Majid Lotfian Delouee, Daria G. Pernes, Victoria Degeler, Boris Koldehofe. Poster: Towards Federated LLM-Powered CEP Rule Generation and Refinement. In Proceedings of the 18th ACM International Conference on Distributed and Event-Based Systems (DEBS'24), 2 pages, ACM press.

Poster: Towards Federated LLM-Powered CEP Rule Generation and Refinement

Majid Lotfian Delouee
University of Groningen

Daria G. Pernes
University of Groningen

Victoria Degeler
University of Amsterdam

Boris Koldehofe
Technische Universität Ilmenau

ABSTRACT

In traditional event processing systems, patterns representing situations of interest are typically defined by domain experts or learned from historical data. These approaches often make rule generation reactive, time-consuming, and susceptible to human error. In this paper, we propose and investigate the integration of large language models (LLMs) to automate and accelerate query translation and rule generation in event processing systems. Furthermore, we introduce a federated learning schema to refine the initially generated rules by examining them over distributed event streams, ensuring greater accuracy and adaptability. Preliminary results demonstrate the potential of LLMs as a key component in proactively expediting the autonomous rule-generation process. Moreover, our findings suggest that employing customized prompt engineering techniques can further enhance the quality of the generated rules.

KEYWORDS

Autonomous Rule Generation, Complex Event Processing, Rule Refinement, Federated Learning, Large Language Models.

1 INTRODUCTION

Real-time data analysis is vital in today's fast-paced world, especially in fields like e-commerce, healthcare, and the Internet of Things (IoT). Data quickly loses its usefulness in predicting or preventing problems, so it needs to be analyzed almost immediately. Distributed Complex Event Processing (DCEP) is a well-known real-time method for analyzing this data. It takes streams of *simple events* (e.g., IoT sensor readings) and analyzes them to identify meaningful patterns or *complex events*, which signify situations of interest (e.g., road congestion) [2]. This analysis uses CEP *rules*, i.e., instructions explaining the connection between observed events and the user's situations of interest. While the accurate generation of CEP rules significantly impacts detection performance, these rules are traditionally formulated based on the knowledge of domain experts, making it time-consuming, and susceptible to human errors. Moreover, domain experts are unaware of all dependencies in the environment and might not capture complex correlations or be unable to benefit from event source heterogeneity (as in [3]). These limitations encourage DCEP system developers to involve AI-based techniques to support the generation of CEP rules more effectively based on historical data [7].

Existing Solutions and Their Pitfalls. Recent studies have explored how to automatically create rules in DCEP systems [6, 7], but these methods are often limited to specific domains. Furthermore, they haven't addressed how to proactively create rules for situations that might not have been seen yet in the environment but are of high importance or probability, e.g., unseen weather conditions. Additionally, users might issue queries to discover new situations, but current methods can't quickly and autonomously respond to them, causing delays in processing. This means we must move beyond creating rules based on past data, by developing a hybrid approach combining reactive and proactive rule generation. Furthermore, recent advancements in Large Language Models (LLMs) for Natural Language Processing (NLP) [8] offer the potential to proactively enhance such a query to pattern translation using LLM's knowledge, an area not yet explored within the CEP domain.

In this paper, we propose a new method for creating and refining rules for CEP by integrating the power of LLMs, like ChatGPT, to help initiate these rules. First, the LLM takes a user's query, along with quality and privacy requirements, and translates it into a CEP-like pattern. This initial rule is then tested and improved using a distributed learning framework, which allows the rule to be adjusted on data from multiple sources without compromising privacy. Finally, the rule is tested, and in case of successful results, it will be added to a central rule database where it can be used to update existing rules. In the following, we first elaborate on our approach, present the preliminary experiment results, and conclude the paper.

2 OUR APPROACH

Our proposed mechanism aims to automate, accelerate, and fine-tune reactive and proactive rule generation and management in DCEP systems. As illustrated in the proposed system design (see Figure 1 (left)), we fill the gap for autonomous proactive rule generation in DCEP systems alongside a refinement process using distributed local streams. The first challenge in this research is interacting with LLM to generate the initial version of CEP rules. To do so, we consider the constraints of working with LLMs, including prompt length, input multi-modality, integrating LLM API to the code, LLM response uncertainties/hallucination, number of allowed prompts, and eventually subscription costs.

To overcome these limitations, we designed several levels of prompts (i.e., LLM's input) by involving *basic prompting engineering techniques* (see Figure 1 (middle)), including *i*) Zero-Shot, which is direct prompting without providing examples (i.e., labeled records with Ground Truth), *ii*) Few-Shot, which provides a few examples to the prompt's context to improve the response accuracy, *iii*) Chain of Thoughts (CoT), which appends an instruction to the end of a prompt that mimics human reasoning by asking the LLM not only to generate an end result but also to detail the series of intermediate steps that led to that response, *iv*) Chain of Thoughts Self-Consistency (CoT-SC), which generates multiple CoTs by prompting the model several times and applying a majority voting to make the final decision, and *v*) Tree of Thoughts (ToT), which encourages LLM to explore multiple reasoning paths, like branches of a tree, instead of following a single chain of thought, leading to more creative and comprehensive problem-solving, as the model can evaluate different options and backtrack, if necessary. To achieve more accurate responses,

DEBS '24, June 25-28, 2024, Lyon, France.
© 2024 Association for Computing Machinery.
This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive version of the record will be published in proceedings of the ACM conference (DEBS '24).

Majid Lotfian Delouee, Daria G. Pernes, Victoria Degeler, Boris Koldehofe. Poster: Towards Federated LLM-Powered CEP Rule Generation and Refinement. In Proceedings of the 18th ACM International Conference on Distributed and Event-Based Systems (DEBS'24), 2 pages, ACM press.

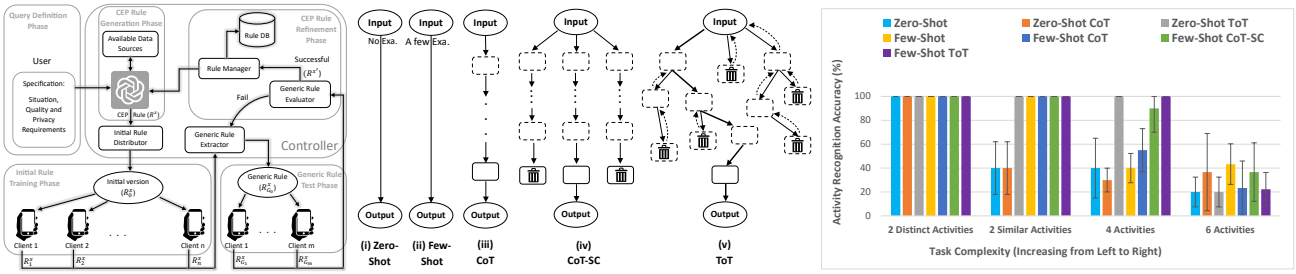

Figure 1: (left) System Design, (middle) Basic Prompt Engineering Techniques, (right) The Initial Results

one can merge two or more basic engineering techniques to benefit from all, e.g., Few-Shot CoT. In the above techniques, a dashed rectangle shows thoughts representing coherent language sequences that serve as intermediate steps toward solving a problem. A solid-line rectangle displays the final decision, delivered as the response. Besides, rectangles marked with recycle bins represent abandoned thoughts, which will not lead to the final decision due to majority voting or backtracking.

The second challenge, where the initial rule should be validated and refined using local streams, brings even more complications since a special aggregation method should be deployed for each rule type to produce a generic rule out of local processing updates. For example, the aggregation method should consider the highest update value for time-based thresholds to cover a maximum number of rule matches in all local streams. In our approach, each client applies the initial version of the rule (i.e., R_0^x) on its simple event streams, including currently available streams alongside historical data. Proactively, each client plays with rule characteristics (e.g., if-else thresholds, window size, etc.) to maximize specified quality metrics (e.g., detection accuracy). The customized rule is then sent to an aggregation module as the client updates (i.e., R_1^x to R_n^x). This module produces a *generic rule* (i.e., $R_{G_0}^x$) according to the aggregation function specified for this rule type. This generic rule is tested over more clients to check its generalizability, and new validation results (i.e., $R_{G_1}^x$ to $R_{G_m}^x$) are created, which are analyzed by *Evaluator* component. This test phase is repeated until the analysis is successful, e.g., most test clients achieve an acceptable detection accuracy. In the end, the refined generic rule (i.e., $R^{x'}$) is submitted to a *rule manager* module to update the CEP rule database and support LLM to improve its responses.

3 PRELIMINARY EXPERIMENTS

Our primary evaluation goals are to: 1) Investigate how accurately LLMs generate the initial version of a CEP rule, 2) Assess to what extent prompt engineering techniques can enhance the quality of generated rules, and 3) Identify the limitations of our approach when applied to real-world datasets.

Simulation Setup: We evaluated our approach by analyzing the *PAMAP2* dataset [5], which contains data on 18 different physical activities (e.g., walking) performed by 9 subjects wearing 3 inertial measurement units and one heart rate monitor. We introduced and applied seven hybrid prompt engineering techniques including: 1) Zero-Shot, 2) Zero-Shot with CoT, 3) Zero-Shot with ToT, 4) Few-Shot, 5) Few-Shot with CoT, 6) Few-Shot with CoT-SC, and 7) Few-Shot with ToT. In the initial assessment, our experiments used the *GPT-4* [4]. Attempts to obtain suitable responses from *Google Gemini* [1] were unsuccessful since it delivered rule generation's instructions, not the rule definitions. Each hybrid technique was applied at least five times, and the final results represent the average of these iterations. The error bars displayed illustrate the confidence intervals of our experimental findings. We tasked the LLM with generating CEP-like patterns to identify specific activities within the dataset. To simplify the process, we provided the LLM with a sample from the PAMAP2 dataset and requested it classify the activity for each record based on the generated patterns. The activity recognition accuracy was then evaluated by comparing the LLM's results against the Ground Truth labels in the dataset. Note that in zero-shot, the LLM's lack of examples can lead to incorrect solutions, and CoT-SC may worsen this by favoring the most consistently wrong answer over the most accurate one. That's why we exclude Zero-Shot CoT-SC from our experiments.

Initial Results: Figure 1 (right) presents the initial results of utilizing the aforementioned techniques to generate CEP rules and classify activities. The results demonstrate that GPT-4 generated rules possess adequate capacity to differentiate between two distinct activities. However, as the number of activities increased (i.e., increasing task complexity), GPT-4 tended to misclassify activities. This suggests that the generated rules may require further refinement (e.g., adjusting thresholds), particularly for activities with greater similarity (e.g., sitting and lying). While CoT, CoT-SC, and ToT were expected to enhance rule accuracy, this was not consistently observed. Although Few-Shot techniques generally outperformed Zero-Shots, the specific conditions favoring each group remain unclear. Additionally, the lack of labeled data in some use cases definitely limits the applicability of Few-Shot approaches.

In conclusion, investigating the applicability of our approach to other use cases would expand the scope of this research and uncover further insights. Additionally, exploring rule refinement through a federated framework applied to distributed data could potentially yield even more accurate and robust CEP rules.

REFERENCES

[1] Google. 2024. Google Gemini: Multimodal AI model. https://ai.googleblog.com. Accessed: 2024-05-21.
[2] M. Lotfian Delouee, V. Degeler, P. Amthor, and B. Koldehofe. 2024. APP-CEP: Adaptive Pattern-level Privacy Protection in Complex Event Processing Systems. In *ICISSP'24*. SCITEPRESS, 12 pages. in press.
[3] M. Lotfian Delouee, B. Koldehofe, and V. Degeler. 2023. AQuA-CEP: Adaptive Quality-Aware Complex Event Processing in the Internet of Things. In *DEBS'23*. ACM, 13–24.
[4] OpenAI. 2023. GPT-4: OpenAI's language model. https://openai.com/research/gpt-4. Accessed: 2024-05-21.
[5] Attila Reiss. 2012. PAMAP2 Physical Activity Monitoring. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5NW2H.
[6] J. Roldán-Gómez, J. Boubeta-Puig, J. Carrillo-Mondéjar, J.M.C. Gómez, and J.M. del Rincón. 2023. An Automatic Complex Event Processing Rules Generation System for the Recognition of Real-time IoT Attack Patterns. *Engineering Applications of Artificial Intelligence* 123 (2023), 106344.
[7] M.U. Simsek, F. Yildirim Okay, and S. Ozdemir. 2021. A Deep Learning-based CEP Rule Extraction Framework for IoT Data. *The Journal of Supercomputing* 77 (2021), 8563–8592.
[8] Y. Zhou, N. Yu, and Z. Liu. 2023. Towards Interactive Research Agents for Internet Incident Investigation. In *HotNET'23*. 33–40.