ICML
International Conference
On Machine Learning

Test-Time Learning for Large Language Models

Jinwu Hu*, Zitian Zhang*, Guohao Chen*, Xutao Wen, Chao Shuai,
Wei Luo*, Bin Xiao✉, Yuanqing Li✉, Mingkui Tan✉

South China University of Technology, Pazhou Laboratory, Zhejiang University,
South China Agricultural University, Chongqing University of Posts and Telecommunications,
China Key Laboratory of Big Data and Intelligent Robot, Ministry of Education, China.

Contents

Contents

Background

- LLMs face significant challenges when deployed in real-world environments with dynamic and diverse data distributions.

  - Vertical Domain Shift: This occurs when test data contains domain-specific terminology, such as in medical, legal, or technical fields, which the model was not explicitly trained on, impairing its performance.
  - Distributional Shift in Non-Specific Domains: Variations in user intent and linguistic diversity, including dialects and slang, lead to distributional discrepancies that negatively affect the model's comprehension and response generation.

Background

- Some advanced techniques to adapt trained models to potentially shifted test domains.

Table 1. Characteristics of problem settings for adapting trained models to potentially shifted test domains.

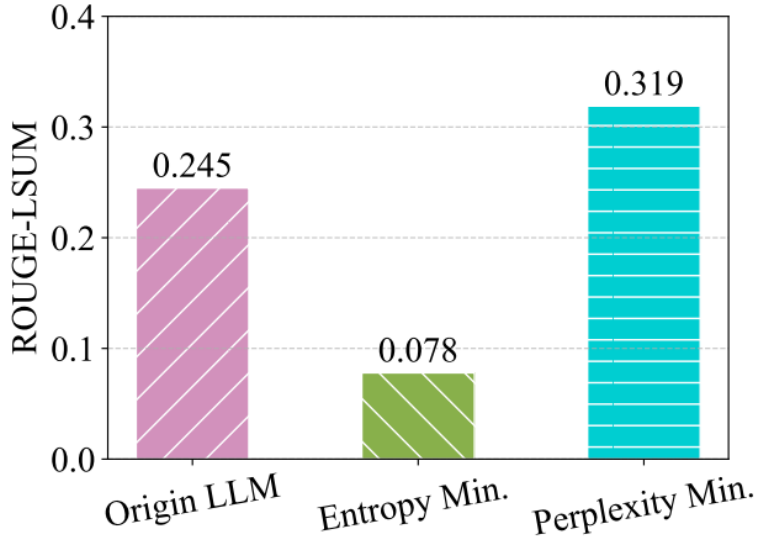
Setting	Knowledge	Source Data	Target Data	Training Loss	Testing Loss	Learning Type
Fine-tuning	✗	✗	x^t, y^t	$\mathcal{L}(x^t, y^t)$	–	Supervised
Retrieval-Augmented Generation (Fan et al., 2024)	✓	✗	x^t	–	–	–
Test-Time Adaptation (Wang et al., 2021)	✗	✗	x^t	✗	$\mathcal{L}(x^t)$	Unsupervised
Test-Time Training (Hardt & Sun, 2024)	✓	x^s, y^s	x^t	✗	$\mathcal{L}(x^t; x^s, y^s)$	–
Test-Time Learning (Ours)	✗	✗	x^t	✗	$\mathcal{L}(x^t)$	Self-supervised

Motivation

- Recent attempts like finetuning, RAG, TTA, and TTT have the following limitations:

  - Difficulty in acquiring labeled data
  - Neglecting autoregressive dependencies
  - High training overhead with catastrophic forgetting.

Dynamically adapts LLMs using only unlabeled test data.



(a) Effectiveness of Entropy and Perplexity Minimization Strategies.

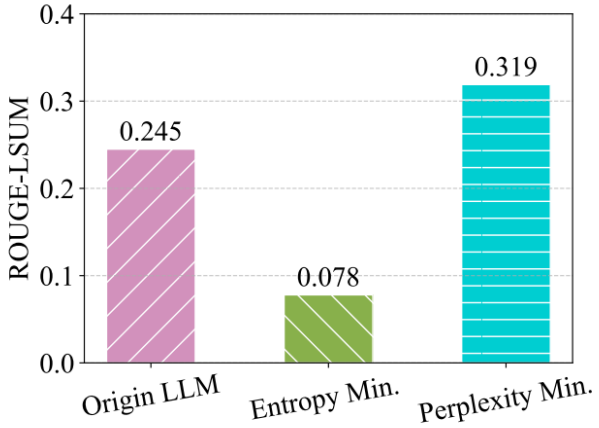
- Perplexity minimization improves the performance of LLMs.

  - A lower perplexity indicates that the model's predictions are more confident and closely align with the true distribution of the data, which implies better model fitting.

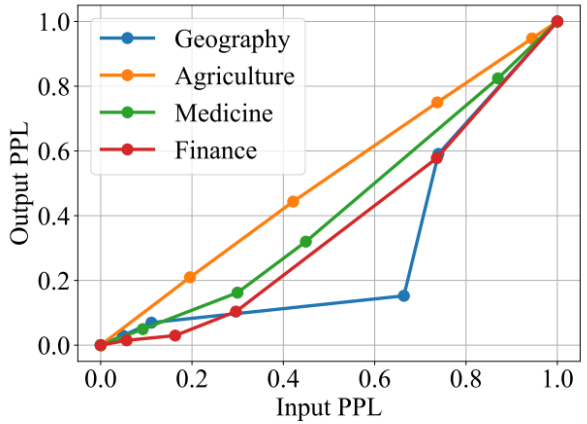
For a given question-answer pair {x,y}, minimizing the perplexity of the model's response y can enhance the model's ability to fit the target data distribution.

Contents

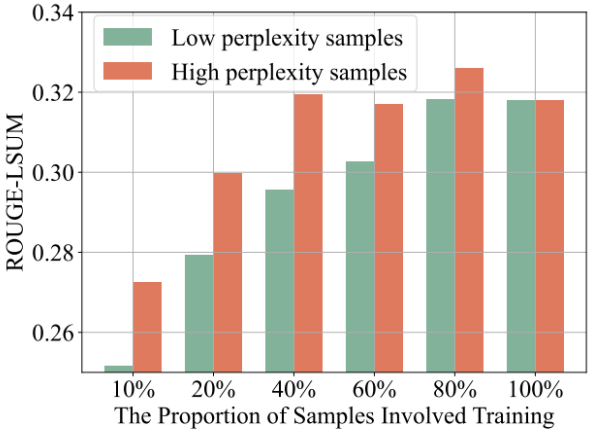
Test-Time Learning for LLMs (TLM)



(a) Effectiveness of Entropy and Perplexity Minimization Strategies.



(b) Trends in perplexity to input and perplexity to output under Llama3.1-8B-Instruct.



(c) Effect of different test samples in test-time perplexity minimization.

Observation 1: Trend of LLM's perplexity to the input $P(x; \Theta)$ and perplexity to the output $P(y|x; \Theta)$ is the same.

- This indicates that reducing output perplexity is possible by minimizing input perplexity in LLMs.

Observation 2: High-perplexity samples contribute more to LLM updates than low-perplexity ones

- training the test samples with high-perplexity makes more contribution than low-perplexity ones
- training on test samples with very low-perplexity may hurt performance.

Test-Time Learning for LLMs (TLM)

Algorithm 1 The pipeline of proposed TLM.

Input: Test samples $\mathcal{D}_{Test} = \{x_j\}_{j=1}^{M}$, the trained LLM $f_\Theta(\cdot)$, LoRA $\Delta\Theta$ with trainable parameters $\mathcal{B}$ and $\mathcal{A}$, batch size B.

Initialize LoRA parameters $\Delta\Theta$.
Add LoRA parameters to trained LLM $\tilde{\Theta} = \Theta + \Delta\Theta$.
for a batch $\mathcal{X} = \{x_b\}_{b=1}^{B}$ in $\mathcal{D}_{Test}$ **do**
    Calculate predictions $\tilde{y}$ for all $x \in \mathcal{X}$ via $f_\Theta(\cdot)$.
    Calculate sample selection score $S(x)$ via Eqn. (6).
    Update LLM ($\tilde{\Theta}$) with Eqn.(5).
end for

Output: The final LLM ($\tilde{\Theta}$).

- *Input Perplexity Minimization Objective*: Inspired by the strong correlation between input perplexity and output perplexity, we adopt input perplexity minimization as the optimization objective.
- *Sample-Efficient Learning Strategy*: Not all test samples equally impact model updates. Employing a perplexity based weighting scheme, the model actively selects and emphasizes high-perplexity test samples for backpropagation, thereby enabling efficient parameter updates during Test-Time Learning.
- *Lightweight Parameter Updates via LoRA*: To mitigate catastrophic forgetting and reduce computational costs, we integrate LoRA into TTL.

A gradient-based theoretical analysis

- Assumption 1 (Autoregressive Property): The LLM generates each token y_t based on the input x and previously generated tokens $y_{1:t-1}$: $\mathcal{P}(y_t|x, y_{1:t-1}; \Theta)$. The standard next-token prediction objective makes model predictions inherently conditional on previous context quality.

- Assumption 2 (Shared Parameter Influence): LLM parameters Θ influence both the input perplexity $\mathcal{P}(x; \Theta)$ and the conditional output perplexity $\mathcal{P}(y|x; \Theta)$. This assumption is valid across various LLM architectures, such as encoder-only and decoder-only models.

Let $\Theta' = \Theta - \eta\nabla_\Theta\big(-\log P(x; \Theta)\big)$ denote the updated parameters after a single TTL step. Using a first-order Taylor expansion:

$$\log P_{\Theta'}(y|x) \approx \mathcal{O}(\eta^2) + \log P_\Theta(y|x) + \eta[\nabla_\Theta \log P(x; \Theta)]^\top \nabla_\Theta \log P_\Theta(y|x),$$

$$\text{Cross-gradient term}$$

where y is the answer to the question x. Our core assumption is that $\langle \nabla_x, \nabla_y \rangle = [\nabla_\Theta \log P(x; \theta)]^\top \nabla_\Theta \log P_\Theta(y|x) \geq 0$ for question-answer pairs with strong semantic alignment. Under this condition, the cross-gradient term becomes non-negative, guaranteeing: $\log P_{\Theta'}(y|x) \geq \log P_\Theta(y|x)$ for small η.

Contents

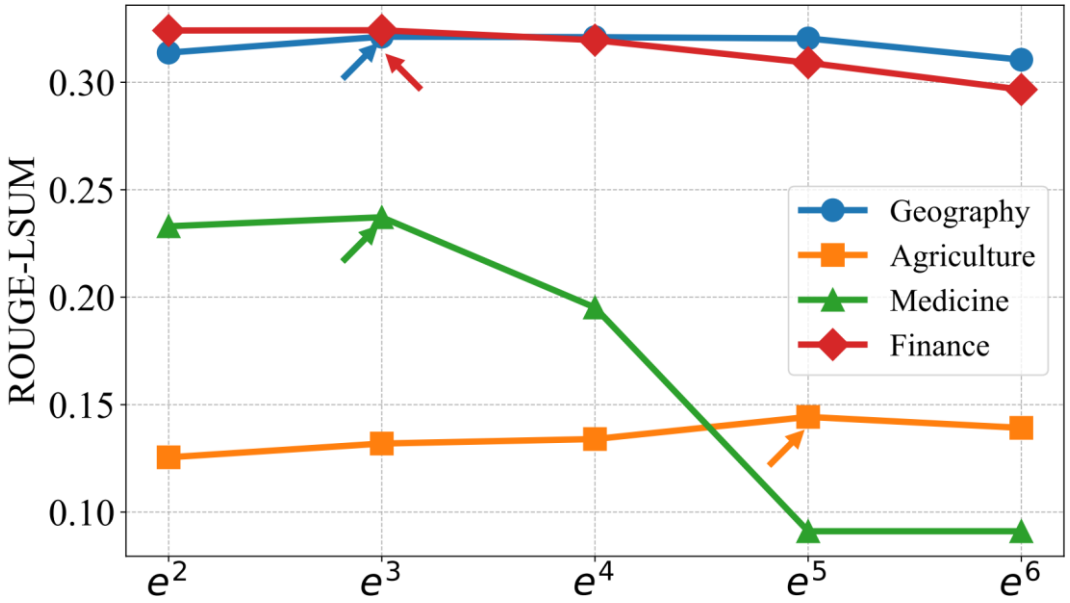
Comparison Experiments

- We compare our proposed TLM, the original LLM, Tent, EATA, and COME to demonstrate the superior performance of our method. We conduct experiments on different types of datasets, including DomainBench, InstructionBench, and ReasoningBench.

Method	DomainBench				InstructionBench		
	Geography	Agriculture	Medicine	Finance	Alpaca-GPT4	Dolly	InstructionWild
Llama3.2-3B-Instruct	0.2395	0.0850	0.1411	0.2229	0.3564	0.3378	0.2562
• Tent	0.1825	0.0150	0.1571	0.1093	0.0336	0.2105	0.0264
• EATA	0.0064	0.0227	0.0259	0.0149	0.1410	0.0090	0.0122
• COME	0.1000	0.1181	0.1542	0.1200	0.0437	0.2186	0.0697
• TLM (Ours)	**0.2893**	**0.1687**	**0.2308**	**0.2953**	**0.3883**	**0.3470**	**0.2824**
Llama3-8B-Instruct	0.2450	0.0834	0.1265	0.2329	0.3752	0.3671	0.2608
• Tent	0.0778	0.0067	0.0105	0.0372	0.2001	0.0036	0.0820
• EATA	0.2081	0.0017	0.0127	0.1257	0.1397	0.1725	0.1088
• COME	0.0048	0.0039	0.0301	0.0328	0.1424	0.0700	0.0240
• TLM (Ours)	**0.3212**	**0.1319**	**0.2372**	**0.3242**	**0.4274**	**0.3785**	**0.2932**
Llama2-13B-chat	0.2182	0.0840	0.1315	0.2382	0.3741	0.2892	0.2781
• Tent	0.0320	0.0196	0.1131	0.0049	0.0955	0.0076	0.1108
• EATA	**0.2800**	0.0771	0.1348	0.1155	0.0811	0.0513	0.1006
• COME	0.1981	0.0380	0.1239	0.0172	0.0806	0.0000	0.0189
• TLM (Ours)	0.2668	**0.1013**	**0.2179**	**0.2760**	**0.3966**	**0.3007**	**0.2865**
Qwen2.5-7B-Instruct	0.2649	0.0981	0.1313	0.2739	0.4439	0.3121	0.2866
• Tent	0.2362	0.1180	0.0524	0.1648	0.2132	0.1946	0.1710
• EATA	0.2109	0.1203	0.1334	0.2846	0.0000	0.2056	0.1710
• COME	0.2306	0.1180	0.0463	0.1780	0.3781	0.2182	0.1710
• TLM (Ours)	**0.3081**	**0.1652**	**0.2394**	**0.3311**	**0.4608**	**0.3177**	**0.3482**

Method	ReasoningBench		
	GSM8K	MetaMath	Logiqa
Llama3.2-3B-Instruct	0.7756	0.7976	0.4194
• Tent	0.7726	0.7412	0.4012
• EATA	0.0032	0.0310	0.0284
• COME	0.7710	0.7308	0.4196
• TLM (Ours)	**0.9096**	**0.8818**	**0.4572**
Llama3-8B-Instruct	0.7610	0.6912	0.4550
• Tent	0.7578	0.6550	0.4378
• EATA	0.0250	0.5454	0.2192
• COME	0.7479	0.6460	0.2180
• TLM (Ours)	**0.8074**	**0.7006**	**0.4868**
Llama2-13B-chat	0.3458	0.2498	0.3992
• Tent	0.2706	0.0040	0.2566
• EATA	0.3392	0.0572	0.2606
• COME	0.3272	**0.2646**	0.2462
• TLM (Ours)	**0.3508**	0.2576	**0.4124**
Qwen2.5-7B-Instruct	0.8378	0.7430	0.5952
• Tent	0.8455	0.7412	0.5934
• EATA	0.7098	0.0070	0.2172
• COME	**0.8556**	0.7559	0.5908
• TLM (Ours)	0.8424	**0.7560**	**0.6046**

Effects of different perplexity margins



$$S(x) = \lambda \cdot e^{\{[\log \mathcal{P}(x;\Theta) - \log \mathcal{P}_0]\}} \cdot \mathbf{I}_{\{\mathcal{P}(x;\Theta) > \mathcal{P}_0\}}(x)$$

To explore the optimal threshold for $\mathcal{P}_0$, we conduct experiments with values of $\mathcal{P}_0$ set to $\{e^2, e^3, e^4, e^5, e^6\}$. When $\mathcal{P}_0 = e^3$, our method achieves the best performance on three datasets, namely Geography, Medicine, and Finance, while also showing near-optimal performance on the Agriculture dataset. Therefore, we select $\mathcal{P}_0 = e^3$ for all experiments. When $\mathcal{P}_0$ is set too high or too low, it negatively affects performance.

Contents

Conclusion

- A novel Test-Time Learning method TLM. Aims to adapt LLMs efficiently using only unlabeled test data, enhancing robustness in target domains.
- Input Perplexity Minimization. Reducing output perplexity can be achieved by minimizing the input perplexity of unlabeled test data.
- Sample-Efficient Learning Strategy. High-perplexity test samples are more informative for model updates than low-perplexity ones.
- Lightweight Adaptation via LoRA. LoRA is more effective than full parameter updates in mitigating catastrophic forgetting.

Code is available at https://github.com/Fhujinwu/TLM

Thank you for your attention!