RULER : A Model-Agnostic Method to Control Generated Length for Large Language Models

Jiaming Li[1,2] **Lei Zhang**[1,2] **Yunshui Li**[1,2] **Ziqiang Liu**[1,2] **Yuelin Bai**[1,2]
Run Luo[1,2] **Longze Chen**[1,2] **Min Yang**[1†]
[1]Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
{jm.li4, min.yang}@siat.ac.cn

Abstract

The instruction-following ability of large language models enables humans to interact with AI agents in a natural way. However, when required to generate responses of a specific length, large language models often struggle to meet users' needs due to their inherent difficulty in accurately perceiving numerical constraints. To explore the ability of large language models to control the length of generated responses, we propose the Target Length Generation Task (*TLG*) and design two metrics, Precise Match (PM) and Flexible Match (FM) to evaluate the model's performance in adhering to specified response lengths. Furthermore, we introduce a novel, model-agnostic approach called RULER, which employs Meta Length Tokens (*MLTs*) to enhance the instruction-following ability of large language models under length-constrained instructions. Specifically, RULER equips LLMs with the ability to generate responses of a specified length based on length constraints within the instructions. Moreover, RULER can automatically generate appropriate *MLT* when length constraints are not explicitly provided, demonstrating excellent versatility and generalization. Comprehensive experiments show the effectiveness of RULER across different LLMs on Target Length Generation Task, e.g., at *All Level* 27.97 average gain on PM, 29.57 average gain on FM. In addition, we conduct extensive ablation experiments to further substantiate the efficacy and generalization of RULER. Our code and data is available at https://github.com/Geaming2002/Ruler.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across a variety of natural language tasks and are increasingly being utilized in various fields (Vaswani et al., 2017; Devlin et al., 2019; Brown et al., 2020). A primary area of interest is the instruction following ability, referring to their capability to execute tasks or generate outputs based on instructions (Ouyang et al., 2022; Wei et al., 2022a). It reflects the model's effectiveness in understanding and responding to instructions.

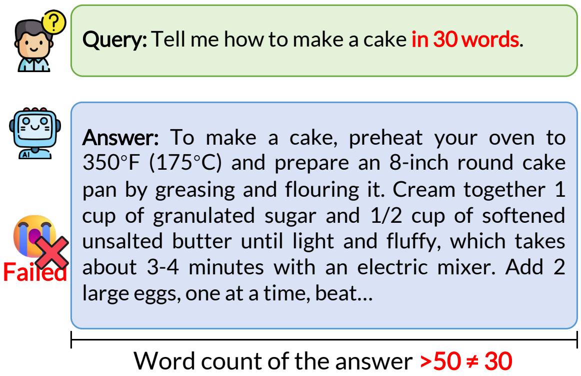

Figure 1: Existing LLMs lack the capability to follow instructions for generating texts of a specified length.

The practical challenges highlight the complexity of achieving precise instruction following, particularly when users require control over the output's length. Users frequently give LLMs various instructions, such as "Tell me how to make a cake in 20 words", "Use 50 words to write a post", "Write a 300-word story for me" and so on. These instructions challenge the instruction following capability of LLMs. To explore how well LLMs handle such challenges, we focus on the scenario where users specify the target length of the responses. The question is posed, "Can LLMs accurately generate with target length?" and introduce the *Target Length Generation Task (TLG)*. We create a test dataset with various target lengths and introduce two evaluation metrics: Precise Match (PM) and Flexible Match (FM). Our findings reveal that current LLMs generally perform poorly in this task, indicating considerable room for improvement. A discussion on the underlying causes is conducted, primarily attributing it to tokenization schemes and

[†]Min Yang is the corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 3042–3059
November 12-16, 2024 ©2024 Association for Computational Linguistics

model training strategy.

To address aforementioned issues, we introduce RULER, a model-agnostic approach designed to enhance the instruction-following capability of LLMs through *Meta Length Tokens (MLTs)*. *MLTs* are designed to control model's responses. By utilizing RULER, LLMs can generate responses that meet target lengths. We create a dataset with *MLTs* $\mathcal{D}_{MLT}$ for end-to-end training of LLMs. LLMs learn to generate *MLT* and the corresponding length response after training. During inference, if a target length is provided, RULER can transform it into a *MLT* and generate responses that meet the requirement. If no target length is specified, it first generates a *MLT*, then the response, ensuring its length aligns with the generated *MLT*.

We apply RULER to various large language models and test them on *TLG*. Each model demonstrates significant improvements. Across all evaluated models, we observe a consistent improvement in both PM and FM scores at all *Levels*. The PM and FM scores across *All Level* showed an average improvement of 27.97 and 29.57.Furthermore, to rigorously test the capabilities of RULER, we randomly sample the dataset provided by Li et al. (2024a) and assess RULER on multi *MLT* generation and self-generated *MLT* experiment to show the its effectiveness and generalizability. Additionally, RULER is tested on six benchmarks to observe whether the models' overall performance is affected.

Our contributions can be summarized as follows:

- We introduce the *Target Length Generation Task (TLG)*, which designed to assess the instruction following capability of LLMs. It evaluates how well models generate responses of target lengths as directed by instructions.

- We propose RULER, a novel and model-agnostic approach which employs the *Meta Length Tokens (MLTs)*. Through end-to-end training, it enables models to generate response matching the target lengths indicated by *MLTs*.

- We demonstrate that RULER significantly enhances the performance of various models on the *TLG*. Further experiments have also validated the effectiveness and generalizability of RULER.

2 Related Work

2.1 Large Language Model

The advent of LLMs has revolutionized the field of natural language processing and become a milestone (Vaswani et al., 2017; Devlin et al., 2019; Brown et al., 2020; Zhang et al., 2023a). Large language models have achieved success across various NLP tasks. Models such as GPT-4(Achiam et al., 2023), Llama-3(AI@Meta, 2024), and Qwen(Bai et al., 2023), known for their powerful capabilities, are increasingly serving as the foundation for various applications and making significant inroads into diverse fields, exerting a substantial impact. In-context learning enables LLMs to infer and generate responses solely based on the contextual information provided within a prompt(Dong et al., 2022; Wei et al., 2022b). This capability allows the models to exhibit a high degree of flexibility and adaptability across a variety of tasks(Levine et al., 2022; Chen et al., 2022; Zhao et al., 2021). CoT further excavates and demonstrates the powerful logical reasoning capabilities of LLMs(Wei et al., 2022c; Huang and Chang, 2023; Zhang et al., 2023b).

2.2 Instruction Following

Instruction following refers to the ability of large language models to comprehend and execute given natural language instructions (Brown et al., 2020; Ouyang et al., 2022; Wei et al., 2022a; Zhou et al., 2023a). This capability enables the models to perform a broad spectrum of tasks, from simple query responses to complex problem-solving and content generation, tailored to specific user requests.

In practical deployments, models may not adhere to comply with user instructions, exhibiting behaviors that deviate from anticipated outcomes. This includes generating responses unrelated to explicit instructions, emitting redundant or erroneous information, or entirely ignoring specified directives (Gehman et al., 2020; Kenton et al., 2021; Wei et al., 2024). To enhance the instruction following capability of LLMs, open-domain instruction following data is frequently used for training. Several prominent studies have explored the construction of instruct-tuning data, to achieve efficient and cost-effective results(Li et al., 2024b; Cao et al., 2024; Liu et al., 2024; Xu et al., 2024).

Level	Target Length	Precise Match (PM)	Flexible Match (FM)
Level:0	10	± 10	$(0, 20]$
	30	± 10	$(20, 40]$
	50	± 10	$(40, 60]$
	80	± 10	$(60, 100]$
Level:1	150	± 20	$(100, 200]$
	300	± 20	$(200, 400]$
	500	± 50	$(400, 600]$
Level:2	700	± 70	$(600, 800]$
	>800	$(800, \infty)$	$(800, \infty)$

Table 1: Nine target lengths and their corresponding match ranges categorized as Precise Match (PM) and Flexible Match (FM). Target lengths are classified into three categories, *Level:0*, *Level:1*, and *Level:2*.

2.3 Meta Token

Recently, an increasing number of studies have employed custom tokens within language models to execute specific functions or enhance performance. Todd et al. (2024) report findings that the hidden states of language models capture representations of these functions, which can be condensed into a Function Vector (FV). Furthermore, their research demonstrates that FV can effectively guide language models in performing specific tasks.

Numerous studies have utilized meta tokens to compress prompts, thereby enhancing the inference capability of models (Li et al., 2023; Liu et al., 2023; Zhang et al., 2024). Mu et al. (2023)introduce the concept of "gist tokens", which can be cached and reused for compute efficiency. Further Jiang et al. (2024) utilize a hierarchical and dynamic approach to extend the concept, proposing "HD-Gist tokens" to improve model performance.

3 Can LLMs Accurately Generate with Target Length?

In this section, we examine the capability of LLMs to generate responses of a target length. Initially, we introduce *Target Length Generation Task (TLG)*. Subsequently, we establish various target lengths and two evaluation metrics (§3.1). We then detail the experimental setup and assess the ability of LLMs to generate responses at target lengths (§3.2). Finally, we present the outcomes of the experiments and analysis the underlying reasons(§3.3).

3.1 Target Length Generation Task

To assess the ability of existing LLMs to control the length of generated response, we develop the *TLG*. This task assesses the models' ability in producing responses that match target lengths as directed designed target lengths are detailed in Table 1. Additionally, we divide these nine target lengths into three *levels*: *Level:0*, *Level:1*, and *Level:2*.

Given that generating responses with target lengths is challenging for existing LLMs, we develop two metrics to evaluate the accuracy of response lengths.

- **Precise Match (PM):** This metric requires that the length of the generated response be very close to the target length. For different *Level*, a precise tolerance range is set (± 10, ± 20, $\ldots$) necessitating that the response length stringently conforms to these defined limits.
- **Flexible Match (FM):** This metric requires a broader tolerance interval for target length. For longer texts, the range incrementally widens to meet response generation requirements.

For the N responses, we assess whether response meets the target length, then calculating the PM and FM scores of the model.

$$\mathrm{PM} = \frac{\sum_{i=1}^{N} \mathbb{1}\left(\mathrm{lb}_{\mathrm{TL_i}}^{\mathrm{P}} < L\left(c_i\right) \leq \mathrm{ub}_{\mathrm{TL_i}}^{\mathrm{P}}\right)}{N} \quad (1)$$

$$\mathrm{FM} = \frac{\sum_{i=1}^{N} \mathbb{1}\left(\mathrm{lb}_{\mathrm{TL_i}}^{\mathrm{F}} < L\left(c_i\right) \leq \mathrm{ub}_{\mathrm{TL_i}}^{\mathrm{F}}\right)}{N} \quad (2)$$

where: c_i denotes the i-th response generated by LLM. The function $L(\cdot)$ calculates the word count of the input string. $\mathrm{lb}_{\mathrm{TL_i}}^{\mathrm{P}}$ and $\mathrm{ub}_{\mathrm{TL_i}}^{\mathrm{P}}$ denote the lower and upper bounds of the precise match

Model	Params	*Target Length Generation Task (TLG)*							
		Level:0		*Level:1*		*Level:2*		*All Level*	
		PM	FM	PM	FM	PM	FM	PM	FM
Closed-source Model[1]									
gpt-4-turbo	-	**82.26**	**86.36**	**46.49**	**85.06**	40.72	47.51	61.35	77.35
gpt-4o	-	74.06	79.05	32.32	69.36	36.22	**71.95**	57.75	74.30
gpt-3.5-turbo	-	64.41	69.84	35.06	75.76	38.24	45.93	49.00	66.50
claude-3-haiku	-	48.23	55.21	35.37	73.78	44.12	50.45	43.10	60.25
claude-3.5-sonnet	-	75.17	81.04	42.38	83.08	**62.67**	71.27	**61.65**	**79.55**
Open-source Model									
Mistral	7B	20.29	23.50	16.77	48.32	3.62	5.66	15.45	27.70
Gemma	2B	20.95	23.17	8.69	24.24	0.23	0.23	12.35	18.45
	7B	15.52	18.85	11.74	35.82	0.45	0.45	10.95	20.35
Llama3	8B	34.59	40.02	29.73	65.70	18.10	21.04	29.35	44.25
	70B	**58.76**	**64.52**	**36.59**	**77.90**	**36.43**	**41.18**	**46.55**	**63.75**
InternLM2	7B	6.65	7.21	8.69	27.44	19.68	22.40	10.20	17.20
	20B	8.98	9.87	10.98	34.45	17.42	20.14	11.50	20.20
DeepSeek-LLM	7B	28.16	31.37	17.68	44.36	10.86	13.12	20.90	31.60
	67B	26.94	30.27	17.07	49.54	9.50	11.99	19.85	32.55
Yi-1.5	6B	23.50	25.83	16.46	48.78	18.10	20.36	20.00	32.15
	9B	25.28	29.16	17.38	44.36	24.43	29.41	22.50	34.20
	34B	28.82	33.59	26.07	65.40	21.27	25.79	26.25	42.30
Qwen1.5	7B	24.28	27.38	14.33	46.19	9.05	11.99	17.65	30.15
	14B	28.27	31.49	18.45	43.90	11.09	14.25	21.25	31.75
	32B	32.59	36.25	22.26	49.39	21.49	25.34	26.75	38.15
	72B	35.59	39.69	18.29	49.70	3.85	6.11	22.90	35.55

Table 2: Overall results of different LLMs of *TLG*. All open-source models used are either chat or instruct models. In models belonging to the same series but varying in parameter sizes, those with larger parameters typically exhibit superior performance. The best-performing model in each *Level* is **in-bold**, and the second best is underlined.

range associated with the target length of i-th response. $\mathrm{lb}^{\mathrm{F}}_{\mathrm{TL_i}}$ and $\mathrm{ub}^{\mathrm{F}}_{\mathrm{TL_i}}$ denote the lower and upper bounds of the flexible match range associated with the target length of i-th response.

3.2 Experimental Setup

Dataset. We employ a two-stage data construction method for this study. Initially, we randomly sample 2,000 data from OpenHermes2.5 (Teknium, 2023). To enhance the complexity of the task and prevent data leakage, the second stage involved uses only the questions from these samples. Additionally, we randomly assign one of nine target lengths for the responses. The distribution of target length in the *TLG* dataset is shown in Figure 3. Further details regarding the format of the *TLG* dataset are provided in Appendix A.1.

Models & Prompt Templates. We conduct extensive experiments with both closed and open-source LLMs, specifically the chat or instruct version. The specific models used are listed in Table 8. We evaluate each model using its own prompt template, as detailed in Table 9.

To integrate the target length into the prompt, we modify the sentence `The response should have a word count of {Target Length} words` into each question. For target length >800, we replace this with `more than 800`.

Hardware & Hyperparameters. All experiments are conducted on NVIDIA A100 GPUs. Inference is performed using the `vllm` (Kwon et al., 2023), with `temperature` set to 0 and

[1]The results of all closed-source models are obtained on July 26, 2024.

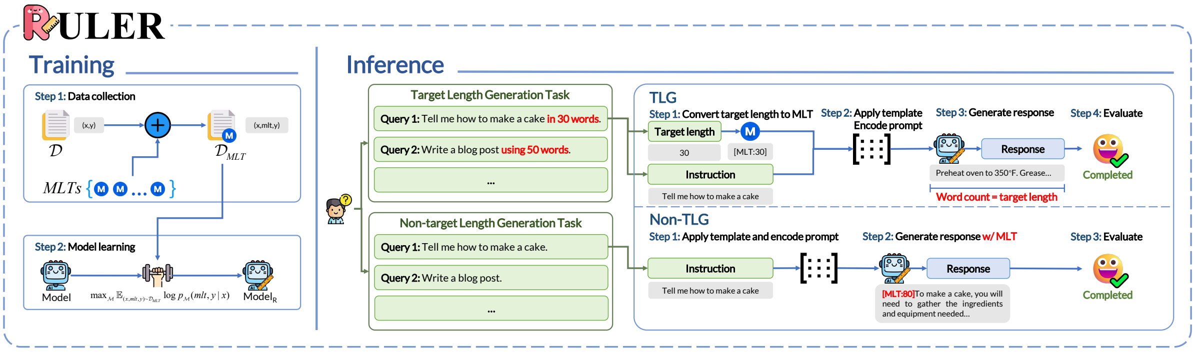

Figure 2: Overview of RULER. The method is divided into two parts: training and inference. The figure illustrates the main content of both sections. Additionally, in the inference section, we show two scenarios: *TLG* and *non-TLG* to show the difference.

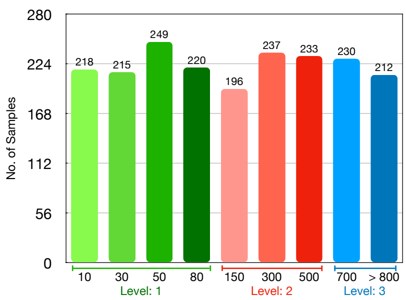

Figure 3: Target length distribution in *TLG* dataset. The count of each target length is approximately 200.

`max_tokens` set to 2,048 in the `SamplingParams`, thereby employing greedy decoding for inference. The `model_max_length` for all models is consistent with their respective configurations, as shown in Table 8.

3.3 Results and Analysis

Table 2 displays the PM and FM scores of open-source models at different *Levels*. Generally, models with advanced capabilities achieve higher PM and FM scores, indicating stronger adherence to instructions. This observation aligns with human expectations.

For most models, scores are lowest at *Level:2*, suggesting significant potential for enhancement in producing longer responses. While, scores at *Level:1* are the highest. This trend may be attributed to the prevalence of shorter responses in the training datasets utilized for model fine-tuning, which influences their generative biases. Desipte potential differences in parameters, a performance gap between closed and open source models remains evident. Notably, claude-3.5-Sonnet achieve the best scores across all models at the *All Level*, with scores of 61.65 and 79.55. Furthermore, the PM and FM scores for each model across various target lengths are detailed in Appendix A.3.

The poor performance in *TLG* can be attributed to a discrepancy between the token counts generated by LLMs and the lengths as understood by humans. The discrepancy between the tokens generated by LLMs and the lengths as understood by humans constirbutes to the issue. This mismatch arises due to several factors:

- **Tokenization Schemes:** LLMs employ subword tokenization schemes that decompose words into smaller units of varying lengths. For example, a single long word might be divided into multiple tokens, complicating the model's ability to equate token counts with human-understood word counts (Gage, 1994).
- **Model Training:** Most LLMs, particularly those trained using autoregressive language modeling, are not explicitly trained with objectives that prioritize output length. As a result, these models often lack strong capabilities for controlling the length of their generated output(Devlin et al., 2019).

4 RULER: Meta Length Token Controlled Generation

In this section, we first introduce RULER, encompassing the design of the *Meta Length Tokens (MLTs)*, the data collection and the learning process associated with the models (§4.1). Subsequently, we detail the difference in the generation of RULER under two scenarios: *TLG* and non-*TLG* (§4.2).

4.1 Method

RULER. We introduce RULER, as illustrated in Figure 2, to effectively control the response length

MLT	Range of Variation	No. in $\mathcal{D}_{MLT}$
[MLT:10]	$[5, 15)$	20,000
[MLT:30]	$[25, 35)$	20,000
[MLT:50]	$[45, 55)$	20,000
[MLT:80]	$[75, 85)$	20,000
[MLT:150]	$[145, 155)$	20,000
[MLT:300]	$[295, 305)$	10,333
[MLT:500]	$[495, 505)$	2,317
[MLT:700]	$[695, 705)$	497
[MLT:>800]	$(800, \infty)$	8,082

Table 3: Meta length tokens in RULER showing their range of variation in data collection and counts in $\mathcal{D}_{MLT}$.

of LLMs using *MLTs*. Ruler employs MLTs to explicitly communicate length requirements within instructions. The *MLTs* represent the model's response length range and aim to enhance its capability on the *TLG* task. Our end-to-end training enables the LLMs to automatically generate *MLTs* in various scenarios, regardless of target length requirements. *MLTs* (Table 3) offer more precise control than traditional text prompt methods, which often prove insufficiently constraining.

Data collection for RULER. For common fine-tuning training datasets, the format typically consist of input-output pairs (x, y). Following Zhou et al. (2023b), we calculate the word count of y for each entry. Based on the predefined *MLTs* in Table 3 and their range of variation, we aim to match each y to a corresponding *mlt* based on its word count. If a match is found, the data is reformatted as (x, mlt, y). This method aids in the construction of the fine-tuning training dataset $\mathcal{D}_{MLT}$, detailed in Algorithm B.

RULER learning. To minimize changes to the model's generation pattern and ensure stability in non-*TLG* scenario, we position the *MLT* immediately before the original response during the construction of fine-tuning data. This strategy maintains the model chat template. Consequently, the combination of mlt and the original response y forms a new complete response y'.

We conduct the training of the RULER $\mathcal{M}$ on the curated corpus $\mathcal{D}_{MLT}$, which is augmented with *Meta Length Tokens* $\mathcal{D}_{MLT}$, employing the standard next token objective:

$$\max_{\mathcal{M}} \mathbb{E}_{(x,mlt,y)\sim\mathcal{D}_{MLT}} \log p_{\mathcal{M}}(mlt, y|x) \qquad (3)$$

We concatenate the *MLT* directly to the beginning of y to compute the loss and use the *MLTs* to expand the original vocabulary $\mathcal{V}$.

4.2 RULER Inference

***TLG* scenario.** In the *Target Length Generation (TLG)* scenario, the user's instruction specifies a target length, decomposed into a question and a target length. The RULER converts this target length into the corresponding *MLT* and appends it to the model chat template. Subsequent to the *MLT*, RULER generates response that aligns with the target length, ensuring compliance with both the user's question and the target length, as illustrated in Figure 2. This approach yields superior results compared to controlling outputs solely through prompts.

non-*TLG* scenario. In the non-*TLG* scenario, users provide straightforward instructions consisting solely of a question. RULER integrates these instructions directly into the model's chat template for generation. Owing to its innovative design and the use of a standard next-token objective in training (Equation 3), RULER autonomously generates a *MLT* prior to producing the textual response. This *MLT* is designed to match the length of the content generated, thereby ensuring normal generation of the model in non-*TLG* scenarios, as illustrated in Figure 2.

5 Experiments

5.1 Experimental Setup

Dataset $\mathcal{D}_{MLT}$. To ensure balanced frequency distribution of each *Meta Length Token (MLT)* in $\mathcal{D}_{MLT}$, we set a maximum occurrence limit of 20,000 for each *MLT*. We construct $\mathcal{D}_{MLT}$ from three datasets: OpenHermes2.5 (excluding data previously used in *TLG*) (Teknium, 2023), LongForm (Köksal et al., 2023), and ELI5 (Fan et al., 2019), in accordance with Algorithm 1. This approach aims to create a diverse dataset, particularly effective for generating longer content that is relatively rare. in total, $\mathcal{D}_{MLT}$ comprises 121,229 entries, with the frequency of each *MLT* in Table 3. Moreover, we calculate the word count for each response in every dataset, allowing us to statistically analyze the *MLT* distribution, as detailed in Table 16.

LLMs. To comprehensively evaluate the performance of RULER across different models, we consider factors such as model size, open-source availability, and overall model performance. We

Model	Target Length Generation Task (TLG)							
	Level:0		*Level:1*		*Level:2*		*All Level*	
	PM	FM	PM	FM	PM	FM	PM	FM
Mistral-7B-Instruct	20.29	23.50	16.77	48.32	3.62	5.66	15.45	27.70
Mistral-7B$_R$	70.18↑49.89	75.06↑51.56	35.52↑18.75	67.84↑19.52	33.71↑30.09	36.43↑30.77	50.75↑35.30	64.15↑36.45
gemma-7b-it	15.52	18.85	11.74	35.82	0.45	0.45	10.95	20.35
gemma-7b$_R$	59.53↑44.01	64.19↑45.34	39.33↑27.59	68.14↑32.32	25.34↑24.89	27.83↑27.38	45.35↑34.40	57.45↑37.10
Llama-3-8B-Instruct	34.59	40.02	29.73	65.70	18.10	21.04	29.35	44.25
Llama-3-8B$_R$	77.27↑42.68	80.71↑40.69	50.76↑21.03	83.84↑18.14	19.23↑1.13	22.85↑1.81	55.75↑26.40	68.95↑24.70
deepseek-llm-7b-chat	28.16	31.37	17.68	44.36	10.86	13.12	20.90	31.60
deepseek-llm-7b$_R$	68.18↑40.02	73.50↑42.13	31.10↑13.42	68.90↑24.54	11.54↑0.68	11.76↓-1.36	43.50↑22.60	58.35↑26.75
Yi-1.5-6B-Chat	23.50	25.83	16.46	48.78	18.10	20.36	20.00	32.15
Yi-1.5-6B$_R$	67.07↑43.57	72.17↑46.34	40.40↑23.94	76.83↑28.05	19.23↑1.13	21.04↑0.68	47.75↑27.75	62.40↑30.25
Qwen1.5-7B-Chat	24.28	27.38	14.33	46.19	9.05	11.99	17.65	30.15
Qwen1.5-7B$_R$	59.09↑34.81	64.41↑37.03	29.88↑15.55	61.28↑15.09	11.54↑2.49	14.25↑2.26	39.00↑21.35	52.30↑22.15

Table 4: Overall results of various LLMs with RULER are presented. Additionally, we also annotate the table with the score changes compared to the chat or instruct model. Consistent improvements in both PM and FM scores are observed across all Levels.

select six LLMs are selected: Mistral-7B-v0.3 (Jiang et al., 2023), gemma-7b (Team et al., 2024), Llama-3-8B (AI@Meta, 2024), deepseek-llm-7b (DeepSeek-AI, 2024), Yi-1.5-6B (AI et al., 2024), and Qwen1.5-7B (Bai et al., 2023). We apply the RULER to these base models and compare the results with their corresponding instruct or chat models.

Evaluation Metric. Consistent with the *TLG* and compared to previous results, we also calculate PM and FM scores to assess the effectiveness of RULER.

5.2 Main Results

Table 4 presents a detailed comparison of PM and FM scores across various LLMs using RULER across different *Levels*. For information on model training see Appendix C.2.

Overall Performance Enhancement. Across all evaluated models, we observe a consistent improvement in both PM and FM scores at all *Levels*. The most significant improvement is observed in gemma-7b$_R$[1], with PM and FM scores increasing by 34.40 and 37.10, respectively. In contrast, the least improvement is noted with PM and FM rising by 21.35 and 22.15. The PM and FM scores across *All Level* showed an average improvement of 27.97 and 29.57. These improvements indicate that RULER effectively enhances the model's ability to generate content of target lengths. This suggests that using *MLT* to control output length is more effective than using prompts, as the model learns to generate content of corresponding lengths during fine-tuning. Additionally, RULER's ability to enhance various models demonstrates its generalizability and scalability.

Different *Level* Analysis. At *Level:0*, all models show significant improvements in both PM and FM scores. Compared to other *Level*, each model achieves the highest PM and FM score improvements at *Level:0*. This enhancement occurs because the models are capable of generating responses of this length; however, their coarse length control impedes precise adherence to target length requirements. Our method significantly improves the models' capacity to accurately control content length at *Level:0* more accurately, better meeting the target length requirements.

Moving to *Level:1*, while the improvements are not as pronounced as at *Level:0*, the models still exhibit significant gains in both PM and FM scores. At *Level:2*, the extent of score improvements varies across models. For instance, Mistral-7B-v0.3$_R$ and gemma-7b$_R$ continue to show substantial score increases. Despite these positive trends, only deepseek-llm-7b-chat$_R$, show a slight decrease in scores at *Level:2*. This is attributed to the insufficient data for *Level:2* in $\mathcal{D}_{MLT}$. The uneven distribution of data likely contributes to the slight decrease in scores.

[1]Model name with $_R$ means base model with RULER

Model	FM of Different Target Length									Avg FM
	10	30	50	80	150	300	500	700	>800	
Mistral-7B-Instruct-v0.3	0.5	0.0	0.5	2.0	18.5	50.5	20.5	3.0	2.5	10.89
Mistral-7B-v0.3$_{\text{R}}$	72.5	68.0	65.5	76.5	76.0	63.0	28.0	24.0	64.5	59.78
gemma-7b-it	13.0	17.0	15.5	26.0	54.5	76.5	17.5	0.0	0.0	24.44
gemma-7b$_{\text{R}}$	58.0	63.5	61.0	69.5	72.5	64.0	42.0	17.0	67.0	57.17
Llama-3-8B-Instruct	23.5	18.0	12.5	28.0	50.5	76.5	57.0	25.5	30.5	35.78
Llama-3-8B$_{\text{R}}$	84.0	84.0	73.0	80.0	87.5	89.5	71.0	14.5	36.5	68.89
deepseek-llm-7b-chat	36.5	16.0	12.5	17.5	23.5	60.5	36.5	16.0	22.5	26.83
deepseek-llm-7b$_{\text{R}}$	64.0	70.0	62.5	73.0	82.0	86.5	27.0	17.0	40.5	58.06
Yi-1.5-6B-Chat	26.5	16.5	14.5	14.5	18.5	42.5	35.0	33.5	28.5	25.56
Yi-1.5-6B$_{\text{R}}$	80.5	66.0	67.0	77.0	83.5	83.5	56.0	22.0	39.5	63.89
Qwen1.5-7B-Chat	13.5	17.0	9.5	16.0	6.5	51.0	57.5	22.5	4.5	22.00
Qwen1.5-7B$_{\text{R}}$	69.0	61.0	46.5	68.5	81.0	80.5	38.5	16.5	36.5	55.33

Table 5: Results in multi *MLT* generation experiment. Generally, the FM scores obtained via RULER surpass those of the baseline models.

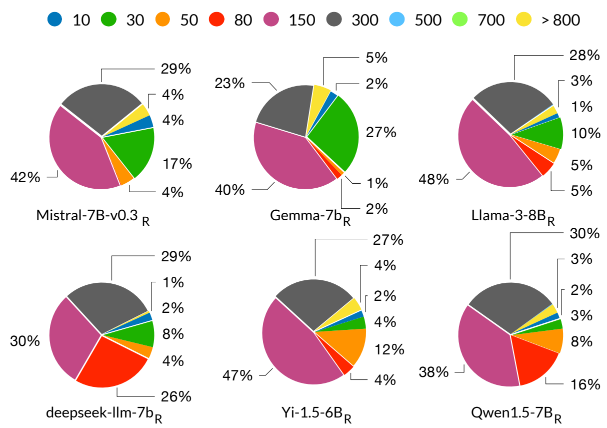

Figure 4: Distribution of *MLTs* generated by RULER in self-generated *MLT* experiment. The models demonstrate a preference for generating responses with lengths of 150 and 300.

Model	FM	Avg WC
Mistral-7B-v0.3$_{\text{R}}$	73.40	279
gemma-7b$_{\text{R}}$	69.00	347
Llama-3-8B$_{\text{R}}$	88.40	215
deepseek-llm-7b$_{\text{R}}$	84.40	187
Yi-1.5-6B$_{\text{R}}$	81.40	236
Qwen1.5-7B$_{\text{R}}$	81.60	245

Table 6: The FM score and average word count of RULER with models in self-generated *MLT* experiment. FM scores are notably high. Specifically, gemma-7b$_{\text{R}}$ recorded the lowest at 69.00, while Llama-3-8B$_{\text{R}}$ achieved the highest at 88.40.

5.3 Do *MLTs* actually influence the length of the generated content?

To further investigate the effectiveness and scalability of *MLTs*, we designed two additional experiments: multi *MLT* generation experiment and self-generated *MLT* experiment.

Multi *MLT* Generation Experiment. To further validate the efficacy and robustness of RULER, we assess its ability to control response length. We randomly sample 200 entries from Arena-Hard-Auto (Li et al., 2024a) and subject each to all target lengths (Table 1), culminating in 1,800 entries at last. Subsequently, we calculate the FM scores for each target length, using the original model as a baseline.

The results presented in Table 5 highlight the enhancements in model performance due to RULER. The FM scores achieved by RULER generally surpass those of the baseline models. Notably, even the well-performing Llama-3-8B$_{\text{R}}$ shows significant improvements. However, when the target length is 700, RULER shows a decline in FM if the baseline model already achieves a certain score. In contrast, RULER enhances performance if the baseline model is underperforming. This phenomenon is likely due to an imbalance in the $\mathcal{D}_{MLT}$, where responses of 700 words are infrequent and differ from the fine-tuning data of the baseline, potentially undermining performance. Overall, RULER

Model	Type	ARC (chanllenge/easy)	HellaSwag	TruthfulQA	MMLU	Winogrande	GSM8K
Mistral-7B-v0.3	vanilla	38.23/67.76	48.57	46.02	34.94	62.04	26.46
-	RULER	37.97/67.85	47.83	47.12	37.88	62.83	27.52
gemma-7b	vanilla	35.75/65.66	45.95	41.13	32.44	57.14	23.58
-	RULER	38.99/67.47	45.40	45.65	31.67	60.30	25.93
Meta-Llama-3-8B	vanilla	48.63/77.48	58.89	51.41	50.91	71.74	44.96
-	RULER	49.23/77.99	59.12	51.90	50.16	71.19	46.63
deepseek-llm-7b-base	vanilla	50.94/79.92	61.48	39.90	48.65	72.93	38.89
-	RULER	51.37/79.55	61.31	38.43	48.81	72.77	37.15
Yi-1.5-6B	vanilla	51.62/79.25	58.79	55.32	54.68	68.51	52.01
-	RULER	51.28/79.46	58.41	49.94	55.13	68.11	50.34
Qwen1.5-7B	vanilla	46.67/77.53	56.39	53.98	54.00	65.98	44.88
-	RULER	47.27/76.68	56.46	50.18	54.59	65.19	47.01

Table 7: Comparison of the overall performance of six models with RULER or vanilla, with scores computed on ARC, HellaSwag, TruthfulQA, MMLU, Winogrande and GSM8K. The overall performance of models using RULER generally remains consistent with the base models with sft.

significantly improves model performance.

Self-generated *MLT* Experiment. To validate RULER in generating *MLT* and responses under a non-*TLG* scenario, we use the Arena-Hard-Auto dataset without providing *MLTs*, thereby necessitating autonomous response generation by the model. We evaluate performance by cataloging the types and proportions of generated *MLTs* (Figure 4) and evaluating response length using FM score at the target lengths corresponding to the *MLTs* (Table 6).

Models show a preference for producing responses with target lengths of 150 and 300. This inclination is likely attributable to the complex nature of the queries in the Arena-Hard-Auto, which require longer responses for problem resolution. In the non-*TLG* scenario, the FM scores are notably high, with the Mistral-7B-v0.3$_{\mathbf{R}}$ recording the lowest at 73.40 and Llama-3-8B$_{\mathbf{R}}$ achieving the highest at 88.40. The word count across all models varies from 187 words to 347 words.

5.4 Evaluation on Overall Performance

To evaluate the impact of RULER on overall performance, we conduct experiments utilizing six benchmark datasets: ARC (Clark et al., 2018), HellaSwag (Zellers et al., 2019), TruthfulQA (Lin et al., 2022), MMLU (Hendrycks et al., 2021), Winogrande (Sakaguchi et al., 2019) and GSM8K (Cobbe et al., 2021). These benchmarks provide a comprehensive assessment across different task types. `lm-evaluation-harness` (Gao et al., 2024) is employed to assess the overall performance. Further details about the experiments on the experiment can be found in Appendix C.4.

Table 7 illustrates that RULER marginally reduces performance on several tasks. Overall performance of models using Ruler generally remains consistent with the original models. The variations in scores are minimal, with changes within a very small range. Moreover, we observe that some models with Ruler actually show improvements in specific tasks. These improvements suggest that Ruler may contribute positively under certain conditions or in certain task types. This indicates that RULER can significantly enhance the model's ability to follow length-based instructions without compromising its performance on the same data.

6 Conclusion

This study initially investigate the instruction following abilities of LLMs and introduces *Target Length Generation Task (TLG)*. Additionally, we propose RULER, a novel and model-angnostic method that controls generated length for LLMs. RULER utilizes the *MLT* and end-to-end training to enhance model performance. Experimental results demonstrate that substantial improvements in PM and FM scores across various models. Moreover, two additional experiments are conducted to further validate the efficacy of the proposed method. Finally, we assess overall performance across six different benchmarks to demonstrate its superiority.

Limitations

With the emergence of large language models (LLMs), an increasing number of applications are now utilizing LLMs. A particularly interesting aspect is the instruction-following capabilities of

LLMs. In this paper, we analyze the capabilities of LLMs solely from the perspective of controlling generated length and propose a solution through RULER. Instructions, which vary widely and represent a real-life scenario or application. We believe addressing the challenges or solving widespread issues across various instructions is crucial. We employ meta token to construct RULER and argue that meta tokens offer more robust control over models than prompts do. Exploring how to develop and utilize models effectively with the help of tokens is a profoundly important question.

Ethical Statements

This study concentrates on managing the output length of Large Language Models (LLMs). While our primary focus is on the length of generated content, we have not assessed the potential for producing toxic content. The research does not involve human participants, nor does it handle personal or sensitive information. We have used only open-source or suitably licensed resources, thereby complying with relevant standards. Additionally, all training data employed are open-source, ensuring the exclusion of any private or sensitive information.

Acknowledgements

This work was supported by National Key Research and Development Program of China (2022YFF0902100), National Natural Science Foundation of China (Grant No. 62376262), the Natural Science Foundation of Guangdong Province of China (2024A1515030166), Shenzhen Science and Technology Innovation Program (KQTD20190929172835662), Shenzhen Basic Research Foundation (JCYJ20210324115614039).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open foundation models by 01.ai.

AI@Meta. 2024. Llama 3 model card.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yihan Cao, Yanbin Kang, Chi Wang, and Lichao Sun. 2024. Instruction mining: Instruction data selection for tuning large language models.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2022. Meta-learning via language model in-context tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 719–730, Dublin, Ireland. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

DeepSeek-AI. 2024. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. In *Proceedings of ACL 2019*.

Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal archive*, 12:23–38.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. A framework for few-shot language model evaluation.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1049–1065, Toronto, Canada. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Yichen Jiang, Marco Vecchio, Mohit Bansal, and Anders Johannsen. 2024. Hierarchical and dynamic prompt compression for efficient zero-shot API usage. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 2162–2174, St. Julian's, Malta. Association for Computational Linguistics.

Zachary Kenton, Tom Everitt, Laura Weidinger, Iason Gabriel, Vladimir Mikulik, and Geoffrey Irving. 2021. Alignment of language agents.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Abdullatif Köksal, Timo Schick, Anna Korhonen, and Hinrich Schütze. 2023. Longform: Effective instruction tuning with reverse instructions.

Yoav Levine, Noam Wies, Daniel Jannai, Dan Navon, Yedid Hoshen, and Amnon Shashua. 2022. The inductive bias of in-context learning: Rethinking pretraining example design. In *International Conference on Learning Representations*.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. 2024a. From live data to high-quality benchmarks: The arena-hard pipeline.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. 2023. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6342–6353, Singapore. Association for Computational Linguistics.

Yunshui Li, Binyuan Hui, Xiaobo Xia, Jiaxi Yang, Min Yang, Lei Zhang, Shuzheng Si, Ling-Hao Chen, Junhao Liu, Tongliang Liu, Fei Huang, and Yongbin Li. 2024b. One-shot learning as instruction data prospector for large language models.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Junyi Liu, Liangzhi Li, Tong Xiang, Bowen Wang, and Yiming Qian. 2023. TCRA-LLM: Token compression retrieval augmented large language model for inference cost reduction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9796–9810, Singapore. Association for Computational Linguistics.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations*.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. 2023. Learning to compress prompts with gist tokens. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open models based on gemini research and technology.

Teknium. 2023. Openhermes 2.5: An open dataset of synthetic data for generalist llm assistants.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. 2024. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2024. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems*, 36.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022a. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022b. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022c. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Lei Zhang, Yunshui Li, Ziqiang Liu, Jiaxi yang, Junhao Liu, and Min Yang. 2023a. Marathon: A race through the realm of long context with large language models.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. 2024. Soaring from 4k to 400k: Extending llm's context with activation beacon. *arXiv preprint arXiv:2401.03462*.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023b. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pages 12697–12706. PMLR.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023a. LIMA: Less is more for alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023b. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

A Target Length Generation Task Deatils

In this section, we present the experimental details of the *Target Length Generation (TLG)*.

A.1 *TLG* Dataset

Dataset constructed for the *TLG*, totaling 2,000 entries.

TLG Dataset

{
"id":"0"
"Instruction":"How can I generate an AI model that can classify articles of clothing as shorts, skirts, or pants based on their descriptions?",
"TargetLength":"50"
}
[...]
{
"id":"1999"
"Instruction":"You will be given several pieces of information about someone, and you will have to answer a question based on the information given.\nJohn is taller than Bill. Mary is shorter than John. Question: Who is the tallest person?",
"TargetLength":"30"
}

A.2 Models & Prompt Templates

In this appendix, we list the models in the *TLG*, including their fullname, params, context length and vocab size. All models are downloaderd from Huggingface[2] and inference is executed using `vllm` (Kwon et al., 2023).

Model	**Model Full Name**	**Params**	**Context Length**	**Vocab Size**
Mistral	Mistral-7B-Instruct-v0.3	7B	32,768	32,768
Gemma	gemma-2b-it	2B	8,192	256,000
	gemma-7b-it	7B	8,192	256,000
Llama3	Meta-Llama-3-8B-Instruct	8B	8,192	128,256
	Meta-Llama-3-70B-Instruct	70B	8,192	128,256
InternLM2	InternLM2-Chat-7B	7B	32,768	92,544
	InternLM2-Chat-20B	20B	32,768	92,544
DeepSeek-LLM	deepseek-llm-7b-chat	7B	4,096	102,400
	deepseek-llm-67b-chat	67B	4,096	102,400
Yi-1.5	Yi-1.5-6B-Chat	6B	4,096	64,000
	Yi-1.5-9B-Chat	9B	4,096	64,000
	Yi-1.5-34B-Chat	34B	4,096	64,000
Qwen1.5	Qwen1.5-7B-Chat	7B	32,768	151,936
	Qwen1.5-14B-Chat	14B	32,768	151,936
	Qwen1.5-32B-Chat	32B	32,768	151,936
	Qwen1.5-72B-Chat	72B	32,768	151,936

Table 8: All models used in *TLG*

[2]https://huggingface.co/

Model	Prompt Template	Eos Tokens
Mistral	<s>[INST] {Instruction} [/INST]	</s>
Gemma	<bos><start_of_turn>user\n{Instruction} <end_of_turn>\n<start_of_turn>model\n	<eos>
Llama3	<|begin_of_text|><|start_header_id|>user <|end_header_id|>\n\n{Instruction}<|eot_id|> <|start_header_id|>assistant<|end_header_id|>\n\n	<|end_of_text|>,<|eot_id|>
InternLM2	<s><|im_start|>user\n{Instruction} <|im_end|>\n<|im_start|>assistant\n	</s>, <|im_end|>
DeepSeek-LLM	<|begin_of_sentence|>User: {Instruction} \n\nAssistant:	<|end_of_sentence|>
Yi-1.5	<|im_start|>user\n{Instruction}<|im_end|> \n<|im_start|>assistant\n	<|im_end|>,<|endoftext|>
Qwen1.5	<|im_start|>system\nYou are a helpful assistant. <|im_end|>\n<|im_start|>user\n{Instruction} <|im_end|>\n<|im_start|>assistant\n	<|im_end|>, <|endoftext|>

Table 9: Prompt templates and Eos tokens for all models used in *TLG*.

A.3 Results on Different Target Length

Here, we present the FM and PM scores of the models at all target lengths.

A.3.1 *Level:0*

The PM and FM scores for each model at *Level:0* are shown in Table 11 and Table 10.

Model	Params	*Level:0*							
		10		30		50		80	
		PM	FM	PM	FM	PM	FM	PM	FM
Mistral	7B	30.73	30.73	18.60	18.60	16.87	16.87	15.45	28.64
Gemma	2B	21.56	21.56	30.23	30.23	20.88	20.88	11.36	20.45
	7B	12.39	12.39	18.14	18.14	18.88	18.88	12.27	25.91
Llama3	8B	45.41	45.41	35.35	35.35	33.73	33.73	24.09	46.36
	70B	**60.55**	**60.55**	**66.05**	**66.05**	**61.45**	**61.45**	**46.82**	**70.45**
InternLM2	7B	17.89	17.89	6.98	6.98	1.20	1.20	1.36	3.64
	20B	20.64	20.64	8.84	8.84	2.81	2.81	4.55	8.18
DeepSeek-LLM	7B	58.26	58.26	25.12	25.12	17.67	17.67	13.18	26.36
	67B	46.79	46.79	20.47	20.47	22.09	22.09	19.09	32.73
Yi-1.5	6B	39.91	39.91	23.72	23.72	20.08	20.08	10.91	20.45
	9B	47.71	47.71	23.72	23.72	17.27	17.27	13.64	29.55
	34B	45.41	45.41	27.44	27.44	20.48	20.48	23.18	42.73
Qwen1.5	7B	31.19	31.19	25.58	25.58	22.89	22.89	17.73	30.45
	14B	45.87	45.87	28.84	28.84	26.51	26.51	12.27	25.45
	32B	46.79	46.79	33.95	33.95	29.32	29.32	20.91	35.91
	72B	39.45	39.45	41.86	41.86	32.53	32.53	29.09	45.91

Table 10: Results of open-source models of *TLG* at *Level:0*. The best-performing model in each target length is **in-bold**, and the second best is underlined.

A.3.2 *Level:1*

The PM and FM scores for each model at *Level:1* are shown in Table 12 and Table 13.

Model	Params	Level:0							
		10		30		50		80	
		PM	FM	PM	FM	PM	FM	PM	FM
gpt-4-turbo	-	**89.45**	**89.45**	**86.98**	**86.98**	**82.33**	**82.33**	**70.45**	87.27
gpt-4o	-	83.49	83.49	80.47	80.47	71.08	71.08	61.82	82.27
gpt-3.5-turbo	-	80.73	80.73	72.09	72.09	57.43	57.43	48.64	70.91
claude-3-haiku	-	69.27	69.27	54.42	54.42	42.17	42.17	28.18	56.82
claude-3.5-sonnet	-	82.57	82.57	74.42	74.42	75.50	75.50	68.18	**92.27**

Table 11: Results of closed-source models of *TLG* at *Level:0*. The best-performing model in each target length is **in-bold**, and the second best is underlined.

Model	Params	Level:1					
		150		300		500	
		PM	FM	PM	FM	PM	FM
Mistral	7B	17.86	41.84	14.77	70.04	17.94	30.94
Gemma	2B	17.35	32.65	7.17	33.33	2.69	7.17
	7B	18.88	42.35	12.24	51.90	4.93	13.00
Llama3	8B	38.27	70.92	**27.00**	78.90	25.11	47.09
	70B	**55.10**	**85.71**	22.36	**88.61**	35.43	**59.64**
InternLM2	7B	9.18	20.92	5.91	37.55	11.21	22.42
	20B	9.69	22.96	9.28	45.99	13.90	32.29
DeepSeek-LLM	7B	15.31	37.24	18.14	60.76	19.28	33.18
	67B	9.18	34.69	19.83	71.73	21.08	39.01
Yi-1.5	6B	18.88	46.94	12.66	62.45	18.39	35.87
	9B	12.76	33.16	12.66	53.59	26.46	44.39
	34B	25.51	58.67	24.05	78.48	28.70	57.40
Qwen1.5	7B	9.69	29.59	7.17	61.60	26.01	44.39
	14B	5.61	16.84	10.97	56.12	**37.67**	54.71
	32B	20.92	43.37	14.77	53.59	31.39	50.22
	72B	13.27	35.20	12.66	64.98	28.70	46.19

Table 12: Results of open-source models of *TLG* at *Level:1*. The best-performing model in each target length is **in-bold**, and the second best is underlined.

Model	Params	Level:1					
		150		300		500	
		PM	FM	PM	FM	PM	FM
gpt-4-turbo	-	68.37	93.88	22.36	83.97	**52.91**	**78.48**
gpt-4o	-	60.20	88.78	15.61	71.31	25.56	50.22
gpt-3.5-turbo	-	54.08	79.59	15.19	81.86	39.46	65.92
claude-3-haiku	-	43.88	76.02	19.41	75.95	44.84	69.51
claude-3.5-sonnet	-	**76.53**	**97.45**	**24.05**	**88.61**	31.84	64.57

Table 13: Results of closed-source models of *TLG* at *Level:1*. The best-performing model in each target length is **in-bold**, and the second best is underlined.

A.3.3 *Level:2*

The PM and FM scores for each model at *Level:2* are shown in Table 14 and Table 15.

Model	Params	*Level:2* 700 PM	700 FM	>800 PM	>800 FM
Mistral	7B	3.04	6.96	4.25	4.25
Gemma	2B	0.00	0.00	0.47	0.47
	7B	0.87	0.87	0.00	0.00
Llama3	8B	16.09	21.74	20.28	20.28
	70B	<u>24.35</u>	**33.48**	**49.53**	**49.53**
InternLM2	7B	18.70	23.91	20.75	20.75
	20B	17.39	22.61	17.45	17.45
DeepSeek-LLM	7B	9.13	13.48	12.74	12.74
	67B	9.13	13.91	9.91	9.91
Yi-1.5	6B	12.61	16.96	24.06	24.06
	9B	22.17	<u>31.74</u>	<u>26.89</u>	<u>26.89</u>
	34B	22.17	30.87	20.28	20.28
Qwen1.5	7B	12.17	17.83	5.66	5.66
	14B	15.22	21.30	6.60	6.60
	32B	**23.91**	31.30	18.87	18.87
	72B	6.09	10.43	1.42	1.42

Table 14: Results of open-source models of *TLG* at *Level:2*. The best-performing model in each target length is **in-bold**, and the second best is <u>underlined</u>.

Model	Params	*Level:2* 700 PM	700 FM	>800 PM	>800 FM
gpt-4-turbo	-	**49.57**	<u>62.61</u>	31.13	31.13
gpt-4o	-	<u>46.09</u>	**64.78**	<u>79.72</u>	<u>79.72</u>
gpt-3.5-turbo	-	35.65	50.43	41.04	41.04
claude-3-haiku	-	39.57	51.74	49.06	49.06
claude-3.5-sonnet	-	36.52	53.04	**91.04**	**91.04**

Table 15: Results of closed-source models of *TLG* at *Level:2*. The best-performing model in each target length is **in-bold**, and the second best is <u>underlined</u>.

B $\mathcal{D}_{MLT}$ Data Creation

C Experiments Details

C.1 *MLT* in Datasets

To obtain data with varying response lengths for composing $\mathcal{D}_{MLT}$, particularly those responses exceeding 500, we integrateg data from OpenHermes2.5 (Teknium, 2023), LongForm (Köksal et al., 2023) and ELI5 (Fan et al., 2019). We calculate the word count for each response in every dataset, allowing us to statistically analyze the *MLT* distribution, shown in Table 16.

Algorithm 1 $\mathcal{D}_{MLT}$ Data Creation

Require: Word count function $L(\cdot)$, meta length tokens $MLTs = \{MLT_0, MLT_1, \cdots\}$
Input: Initial dataset $\mathcal{D}$
Output: $\mathcal{D}_{MLT}$

$\mathcal{D}_{MLT} \leftarrow \{\}$
for each tuple (x, y) in D **do**
  $mlt \leftarrow$ None
  for each MLT in $MLTs$ **do**
    if $L(y) > \mathrm{lb}_{MLT}$ **and** $L(y) \leq \mathrm{ub}_{MLT}$ **then**
      $mlt \leftarrow MLT$
      break
    end if
  end for
  if mlt is not None **then**
    $\mathcal{D}_{MLT} \leftarrow \mathcal{D}_{MLT} \cup \{(x, mlt, y)\}$
  end if
end for
return $\mathcal{D}_{MLT}$

MLT	**OpenHermes2.5** (Teknium, 2023)	**LongForm** (Köksal et al., 2023)	**ELI5** (Fan et al., 2019)
[MLT:10]	28,552	586	3,280
[MLT:30]	16,860	1,428	14,143
[MLT:50]	18,867	1,236	17,597
[MLT:80]	18,014	852	15,926
[MLT:150]	37,515	1,037	19,103
[MLT:300]	7,526	252	2,555
[MLT:500]	1,495	140	682
[MLT:700]	193	101	203
[MLT:800]	1,809	2,465	3,808

Table 16: *MLT* distribution in each dataset. The OpenHermes2.5 excludes the data utilized in *TLG*. The LongForm and ELI5 employs its training, validation, and test sets simultaneously. When multiple answers are available in the dataset, the longest answer is selected as the final response.

C.2 More Details of Training

More details of training. We use 4*A100 with 80GB Nvidia GPUs to train the models. The training utilizes both bf16 and tensor tf32 precision formats. The per-device training batch size is set to 4, with gradient accumulation is 8 steps. A cosine learning rate scheduler is applied, starting with an initial learning rate of 2e-5 and a warmup ratio of 0.05. All models are trained for 3 epochs. Additionally, log is set to print every 5 steps.

Loss. We document the changes in training loss for all models, as shown in Figure 5.

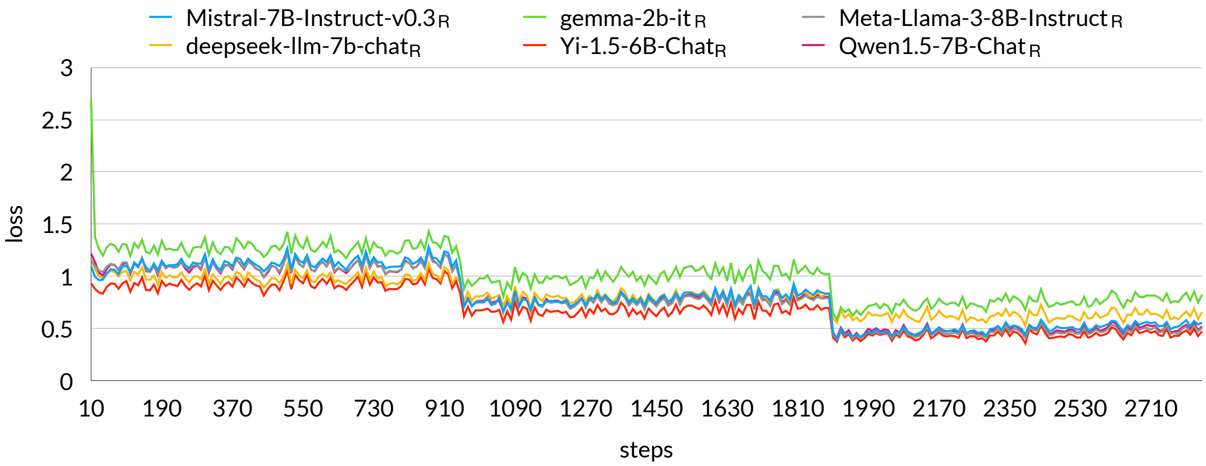

Figure 5: Training loss for models.

C.3 Multi *MLT* generation experiment

Here is the results in multi *MLT* generation experiment.

C.4 More Details of Other Tasks

We tested the RULERon six benchmarks (ARC (Clark et al., 2018), HellaSwag (Zellers et al., 2019), TruthfulQA (Lin et al., 2022), MMLU (Hendrycks et al., 2021), Winogrande (Sakaguchi et al., 2019) and GSM8K (Cobbe et al., 2021)) to examine whether the performance of the fine-tuned models varies on different tasks. We employ 25-shot in ARC, 10-shot setting in Hellaswag, 5-shot setting in MMLU, 0-shot setting in TruthfulQA, 5-shot setting in Winogrande and 5-shot in GSM8K.