Technical Disclosure Commons

Defensive Publications Series

July 2023

Log Exploration and Analytics Using Large Language Models

Hari Bhaskar S

Log Exploration and Analytics Using Large Language Models

ABSTRACT

Log data are typically stored in databases as schema-based entries in a structured format. Conventionally, log exploration requires an understanding of the fields, schema, and query parameters of the database. This disclosure describes techniques that use tabular large language models (LLMs) to process, mine, and make log data amenable to natural language queries. A relatively unsophisticated user with no database skills can query log files using natural language search. The LLMs can be fine-tuned using prompt engineering and causation information. The conventional, tedious mining of logs across multiple systems using database queries is replaced by a simple natural language interface that provides the ability to determine meaningful relationships and context across events captured within the logs. Natural language queries can enable help desks to do a basic level of troubleshooting, saving time for administrators. As more information gets added, querying and analytics of logs are simplified, with a resultant improvement in the speed and quality of troubleshooting.

KEYWORDS

- Log exploration
- Log querying
- Log analytics
- Large language model (LLM)
- Tabular LLM
- Prompt engineering
- Causation information
- Intent processing

BACKGROUND

Log files, which can be very large, are explored and searched through using filters and parameters. Exploration of log files requires knowledge of the tables and fields of the log database. Log files do not lend themselves immediately to natural language queries such as ‘show me the latest high severity incidents by storage, network, database.’

Log mining is not understood currently as a natural language processing problem. While an accurate response to a natural language query (as the aforementioned example) can help a service engineer or help desk (and ultimately, the end customer), the requisite visualizations, intent processing, prompt processing, etc., are not available with current log management and exploration tools.

In many systems, logs are stored in databases as schema-based entries in a structured format. With such systems, logs can be explored using database queries, not natural language queries. Therefore, log exploration requires special skills, and is open only to relatively sophisticated professionals such as administrators, developers, or other experts who have an in-depth understanding of the database fields, schema, and query parameters. It is tedious to mine sequences or determine the causation of events, especially if there are multiple log systems that record events. To provide actionable information, even the first level of service engineer or help desk needs to be trained on the log database, which is a substantial expense.

DESCRIPTION

This disclosure describes techniques that leverage large language models (LLM) to convert tables of log data to tabular LLM methods (text LLM format). An LLM processes the converted tables to generate answers to queries posed in a natural language. Prompt engineering, which can build links and connections between text data, can be used to fine-tune an LLM trained over the log files. For example, a prompt-engineered LLM can digest a log file that includes tabular data relating to storage devices, networks, databases, input-output modules, latencies, etc., to determine potential links between them. Similarly, causation data provided by a user can be used to fine-tune the LLM. Furthermore, if the log data is available in natural language form, a user can formulate interesting queries without having to understand the underlying schema.

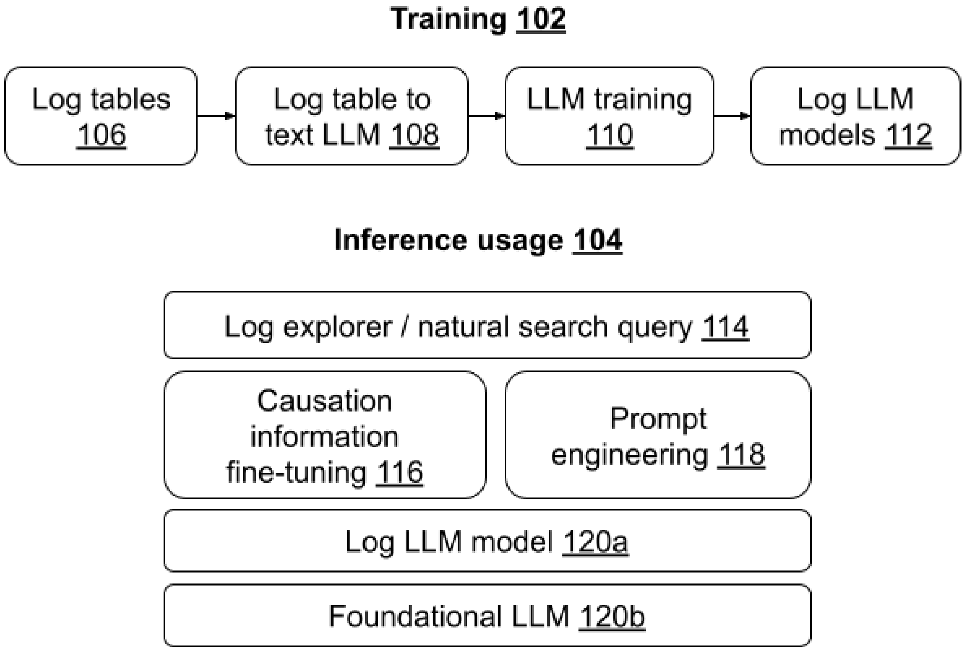

Fig. 1: Natural language queries, log exploration, and analytics using tabular LLMs

Fig. 1 illustrates natural language queries, log exploration, and analytics using tabular LLMs. In a training phase (102), tables of log data (106) are converted to text LLM (108). LLM training (110) to form log LLM models (112). At inference time (104), natural language queries or log explorations (114), are fine-tuned using causation information (116) and prompt engineering (118) to arrive at log LLM model (120a) or foundational LLM model (120b). Fine-tuning with causation information and prompt engineering are explained in greater detail below.

Fine-tuning with causation information

Some examples of causation information include:

- A lengthening queue of network requests leads to an increased serving latency, in turn leading to an I/O problem, a storage failure, and subsequent service outage.
- A database upgrade causes an increased query latency, leading to a downstream performance impact.
- A lack of disk swap space during data transfer across storage zones leads to higher severity.

Causative chains, as in the above examples (lengthening network queue→serving latency→…→service outage; database upgrade→query latency→performance impact; disk swap space→severity) can be specified using a graph, e.g., as illustrated in Fig. 2, to fine-tune the LLM.

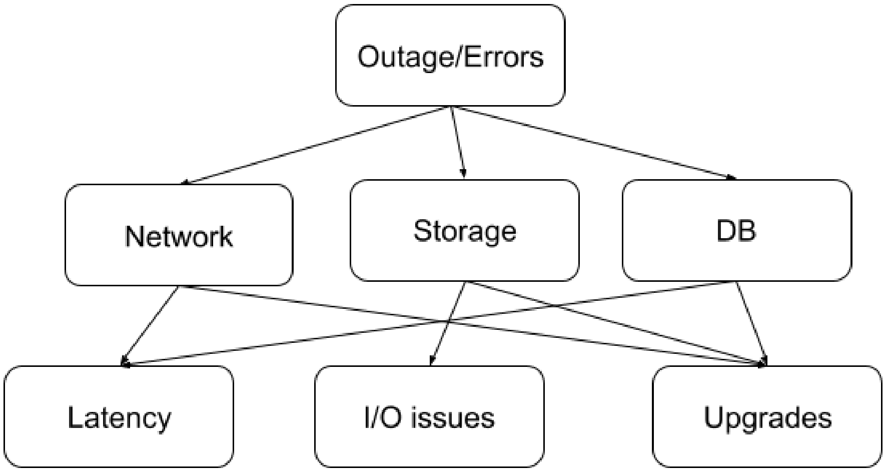

Fig 2: Specifying causation information to fine-tune the LLM

The LLM can be fine-tuned with causation information using active learning or reinforcement learning. For example, the relationships between various log entities can be established. In this manner, by feeding a graph to the LLM, simple fine-tuning based on causation information can be done.

Prompt engineering

Prompt engineering can be designed using factors such as:

- Commonly used natural queries that arise during log mining, e.g., severity, event time, event type, associated entity (storage, network, database etc.).
- Common performance impacts, e.g., latency, outage, reliability service level objectives (SLOs), thresholds, breaches, etc.
- Domain-specific information based on standard terminologies associated with system engineering, reliability, performance, etc.

Effective prompt engineering can enable the LLM to learn the types of queries, associated results, and domain constructs on log mining.

In this manner, large log files can be processed, mined, and made amenable to natural language queries using tabular LLMs. A relatively unsophisticated user with no particular knowledge of the fields, schema, or query parameters of the log database can utilize the trained LLM as described herein to query log data using natural language search.

With active learning, the LLM can be fine-tuned with prompt engineering and causation information. The conventional, tedious mining of logs across multiple systems using database queries is replaced by a simple natural language interface that provides the ability to determine meaningful relationships and context across events captured within the logs. Natural language queries can enable help desks to do a basic level of troubleshooting, saving time for administrators. Log analysis cases that are more demanding can be sent to experts, and even those benefit from troubleshooting details. As more information gets added, querying and analytics of logs are simplified, with a resultant improvement in the speed and quality of troubleshooting.

CONCLUSION

This disclosure describes techniques that use tabular large language models (LLMs) to process, mine, and make log data amenable to natural language queries. A relatively unsophisticated user with no database skills can query log files using natural language search. The LLMs can be fine-tuned using prompt engineering and causation information. The conventional, tedious mining of logs across multiple systems using database queries is replaced by a simple natural language interface that provides the ability to determine meaningful relationships and context across events captured within the logs. Natural language queries can enable help desks to do a basic level of troubleshooting, saving time for administrators. As more information gets added, querying and analytics of logs are simplified, with a resultant improvement in the speed and quality of troubleshooting.

REFERENCES

1. Hegselmann, Stefan, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. "TabLLM: Few-shot classification of tabular data with large language models." In *International Conference on Artificial Intelligence and Statistics*, pp. 5549-5581. PMLR, 2023.