BERT를 활용한 로그 시퀀스 내 로그 단위 이상탐지를 위한 프레임워크*

권태완 ○ 이찬재 정상현 김용신 이태희

오케스트로 주식회사

tw.kwon@oketro.com, cjlee@oketro.com, sh.jung@oketro.com,
ys.kim2@oketro.com, th.lee@oketro.com

Framework for Log-Level Anomaly Detection in a Log Sequence using Bidirectional Encoder Representations from Transformers

Taewan Kwon○ Chanjae Lee Sanghyeon Jung Yongshin Kim Taehee Lee

OKESTRO Co., Ltd.

요 약

로그 기반 이상탐지는 시스템이 동작하는 동안 발생한 이벤트에 의해 동시적으로 발생한 로그 메시지의 집합 또는 시퀀스에 대해 이상 여부를 탐지하는 기술이다. 주어진 로그 시퀀스의 맥락을 더욱 잘 포착하기 위해 최근 로그 기반 이상탐지에 BERT 등의 언어 모델을 채택한 연구들이 증가하고 있으나, 대부분의 연구가 여전히 주어진 로그 집합에 대해 단순히 이진분류 하는 방식에 그치고 있다. 이러한 방식은 해당 시퀀스 내에서 어떤 로그 메시지가 에러에 해당하는지 특정하지 못한다는 문제점이 있다. 본 논문은 BERT를 이용하여 로그 시퀀스에 대한 예측에 더해, 주어진 시퀀스의 맥락에서 각 로그 메시지에 대한 이상탐지까지 가능한, 로그 메시지 단위의 이상탐지 프레임워크 FineLog를 새로이 제시하고자 한다. 그리고 본 연구는 실험을 통해 기존의 시퀀스 단위 이상탐지에서도 FineLog가 높은 성능을 기록할 뿐만 아니라, 로그 단위 이상탐지에서도 높은 성능을 기록하는 것을 보여준다.

1. 서론

로그 기반 이상탐지는 시스템이 동작하는 동안 발생한 이벤트에 의해 동시적으로 발생한 로그 메시지의 집합 또는 시퀀스에 대해 이상 여부를 탐지하는 기술이다. 최근 시스템의 규모와 복잡성이 증가함에 따라, 시스템이 발생시키는 로그의 양은 매우 방대해지고 로그 메시지의 내용 또한 다양해지고 있다. 최근의 연구들은 시스템 로그의 높아지는 복잡성을 해결하기 위해, BERT[1] 등의 자연어 모델을 로그 기반 이상탐지에 도입, 발전시키고 있다[2, 3, 4, 5].

이러한 연구의 초기 흐름은 주로 로그 메시지를 일종의 텍스트 시퀀스로서 보고, 단순히 임베딩 용도로 BERT를 사용하는데 그쳤다[2, 3]. 한편 최근에는 BERT를 시스템 로그라는 특정 도메인에 더욱 피팅시키기 위해 BERT를 시스템 로그 데이터셋에서 사전학습 시키거나[4, 5], 특수한 학습 전략으로 학습시키는 연구가 진행되고 있다[3].

하지만 이처럼 다양한 시도들이 이루어지고 있음에도 불구하고, 대다수의 연구들은 여전히 주어진 시퀀스에 대한 예측만을 수행한다. 이러한 접근법은 주어진 로그 시퀀스의 맥락 안에서 어떤 로그 메시지가 이상에 해당하는지 특정하지 못한다. 반면 본 연구는 BERT를 이용한 대표적인 요약 모델, BertSum[6]을 활용하여, 주어진 로그 시퀀스의 맥락 안에서 각 로그 메시지의 이상 여부까지 판단 가능한, 로그 메시지 단위의 이상탐지 프레임워크를 제시한다.

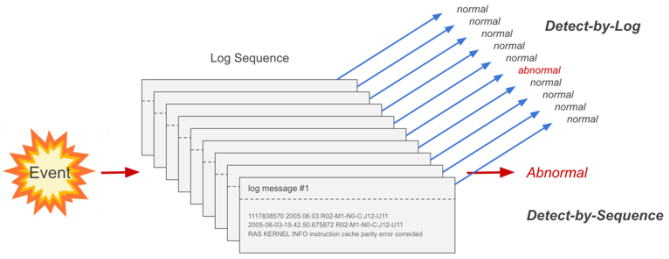

그림 1. FineLog의 전체 프레임워크

(i) BERT와 로그 시퀀스의 맥락 정보로부터 각 로그 키의 표현을 얻어내고 이들 간 연관성을 계산하여, 이들

* 이 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2021-0-00256, 클라우드 자원의 지능적 관리를 위한 이종 가상화(VM + Container) 통합 운용 기술 개발)

각각에 대한 이상탐지를 수행한다.

(ii) BERT를 사전학습 시키거나 단순히 임베딩으로 사용한 기존 연구와 다르게, BERT를 시스템 로그라는 특정 도메인에 파인튜닝 시킨다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 제시하는 모델 FineLog의 구조를 베이스 모델과 인코더로 나누어 설명한다. 3장에서는 BGL 데이터셋에서 진행한 시퀀스 단위 이상탐지와 로그 단위 이상탐지의 실험 결과를 각각 보여주고, 마지막 4장에서 결론을 맺는다.

2. 모델 구조

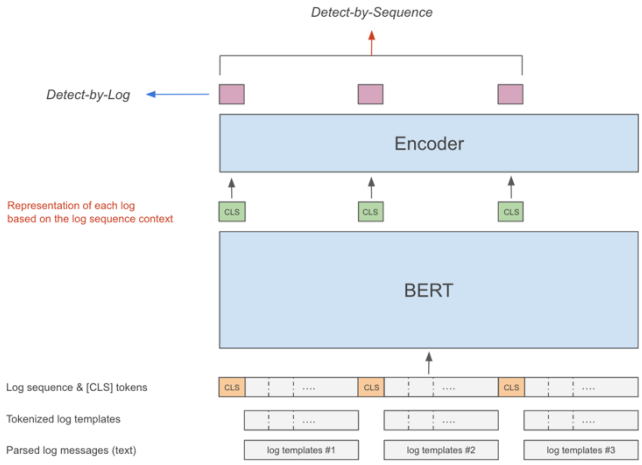

그림 2. FineLog의 모델 구조와 입력 시퀀스 형식

2.1. 베이스 모델

로그 메시지 또는 로그 키를 일종의 텍스트 시퀀스로 간주하고, 이들의 특징을 추출하기 위해 BERT를 채택하는 연구는 이미 많이 공개되었다[2, 3, 4, 5]. 그들은 주로 1개의 로그 메시지를 독립적으로 BERT에 전달, 임베딩 벡터로 변환한 다음, 로그 시퀀스에 포함된 로그 메시지 간 맥락 정보를 포착하기 위해 로그 메시지의 집합을 RNN, LSTM 또는 주로 2개 레이어로 구성된 트랜스포머 등으로 구현된 인코더에 전달한다[2, 3].

로그 기반 이상탐지에서 하나의 로그 메시지 또는 로그 키는 해당 로그 메시지가 포함된 시퀀스에 종속적이다. 하나의 로그 메시지는 자신이 포함된 로그 시퀀스, 더 정확히는 시퀀스를 구성하는 다른 로그 메시지들에 따라 정상일 수도, 이상일 수도 있다. 이는 로그 시퀀스의 이상 여부에 있어 시퀀스의 맥락 정보가 매우 중요함을 의미한다. 반면 BERT가 텍스트 시퀀스에서 높은 성능을 보이는 이유는 그것이 현재 레이어에서 토큰 간 맥락 정보를 포착하고 이를 각 토큰의 임베딩 벡터에 내포시켜, 다음 레이어로 전달하기 때문이다[1].

BERT를 이용해 각 로그 메시지 맥락 정보를 독립적으로 포착하고, 이후 인코더에서 로그 시퀀스의 전체 맥락 정보를 포착하는 기존 연구와 달리, FineLog는 BERT에서부터 로그 시퀀스의 전체 맥락 정보를 포착하고 이를 각 토큰의 임베딩 벡터에 내포시킨다. 이를 위해 본 연구에서는 베이스 모델인 BERT의 입력으로 로그 메시지들을 텍스트 레벨에서 이어 붙인, 로그 시퀀스의 전체 텍스트를 전달한다. 또한 BERT를 통과한 이후 각 로그 키의 적절한 표현을 얻기 위해, 요약 모델에서 매우 일반적으로 사용되는 방식인, 각각의 로그 키 앞에 CLS 토큰을 붙이는 방식을 사용한다[6]. CLS 토큰은 문장의 시작을 알리는 토큰으로, 일반적으로 BERT 인코딩 이후 문장 전체를 표현하기 위한 토큰으로서 사용된다[1]. 이후 인코더에는 각 로그 키를 대표하는 CLS 토큰의 임베딩 벡터들만이 전달된다.

2.2. 인코더

FineLog의 인코더는 두 개의 예측 값을 반환한다. 하나는 로그 시퀀스에 대해 예측하는 'Detect-by-Sequence'이고, 다른 하나는 주어진 로그 시퀀스의 맥락에서 각 로그 메시지마다 이상 여부를 예측하는 'Detect-by-Log'이다.

Detect-by-Log

인코더는 기본적으로 두 개의 트랜스포머 레이어로 구성된다. 하지만 기존 연구의 인코더와 다른 점은 각 로그 키를 대표하는 CLS 토큰에 대해 BERT로부터 얻은 각각의 임베딩 벡터에 대하여 이진 분류 한다는 점이다. 이는 각 로그 메시지가 주어진 시퀀스의 맥락 안에서 정상인지 이상인지를 판별하는 것이다.

Detect-by-Sequence

시스템 로그의 벤치마크 데이터셋에서 하나의 시퀀스에 대한 레이블은, 해당 시퀀스에 포함된 로그 메시지들 중 1개 이상의 이상 레이블을 가진 메시지가 포함되어 있는지 여부에 의해 결정된다. 이에 따라 본 연구는 다소 강건(robust)하게, Detect-by-Log에서 1개 이상의 이상 예측이 존재하는 경우 해당 시퀀스 전체를 이상으로 예측한다.

3. 실험

데이터셋

본 연구의 실험은 4,747,963개의 로그 메시지 중 348,460개의 이상 로그를 가진 벤치마크 데이터셋 BGL에 대해 진행되었다. 본 연구는 파싱을 위해 Drain을 사용했으며, 기존 연구들에 따라 BGL의 전체 데이터셋을 시간 순으로 정렬한 뒤, 윈도우 크기가 20인 슬라이딩 윈도우 기법을 적용했다[2, 3, 4, 5]. 평가 데이터셋으로는 전체 시퀀스 데이터셋의 20%가 사용되었다[2, 3, 4, 5]. 마지막으로, 파싱된 각 로그 키 앞에는 로그 레벨 정보가 추가되어 사용되었다.

트레이닝

학습 과정에서 BERT를 단순히 전처리 단계에서의 임베딩 용도로만 사용하는 NeuralLog[2], TransLog[3] 의 경우, 시스템 로그 도메인에 매우 피팅 된 학습 전략을 사용한다. 또한 LogBERT[4], LAnoBERT[5] 은 BERT를 시스템 로그 데이터셋에서 직접 사전학습 시키는, 비용이 매우 많이 드는 학습 전략을 사용한다. 반면 FineLog는 BERT를 다운스트림 태스크에서 학습시키는 매우 일반적인 방식인 파인튜닝으로 학습되었다. 표 1은 오직 파인튜닝으로 학습시키는 것만으로 시스템 로그 도메인에서 높은 성능을 기록하는 것을 보여준다.

구현

베이스 모델의 가중치로 본 연구는 가장 일반적인 bert-base-uncased을 사용했다[1]. 인코더는 각 MHA(Multi-Head Attention)의 헤드 개수가 8, 중간 피처 2048인 2개의 트랜스포머 레이어로 구성되었다[6]. 최적화는 Adam에 의해 이루어졌고, 학습률 스케줄러의 경우, 다음 식에서 warm-up step 10000으로 설정했다[1, 6].

$$lr = 2e^{-3} \cdot \min(\text{step}^{-0.5}, \text{step} \cdot \text{warmup}^{-1.5})$$

실험은 1개의 Nvidia T4에서 진행되었으며, GPU 메모리를 고려해 배치 사이즈는 16, 그레디언트 축적은 2로 진행했다.

3.1. Detect-by-Sequence

Method	Precision	Recall	F_1 Score
LR	0.13	0.93	0.23
SVM	0.97	0.30	0.46
DeepLog	0.8974	0.8278	0.8612
LogAnomaly	0.7312	0.7609	0.7408
NeuralLog (sup.)	0.61	0.78	0.68
TransLog (sup.)	0.98	**0.98**	**0.98**
LogBERT	0.8940	0.9232	0.9083
LAnoBERT	-	-	0.8749
Ours	**0.9836**	0.9404	**0.9616**

표 1. BGL 데이터셋에서 FineLog의 Detect-by-Sequence 결과

표 1은 FineLog가 기존의 프레임워크에서도 얼마나 잘 작동하는지 보여준다. 우선 FineLog의 F1 스코어는 사전학습 된 LogBERT[4] 보다 약 5 이상 높고, SOTA인 TransLog[3] 보다 약 1.4 정도 낮다. 주목할 만한 점은 예측이 강건함에도 불구하고, 정밀도와 재현율이 높고 균형적이라는 점이다.

3.2. Detect-by-Log

Precision	Recall	F_1 Score
0.9894	0.9654	0.9773

표 2. BGL 데이터셋에서 FineLog의 Detect-by-Log 결과

표 2는 본 연구의 주요 목적인 로그 단위 이상탐지에 대한 실험 결과이다. 로그 단위 이상탐지의 경우, 각 로그 메시지의 이상 여부가 로그 시퀀스에 따라 달라질 수 있어, 해당 시퀀스의 맥락을 잘 파악하는 것이 중요하다. 위 실험 결과는 FineLog가 BERT의 입력으로 전달받은 로그 시퀀스의 맥락을 기반으로, 각 로그 메시지의 이상 여부를 잘 파악함을 보여준다.

4. 결론

본 논문은 기존의 시퀀스 단위 이상탐지에 더해, 시퀀스 내 각 로그 메시지에 대한 이상탐지 예측까지 수행하는, 로그 기반 이상탐지 프레임워크 FineLog를 제시했다. 본 연구의 실험은 FineLog가 간단한 파인튜닝만으로 기존의 이상탐지 방식인 시퀀스 단위 이상탐지에서 높은 점수를 기록할 뿐만 아니라, 본 연구에서 제안한 로그 단위 이상탐지에서도 높은 점수를 기록하는 것을 보여준다. 이로써 본 연구는 최근 로그 기반 이상탐지 분야에서 활발하게 사용되는 BERT을 이용해, 일정 개수의 로그 집합에 대해서만 이상탐지를 수행하는 기존의 방식을 넘어, 해당 시퀀스 내에서 에러에 해당하는 로그 메시지가 무엇인지 특정하는 방식을 제안한다.

참고문헌

[1] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." _arXiv preprint arXiv:1810.04805_ (2018).

[2] Le, Van-Hoang, and Hongyu Zhang. "Log-based anomaly detection without log parsing." _2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)_. IEEE, 2021.

[3] Guo, Hongcheng, et al. "Translog: A unified transformer-based framework for log anomaly detection." _arXiv preprint arXiv:2201.00016_ (2021).

[4] Guo, Haixuan, Shuhan Yuan, and Xintao Wu. "Logbert: Log anomaly detection via bert." _2021 international joint conference on neural networks (IJCNN)_. IEEE, 2021.

[5] Lee, Yukyung, Jina Kim, and Pilsung Kang. "LAnoBERT: System log anomaly detection based on BERT masked language model." _arXiv preprint arXiv:2111.09564_ (2021).

[6] Liu, Yang, and Mirella Lapata. "Text summarization with pretrained encoders." _arXiv preprint arXiv:1908.08345_ (2019).