Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI-25)
Demonstrations Track

PCToolkit: A Unified Plug-and-Play Prompt Compression Toolkit of Large Language Models

Zheng Zhang[1] , **Jinyi Li**[2] , **Yihuai Lan**[1] , **Xiang Wang**[3] , **Hao Wang**[1*]
[1]The Hong Kong University of Science and Technology (Guangzhou)
[2]South China University of Technology
[3]University of Science and Technology of China
zzhang302@connect.hkust-gz.edu.cn, haowang@hkust-gz.edu.cn

Abstract

Prompt engineering enables Large Language Models (LLMs) to perform a variety of tasks. However, lengthy prompts significantly increase computational complexity and economic costs. To address this issue, prompt compression reduces prompt length while maintaining LLM response quality. To support rapid implementation and standardization, we present the Prompt Compression Toolkit (PCToolkit), a unified plug-and-play framework for LLM prompt compression. PCToolkit integrates state-of-the-art compression algorithms, benchmark datasets, and evaluation metrics, enabling systematic performance analysis. Its modular architecture simplifies customization, offering portable interfaces for seamless incorporation of new datasets, metrics, and compression methods. Our code is available at https://github.com/3DAgentWorld/Toolkit-for-Prompt-Compression. Our demo is at https://huggingface.co/spaces/CjangCjengh/Prompt-Compression-Toolbox.

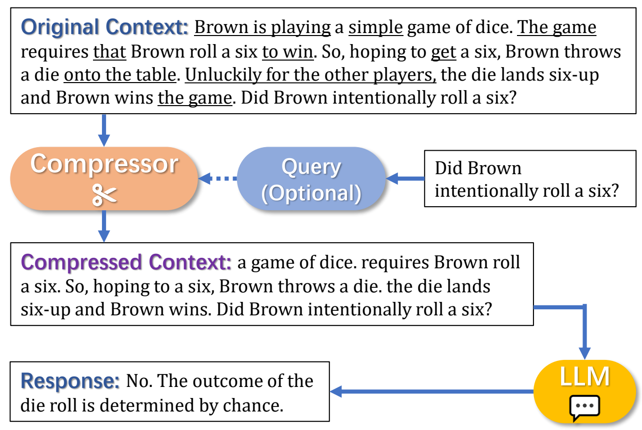


Figure 1: **Illustration of prompt compression.** The original context is distilled into a more concise form while preserving pertinent information for LLMs to process. Some methods compress the context based on the query, while others do not. Words that are underlined in the original text denote the segments that are trimmed by the compressor.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable generalization capabilities [Grosse *et al.*, 2023; Yang *et al.*, 2024], allowing them to adapt to a wide range of tasks through prompt engineering techniques such as CoT [Wei *et al.*, 2024], ICL [Dong *et al.*, 2024], and RAG [Lewis *et al.*, 2020] without necessitating fine-tuning. However, this advantage comes with an obvious drawback: increasing the length of prompts to encompass the necessary information, which subsequently escalates computational overhead [Wang *et al.*, 2024]. Also, for online models such as ChatGPT and Claude, lengthy prompts inflate the economic cost associated with API calls.

To address this issue, prompt compression is the most straightforward strategy. As illustrated in Figure 1, it aims to reduce the length of prompts while retaining the essential information. However, the deployment of prompt compression methods varies between different approaches. There is not yet a general toolkit that can invoke multiple types of compressors. Thus, with the aim of providing plug-and-play services, easily customizable interfaces, and supporting common datasets and metrics, we propose Prompt Compression Toolkit (PCToolkit), a unified plug-and-play toolkit for Prompt Compression of LLMs, making prompt compression methods accessible and portable to a wider audience. Our plug-and-play design enables users to deploy and use the toolkit without any further model training. Meanwhile, users are also able to plug in their custom-trained models in PCToolkit.

Key features of PCToolkit include:

(i) **Reproducible methods.** PCToolkit offers a unified interface for six different compressors: KiS [Laban *et al.*, 2021], SCRL [Ghalandari *et al.*, 2022], Selective Context [Li *et al.*, 2023], LLMLingua [Jiang *et al.*, 2023], LongLLMLingua [Jiang *et al.*, 2024], and LLMLingua-2 [Pan *et al.*, 2024].

(ii) **Modular design.** Featuring a modular structure that simplifies the transition between different methods, datasets, and metrics, PCToolkit is organized into four distinct modules: Compressors, Datasets, Metrics and Runner.

*Corresponding author.

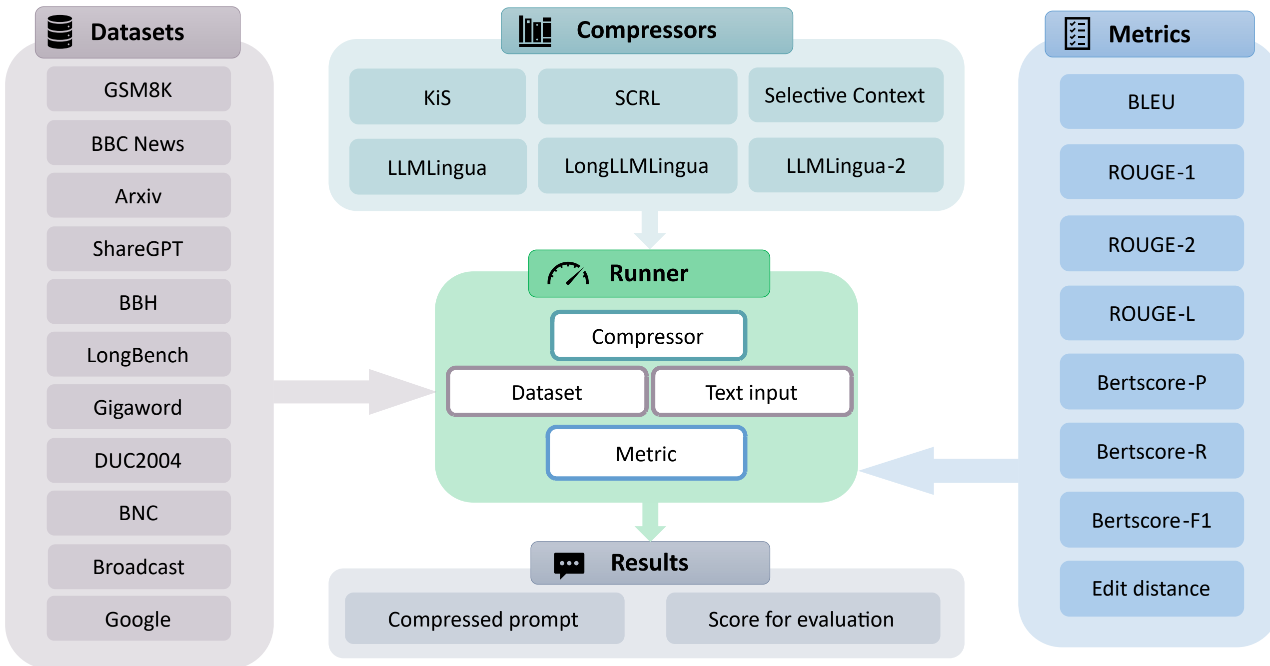


Figure 2: **Architecture of PCToolkit.** The *compressors* module encompasses prompt compression methods that can be accessed through a unified interface with customizable parameters. The *datasets* module includes diverse datasets. The *metrics* module comprises primary metrics utilized for evaluating the performance of compressors. The *runner* module offers a generalized interface for executing evaluations or simply retrieving the compressed prompt generated by the compressors.

(iii) **User-friendly interface.** Facilitating portability and ease of adaptation to different environments, the interfaces within PCToolkit are designed to be easily customizable.

2 PCToolkit

2.1 Modular Design

As shown in Figure 2, PCToolkit is designed with a modular architecture, consisting of Compressors, Datasets, Metrics and Runner.

Compressors. `pctoolkit.compressors` module encompasses six compression methods tailored for prompt optimization. All compressors can be invoked through a unified interface shown in Section 2.2. Figure 3 divides them into three categories: (1) *RL-based:* KiS, SCRL, (2) *LLM scoring-based:* Selective Context, and (3) *LLM annotation-based:* LLMLingua, LongLLMLingua, LLMLingua-2. Among them, KiS does not typically trim words but uses an autoregressive approach to regenerate a shorter context.

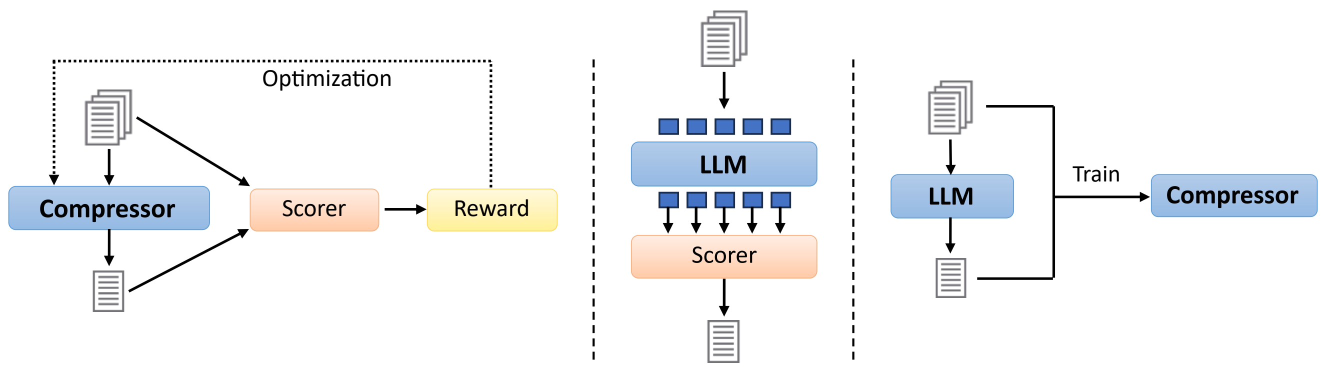
Datasets. `pctoolkit.datasets` module includes a diverse collection of datasets, each curated to cover a wide array of natural language tasks. As shown in Table 1, the datasets are systematically organized by task requirements. For instance, reconstruction tasks leverage domain-specific corpora like BBC and Arxiv, while complex reasoning tasks utilize mathematical benchmarks like GSM8K. From tasks like reconstruction, summarization, question answering, to more specialized domains such as code completion and lies recognition, PCToolkit offers a comprehensive testing ground for assessing prompt compression techniques.

Task	Datasets
Reconstruction	BBC, ShareGPT, Arxiv, GSM8K
Mathematical Problems	GSM8K, BBH
Boolean Expressions	BBH
Multiple Choice	BBH
Lie Detection	BBH
Summarization	BBC, Arxiv, Gigaword, DUC2004, BNC, Broadcast, Google, LongBench
Question Answering	BBH, LongBench
Few-Shot Learning	LongBench
Synthetic Tasks	LongBench
Code Completion	LongBench

Table 1: **Task-dataset mapping in PCToolkit.** The table illustrates a structured breakdown of supported NLP tasks and their corresponding evaluation datasets across reconstruction, reasoning, and generation paradigms.

Metrics. `pctoolkit.metrics` module quantifies the performance of the compression methods across different tasks. Key metrics include accuracy, BLEU [Papineni *et al.*, 2002], ROUGE [Lin, 2004], BERTScore [Zhang* *et al.*, 2020], Token-F1 [Bai *et al.*, 2024], and edit-distance. All necessary metrics can be easily organized into a list, which instructs the Runner on what to measure. As detailed in Table 2, task requirements dictate metric selection: accuracy dominates mathematical and reasoning evaluations (GSM8K, BBH), while text generation tasks (BBC, Arxiv) employ composite metrics including ROUGE and BERTScore.

Runners. `pctoolkit.runners` module serves as the engine that drives the evaluation process. Users can seam-



(a) Reinforcement Learning (RL) (b) LLM Scoring (c) LLM Annotation

Figure 3: **Categories of prompt compression methods.** These methods can be grouped into three main categories: (a) RL-based methods, which use heuristic rewards to optimize the compressor, (b) LLM scoring-based methods, which use another language model to score each token in a single autoregressive step and decide to keep or discard each token based on its score, and (c) LLM annotation-based methods, which use LLMs to annotate data for training a small model specifically designed for prompt compression.

Dataset	Metrics
BBH	Accuracy
Gigaword	ROUGE, Token-F1
BNC	ROUGE, Token-F1
DUC2004	ROUGE, Token-F1
Broadcast	ROUGE, Token-F1
Google	ROUGE, Token-F1
GSM8K	Accuracy, BLEU, ROUGE, BERTScore
BBC News	BLEU, ROUGE, BERTScore
Arxiv articles	BLEU, ROUGE, BERTScore
ShareGPT	BLEU, ROUGE, BERTScore
LongBench	Accuracy, BLEU, ROUGE, BERTScore, Edit-distance

Table 2: **Dataset-metric mapping in PCToolkit.** The table presents correspondences between evaluation datasets and their specialized metrics, emphasizing accuracy for reasoning tasks (e.g., BBH) and text generation metrics (e.g., ROUGE) for summarization.

lessly execute experiments, compare results, and analyze the performance of different compression techniques using the Runner component.

2.2 Unified Interface

In PCToolkit, a unified interface for invoking prompt compression methods is provided. In the following example, we show how to simply invoke the compressing methods within few lines.

```
from pctoolkit.compressors import
    PromptCompressor

compressor = PromptCompressor(
    type='SCCompressor', device='cuda')

prompt = 'This is a prompt.'
ratio = 0.5
result = compressor.
         compressgo(prompt, ratio)
```

For simple compression task, one compressor is selected. Following the example given above, the original prompt is input to the compressor, and the compressor outputs the compressed prompt. For datasets evaluation, one datasets and multiple metrics are selected, along with the compressor chosen, these three parts are deployed in Runner. The Runner will provide the evaluation results according to the metrics list. The following example shows how to use PCToolkit to evaluate a dataset.

```
from pctoolkit.runners import run
from pctoolkit.datasets import
    load_dataset
from pctoolkit.metrics import
    load_metrics

compressor = PromptCompressor(
    type='SCCompressor', device='cuda')
dataset_name = 'arxiv'
dataset = load_dataset(dataset_name)

run(compressor=compressor,
    dataset=dataset,
    metrics=load_metrics,
    ratio=0.5)
```

Currently, the supporting dataset calls are implemented inside `run`. Users can also following the format in `run` to adapt their own datasets or metrics.

3 Conclusion

We introduced PCToolkit, an open-source project designed for prompt compression and evaluation. This toolkit provides a user-friendly and comprehensive resource, featuring six common compression methods and over ten diverse datasets that encompass a wide range of natural language tasks.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (No. 62406267), the Guangzhou-HKUST(GZ) Joint Funding Program (Grant No.2025A03J3956), the Guangzhou Municipal Science and Technology Project (No. 2025A04J4070), and the Guangzhou Municipal Education Project (No. 2024312122).

References

[Bai *et al.*, 2024] Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[Dong *et al.*, 2024] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A survey on in-context learning, 2024.

[Ghalandari *et al.*, 2022] Demian Ghalandari, Chris Hokamp, and Georgiana Ifrim. Efficient unsupervised sentence compression by fine-tuning transformers with reinforcement learning. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1267–1280, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[Grosse *et al.*, 2023] Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, Evan Hubinger, Kamil Lukoit, Karina Nguyen, Nicholas Joseph, Sam McCandlish, Jared Kaplan, and Samuel R. Bowman. Studying large language model generalization with influence functions, 2023.

[Jiang *et al.*, 2023] Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing prompts for accelerated inference of large language models. In *Conference on Empirical Methods in Natural Language Processing*, 2023.

[Jiang *et al.*, 2024] Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[Laban *et al.*, 2021] Philippe Laban, Tobias Schnabel, Paul Bennett, and Marti A. Hearst. Keep it simple: Unsupervised simplification of multi-paragraph text. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6365–6378, Online, August 2021. Association for Computational Linguistics.

[Lewis *et al.*, 2020] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[Li *et al.*, 2023] Yucheng Li, Bo Dong, Chenghua Lin, and Frank Guerin. Compressing context to enhance inference efficiency of large language models. In *Conference on Empirical Methods in Natural Language Processing*, 2023.

[Lin, 2004] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Annual Meeting of the Association for Computational Linguistics*, 2004.

[Pan *et al.*, 2024] Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics ACL 2024*, pages 963–981, Bangkok, Thailand and virtual meeting, August 2024. Association for Computational Linguistics.

[Papineni *et al.*, 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[Wang *et al.*, 2024] Xindi Wang, Mahsa Salmani, Parsa Omidi, Xiangyu Ren, Mehdi Rezagholizadeh, and Armaghan Eshaghi. Beyond the limits: A survey of techniques to extend the context length in large language models. In Kate Larson, editor, *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 8299–8307. International Joint Conferences on Artificial Intelligence Organization, 8 2024. Survey Track.

[Wei *et al.*, 2024] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2024. Curran Associates Inc.

[Yang *et al.*, 2024] Haoran Yang, Yumeng Zhang, Jiaqi Xu, Hongyuan Lu, Pheng-Ann Heng, and Wai Lam. Unveiling the generalization power of fine-tuned large language models. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 884–899, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[Zhang* *et al.*, 2020] Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020.