not prevail under all circumstances. As shown in Tab. 2, LR probe on PLM also degrades to majority when the number of layers is 48 or greater. This could possibly be attributed to the lack of tuning in learning rates, or other hyperparameter settings.

Depth	Majority	Logistic Regression	
		PLM	RLM
1		0.9435 (7)	0.4294 (61)
4		0.9464 (20)	0.4282 (36)
8		0.9414 (31)	0.4245 (52)
16		0.8946 (43)	0.3265 (60)
32	0.1591	0.7006 (21)	0.2206 (40)
48		0.1591 (25)	0.1591 (1)
64		0.1591 (25)	0.1591 (30)
96		0.1591 (22)	0.1591 (5)
128		0.1591 (5)	0.1591 (5)

Table 2: Test accuracy of LR probes on PLM and RLM with varying number of hidden layers. Width is fixed at 512. Integers within parentheses indicate the number of training epochs. When depth is less than 8, the accuracy maintains at a high value (around 93%) regardless of depths. When depth is larger than 48, both probes on PLM and RLM degrade to majority of 15.91%. The accuracy of the LR probe on RLMs continuously declines with increasing depth, until it entirely degrades to a majority classifier.

As in Tab. 1 and Tab. 2, the accuracy of MLP probe on RLM is higher than that of LR probe. This aligns with our intuition, as the expressivity of MLP probe is higher than that of the LR probe. Thus, the conclusion that a probe which is more complex than necessary is favored, holds true only within probes of the same class.

A.4 Prompting via Probing

A.4.1 Architecture

Li et al. (2022) proposed the Probing via Prompting (PP) framework, using question-answer-pair reformatted datasets and prefix tuning (Li and Liang, 2021) to instruct the LM answering probing tasks. Prefix Tuning uses continuous word embeddings, which surpasses the limitation that the embeddings of actual words are discrete and finite. Hence, PP possesses higher expressivity, but at the cost of increased complexity because of the bottleneck MLP reparametrization.

Following the nomenclature in Schick and Schütze (2021), in PP framework (Li et al., 2022), the vocabulary Σ is augmented into $\Sigma \cup \{\texttt{SEP}, \texttt{EOS}, \texttt{CLS}[1], \ldots, \texttt{CLS}[|\mathcal{Y}|]\}$ with distin-

[3] https://huggingface.co/transformers/v4.6.0/_modules/transformers/models/gpt2/modeling_gpt2.html

guished symbols $\texttt{CLS}[y]$ for each label $y \in \mathcal{Y}$. The sentence and the span are concatenated into a **pattern** with a task-specific **prefix**, i.e.,

$$\text{pat} = \text{pfx} \circ \texttt{SEP} \circ \mathbf{x} \circ \texttt{SEP} \circ \mathbf{x}_{i:j} \circ \texttt{EOS} \quad (2)$$

where string concatenation is denoted by $\circ$, pat stands for pattern, and pfx for prefix.

When predicting, the class whose verbalizer $\text{vb}\colon \mathcal{Y} \to \{\texttt{CLS}[1], \cdots, \texttt{CLS}[|\mathcal{Y}|]\}$ has the highest next-token probability:

$$\hat{y} = \underset{z \in \mathcal{Y}}{\text{argmax}}\, \mathbb{P}\left[\text{LM}\left(\text{pat}\right)_{:1} = \text{vb}(z) \mid \text{pat}\right] \quad (3)$$

where $\text{LM}\left(\text{pat}\right)_{:1}$ denotes the first token in the LM's response given a pattern as input.

Li and Liang (2021) find that using natural language (discrete) prompt[4] fails for moderately-sized PLMs and Li et al. (2022) replicates that that discrete prefixes perform poorly on GPT-2 (Radford et al., 2019). Prefix tuning (Li and Liang, 2021) (namely continuous prompt) achieves a higher degree of expressiveness, which entails the addition of trainable prefix vectors, denoted as $\mathbf{P^V}$ and $\mathbf{P^K}$. They are inserted at the beginning of the input keys and values of the attention heads, respectively and the prefix length l_{pfx} is customized:

$$\text{head}_i^{\text{pfx}} = \text{Attn}\left(\mathbf{Q}\,\mathbf{W}_i^{\mathbf{Q}}, \left[\mathbf{P}_i^{\mathbf{K}}, \mathbf{K}\,\mathbf{W}_i^{\mathbf{K}}\right], \left[\mathbf{P}_i^{\mathbf{V}}, \mathbf{V}\,\mathbf{W}_i^{\mathbf{V}}\right]\right) \quad (4)$$

Note that in Li and Liang (2021), the prefix vectors are not subject to direct optimization, but rather undergo reparameterization through a bottleneck MLP. Since in their preliminary experiments, they discover that optimizing prefix vectors directly is highly susceptible to initialization, resulting in unstable optimization and a minor decline in performance. Therefore, reparametrization MLP is crucial for the performance of PP. We kindly ask readers to refer to Li and Liang (2021); Li et al. (2022) for further details.

A.4.2 Experiments

We focus on changing the complexity of PP by adjusting the prefix length l_{pfx}. For PPs on RLM, Acc_{dev} has yet to stabilize within 256 epochs, making early stopping inapplicable in this scenario. Due limited computational resources, we opt not to increase the number of epochs further, but instead fit Eq. (5) (Kolachina et al., 2012) on both Acc_{dev} and Acc_{test}. Then we calculate the epoch T_{dev} at which the first derivative $\hat{f}'_{\text{dev}}(\cdot)$ falls below the threshold of 0.0001, and take $\hat{f}'_{\text{test}}\left(T_{\text{dev}}\right)$ as the converged test accuracy, even though we do not actually train up to T_{dev}.

$$\text{Acc} = \hat{f}(t) = a \cdot \exp(-b \cdot t) + c \quad (5)$$

Prefix Length	PLM	RLM
50	0.9587	0.7472 (398)
100	0.9574	0.7263 (740)
200	0.9543	0.5526 (416)

Table 3: Test accuracy of PP probes on PLM and RLM with varying prefix length. Integers within parentheses indicate the number of training epochs. For PLM, the accuracy maintains at a high value (around 95%) regardless of prefix lengths, whereas for RLM, accuracy decreases as the complexity increases.

As illustrated in Tab. 3, the prefix length increases from 50 to 200, with the probing accuracy on PLMs remaining stable at around 95%, while on RLMs it declines from 74.72% to 55.26%. Hence, PP also exhibits characteristics similar to those of DP-MLP, displaying robustness on PLMs, while its performance on RLMs declines with increased complexity. Given that the training difficulty of PP surpasses that of DP-MLP, PP can be regarded as an extremely complex probe.

A.5 DP-MLP (Wide)

In Tab. 4, for PLMs, the MLP probes with varying widths exhibit a similar robustness as those with varying depths, maintaining a probing accuracy of around 93% as the width increases from 64 to 4096. However, on RLMs, the performance of MLP probes with varying widths slightly diverges, as there is no apparent trend of increasing or decreasing accuracy. This still contradicts the common belief that more complex models perform better.

[4] e.g., "Give the NER label of the span words."

HiddenDim	PLM	RLM
128	0.9337	0.6856 (111)
256	0.9364	0.6977 (102)
512	0.9341	0.7076 (88)
768	0.9307	0.7092 (102)
1024	0.9346	0.6979 (57)
2048	0.9320	0.7049 (70)
3072	0.9312	0.6966 (78)
4096	0.9308	0.6807 (86)

Table 4: Test accuracy of MLP probes on PLM and RLM with varying hidden dimension (width). The number of hidden layers is fixed at 8. Integers within parentheses indicate the number of training epochs. The robustness on PLM is again pronounced: the probing accuracy stays at around 93%. On RLM, the accuracy exhibits an initial increase followed by a decline as complexity escalates.