Poster: Intelligent Network Management: RAG-Enhanced LLMs for Log Analysis, Troubleshooting, and Documentation

Shaghayegh Shajarian
North Carolina Agricultural and Technical State University
Greensboro, North Carolina, USA
sshajarian@aggies.ncat.edu

Sajad Khorsandroo
North Carolina Agricultural and Technical State University
Greensboro, North Carolina, USA
skhorsandroo@ncat.edu

Mahmoud Abdelsalam
North Carolina Agricultural and Technical State University
Greensboro, North Carolina, USA
mabdelsalam1@ncat.edu

Abstract

Modern network management is increasingly complex, requiring administrators to handle vast amounts of log data from diverse sources, leading to inefficiencies, errors, and operational challenges. In this work, we propose a novel AI-driven framework that integrates Large Language Models (LLMs) with Retrieval-Augmented Generation (RAG) and human-in-the-loop process to automate network management tasks such as log analysis, troubleshooting recommendations, and documentation generation. This study aims to enhance network reliability, reduce operational complexity, and move forward to autonomous network management.

Keywords

Network Management, Large Language Models (LLMs), Retrieval-Augmented Generation (RAG).

ACM Reference Format:
Shaghayegh Shajarian, Sajad Khorsandroo, and Mahmoud Abdelsalam. 2024. Poster: Intelligent Network Management: RAG-Enhanced LLMs for Log Analysis, Troubleshooting, and Documentation. In *Proceedings of the 20th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '24), December 9–12, 2024, Los Angeles, CA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3680121.3699887

1 Introduction

In the modern complex networking landscape, networks comprise both vendor-specific and vendor-agnostic equipment operating in physical and virtual environments. These networks are deployed across a wide range of settings, from data centers, cloud infrastructures, IoT devices, Wide Area Networks (WANs), and the core and access networks of mobile systems, each serving specific requirements and purposes. The devices within each domain generate a huge amount of technical logs in various formats and granularity.

Traditional network management approaches require administrators to navigate a maze of these log files, which often leads to misconfigurations, security vulnerabilities, and high operational overhead. The emergence of self-driving networks, which aim to automate many network operations, further underscores the need for advanced solutions [1].

Specifically, log analysis methods often struggle to cope with the high volume and complexity of data, which hinders the extraction of important information and effective diagnosis [2]. Despite the advancements in tools such as Splunk and Google's Logs Explorer, network operators still face challenges in utilizing logs effectively. This is because, for log analysis, operators need to learn proprietary query languages, such as Splunk's Search Processing Language (SPL) or Google's Logging Query Language. Moreover, for diagnosis and troubleshooting, operators have to consider the correlation of events in these different logs needs. Furthermore, accurate and well-documented technical records and documentation are valuable for decision-making and future troubleshooting.

Given the advancements in Artificial Intelligence (AI), particularly Large Language Models (LLMs), there is a significant opportunity to facilitate network management for operators who deal with log analysis, troubleshooting, and documentation. In this research, we aim to develop an intelligent framework to automate log analysis, provide troubleshooting recommendations to operators based on detected anomalies or failures, leverage professionals' knowledge to modify the troubleshooting steps, and, eventually, generate a comprehensive document of the detected issues and recommendations.

2 Proposed Framework

In this section, we propose a *"human assisted partially-automated closed-loop framework"* with four main components: ***1) Log Analyst Agent***, ***2) Troubleshooting Recommender Agent***, ***3) Human Validator***, and ***4) Documenter Agent***.

The proposed framework utilizes LLMs within three of its agents to perform various tasks. While LLMs have shown their effectiveness in many applications, they also have certain limitations, such as generating hallucinated information and lacking the necessary domain-specific knowledge. To overcome these challenges, we integrate the RAG technique into our LLM-based agents [3]. RAG enhances the performance of LLMs by coupling them with an external retrieval system, allowing the model to search for and retrieve relevant information from predefined knowledge sources. In our scenario, by retrieving historical logs, best practices, and relevant documentation from past events, the RAG-based framework is equipped with the necessary information to produce contextually grounded responses. The main advantage of our multi-agent architecture is the ability to customize each agent's model, specifications, and knowledge sources.

In this work, we focus specifically on log analysis for troubleshooting purposes. For simplicity, we do not cover the earlier phases of log management because our approach assumes that logs

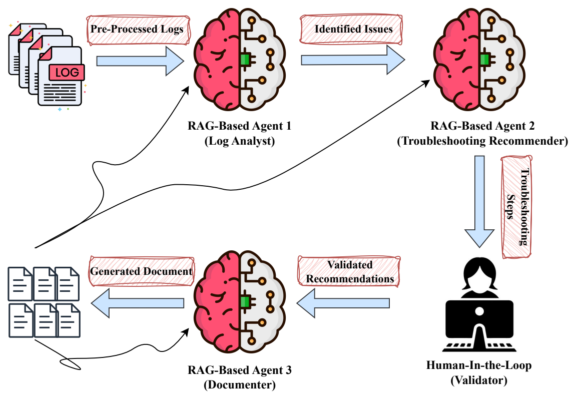

Figure 1: Proposed Framework

have already been collected, ingested, and pre-processed. Instead, our focus is on analyzing the processed logs to generate meaningful insights and recommendations for resolving issues.

As shown in Figure 1, the first LLM agent, Log Analyst, analyzes the network log files and, based on its information and the retrieval information from RAG's knowledge base, identifies any issues or concerns. These findings are then passed to the second LLM agent, the Troubleshooting Recommender, which also utilizes the RAG knowledge base to generate natural language recommendations for troubleshooting.

The recommendations generated by this agent are validated through a human-in-the-loop process, where network operators provide feedback. They can either approve or modify the recommendations based on their expertise. This component has two benefits for the proposed framework. First, it ensures that AI-generated recommendations are vetted by experts and the accuracy and relevance of the recommendations will be checked before they are documented. Additionally, by tracking how often users modify or accept the recommendations, we assess the overall accuracy and effectiveness of the proposed framework.

After validating the recommendation, the last agent, which is implemented as a multi-modal LLM, generates a well-documented report. It generates both textual and visual documentation, such as diagrams or network topology visualizations, to create more comprehensive documentation. As this framework is closed-loop, the generated documents are utilized in post-hoc analysis and support. They serve as entries in the RAG knowledge base, where each LLM agent can access detailed records of past network issues and their solutions and thus be able to obtain accurate and up-to-date information.

3 Implementation

Our preliminary implementation, built using Langchain [1], facilitates efficient retrieval of relevant information from various document formats, including PDFs, text files, and JSON files. The documents are loaded and split into manageable chunks to facilitate efficient retrieval and generation. The system generates vector embeddings and uses a vector index to store these embeddings, allowing for rapid similarity-based search. The specific parameters used in this implementation are outlined in Table 1, which provides an overview of the key settings that optimize the retrieval process.

[1]https://github.com/langchain-ai

Table 1: Key Parameters of the Implementation

Parameter	Value
Language Model	GPT-4o
Temperature	0
Chunk Size	1000 chars
Chunk Overlap	200 chars
Embedding Model	OpenAIEmbeddings
Vector Storage	FAISS

4 Conclusion and Future Work

We proposed a RAG-enhanced LLM framework that automates network operations, i.e., log analysis, troubleshooting recommendations, and documentation generation. By combining AI with expert validation, the framework enhances network reliability, reduces operational complexity, and efficiently handles the vast generated log data. This work is a step towards fully autonomous network management. In the future, the proposed framework will be evaluated using qualitative and quantitative approaches to ensure the framework's applicability in real-world environments. Moreover, by integrating new techniques such as Chain-of-Thought (CoT) and Reasoning and Acting (ReAct), the framework can improve both the depth and accuracy of the LLM agents' outputs.

Acknowledgments

This work is being supported in part by the National Science Foundation (NSF) Grants #2113945 and #2200538. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agency.

References

[1] Nick Feamster and Jennifer Rexford. 2017. Why (and how) networks should run themselves. *arXiv preprint arXiv:1710.11583* (2017).
[2] Shilin He, Xu Zhang, Pinjia He, Yong Xu, Liqun Li, Yu Kang, Minghua Ma, Yining Wei, Yingnong Dang, Saravanakumar Rajmohan, et al. 2022. An empirical study of log analysis at Microsoft. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 1465–1476.
[3] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.