

Research on the Length Controllability of Chinese Large Language Model Outputs Based on Post-Processing Correction

Yiheng Xu*

School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China

**Corresponding author: Yiheng Xu.*

Abstract

Large language models often deviate from the target word count when following length constraints such as “generate X words.” Existing studies mostly focus on content quality, while paying relatively less attention to format control, especially the accuracy of length control. To address this issue, this study builds a Chinese text generation benchmark covering 5 topics and 4 word-count levels, and evaluates three representative models: DeepSeek-V3, Tongyi Qianwen (Qwen2.5), and Kimi. Based on this, a lightweight post-processing framework, LAPP (Length-Aware Post-Processing), is proposed. When the output is too long, LAPP uses sentence-level intelligent truncation; when the output is too short, it calls Qwen2.5-1.5B-Instruct for semantics-preserving expansion, and controls the length error within $\pm 10\%$ through iterative correction. The experimental results show that the original average deviation rates of the three models are 11.93%–38.70%, which are reduced to about 5% after LAPP correction, and the pass rate increases from 20%–70% to 85%–100%. The semantic preservation results show an average BERTScore of 0.97 between the corrected and original texts, suggesting that the framework improves length control while largely keeping the original meaning. This study shows that the “rough generation + precise correction” approach is effective for word-count control, and offers a low-cost, transferable way to improve the instruction-following ability of large language models.

Keywords

large language models, length control, instruction following, post-processing correction, controllable text generation

1. Introduction

Large language models (LLMs), including GPT-4, DeepSeek-V3, Tongyi Qianwen, and Kimi, are widely used for writing, coding, and question answering [1]. However, they often fail to follow word-count instructions. A request for 200 words, for example, may produce 500 words or only 80. Since academic papers, official documents, and social media posts often have length limits, accurate word-count control is a basic practical requirement [2].

However, current mainstream large language models still have difficulty meeting practical requirements when following “numerical constraint instructions.” Existing studies pay more attention to knowledge accuracy, reasoning ability, and safety [3,4], while systematic evaluation and improvement of “word count control,” a basic instruction-following ability, are still relatively insufficient. Based on this, this study focuses on Chinese text generation scenarios, builds a word count control evaluation benchmark, systematically evaluates the word count

following ability of three mainstream large language models, and proposes a lightweight post-processing correction framework. This framework does not need to adjust the internal parameters of the models, but corrects the length of generated results through two strategies, “truncation” and “expansion,” so as to verify the effectiveness of the “generation + correction” paradigm in controllable text generation tasks [5].

Ouyang et al. [1] improved instruction following through RLHF, yet Zhou et al. [6] found persistent problems with format constraints. Studies on model evaluation [3], Chinese generation quality [4], controllable generation [2], and automatic evaluation [7] provide useful methods but give limited attention to character-count control. Decoding-based approaches can control length [8], though changes to the generation process complicate deployment across models. This study therefore evaluates DeepSeek-V3, Tongyi Qianwen, and Kimi on a Chinese character-count benchmark using LDR and EHR. It introduces LAPP, a lightweight post-processing framework designed to bring length errors within $\pm 10\%$ through iterative correction, and measures semantic fidelity to assess changes in meaning.

2. Related Work and Comparison of Technical Routes

2.1 Research Progress on Evaluation Benchmarks for Large Model Instruction Following

Based on the Transformer architecture [9], large language models carry out self-supervised pre-training on massive texts, and learn language statistical features and semantic knowledge from them. BERT [10] and the GPT series are representative models in this direction. With the rapid development of Chinese large language models, models such as DeepSeek-V3, Tongyi Qianwen Qwen2.5, and Kimi have shown good performance in Chinese understanding and generation tasks [11]. Instruction-following ability describes the level at which a model completes target tasks according to natural language instructions. Ouyang et al. [1] significantly improved the instruction-following ability of GPT-3 through instruction tuning and RLHF, but existing models still have obvious defects when responding to numerical constraint instructions such as output length [6].

2.2 Built-in Controllable Text Generation Methods

Current mainstream length control methods mainly include three types: prompt constraints, decoding constraints, and model fine-tuning. Prompt constraints add word count requirements in the prompt. They are easy to use, but the generation effect fluctuates greatly. Decoding constraints introduce length penalty or dynamic truncation mechanisms during the generation stage [8]. They require changes to the model inference process, so they are difficult to implement and have limited generalization ability. Model fine-tuning relies on special datasets for secondary training to improve length control, but the training cost is high and cross-scenario transfer is difficult. The length-controllable generation method proposed by Sun et al. [8] realizes output constraints by adjusting the length penalty term in the decoding strategy, but this method needs to change the internal parameters of the model, so the threshold for actual deployment is relatively high.

2.3 Advantage Analysis of the Post-processing Text Correction Paradigm

Compared with built-in control methods, the post-processing correction method has several clear features. This method is independent of the model itself, and does not need to change the internal parameters or inference logic of large language models. It can be used as a general plug-in to adapt to different models called through APIs or deployed locally. In the deployment stage, it does not need new training datasets or extra computing power, and can be used directly. The text correction process is clear and transparent, with good interpretability, making it convenient for debugging and iteration. At the same time, it can be combined with any generation model, and will not be affected by model version updates. This paper adopts the post-processing technical route, builds a lightweight correction framework LAPP, and verifies the practical value of this paradigm in the scenario of text word count control.

3. Benchmark for Length Control in Chinese Outputs

3.1 Evaluation Dataset Design

To evaluate how well large language models follow word count constraint instructions, this paper refers to existing evaluation schemes [3,4] and builds the evaluation dataset from two dimensions: topic type and target

text length. The test topics include five Chinese writing themes: artificial intelligence, environmental protection, educational equity, technological innovation, and healthy living, covering the fields of technology, society, and people’s livelihood. The length settings include four levels: 50 words, 100 words, 200 words, and 500 words, which correspond to ultra-short, short, medium, and long text generation scenarios. To reduce the random error caused by a single output, each topic and word count combination is repeatedly sampled 3 times, and 60 test samples are finally obtained ($5 \times 4 \times 3$). All test prompts use direct instructions in a unified form. An example is as follows: “Please write an article about ‘artificial intelligence,’ with the word count controlled at around 200 words. Only output the main text, and do not add any extra explanation.”

3.2 Selection of Models

Three mainstream LLMs representative of Chinese-language applications were selected, as shown in Table 1. (1) DeepSeek-V3, developed by DeepSeek, provides strong Chinese-language understanding and relatively low API costs. (2) Qwen2.5, developed by Alibaba’s DAMO Academy, performs well on Chinese generation tasks and has a mature open-source ecosystem. (3) Kimi, developed by Moonshot AI, is known for long-context processing and has a large user base. Generation temperature and all other hyperparameters were left at their default values for every model to support a fair and reproducible comparison.

Table 1: Basic Information on the Evaluated Models

Model	Developer	Key characteristics
DeepSeek-V3	DeepSeek	Strong Chinese-language understanding and relatively low API cost
Qwen2.5	Alibaba DAMO Academy	Strong Chinese generation performance and a mature open-source ecosystem
Kimi	Moonshot AI	Known for long-context processing and a large user base

3.3 Quantitative Evaluation Metrics

Three metrics are used in this study. LDR measures the relative difference between the actual and target character counts:

$$LDR = \frac{|L_{actual} - L_{target}|}{L_{target}} \times 100\% \quad (1)$$

where L_{actual} is the actual output length and L_{target} is the target length. An output qualifies when its deviation rate is no greater than 10%.

EHR gives the proportion of qualified samples:

$$EHR = \frac{N_{qualified}}{N_{total}} \times 100\% \quad (2)$$

where $N_{qualified}$ is the number of samples with a deviation rate no greater than 10%, and N_{total} is the total number of samples.

Semantic Fidelity (SF) is measured with BERTScore [12] by comparing the corrected text with the original. Since BERTScore uses contextual embeddings rather than n-gram overlap, it can better identify whether the original meaning is retained.

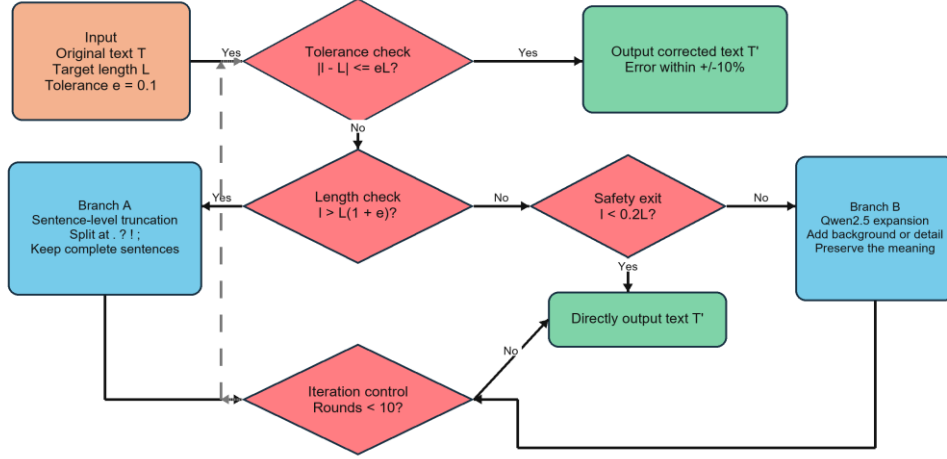
4. Lapp a Lightweight Length-Aware Post-Processing Framework

4.1 Overall Architecture

The measured data show that the word count deviation rates of DeepSeek-V3, Qwen2.5, and Kimi range from 11.93% to 38.70%, and the overall results show that the deviation is larger in long-text generation. This problem comes from the inherent mechanism of autoregressive word-by-word generation in large language models, which cannot make a global prediction of the final text length [5]. For this reason, this paper proposes a lightweight post-processing correction framework, LAPP, and adopts the optimization idea of “rough generation + precise correction.” This method lets the model first ensure the quality of generated content, and

then uses an independent external module to accurately correct word count deviation. Since it does not need to change the internal structure of the model, LAPP can be directly adapted to various large model interfaces, with simple deployment and strong generality. The overall workflow is shown in Figure 1.

Figure 1: Workflow of the LAPP Framework



4.2 Lapp Correction Process

When the character count is higher than $L \times (1 + \varepsilon)$, LAPP cuts the text by complete sentences instead of cutting directly at L , using periods, question marks, exclamation marks, and semicolons as boundaries. When the character count is lower than $L \times (1 - \varepsilon)$, it calls Qwen2.5-1.5B-Instruct to expand the text based on the original content, with the prompt requiring about $[L]$ Chinese characters, preserved meaning, added details, and no extra explanation. In each iteration, LAPP counts the current length l . The text is returned if $|l - L| \leq \varepsilon \cdot L$, truncated when $l \geq L \cdot (1 + \varepsilon)$, and expanded when $0.2L \leq l < L \cdot (1 - \varepsilon)$. If $l < 0.2L$, the correction stops and the original text is kept to avoid excessive expansion and semantic distortion.

5. Experimental Evaluation and Results

5.1 Experimental Setup Benchmark and Hyperparameters

The experiments were conducted in Python 3.11, with Qwen2.5-1.5B-Instruct running locally on a CPU. The benchmark contained 60 samples from five topics, four target lengths, and three runs per setting. DeepSeek-V3, Qwen2.5, and Kimi were evaluated using LDR, EHR with a $\pm 10\%$ tolerance, and BERTScore-based SF. For post-processing, ε was set to 0.1 and the maximum number of iterations to 10, while Qwen expansion was optional.

5.2 Baseline Length Control of Mainstream LLMs

Table 2 reports the uncorrected length deviation rate for each model at each target length.

Table 2: Original Length Deviation Rates at Different Target Lengths (%)

Model	50 characters	100 characters	200 characters	500 characters	Overall mean
DeepSeek-V3	8.0	22.3	30.0	31.2	22.88
Qwen2.5	28.0	49.3	45.0	32.5	38.70
Kimi	7.5	8.8	12.5	18.9	11.93

The results show that Kimi has the best overall performance, with an average deviation rate of 11.93%. Its advantage is more obvious in short-text tasks, and the deviation rates of the 50-word and 100-word tasks are both controlled within 10%. In comparison, Tongyi Qianwen has the highest deviation rate, reaching 38.70%.

In its 100-word task, the deviation is close to 50%, showing that its response to short-text length constraints is not stable enough. Overall, all three models show a trend that the longer the target word count, the larger the deviation, which indicates that length planning is more difficult in long-text generation. It is worth noting that Tongyi Qianwen has a lower deviation rate in the 50-word task than in the 100-word task. This may be related to its tendency to generate more complete and detailed answers; under a shorter constraint, the model is more likely to trigger compressed output.

5.3 Effect of Lapp Correction

Table 3 compares the overall deviation rates before and after LAPP correction.

Table 3: Overall Deviation Rates Before and After LAPP Correction

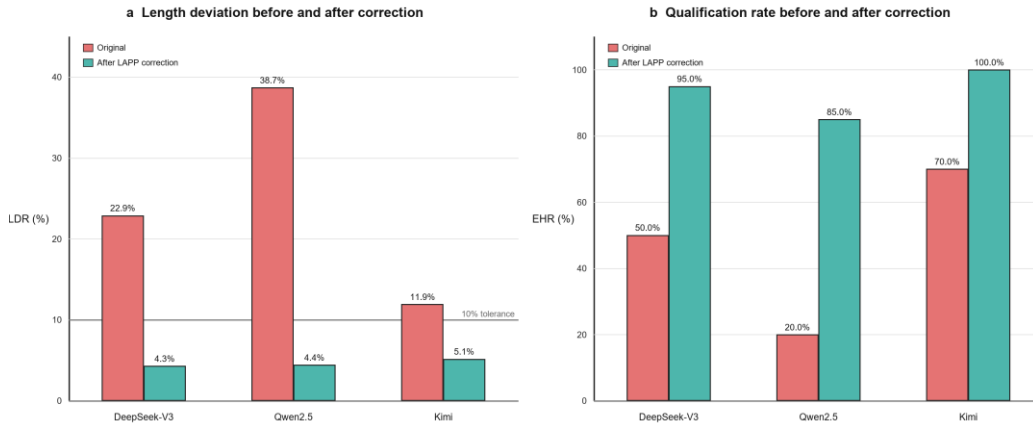
Model	Original LDR (%)	Corrected LDR (%)	Reduction (%)
DeepSeek-V3	22.88	4.28	18.60
Qwen2.5	38.70	4.42	34.28
Kimi	11.93	5.14	6.79

Table 4 compares the qualification rates before and after correction. An output qualifies when its deviation is no greater than 10%.

Table 4: Qualification Rates Before and After Correction

Model	Original EHR(%)	Corrected EHR(%)	Increase(%)
DeepSeek-V3	50.0	95.0	45.0
Qwen2.5	20.0	85.0	65.0
Kimi	70.0	100.0	30.0

Figure 2: Model Performance Before and After LAPP Correction



Tests across the three models showed that LAPP can work as a general correction tool. The largest improvement was found in Tongyi Qianwen, whose deviation rate dropped from 38.70% to 4.42%. After correction, the pass rate exceeded 85% for every model, reaching 100% for Kimi and 95% for DeepSeek-V3. Even in Tongyi Qianwen's 50-character task, the deviation fell from 28% to below 6%, suggesting that LAPP remains effective on difficult samples.

5.4 Semantic Fidelity and Generated Content Quality

To examine whether length correction affects text quality, this paper uses BERTScore [12] to measure SF and compares 20 samples from each of the three models. The results are shown in Table 5.

Table 5: Semantic Fidelity After LAPP Correction Measured by BERTScore

Model	Truncated-sample SF	Expanded-sample SF	Overall mean SF
DeepSeek-V3	0.9479	0.9748	0.9745
Qwen2.5	0.9206	0.9547	0.9513
Kimi	1.0000	0.9806	0.9961

The average scores of all three models are above 0.95, indicating that the original meaning is largely retained after correction. The scores of truncated samples range from 0.92 to 1.00 and are generally higher than the 0.95–0.98 scores of expanded samples, possibly because expansion adds some new information. Kimi has the highest score of 0.996, which may be related to its smaller initial deviation and fewer changes. Expanded samples still score above 0.95, showing no obvious semantic drift in the content generated by Qwen2.5-1.5B. Overall, LAPP does not noticeably reduce content quality when adjusting text length.

5.5 Case Studies and Error Analysis

Table 6 shows typical correction cases, which can more directly explain the actual effect of LAPP.

Table 6: Representative Correction Cases

Case	Model	Topic	Target	Original	Original deviation(%)	Corrected	Corrected deviation(%)
1	DeepSeek-V3	Artificial intelligence	200	118	41.0	183	8.5
2	Qwen2.5	Artificial intelligence	100	57	43.0	93	7.0
3	Kimi	Artificial intelligence	500	394	21.2	473	5.4
4	DeepSeek-V3	Artificial intelligence	50	50	0.0	50	0.0
5	Qwen2.5	Environmental protection	200	73	63.5	199	0.5

In Case 5, Tongyi Qianwen generated only 73 words for a 200-word task, with a deviation rate of 63.5%. After nine rounds of expansion, the text reached 199 words and the deviation rate fell to 0.5%, showing that LAPP can correct large initial deviations. However, the framework still has some limitations. Texts shorter than 15 words contain too little information for coherent expansion, and Qwen2.5-1.5B may sometimes generate repetitive or off-topic content. Some samples also require more than eight iterations, which increases computational cost. In addition, this study counts only Chinese characters and excludes punctuation and English characters, so the counting rules should be adjusted for specific applications.

6. Conclusions and Future Work

Tests on DeepSeek-V3, Tongyi Qianwen, and Kimi showed initial length deviations of 11.93%–38.70%, with larger errors in longer texts. By truncating long outputs and expanding short ones, LAPP reduced the deviation to about 5% and raised pass rates to 85%–100%, while an average BERTScore of 0.97 indicated little semantic loss. However, the method adds inference time, counts only Chinese characters, and was tested on just 60 samples. Future work should improve efficiency, expand multilingual testing, and explore lightweight deployment and decoding-based control.

References

- [1] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., ... & Lowe, R. (2022). Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35, 27730-27744. <https://doi.org/10.52202/068431-2011>
- [2] Wang, J., & Sun, Y. (2024). A review of controllable text generation techniques [In Chinese]. *Journal of Chinese Information Processing*, 38(10).
- [3] Luo, W., & Wang, H. (2024). A review of large language model evaluation [In Chinese]. *Journal of Chinese Information Processing*, 38(1), 1-23..
- [4] Zhang, W., Zhang, Y., et al. (2024). An indicator system for evaluating the generation quality of Chinese large language models [In Chinese]. *Journal of Chinese Information Processing*, 38(3), 1-15.
- [5] Liu, K., Qu, X., Tan, F., et al. (2024). A review of large language model alignment research [In Chinese]. *Telecommunications Science*, 40(6), 173-194.

- [6] Zhou, J., Lu, T., Mishra, S., Brahma, S., Basu, S., Luan, Y., Zhou, D., & Hou, L. (2023). *Instruction-following evaluation for large language models*. <https://doi.org/10.48550/arXiv.2311.07911>
- [7] Ma, Q., Zhang, J., & Liu, Q. (2018). A machine translation automatic evaluation method based on fusion strategy [In Chinese]. *Journal of Chinese Information Processing*, 32(9), 11-19.
- [8] Bastan, M., Surdeanu, M., & Balasubramanian, N. (2023). Neurostructural decoding: Neural text generation with structural constraints. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics* (pp. 9496-9510). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2023.acl-long.528>
- [9] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 5998-6008. <https://doi.org/10.48550/arXiv.1706.03762>
- [10] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 4171-4186). Association for Computational Linguistics. <https://doi.org/10.18653/v1/N19-1423>
- [11] Kikuchi, Y., Neubig, G., Sasano, R., et al. (2016). Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (pp. 1328-1338). Association for Computational Linguistics. <https://doi.org/10.18653/v1/D16-1140>
- [12] Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2020). BERTScore: Evaluating text generation with BERT. *International Conference on Learning Representations*. <https://doi.org/10.48550/arXiv.1904.09675>

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.

Copyrights

Copyright for this article is retained by the author (s), with first publication rights granted to the journal. This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).