LogPrompt: Prompt Engineering Towards Zero-Shot and Interpretable Log Analysis

Yilun Liu, Shimin Tao, Weibin Meng, Feiyu Yao, Xiaofeng Zhao, Hao Yang
Huawei, China
{liuyilun3,taoshimin,mengweibin3,yaofeiyu1,zhaoxiaofeng14,yanghao30}@huawei.com

ABSTRACT

Automated log analysis plays a crucial role in software maintenance as it allows for efficient identification and resolution of issues. However, traditional methods employed in log analysis heavily rely on extensive historical data for training purposes and lack rationales for its predictions. The performance of these traditional methods significantly deteriorates when in-domain logs for training are limited and unseen log data are the majority, particularly in rapidly changing online environments. Additionally, the lack of rationales hampers the interpretability of analysis results and impacts analysts' subsequent decision-making processes. To address these challenges, we proposes LogPrompt, an novel approach that leverages large language models (LLMs) and advanced prompting techniques to achieve performance improvements in zero-shot scenarios (*i.e.*, no in-domain training). Moreover, LogPrompt has garnered positive evaluations from experienced practitioners in its log interpretation ability. Code available at https://github.com/lunyiliu/LogPrompt.

ACM Reference Format:
Yilun Liu, Shimin Tao, Weibin Meng, Feiyu Yao, Xiaofeng Zhao, Hao Yang. 2024. LogPrompt: Prompt Engineering Towards Zero-Shot and Interpretable Log Analysis. In *2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion '24), April 14–20, 2024, Lisbon, Portugal.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3639478.3643108

1 INTRODUCTION

Automated log analysis plays a significant role in ensuring the reliability of services in software system development and maintenance. With the help of current techniques, engineers can effectively parse logs and detect anomalies, thereby easing the workload of Operations and Maintenance (O&M) teams. The task of log parsing differentiate between static segments (*i.e.*, template) and dynamic segments (*i.e.*, variable), leading to a reduction in log size while preserving important information. The task of anomaly detection identifies anomalies in historical logs.

There are two challenges of applying existing log analysis technologies. First, the performance of existing log analysis technologies is significantly affected when there is a lack of training data, particularly in online situations where most confronted logs are new and unseen. Our preliminary experiment reveals that the performance can decrease by up to 62.5% when the ratio of training to test data changes from 9:1 to 1:9. In some cases, there may be no in-domain training data available at all, highlighting the need for a zero-shot log analysis approach. Second, the practical use of log analysis tools is hindered by their limited interpretability. Current methods often provide results without clear reasoning or enough context, only indicating variable positions for log parsing or giving binary yes/no answers for anomaly detection. This limitation requires users to invest additional effort in confirming results and locating root causes. In contrast, an interpretable output that includes explanations for parsing variables and rationales for identified anomalies would greatly enhance the log analysis process by facilitating the identification of false alarms and informing subsequent actions.

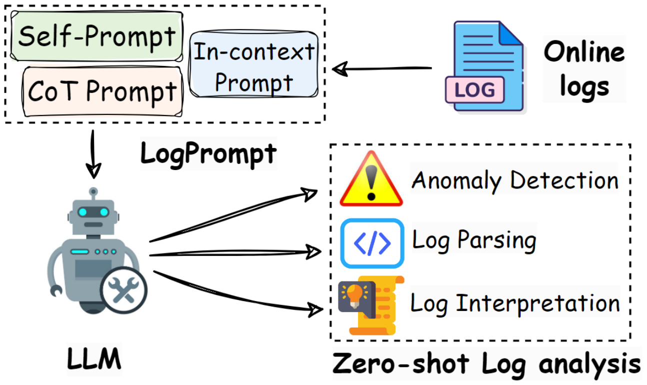

Figure 1: Workflow of LogPrompt in online log analysis.

This poster aims to address the two challenges in log analysis, which are zero-shot scenarios and limited interpretability. To combat these challenges, we leverage the powerful language ability of large language models (LLMs) and polish prompt strategies to adapt the ability of LLMs to log analysis tasks. As shown in Fig. 1, we propose a novel log analysis approach called LogPrompt, which focuses on prompt engineering with three strategies: self-prompt, chain-of-thought (CoT) prompt, and in-context prompt. Experimental results demonstrate the effectiveness of the prompt strategies in performing zero-shot log analysis tasks and log interpretation, which could aid O&M engineers in swiftly analyzing online logs, locating root causes and making mitigation decisions.

2 PROPOSED METHOD

As shown in Fig. 1, LogPrompt provides effective prompts for LLMs that enhance the ability of zero-shot log analysis and log interpretation. A textual prompt consists of a prompt prefix for task-specific instructions, an input slot for the task inputs, and an answer slot for the model's predictions. Firstly, LogPrompt standardizes the formats of input slot and answer slot, in order to mitigate uncertainty in the generated solutions. Subsequently, a tailored prompt strategy is employed to construct the prompt prefix. The prompt is assembled by concatenating the prompt prefix, input slot, and answer slot, with the final response acquired from the LLM.

For constructing the prompt prefix, three strategies are utilized in the experiment, founded on unique underlying principles:

Self-prompt. Inspired by Jiao *et al.*[2], this strategy involves the LLM suggesting its own prompt prefixes. A meta-prompt describing the task asks the LLM to generate prompt prefix candidates. These candidates are then tested on a specific log dataset, with the most effective prompt chosen based on performance.

CoT Prompt. Modeled after human problem-solving as per Wei *et al.* [3], the CoT prompt guides the LLM through a problem with reasoning steps, both *implicitly* and *explicitly*. (1) *Implicitly*: the LLM is asked to justify each decision, implicitly forming a logical thought chain. (2) *Explicitly*: intermediate steps for analysis tasks are manually defined and explicitly expressed in the prompt.

In-context Prompt. This approach uses several samples of labeled logs to set the context for the task. The LLM then predicts on new logs, using the context from the sample logs. For example, in anomaly detection, it classifies logs based on a few given examples of normal and abnormal logs.

3 RESULTS

This section presents our results on the efficacy of LogPrompt, implemented with the three strategies, for conducting zero-shot log analysis and log interpretation, tested on real-world log datasets from [1]. ChatGPT (version: gpt-3.5-turbo-0301) served as the underlying LLM, without any in-domain training (*i.e.*, zero-shot). For log parsing, the reported score is the average of RandIndex and F1-score. For anomaly detection, the metric is session-level F1-score (S-F1) and template-level F1-score (T-F1), which represents the accuracy of anomalies in a group of logs or single pieces of logs.

Self-prompt. Fig. 2 displays the log parsing performance of five prompt candidates suggested by ChatGPT, tested on a small subset of logs. Prompt 2 outperformed other candidates, and thereby is chosen in the full experiment involving eight datasets, where LogPrompt averagely outperformed existing SOTAs by 32.8%.

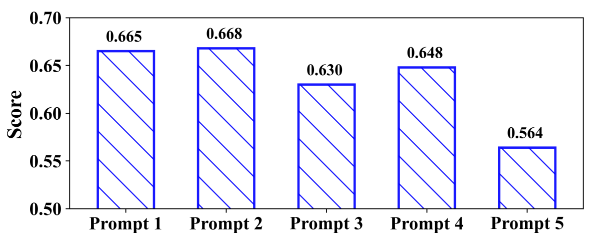

Figure 2: Candidates selection in self-prompt for log parsing.

CoT Prompt. As shown in Table 1, both the implicit and explicit CoT improved the zero-shot anomaly detection ability of LLM. In the full experiment, LogPrompt equipping the explicit CoT strategies averagely outperformed SOTA methods by 55.9% in F1-score.

Table 1: Performance of CoT Prompt in Anomaly Detection.

Methods	BGL		Spirit	
	S-F1	T-F1	S-F1	T-F1
ChatGPT (simple prompt)	0.189	0.115	0.445	0.086
+Implicit CoT	0.307	0.183	0.443	0.096
+Explicit CoT	**0.384**	**0.321**	**0.450**	**0.116**

In-context Prompt. In Fig. 3, both session-level F1-score and Precision for anomaly detection improved with more sample logs provided in the in-context prompt, up to 20, demonstrating the benefit of context demonstration. Performance declined when the context exceeds 20 samples, suggesting excessive information may distract the model from handling input logs.

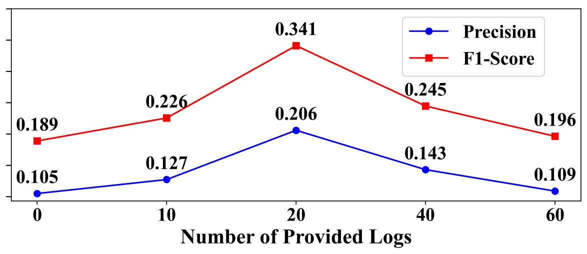

Figure 3: In-context prompt applied in anomaly detection.

Log Interpretation. Six senior practitioners evaluated LogPrompt's interpretations by rating its usefulness and readability on a one to five scale for 100 random logs in log parsing (*i.e.*, explanations for variables) and 100 in anomaly detection (*i.e.*, rationales for anomalies). As shown in Table 2, the mean score and HIP (*i.e.*, percentage rated above four) were consistently high, suggesting that practitioners found the generated content to be both helpful and readable.

Table 2: Interpretability of LogPrompt Rated by Reviewers.

Raters	Log Parsing				Anomaly Detection			
	Usefulness		Readability		Usefulness		Readability	
	Mean	HIP	Mean	HIP	Mean	HIP	Mean	HIP
R1	4.19	76.00%	4.70	94.00%	4.27	73.00%	4.50	84.00%
R2	4.36	81.00%	4.78	95.00%	4.42	79.00%	4.63	90.00%
R3	4.13	91.00%	4.18	95.00%	4.12	86.00%	4.35	94.00%
R4	4.20	73.00%	4.74	98.00%	4.40	85.00%	4.71	95.00%
R5	4.06	91.00%	4.41	100.00%	4.12	94.00%	4.24	98.00%
R6	4.46	86.00%	4.77	97.00%	4.48	90.00%	4.78	99.00%
Avg.	4.23	83.00%	4.60	96.50%	4.30	84.50%	4.54	93.33%

REFERENCES

[1] Shilin He, Jieming Zhu, Pinjia He, and Michael R Lyu. 2020. Loghub: a large collection of system log datasets towards automated log analytics. *arXiv preprint arXiv:2008.06448* (2020).
[2] Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Xing Wang, and Zhaopeng Tu. 2023. Is ChatGPT a good translator? A preliminary study. *arXiv preprint arXiv:2301.08745* (2023).
[3] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903* (2022).