Vol. 14 (2026): Proceedings of the 2nd International Conference on Artificial Intelligence, Modern Engineering and Environmental Sustainability (IC-AIMEES 2026)
E-ISSN: 3105-0522 P-ISSN：3105-0514
DOI: https://doi.org/10.70267/ ic-aimees.20260226231

Analysis of the Bottlenecks and Solutions for Enhancing the Capabilities of Large Language Models

Yihan Wang*

Sehool of Statistics and Data Science, Lanzhou University of Finance and Economics, Lanzhou 730101, China

**Corresponding author: Yihan Wang.*

Abstract

Large language models, with their ability to understand human language, have become intelligent assistants and efficiency-enhancing tools for people in learning, medical care, entertainment, and more, driving the intelligent development of society. This paper conducts relevant research on the current problems in the improvement of capabilities of large language models from three aspects: data, algorithms, and the models themselves, and reviews them successively from the perspectives of problems and solutions. Through the study of the problems and solutions from the above three perspectives, this paper finds that there are common problems in the process of capability improvement of large language models, such as insufficient data diversity, redundant and low-quality generated text, inability to stably identify their own errors, inability to achieve efficient iterative evolution, model fragility and reduced generalization ability. In the future, the focus of development of large language models can be shifted to improving their accuracy, security, and controllability, and researchers can mainly focus on breaking through their self-thinking, self-correction, and efficient reasoning capabilities. This paper aims to provide researchers in related fields with ideas and theoretical references for model optimization to help large language models break through development bottlenecks.

Keywords

large language models, data, algorithms, bottlenecks and solutions

1. Introduction

Large language models, which are widely studied in current natural language processing, are further extensions based on Transformer. Common methods include pre-training, fine-tuning, retrieval enhancement production (RAG), model quantization, and knowledge distillation. Commonly used models at home and abroad include GPT, Deepseek, Doubao, Qianwen, etc. Because large languages have the ability to understand and reason about text, their application scope is not limited to their own field but is also widely used in other fields. For example, in the medical field, it can be used for intelligent diagnosis and clinical auxiliary diagnosis; In education, it can assist in correcting homework and providing one-on-one targeted tutoring; Stock forecasting and risk assessment in the financial industry; Intelligent customer service and information consulting in the service industry. The improvement of its capabilities is key to ensuring reliability, practicality and security in applications across all fields. Current research on large language

model optimization mainly focuses on basic capabilities and architecture expansion, training inference optimization, semantic data expansion, etc. However, traditional improvement methods rely on manual annotation and external data, are costly, have superficial modeling of human deep learning mechanisms, and have limited performance improvement capabilities when dealing with complex inference tasks.

This paper reviews the core constraints for enhancing the capabilities of large language models, integrates current cutting-edge optimization methods, and can help developers reduce the cost and cycle of model optimization, improve the reliability and practicality of its application in various fields, and has important theoretical and practical value.

2. Core Issues in Enhancing the Capabilities of Large Language Models

The current improvement in the capabilities of large language models is not a single technological advancement or expansion of scope, but rather a greater focus on how to make the models think more human-like and generate more accurate and practical text. The core goal is to reduce human intervention, allowing the model to be optimized based on the data or feedback it generates to make it more similar to human thinking. But the current development faces many problems, mainly focusing on three dimensions: data, algorithms, and the model itself.

At the data level, the data generated by models in the process of self-learning and self-reflection often faces problems of data diversity and generation efficiency. The problem of data diversity mainly refers to insufficient data diversity, where the model only uses the data it prefers to produce, over-sampling when dealing with simple problems and severely missing samples when dealing with complex problems, and this phenomenon becomes more obvious as the number of iterations increases, resulting in the degradation of the model's generalization and reasoning ability [1]. The problem of data efficiency mainly refers to the high training cost and low marginal benefit due to redundant, low-quality or even harmful samples in the data, as well as the model's own insufficient ability to evaluate data quality. The bias of large language models leads to inaccurate scoring and easy misjudgment, resulting in the inability to accurately and efficiently screen out high-quality samples [2].

At the algorithmic level, the algorithm is the core engine that drives the improvement of the model's implementation capabilities. It often encounters problems with insufficient self-correction and iterative evolution capabilities. Currently, model tasks fail mostly because they are unable to adjust themselves and cannot keep up with the iterations. They cannot stably identify their own errors, cannot accurately find the root cause of the errors, and the accuracy of self-validation is insufficient. During the self-correction and reflection stage, the model is prone to fall into a "self-loop", repeatedly proposing non-progressive actions. At the same time, effective self-correction requires a large amount of external data, as well as considerable training time and costs [3]. The main problem with model iterative evolution is that the traditional reinforcement learning framework usually assumes the maximum depth of iterations, limiting their flexibility and potentially leading to resource waste and increased costs. Additionally, traditional frameworks have slow training speeds when dealing with multi-step reasoning, and the context length and memory are prone to depletion, making iterative evolution difficult to carry out stably and efficiently [4].

The main limitations at the model level and within the model itself are model fragility, capability degradation, and imbalance. The issue of model fragility mainly refers to the high sensitivity of the model to input changes, making it prone to errors. The model is highly dependent on a few "key neurons". These neurons are concentrated in the external MLP down_proj module and are vulnerable to attacks. If these neurons are damaged, the model will completely collapse, its performance will sharply decline, and it will threaten the security of the model [5]. The issue of model capability degradation and imbalance refers to the possibility of a decline in model performance after multiple iterations. The reason for this situation lies in the superficial nature of the model's reasoning. It seems to be learning reasoning, but in fact, it is more like memory. The updates of the model are concentrated in certain parts, which do not provide much help for reasoning, and the updated weights are relatively small in the key parts of reasoning, resulting in insufficient generalization ability of the model's reasoning [6].

3. Analysis of Existing Methods and Feasibility for Enhancing the Capabilities of Large Language Models

3.1 Data-level solutions and feasibility analysis

The common solutions to the common problems of Data Diversity and generation efficiency at the data level are the Seed-Driven Growth Technique (SDGT) and the diverse-aware Score Curation for Data Selection (DS2) method.

Among them, Gao et al. proposed the Seed-Driven Growth Technique (SDGT) for issues such as data diversity in large language models [7]. Based on a small amount of high-quality seed data, the method achieves automatic expansion of high-quality training data through diversity control and consistency control, thereby reducing the cost of manual annotation while improving data generation efficiency. Experiments show that the performance of SDGT-generated data in tasks such as summary generation, reading comprehension, and reasoning is on average 88% and up to 114% higher than that of manually high-quality labeled data, indicating strong application potential in self-generated data construction. The method is particularly suitable for scenarios such as continuous fine-tuning of large language models, self-improvement, and low-cost data expansion.

For data efficiency issues, Zhang et al. proposed Diversity-aware Score Curation for Data Selection (DS2) [8]. This approach improves data efficiency by revising the quality scoring results of large language models, enhances the accuracy of the quality scoring of data samples by using the K-NN clustering assumption, calculates the Long-tail Diversity Score, selects samples with low similarity to make the samples more distinctive, and solves the problem of lengthy and monotonous text. By fine-tuning a small number of high-quality samples, efficiency is improved while quality is guaranteed. Experiments showed that SD2, fine-tuned with 3.3% of the samples, trained better than a complete dataset of 300,000 samples. It shows that high-quality small sample data can avoid the problem of data redundancy and low quality. This method is applicable to scenarios where data resources are limited and small sample modeling is used.

3.2 Algorithm-level solutions and feasibility analysis

Common solutions to the problem of insufficient self-correction and iterative evolution capabilities at the algorithmic level include the S2R framework, Thought-ICS, and the exploratory iteration (ExIt) approach.

Among them, Ruotian Ma et al. proposed the S2R framework for the problem of self-correction ability [9]. This approach enables the model to have self-validation and error correction behavior through supervised fine-tuning, and further enhances this ability through reinforcement learning and reward mechanisms at both the result and process levels, thereby enabling it to learn to actively validate and correct errors in reasoning. Experiments show that the S2R method can improve the accuracy of Qwen2.5-math-7B from 51.0% to 81.6% with just 3.1k training samples and outperforms long chain of thought data distillation. It shows that this method significantly improves the model's error correction ability while using fewer resources, without the need for manual annotation and large-scale distillation, saving time costs. This method is applicable to scenarios such as complex logical reasoning, tasks with scarce data and expensive annotation. Samanta et al. proposed Thought-ICS [10]. The method achieves self-correction by locating the first wrong step and then rereasoning from the last correct step. Experiments show that the Thought-ICS method can achieve a 20%-40% improvement in self-correction. It shows that the method is capable of precisely locating and correcting errors

Jiang [4] et al. proposed the Exploratory iteration (ExIt) method to address issues such as high cost and long training time for multi-step reasoning during model iterative evolution. This method achieves multi-step self-optimization through single-step self-improvement transformation, prioritizing the output of the previous round as the starting point for training, and continuously expanding the task space to automatically generate data augmentation. This avoids the problems of being overly strict and costly with a fixed iteration depth. Experiments show that the ExIt method can significantly increase reasoning accuracy and improve generalization ability to gradually solve complex tasks. This method is applicable to scenarios where multi-step iterative optimization is required during reasoning, where training data is scarce, and where diverse results need to be generated.

3.3 Model-level solutions and feasibility analysis

Common solutions to the common problems of model fragility and model capability degradation and imbalance at the model level include perturbed key neuron causal identification, forgetting and reconnection, and iterative model fusion.

For model fragility, Qin et al. proposed a perturbation-based causal identification method for locating key neurons [4]. This method identifies key neurons by combining noise sensitivity analysis with causal verification, disperses key neurons concentrated in the outer layer to more layers, and regularizes during training to reduce the model's dependence on a small number of neurons. Real-time monitoring is carried out to enhance defense mechanisms. Experiments have shown that the method can prevent the model from crashing when a small number of key neurons are damaged, and it has been validated through 21 models and multiple datasets as feasible. It can be applied to fields with high security requirements such as healthcare and finance. Another approach is the Forgetting and Reconnection (FaR) mechanism proposed by Yuan et al. [11], which reassigns critical functions to non-critical neurons, hides critical weights, and reduces sensitivity to critical neurons and attackers' identification of vulnerable neurons. Experiments showed that the method reduced the success rate of bit-flip attacks by 1.4 to 4.2 times, and the obfuscation rates against expert attackers and basic attackers were up to 84% and 91% respectively, while the accuracy loss was less than 2%, enhancing robustness.

Yuan et al. proposed an iterative model fusion (IMM) method for the problem of capability enhancement generalization degradation [6]. This approach strikes a balance between performance improvement and maintenance of generalized reasoning ability by strategically combining the weights of the original model and the self-improved model and increasing the update weights for the key parts. Experiments have shown that the IMM approach ensures that the model's performance improves or remains stable after iteration without model crashes. It is applicable to scenarios such as multilingual reasoning, academic reasoning, and complex logic programming.

3.4 A common metric for improving the performance of large language models

The metrics for the core reasoning ability of the model mainly include accuracy in mathematical reasoning and common sense reasoning; The accuracy rate between the generated content and the facts; Average inference time for a single sample; and model storage overhead.

For model security issues, the success rate of bit-flip attacks and the obfuscation rate of key parameters are often used to assess a model's ability to defend against external attacks.

For model stability issues, common metrics include the number of iterations of adversarial samples and the perplexity of the model after masking key neurons.

Common evaluation indicators for the self-correction and self-verification capabilities of models include the accuracy rate of locating error steps. Error recall rate: The proportion of marked errors; Specificity: The proportion that retains correct reasoning; Error correction rate; Correct error rate.

Common metrics for evaluating data diversity and quality include manual annotation consistency, data information density, difficult sample retention rate, semantic space coverage, and lexical level generated text diversity.

4. Conclusions

This article reviews the bottlenecks in enhancing the capabilities of large language models themselves and the existing solutions. In terms of existing problems, there are common issues of data diversity and generation efficiency at the data level, insufficient self-correction and iterative evolution capabilities at the algorithm level, and model fragility and degradation and imbalance of model capabilities at the model level. The existing solutions include SDGT, DS2, the S2R framework, Thought-ICS, ExIt, key neuron causality identification, forgetting and reconnection, and iterative model fusion. Existing performance metrics for large language models include inference accuracy, success rate of bit-flip attacks, error recall, and other performance evaluations in terms of inference ability, model security, stability, self-correction ability, data diversity and quality. This review has limitations due to its limited coverage of the literature and insufficient

organization of multiple perspectives. In the future, people can further analyze and construct data-algorithm-model multi-dimensional optimization schemes and performance evaluation schemes based on the problems of the model in terms of data, algorithms, models, etc., in combination with current solutions for specific problems, to improve the practicability, security and accuracy of the model and enable the model to be applied in more fields.

References

[1] Ding Y, Xi Z, He W, et al. Mitigating Tail Narrowing in LLM Self-Improvement via Socratic-Guided Sampling[C]//Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). 2025: 10627-10646.

[2] Pang J, Wei J, Shah A P, et al. Improving data efficiency via curating llm-driven rating systems[J]. arXiv preprint arXiv:2410.10877, 2024.

[3] Maity A, Potamitis N, Arora A. Reconciling Divergent Views Through a Critical Analysis of Iterative Self-Improvement in LLMs[J]. 2025.

[4] Jiang M, Lupu A, Bachrach Y. Bootstrapping task spaces for self-improvement[J]. arXiv preprint arXiv:2509.04575, 2025.

[5] Qin Z, Lyu K, Yu Q, et al. The Achilles' Heel of LLMs: How Altering a Handful of Neurons Can Cripple Language Abilities[J]. arXiv preprint arXiv:2510.10238, 2025.

[6] Yuan X, Zhang C, Liu Z, et al. Superficial self-improved reasoners benefit from model merging[C]//Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing. 2025: 5912-5932.

[7] Gao D, Dai J, Liu S, et al. SDGT: LLMs fine-tuning with seed-driven growth technology based on GPT-4 data expansion[J]. Neurocomputing, 2026: 132766.

[8] Zhang J, Zhang C X, Liu Y, et al. D3: Diversity, difficulty, and dependability-aware data selection for sample-efficient llm instruction tuning[J]. arXiv preprint arXiv:2503.11441 2025.

[9] Ruotian Ma, Peisong Wang, Cheng Liu, Xingyan Liu, Jiaqi Chen, Bang Zhang, Xin Zhou, Nan Du, and Jia Li. 2025. S2R: Teaching LLMs to Self-verify and Self-correct via Reinforcement Learning. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 22632–22654, Vienna, Austria. Association for Computational Linguistics.

[10] Samanta A, Magesh A, Jain A, et al. Structure Enables Effective Self-Localization of Errors in LLMs[J]. arXiv preprint arXiv:2602.02416, 2026.

[11] Nazari N, Makrani H M, Fang C, et al. Forget and rewire: Enhancing the resilience of transformer-based models against {Bit-Flip} attacks[C]//33rd USENIX Security Symposium (USENIX Security 24). 2024: 1349-1366.

Funding

This research received no external funding.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgment

This paper is an output of the science project.

Open Access