August 2023

NETWORK DEVICE SYSTEM LOGGING SUMMARIZATION BASED ON LOW-RANK ADAPTATION AND CONTRASTIVE LEARNING

Cheng Jiao

Bruce Yang

Lynn Chen

Xinqi Wang

Qihong Shao

NETWORK DEVICE SYSTEM LOGGING SUMMARIZATION BASED ON LOW-RANK ADAPTATION AND CONTRASTIVE LEARNING

AUTHORS:
Cheng Jiao
Bruce Yang
Lynn Chen
Xinqi Wang
Qihong Shao

ABSTRACT

Techniques are presented herein that support the automatic generation of refined and summarized text from a system logging (syslog) message sequence. Aspects of the presented techniques employ an abstractive syslog summarization large language model (LLM) that is trained with contrastive learning and then fine-tuned using a Low-Rank Adaptation (LoRA) methodology. Under further aspects of the presented techniques, auxiliary text (such as network incident reports and application incident reports) is added to the prompt of the input of the LLM model to help the model generate a richer syslog summarization.

DETAILED DESCRIPTION

The system logging (syslog) facility supports the centralized recording of a wide range of messages. Such a collection of messages may describe a series of events that are generated by network devices and help a network engineer track and debug software and hardware issues that are associated with those devices. When a network failure occurs, a network engineer typically needs to gather and read through the raw syslog files to analyze the root cause of the failure. However, analyzing raw syslog messages is difficult and very time consuming.

The content of a syslog message is basically unstructured or semi-structured free text, and the format of such a message varies across network device vendors and device operating systems. Additionally, syslog messages are low-level data and cannot be directly translated into actual events in a network (i.e., network events) because they are not abstracted and aggregated. Further, not every error-likely syslog message from a network device indicates that there is really an event impacting the related network services. For example, some router's syslog messages are generated purely for debugging purposes and have no impact on network services.

Although automated syslog analysis tools for diagnosis can assist a network engineer in locating problems, during troubleshooting such an engineer will still want to understand individual network events as expressed in a massive syslog sequence. Therefore, syslog summarization (i.e., the automatic generation of refined and summarized text from a syslog message sequence) becomes particularly important. Such a summarization process may have a number of purposes, including the removal of redundant information (i.e., deduplication), the removal of unimportant messages (such as debug information), the extraction of key information (such as device names, port numbers, and other entities and their actions), and a reorganization of the extracted information into a natural language that is easy for a human to read.

However, existing summarization methods are either rule-based (employing, for example, regular expressions) and semi-manual or based on information extraction methods that retrieve keywords from a syslog message. The former is inefficient and not suitable for summarizing a massive syslog sequence, while the latter is an extractive summarization method. One obvious problem with such a method is that it lacks fluency and sentences cannot naturally transition from one to another. Moreover, the main topic of a syslog sequence may be buried in the raw syslog messages, which cannot be captured in a single syslog artifact, so the representativeness and comprehensiveness of the summarization are limited.

Techniques are presented herein that address the above-described challenge by offering three key functionalities.

First, an abstractive syslog summarization model is fine-tuned using a Low-Rank Adaptation (LoRA) methodology that is based on a generative large language model (LLM) to solve the problem of the extractive syslog summarization output not being smooth and fluent (which can lead to a network engineer easily missing key information).

Second, abstractive syslog summarization is often based on supervised machine learning methods. However, syslog summarization lacks manually labeled data. Therefore, the presented techniques employ contrastive learning, a self-supervised learning method, to improve the quality of the syslog summarization that is generated by the instant model in the absence of labeled data. Aspects of the presented techniques may employ elements of a logging management or analysis platform to identify whether or not a particular syslog message is normal before such a message is used as training data for the contrastive learning process.

Third, previous syslog summarization methods often only focused on the network device syslog sequence itself and ignored other network event-related textual data such as network incident reports and application incident reports (such as, for example, meeting management system incident reports). Therefore, the presented techniques add that auxiliary textual data to the prompt input to the instant LLM so that the generated syslog summarization not only contains a summary description of the syslog sequence but also includes other information such as a resolution, affected services, and a root cause analysis of the syslog sequence.

Figure 1, below, presents a high-level overview of elements of a model that is possible according to the presented techniques during the training stage of that model.

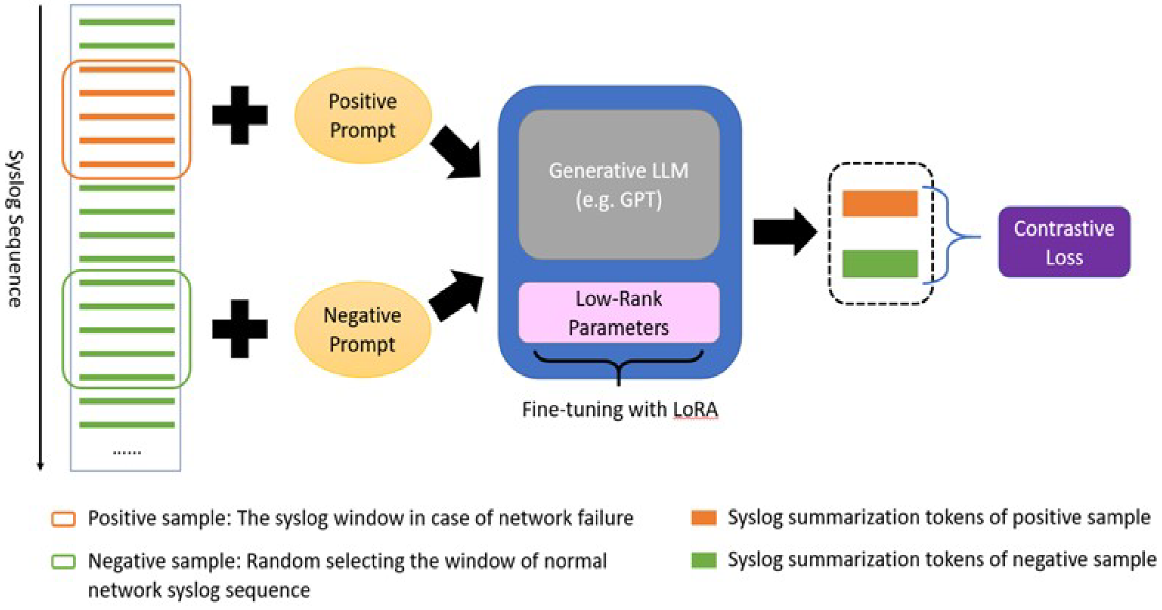
Syslog Sequence
......
Positive Prompt
Negative Prompt
Generative LLM (e.g. GPT)
Low-Rank Parameters
Fine-tuning with LoRA
Contrastive Loss
□ Positive sample: The syslog window in case of network failure
□ Negative sample: Random selecting the window of normal network syslog sequence
■ Syslog summarization tokens of positive sample
■ Syslog summarization tokens of negative sample

Figure 1: Model Training

As depicted on the left-hand side of Figure 1, above, a first key element of the presented techniques encompasses a data selection process. As introduced above, contrastive learning is a type of self-supervised learning that requires both positive and negative samples during training. Since the presented techniques focus mainly on a summarization when a network failure occurs, a syslog sequence (with a fixed window size) that is closely related to such a failure may be selected as a positive sample and a syslog sequence (having the same window size) when the network is normal may be randomly selected (from, for example, a database) as a negative sample. Automated tools or machine learning may be used to identify positive and negative samples in a raw syslog sequence, or manual labeling may be performed.

Since the presented techniques employ a generative LLM, corresponding prompts are also required in addition to positive and negative syslog sequences. Those prompts may include auxiliary text such as a network incident report and an application system incident report. The presented techniques employ a template for such information where a syslog sequence is delimited by single backticks (i.e., `` `{syslog sequence}` ``), a network incident report is delimited by double backticks (i.e., ``` ``{network incident report}`` ```), and a meeting management system incident report is delimited by triple backticks (i.e., ```` ```{application system incident report}``` ````).

According to the presented techniques, the above-described information may be processed and summarized to generate a network troubleshooting report. Such a report may include, possibly among other things, the different sections that are identified in Table 1, below.

Table 1: Network Troubleshooting Report

Section	Content
1	Incident title
2	Date and time
3	Date and time of restoration
4	Summary
5	Timeline of events
6	Resolution
7	Affected services
8	Root cause analysis
9	Conclusion

A second key element of the presented techniques encompasses a loss function. For the objective function of contrastive learning, the presented techniques employ the Normalized Temperature-scaled Cross Entropy loss (NT-Xent loss) function as shown in the following formula:

$$\mathcal{L}_{CL} = -\log \frac{e^{\mathrm{sim}\left(\mathbf{h}_i,\mathbf{h}_i^{+}\right)/\tau}}{\sum_{j=1}^{N} e^{\mathrm{sim}\left(\mathbf{h}_i,\mathbf{h}_j^{-}\right)/\tau}}$$

In the above formula, h is a sentence vector that is composed of the summation of LLM output tokens, h_i and h^+ in the numerator are a pair of positive samples from a mini-batch, h^- in the denominator is a negative sample from the same mini-batch, and N represents the number of samples in the mini-batch.

A third key element of the presented techniques encompasses a fine-tuning process. Since a LLM typically has billions, or even more, parameters, in order to save training time and computational resources the presented techniques employ a LoRA methodology to fine-tune the above-described model. Figure 2, below, presents elements of such an approach.

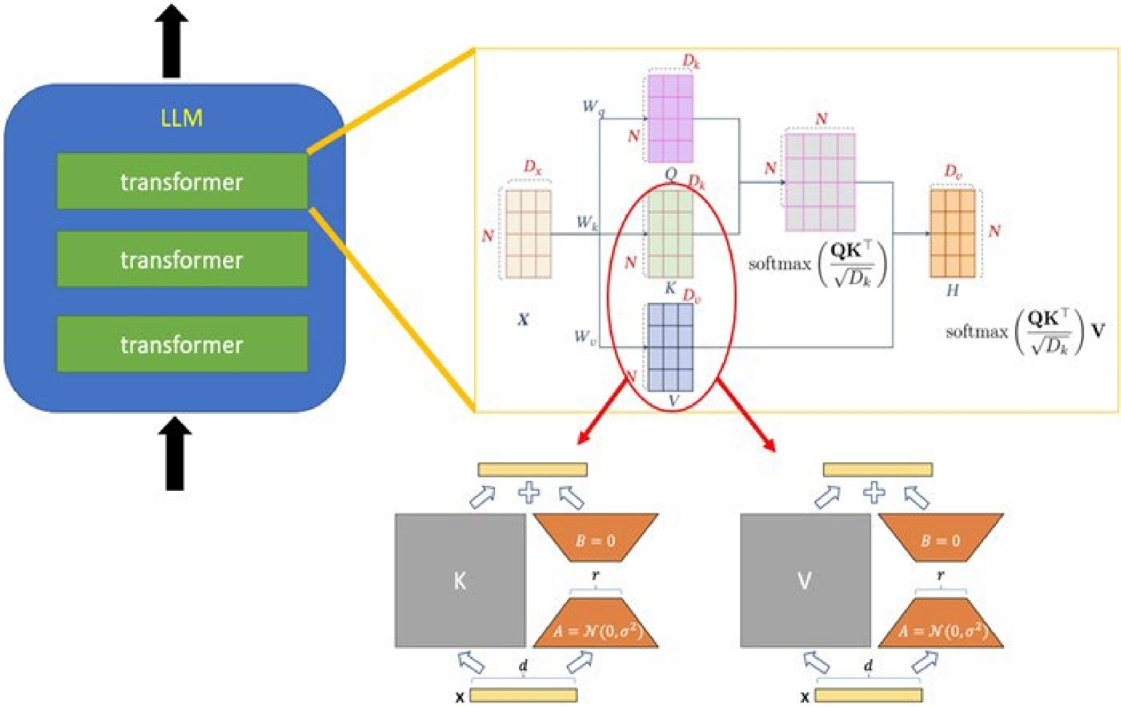
LLM
transformer
transformer
transformer
W_q
D_k
N
Q
D_x
N
W_k
X
K
D_v
W_v
V
N
$\mathrm{softmax}\left(\frac{\mathbf{QK}^\top}{\sqrt{D_k}}\right)$
D_v
H
$\mathrm{softmax}\left(\frac{\mathbf{QK}^\top}{\sqrt{D_k}}\right)\mathbf{V}$
K
$B = 0$
r
$A = \mathcal{N}(0,\sigma^2)$
d
x
V
$B = 0$
r
$A = \mathcal{N}(0,\sigma^2)$
d
x

Figure 2: Fine Tuning

In an LLM, and as depicted in Figure 2, above, low-rank parameter matrices A and B are added to the sides of the K and V weights in each layer of the transformer structure, where the number of parameters in A and B are much smaller than that of K and V. During fine-tuning, only the weights of A and B are optimized while the weights of the LLM itself remain fixed.

The techniques presented herein, as described and illustrated above, may be further understood through an illustrative example which includes a sample prompt that is input to the above-described model and the desired output that is generated by the trained model.

As described above, the presented techniques employ a template for input information. Using that template under the illustrative example, the sample prompt (or model input) may comprise:

```
`
2022-07-06 12:01 DEVICE001 % PLATFORM: R0/0: kernel: [28463270.988920] free:13891 slab_reclaimable: 1442 slab_unreclaimable:3050
2022-07-06 12:43 DEVICE001 %PLATFORM: RO/0: kernel: [23371934.464163] active_anon:596349 inactive_anon:163542 isolated_anon:1
2022-07-06 12:01 DEVICE001 % PLATFORM: R0/0: kernel: [28463280.436395] smand invoked oom-killer: gfp_mask=0x44d0, order=2, oom_adj=4
2022-07-06 13:53 DEVICE001 %PLATFORM: R0/0: kernel: [28469996.861787] HighMem free:3480kB min:512kB low:3744kB high:6980kB active_anon:2267560kB in...
2022-07-06 13:53 DEVICE001 % PLATFORM: R0/0: kernel: [28463282.762005] lowmem reserve[]: 0 0 2788 2788
2022-07-06 12:01 DEVICE001 % PLATFORM: R0/0: kernel: [28463282.762005] lowmem reserve[]: 0 0 2788 2788
2022-07-06 13:29 DEVICE001 %PLATFORM: R0/0: kernel: [28468568.991413] HighMem free:3736kB min:512kB low:3744kB high:6980kB active_anon:2266964kB in...
2022-07-06 12:35 DEVICE001 % PLATFORM: RO/0: kernel: [25178575.022366] DMA per-cpu:
2022-07-06 12:47 DEVICE001 % PLATFORM: RO/0: kernel: [25178575.022366] DMA per-cpu:
2022-07-06 13:29 DEVICE001 %PLATFORM: RO/0: kernel: [25179289.100588] [e3e7bd90] [c0006ecc] show_stack+0x44/0x160 (unreliable)
```

~~~
2022-07-06 12:35 DEVICE001 %PLATFORM: R0/0: kernel: [25178585.636043] HighMem free:3908kB min:512kB low:3744kB high:6980kB active_anon:2267528kB in...
2022-07-06 13:56 DEVICE001 %PLATFORM: RO/0: kernel: [25179289.100588] [e3e7bd90] [c0006ecc] show_stack+0x44/0x160 (unreliable)
2022-07-06 12:01 DEVICE001 %PLATFORM: R0/0: kernel: [28470168.677831] Call Trace:
2022-07-06 13:57 DEVICE001 %PLATFORM: RO/0: kernel: [25179289.100588] [e3e7bd90] [c0006ecc] show_stack+0x44/0x160 (unreliable)
`
``
~~~

Incident Report: XYZ 123 Unexpected Reboot

Report Date: 2022-07-06

Report Time: 15:30 PM

Incident Summary: On July 6th, 2022, at approximately 14:00 PM, an unexpected reboot occurred on the XYZ 123 device. The issue was first detected around 12:00 PM, leading up to the device reboot at 14:00 PM. Initial investigation suggests that the root cause of the reboot was a device defect and an out of memory condition.

Timeline of Events:

1. 12:00 PM: Abnormal behavior observed on the XYZ 123 device. Performance degradation and intermittent connectivity issues were reported.
2. 13:00 PM: Network monitoring systems alerted the network operations team about the ongoing issues with the 123 device.
3. 13:30 PM: Troubleshooting efforts began to identify the root cause of the problem. System logs were analyzed, and device diagnostics were performed.
4. 14:00 PM: The XYZ 123 device unexpectedly rebooted, resulting in a temporary disruption of network services.
5. 14:10 PM: Post-reboot, the 123 device successfully recovered, and network connectivity was restored.
6. 14:30 PM: Detailed analysis of the device logs and diagnostic information indicated a device defect and an out of memory condition as possible causes for the unexpected reboot.
7. 15:00 PM: The incident was escalated to XYZ technical support for further investigation and assistance in resolving the device defect.

Affected Systems: The unexpected reboot of the XYZ 123 device impacted the following systems and services:

1. Network connectivity within the affected device's domain.

2. Services relying on the 123 device, such as routing, switching, and traffic forwarding.

Conclusion: The unexpected reboot of the XYZ 123 device on July 6th, 2022, was caused by a device defect and an out of memory condition. The network operations team is actively working with XYZ technical support to resolve the underlying issues and prevent a recurrence. Regular updates will be provided as the investigation progresses and mitigation measures are implemented.

` `

` ` `

Incident Report: Meeting management system outage and active meeting control failed.

Date and Time: July 6, 2022, at 14:00 PM (UTC)

Date and Time of Restoration: July 6, 2022, at 14:10 PM (UTC)

Summary: On July 6, 2022, at 14:00 PM (UTC), an unexpected issue occurred within the system, affecting a total of 23 video conferences and 200 video endpoints. The incident resulted in the inability of three meeting operators to perform active meeting control for meetings numbered 2004, 2007, and 3400. The root cause of the issue was suspected to be a network-related problem.

Timeline of Events:

8. 14:00 PM (UTC): The incident was reported as meeting operators started experiencing difficulties in actively controlling meetings 2004, 2007, and 3400. Symptoms suggested a network-related issue.

9. 14:02 PM (UTC): The system management team was immediately notified about the ongoing incident, and an investigation was initiated.

10. 14:04 PM (UTC): Initial analysis indicated a potential network issue affecting the meeting control capabilities for the specified meetings. The team began troubleshooting the network infrastructure.

11. 14:07 PM (UTC): Further investigation revealed a network failure in the routers responsible for handling the traffic of the affected video endpoints.

12. 14:10 PM (UTC): The network service is restored, and the meeting operators regained active meeting control for all affected meetings, including meetings 2004, 2007, and 3400.

Resolution: The incident was resolved at 14:10 PM (UTC) on July 6, 2022, after affected network routers rebooted and restarting the affected services, the system management team was able to restore the active meeting control functionality for all impacted video conferences.

Affected Services:

3. Active meeting control for meetings 2004, 2007, and 3400.

4. Network connectivity for 200 video endpoints participating in the affected meetings.

Root Cause Analysis: The root cause of the incident was identified as a network issue in the routers responsible for handling the traffic of the affected video endpoints. The network issue caused disruptions in the active meeting control capabilities, preventing the meeting operators from effectively managing meetings 2004, 2007, and 3400.

Conclusion: The network-related issue impacting 23 video conferences and 200 video endpoints, resulting in the inability of meeting operators to actively control meetings 2004, 2007, and 3400, was successfully resolved by the system management

` ` `

Under the presented techniques, the above-described prompt (or model input) may be summarized to generate a network troubleshooting report where such a report may contain the different sections that were identified in Table 1, above. Following that paradigm, under the illustrative example the model output may comprise:

Incident Title: Network Failure on Routers Affecting Meeting Management System

Date and Time: July 6, 2022, at 14:00 PM (UTC)

Date and Time of Restoration: July 6, 2022, at 14:10 PM (UTC)

Summary: A network failure occurred on the routers responsible for handling the traffic of 200 video endpoints, affecting the meeting management system and causing three meeting operators to lose active meeting control for meetings numbered 2004, 2007, and 3400. The network service was restored after affected network routers rebooted and restarting the affected services. The root cause of the network failure was suspected to be a memory issue on the XYZ 123 device, as indicated by the syslog messages collected from the device.

Timeline of Events:

At 13:55 PM (UTC), the network monitoring system detected a high memory usage on the XYZ 123 device and alerted the network administrator.

At 14:00 PM (UTC), the XYZ 123 device ran out of memory and invoked the oom-killer process to free up some memory. This caused the device to crash and reboot unexpectedly.

At 14:01 PM (UTC), the network failure affected the routers connected to the XYZ 123 device and disrupted the traffic of 200 video endpoints. This resulted in three meeting operators losing active meeting control for meetings numbered 2004, 2007, and 3400.

At 14:05 PM (UTC), the XYZ 123 device completed rebooting and restored its normal function.

At 14:07 PM (UTC), the network administrator restarted the affected routers and verified their connectivity and performance.

At 14:10 PM (UTC), the network service was fully restored and the meeting management system resumed its normal operation. The three meeting operators regained active meeting control for their meetings.

Resolution: The network service was restored by rebooting the affected network routers and restarting the affected services.

Affected Services: The network failure affected the following services: Meeting management system, active meeting control, video endpoint traffic.

Root Cause Analysis: The root cause of the network failure was suspected to be a memory issue on the XYZ 123 device, as indicated by the syslog messages collected from the device. The device was running out of memory and invoked the oom-killer process to free up some memory. This could be related to the device defect and the out of memory condition that caused the unexpected reboot.

Conclusion: The network failure was a rare and unexpected event that caused a temporary disruption of the network service and affected some meeting operators. The network administrator took prompt actions to restore the service and minimize the impact. The root cause of the network failure was suspected to be a memory issue on the XYZ 123 device, which needs further investigation and resolution.

In summary, techniques have been presented that support the automatic generation of refined and summarized text from a syslog message sequence. Aspects of the presented techniques employ an abstractive syslog summarization LLM that is trained with contrastive learning and then fine-tuned using a LoRA methodology. Under further aspects of the presented techniques, auxiliary text (such as network incident reports and application incident reports) is added to the prompt of the input of the LLM model to help the model generate a richer syslog summarization.