International Conference on Actuarial Sciences (ICActS)
Available on-line at: www.sicash.sliit.lk

Automated Log Parsing and Anomaly Detection Using BERT and GPT-2: A Large Language Model Approach for IT Systems

W. W. N. C. Sathyanjana[1*], H. M. K. T. Gunawardhane[1], B. T. G. S. Kumara Samantha[2], S. Perera[3]

[1]*Department of Computing & Information Systems, Faculty of Computing, Sabaragamuwa University of Sri Lanka, Sri Lanka.*
[2]*Department Of Data Science, Faculty of Computing, Sabaragamuwa University Of Sri Lanka, Sri Lanka.*
[3]*WSO2 Inc.*

Corresponding author*: chathuminasathyanjana147741@gmail.com

Abstract

Logs are important for diagnosing and understanding the security and operations of IT systems. In these spectacles, the sheer volume of data and their inherent complexity do not allow for an isolated approach. Issues of scalability and adaptability majorly divest most rule-based systems in log analysis. This paper proposes an automatic approach that employs state-of-the-art Large Language Models to detect anomalies, suggest parsing templates, and improve log quality. The suggested system will try to integrate and analyse an Anomaly Detection module for identifying outliers and threats to security, a Pattern Recognition Engine for identifying semantic relations, and a Log Parsing Module for deriving structured patterns. All three collectively serve to enhance efficiency, adaptability, and real-time detection of the log analysis process. Before any LLM-based processing, the results of these pre-processing steps are put through tokenization and normalization. The system was evaluated in a combination of 16 log sources with over 32,000 entries. The model attained an accuracy of 96% in classification; this shows that it performs well in identifying complex log structures and detecting anomalies. Compared to standard approaches, the framework reduces manual processes and increases interpretability on a large scale across diverse environments of IT. The paper describes a structured approach in AI-powered log analysis, which automates essential procedures to offer improved system reliability, as well as real-time security monitoring. Further directions include real-time streaming analysis, addressing ethical concerns in log data processing, and enhancing explain ability.

Keywords: Anomaly Detection, Large Language Models (LLM), Log Analysis

Introduction

System logs are key to modern IT infrastructures and are essential to their monitoring, diagnostics, performance tracking, and security assurance. They record in real-time the events, errors, and user activities occurring in the system and are used by IT administrators to notice faults, check for any outliers, and keep an overall health status of the system. However, with the digital landscape growing by leaps and bounds, the volume, variety, and velocity of log data have grown manifold (Zhang et al., 2024). This ever-increasing volume of logs throws serious challenges in front of traditional parsing and analysis mechanisms, particularly those being manual or rule-based (Boyagane et al., 2022). The rule-based systems, although effective in well-defined environments, are also inefficient at matching diverse log formats generated by the heterogeneous systems (Vaarandi, 2003). There is very little in their defence that they lack generalization of new log patterns, and then they are only good if a human is

https://doi.org/10.54389/NIAW2551

present to continuously refine the log parser. And while the classical machine learning techniques are advantageous to a certain degree, they give a lesser degree of flexibility, break under the enormous weight of scaling data, and are not sufficient, for all practical purposes, to directly analyse an end-of-the-day cohesive situational state ensuring suitable amelioration (Ma et al., 2023).

To overcome the challenge, novel artificial intelligence (AI) tools, particularly Large Language Models (LLMs), have shown great promise in the field of unstructured text comprehension. BERT (Bidirectional Encoder Representations from Transformers) and GPT-2 (Generative Pretrained Transformer-2) have indeed proven very resourceful in a number of natural language processing (NLP) tasks (Boyagane et al., 2022). These models comprehend context, identify patterns, and generalize well over unseen data, thus forming very attractive candidates for the task of automating log parsing and analysis (Boyagane et al., 2022; Vaarandi, 2003).

In this paper, we exploit the benefits of LLMs in creating a framework to assist in log template extraction and anomaly detection. We use BERT and GPT-2 specifically to translate unstructured logs into structured formats that offer real-time monitoring and intelligent anomaly detection (Shah et al., 2022). The performance of the two models was evaluated on a diverse dataset of more than 20,000 log entries collected from real-world platforms such as Log hub, GitHub, and Kaggle (Boyagane et al., 2022; Ji et al., 2024).

Methodology

The proposed methodology uses a six-stage pipeline for automatic log parsing and anomaly detection using transformer-based LLMs (BERT and GPT-2). The process starts with dataset collection, normalization, tokenization, and timestamp-formatting. Template identification is performed using label encoding and vectorization. Both models are fine-tuned for classifying the log entries, with the anomaly detection capacity added to it through the application of the Isolation Forest algorithm. The system is evaluated with a variety of performance metrics, giving a comprehensive view for comparison between BERT and GPT-2.

Data and data pre-processing

The preparation of training data for this research consisted of a diverse and comprehensive set of more than 32,000 logged events emanating from 16 different publicly accessible sources. The dataset comprised sources such as LogHub, GitHub repositories of log files, and Kaggle log analyses, whereby the datasets themselves were varied in terms of system, application, or security logs, duly wrought in different formats with different message patterns and levels of severity. For ensuring data quality and model compatibility with transformer-based frameworks, the following preprocessing horizon was applied:

Pre-processing steps

Noise Removal: recognition of irrelevant elements such as IPs, hexadecimal number formats (e.g., 0xA0B1), stack traces, etc., and removal therefrom. **Timestamp Normalization:** All timestamps were converted into ISO 8601 format (YYYY-MM-DD HH:MM:SS) to conform to a unified temporal structure. **Structural Parsing:** Parsing of the logs to extract elements such as log levels (INFO, WARN, ERROR), component identifiers, message content, and thread identifiers. **Domain Normalization:** Mapped technical jargon (HTTP codes, SQL states, etc.) to a uniform representation. **Tokenization:** Used model-specific tokenizers (BERT and GPT-2), inserting [CLS] and [SEP] tokens for classification. **Padding & Truncation:** All log entries were padded or truncated to the length of 128 tokens to maintain

uniform input length, which were important considerations when batch processing during training. The steps ensured that the models accommodated syntactic irregularities while preserving semantic integrity in varied log structures.

Template identification

To perform classification and anomaly detection, the logs were mapped to templates using label encoding. Each log message was then assigned a numerical class using the method, Pandas. astype('category').cat.codes to allow efficient learning of the event types. The dataset was segregated with a 0.8 and 0.2 ratio with stratified sampling so that there were equal label distributions. The following two complementary feature extraction mechanisms wereused.

- TF-IDF Vectorization: Capture statistical significance of terms across logs. Top 5,000 terms were retained following feature selection for better model generalization.
- Word Embeddings: Serve as model-specific embeddings capturing contextual nuances in log entries.

A custom PyTorch Dataset class was built to generate input dictionaries with input_ids, attention_masks, and encoded labels, which allowed efficient streaming of data into both models during fine-tuning. Special care was taken to preserve meaningful sequences and detect variable components within log templates as these are vital for accurate classification and anomaly detection.

Model architecture

The fine-tuning process involved BERT- and GPT-2-based models for the downstream log template classification task. Both models, having been trained beforehand on large-scale corpora, were chosen to exploit their complementary strengths, namely BERT's bidirectional understanding and GPT-2's ability to learn in an autoregressive manner. Model Architecture: - **Transform Layers**: There were altogether 12 layers in each model, having 768 hidden units and 12 multi-attention heads. **Classification Head**: A fully connected layer with SoftMax was added to transform the outputs of the transformer into probabilities of the classes. **Fine-tuning**: Both were further fine-tuned on the labelled log dataset via transfer learning to adapt the pre-trained language knowledge to log-specific language formats.

Training configuration

These models were trained by employing AdamW optimizer with a learning rate of 5e-5 and the linear warmup for 500 steps. The main hyper-parameters are shown in Table 1.

Table 1: *Key hyperparameters*

Parameter	Value
Epochs	3
Batch Size	32
Max Sequence Length	128 Tokens
Optimizer	AdamW
Warmup Steps	500

Training was done in Google Colab, aiming to strike a balance between performance and resources. A single epoch was trained to keep computational cost down, with early stopping put in place to prevent overfitting.

Anomaly detection

Anomaly detection in a system could be implemented by interfacing the Isolation Forest algorithm with the model outputs. The features used to pinpoint anomalies included:

- **Predicted Template Labels**: Logs belonging to infrequent or unexpected templates could be considered anomalies., **Log Occurrence Frequency**: The frequency with which log events occur within certain time frames was monitored for unusual deviations, **Sequential Dependencies**: Logs were considered as sequences to detect irregular transitions in event flow.

So, Isolation Forest isolates data points by means of random partitioning. Labels -1 were assigned to anomalies (outliers), while normal entries were labelled 1. Validation of the detection
was performed with visualization and statistical frequency analysis.

Evaluation matrix

The performance of model was assessed using: Accuracy, Precision, Recall, F1-Score: For classification reliability, MAE and MSE: For prediction error analysis, Confusion Matrix: For class-specific performance, AUC-ROC: For anomaly detection effectiveness.

Results and Discussion

Metrics such as accuracy, precision, recall, F1-Score, MAE, MSE, confusion matrices, and anomaly detection outputs measure the performance of the automated log parsing framework using LLMs.

Parsing Accuracy and Classification Performance

Between the two they tested, GPT-2 obtained the most accuracy in the parsing domain: a scoring of 97.44% against BERT's 96.61%. Whereas BERT might shine in cleaner data and in lower noise levels, GPT-2 seems to be better accommodating the heavy noise and unstructured nature of log data in the real-world system-monitoring domain. Hence, with generative pretraining, GPT-2 may adapt flexibly to any log format without any kind of prior fixed structure in place. On the other side of the coin, BERT is said to excel in semantic understanding of logs, as witnessed by lower errors (i.e., MAE and MSE). The bidirectional attentions of BERT likely help capture contextual dependencies better, especially when the log entries are semantically rich, such as system diagnostics or application-level logs.

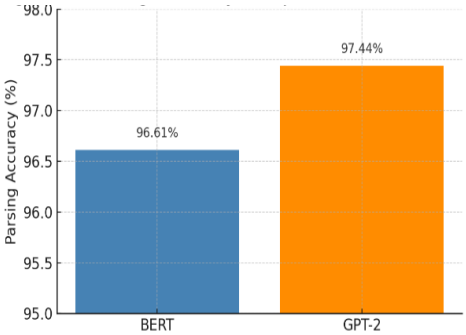

Figure. *1: Accuracy - BERT vs GPT-2*

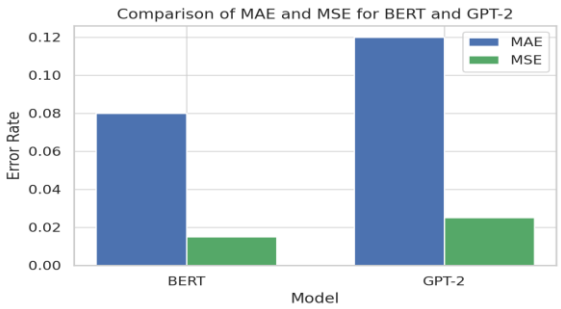

Figure. 2: *Comparison of MAE and MSE*

Confusion matrix and ROC curves

The confusion matrices for the models show their respective performance in classifying log events into three classes: Routine, Warning, and Critical. Both showed true positives at high rates in all classes, but

BERT gave a higher false-positive rate for the Critical class, reinforcing that it is more sensitive to severe log events.

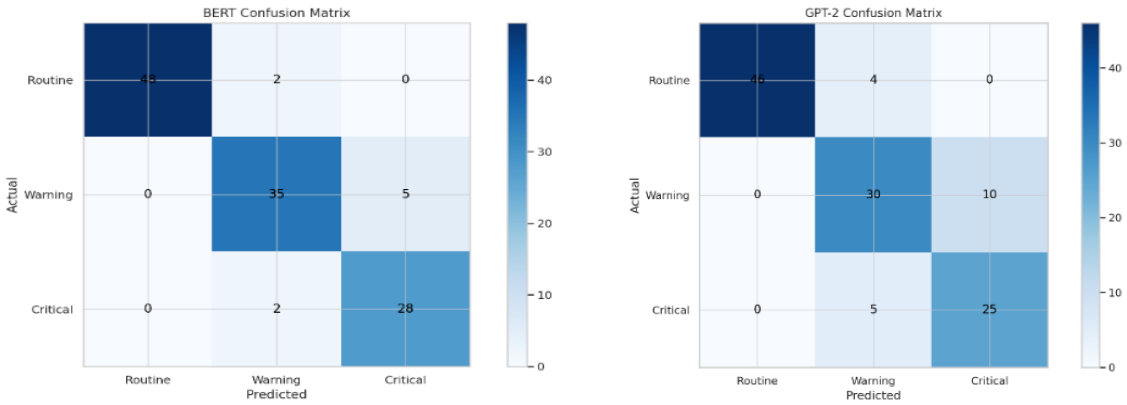

Figure 3: *Confusion matrix of BERT and GPT-2*

The ROC curve shows that both models achieved an AUC close to 0.98, indicating excellent overall classification performance. However, GPT-2 slightly surpassed BERT in micro-average AUC, affirming its reliability in multi-class classification tasks involving unbalanced datasets.

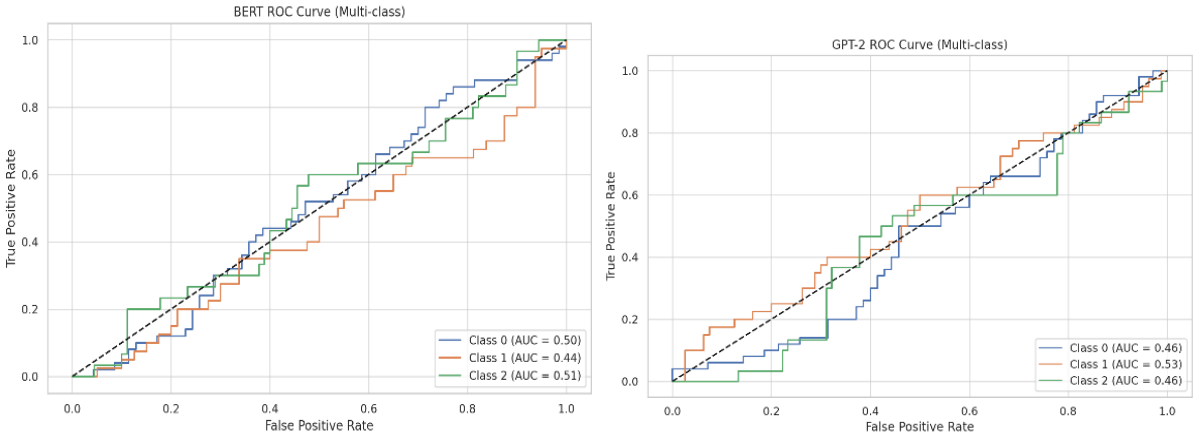

Figure 4: *BERT and GPT-2 ROC Curves*

Anomaly detection insights

Anomaly detection was done using the Isolation Forest algorithm directly on the outputs of the models. GPT-2 shows a different level of sensitivity to anomalies than BERT, recognizing many more anomalies in log sequences: about 2,000 anomalies from 30,000 entries, in contrast to 100 anomalies from 2,000 entries by BERT. This shows that GPT-2's broader interpretation of anomalies lies especially in longer and less structured logs. In contrast, BERT has more precision in the anomalies it identifies, with fewer false-positive alarms on structured event sequences.

Model Trade-offs and Practical Implications

A key takeaway from the evaluation is the inherent trade-off between contextual accuracy and generalization speed. BERT supplies deeper semantic understanding and consistent parsing for structured logs; conversely, its computational cost and longer inference time weigh on the model. GPT-2, on the other hand, may inject noise into high-context log sequences given its autoregressive nature but is faster, more adaptive, and scales well for real-time or streaming logs. The findings strongly

suggest that BERT is more suited for log environments requiring high-level interpretability and structured consistency, while GPT-2 is drilled down for large-scale deployments and dynamic log sources, such as cloud infrastructure or network systems.

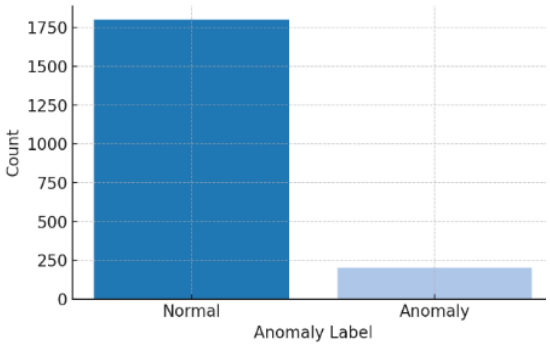

Figure 5*: BERT Anomaly*

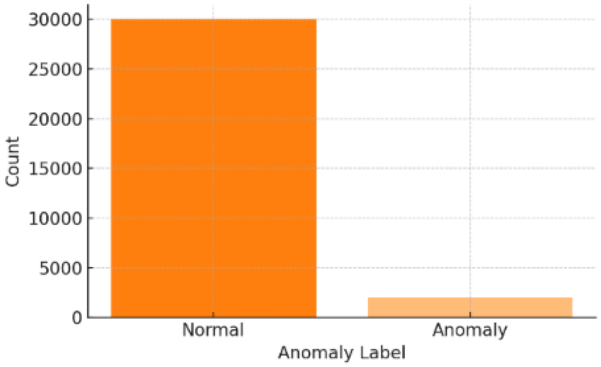

Figure 6 : *GPT-2 Anomaly*

Conclusions

An automated framework for log parsing and anomaly detection was presented in this study, applying the Large Language Models (BERT and GPT-2). By utilizing the context-based understanding and language modelling abilities of the proposed transformer architectures, the system found out log templates and anomalous patterns across diversely logged real-world data. The comparative study revealed that GPT-2 had better parsing accuracy of 97.44%, thus adapting better to noisy, unstructured log environments. However, BERT performed better on error metrics like MAE and MSE, meaning that it was more semantically consistent and contextually accurate in structured log settings. The use of an Isolation Forest on the output of these models further improved the detection of anomalies, leading to the robust detection of irregular system behaviour. A few important constraints like few training epochs, availability of labelled datasets, and class imbalance are acknowledged, and solutions considered can include the pruning of the model, semi-supervised training, and integration into monitoring tools in real-time.

Acknowledgements

We thanked Mr. H.M.K.T. Gunawardhana, Prof. B.T.G.S. Kumara, Mr. Srinath Perera and Department of Computing and Information Systems Sabaragamuwa University of Sri Lanka for their support.

References

Boyagane, I., Katulanda, O., Ranathunga, S., & Perera, S. (2022). Vue4logs – Automatic structuring of heterogeneous computer system logs. *arXiv*. https://doi.org/10.48550/arXiv.2202.07504

Ji, Y., Liu, Y., Yao, F., He, M., Tao, S., Zhao, X., Chang, S., Yang, X., Meng, W., Xie, Y., Chen, B., & Yang, H. (2024). Adapting large language models to log analysis with interpretable domain knowledge. *arXiv*. https://doi.org/10.48550/arXiv.2412.01377

Ma, Z., Chen, A. R., Kim, D. J., Chen, T.-H., & Wang, S. (2023). LLM Parser: An exploratory study on using large language models for log parsing. *Proceedings of the ACM Web Conference 2023* (Article No. 99, pp. 1-13). ACM. https://doi.org/10.1145/3597503.3639150

Shah, A. H., Pasha, D., Zadeh, E. H., & Konur, S. (2022). Automated log analysis and anomaly detection using machine learning. *Fuzzy systems and data mining VIII* (pp.137–147). IOS Press. https://doi.org/10.3233/FAIA220378

Vaarandi, R. (2003). A data clustering algorithm for mining patterns from event logs. *Proceedings of the 2003 IEEE Workshop on IP Operations and Management (IPOM)* (pp. 119–126). IEEE. https://doi.org/10.1109/IPOM.2003.1251233

Zhang, L., Jia, T., Jia, M., Wu, Y., Liu, A., Yang, Y., Wu, Z., Hu, X., Yu, P. S., & Li, Y. (2024). A survey of AIOps for failure management in the era of large language models. *arXiv.* https://doi.org/10.48550/arXiv.2406.11213