Self-Attentive Classification-Based Anomaly Detection in Unstructured Logs (2020)

20201011 이연재, 김미주

Index

Logsy

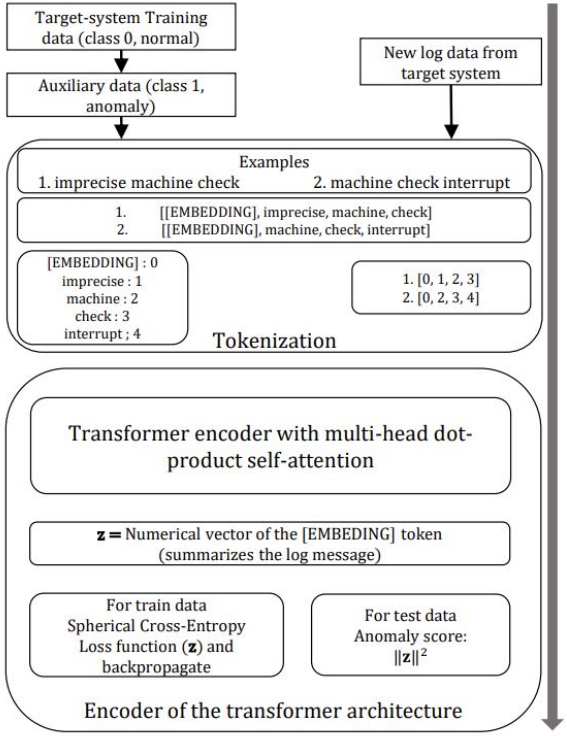


Fig. 1. Overview of the architecture and component details of Logsy.

- log message tokenization
- neural network model

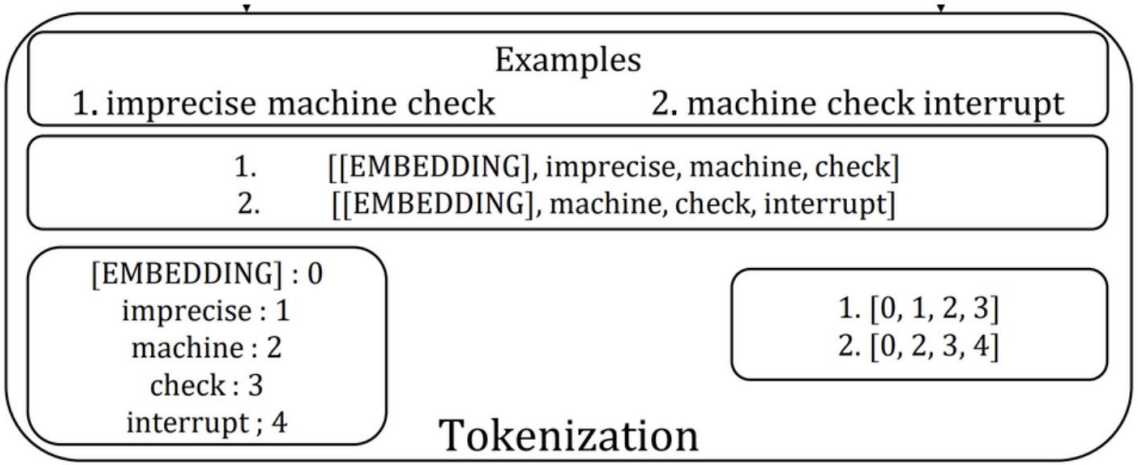
Tokenization

- log message 는 HTTP 및 system path endpoints 에 대해 **filtering** 됨.
    - (e.g., /p/gb2/stella/RAPTOR/)
- 모든 대소문자는 **소문자로 변환**되고 **ASCII 특수문자는 제거**됨.
- Log message 는 **word token 으로 분할(split)** 됨.
- numerical character 가 포함된 모든 token 제거 (log message 에서 변수를 나타냄 —> 별로 중요한 정보가 아님)
- 가장 자주 사용되는 English word 제거(NLTK 의 stop words dictionary 에 포함된 단어들..)
    - (e.g., the and is)
- tokenized log message 앞에는 **'[EMBEDDING]' 라는 special token** 이 붙는다.
    - '[EMBEDDING]' token : 모델이 vector 표현에서 log message 의 context 를 summarize 할 수 있게 한다.
- log message 의 모든 token 은 크기 |V| 인 Vocabulary V 를 형성한다.
    - 각각의 token 은 **integer label $i \in 0, 1, \ldots, |V| - 1$** 로 표현됨.

Tokenization

model 의 direct input : tokenized log message

▼ tokenization 예시



Model

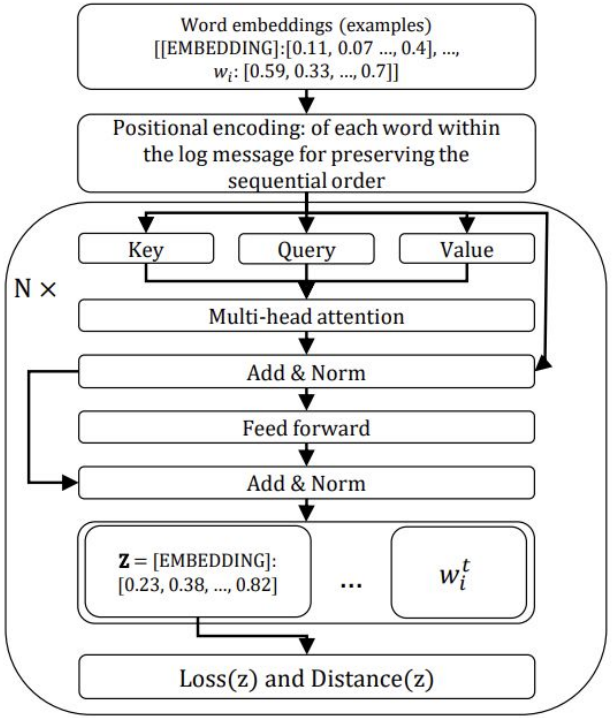


Fig. 2. Transformer encoder architecture with multi-head self-attention.

model 은 input token 에 두 가지 연산을 적용

1. token vectorization(word embedding)
2. positional encoding

multi-head self-attention 기반 transformer encoder

➡ 위 연산을 거친 결과의 벡터를 입력값으로 받는 모델이다.

각각의 log 벡터 표현 'z' 와 각 message 에 대한 anomaly score 를 생성한다.

Model

- word embedding : 입력 토큰(input token) 을 무작위로 수치화된 **벡터 x** 로 변환
- positional encoding : log message 내의 위치를 고려하여 encode
  - positional encoding block 은 sin 및 cos 함수를 기반으로 토큰의 상대 위치를 나타내는 **벡터 n** 을 계산한다.

$$n_{2k} = sin\left(\frac{j}{10000^{\frac{2k}{d}}}\right), \quad n_{2k+1} = cos\left(\frac{j}{10000^{\frac{2k+1}{d}}}\right).$$

  - **d** : 벡터의 size
  - **k** = 0, 1, . . . , d − 1 (index of each element in **n**)
  - **j** = 1, 2, . . . , |ri| (각 token 의 positional index) , |ri| : log message 내 token 의 개수
- x'= word embedding(x) + positional encoding(n)

Model

- 모든 벡터를 한번에 행렬연산으로 처리하기 위해 x' 를 전치한 행렬 X' 을 생성 $\mathbf{x'}^T \in X'$
- 각 입력(X')에 가중치 행렬(W)을 곱하여 Q, K, V 행렬 생성

$$Q_l = X' \times W_l^Q,\ K_l = X' \times W_l^K,\ V_l = X' \times W_l^V$$

- attention 구하기

$$X_l'' = softmax\left(\frac{Q_l \times K_l^T}{\sqrt{w}}\right) \times V_l,\ \text{for } l = 1, 2, \ldots, L.$$

- 축적된 Xl'' 합쳐서 하나의 행렬 X''(M*d) 생성
- original input X' 와 어텐션 과정을 거친 행렬 X'' 을 합쳐서 normalization
  - X' = norm(X' + X'')
- 모델의 마지막 부분에 single linear layer ➔ **log vector representation**
  - **X'** 를 입력으로 받아서 [EMBEDDING] token 이 있는 **행렬의 첫 행(row)**만 추출

Objective function

spherical loss function 제안

- radial classification loss 사용
  - enforces a compact hyperspherical decision region for the normal samples.
  - normal sample 에 대해 compact 한 초구형 결정 지역을 가지게 함.
- normal sample 들의 compact 함을 보장.
- normal sample 들이 구의 중심 c=0 에 모이도록 함

$$\frac{1}{n}\sum_{i=1}^{n}(1-y_i)\|\phi(\mathbf{x_i};\theta)\|^2 \; -y_i \log(1-\exp(-\|\phi(\mathbf{x_i};\theta)\|^2))$$

- normal sample(yi=0) 의 경우, loss function 은 c(중심) 까지의 거리를 최소화 한다.
  - 왼쪽 항의 값이 작아짐.
- anomaly sample(yi=1) 인 경우
  - 오른쪽 항의 값이 커짐.
- spherical classifier 에서의 possible problem → 제안된 function 은 trivial solution 을 해결하려고 하지 않음.

Anomaly score and detecting anomalies

Anomaly score A(xi) : **log vector** ('EMBEDDING' 토큰에서 얻은 값)와 **구의 중심 c 까지의 거리**로 정의한다.

$$A(\mathbf{x_i}) = \|\phi(\mathbf{x_i}; \theta)\|^2$$

- threshold E 를 사용하여 그보다 높으면 anomaly, 낮으면 normal 로 봄.

$$A(\mathbf{x_i}) > \mathcal{E}$$

- anomaly scores : A(xi) > E ➔ the sample is an anomaly
- anomaly scores : A(xi) < E ➔ the sample is an normal

Including expert knowledge

컴퓨터 시스템 관리자가 log event 의 일부를 수동으로 검사하고, label 을 제공해왔음

➡ Logsy 를 그러한 label 된 log 를 통합하는데 활용할 수 있다!

operator-labeled sample

- 현실적이고 고가의 anomaly samples —> 추가하면 성능 향상
- operator-labeled sample 은 보조데이터와 함께 추가되어 retrain 되어야 함/ 또는 일반데이터+보조데이터로 model pre-training 후에 labeled-sample 로 fine-tunning 해야 한다.
- 보조데이터를 labeled sample 로 대체하는 경우
  - model 은 몇 개의 epoch 에서만 parameter 를 튜닝할 수 있음
  - This preserves the already learned information from the larger auxiliary dataset as a bias to the fine-tuning procedure.

Vector representations of the logs

- Logsy 에서 얻은 numerical log representations 은 다른 log anomaly detection method 에도 사용할 수 있다. ➜ 성능 향상
  - ex.) PCA (TF-IDF in previous log-based anomaly detection methods)
- The transformed vector of the [’EMBEDDING’] token
  - 임베딩 토큰의 변환된 벡터는 log message 의 context 를 요약해준다.
- spherical classification decision boundary(구형의 분류 결정 경계;점선) 를 사용하여
  - normal sample 들은 구의 중심에 가깝도록, 그리고 각자끼리도 가깝도록 한다.

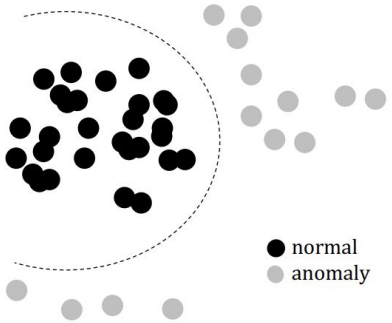


Fig. 3. Ideal distribution of the log vector representations in space.

5. Evaluation

Experimental Setup

평가를 위한 target system dataset	**additional dataset**
Blue Gene/L, Spirit, Thunderbird (HPC system 의 open real-world dataset)	HPC **RAS** log dataset → 보조데이터의 풍부함을 위해 추가하는 데이터

target vs auxiliary 분할

- 각 target dataset 에 대해 anomaly class 을 나타내는 보조 데이터로서 나머지 데이터 세트의 로그를 사용한다.
- target vs auxiliary 분할을 신경써서 해야 함(정보가 누출되지 않도록)

예시) target system(관심있는 system) : Blue Gene/L 이면,

Thunderbird, Spirit, and RAS(나머지 system) 의 negative samples 부분 ⇒ auxiliary dataset 으로 사용됨(anomaly class 나타냄)

Experimental Setup

dataset 의 주요 특성 정리

DATASET DETAILS.

System	#Messages	#Anomalies	#Anomalies5m	#Unique Log messages in test and not in train for every split					total unique messages
				10%	20%	40%	60%	80%	
Blue Gene/L	4747963	348460	348460	2679	2621	2256	2231	465	4486
Thunderbird	211212192	3248239	226287	334	127	71	27	12	3279
Spirit	272298969	172816564	764890	1091	1028	297	129	73	3441

- Thunderbird and Spirit : 2억 개 이상의 꽤 큰 dataset
- 처음 500만개의 log 로 데이터 사이즈 제한(계산 시간 목적을 위해/ timestamp, log message 를 기준으로)
- 분할한 dataset 에서 new unseen logs 이 있음을 보장함.
- Blue Gene/L : 500만개 미만이므로 그대로 모두 유지
- #Anomalies5m : 500만개 메시지 중 anomalous log messages 의 개수
- Logsy의 견고성과 일반화를 자세히 평가하기 위해 target data set 에서 서로 다른 train, test 분할로 여러 실험을 수행한다.
  - ex.) 10% training; 90% test data, 20% training – 80% test, 40% training – 60% test, 60% training – 40% test, and 80% training – 20% test.

Experimental Setup

1) Evaluation Methods

- 이전 work 와의 성능 비교를 위해 **standard evaluation scores** 를 채택
- F1-score, precision, recall, accuracy 로 평가
- The positive class of 1 => anomalous log.

2) Baselines

- PCA , Deeplog 와 비교

Experimental Setup

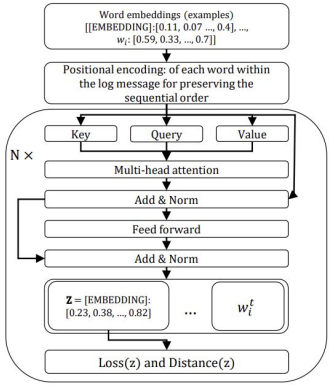


Fig. 2. Transformer encoder architecture with multi-head self-attention.

3) Logsy

- max(|ril) = 50 tokens.
- 두 개의 transformer encoder layer (N=2)
- words are embedded with **16 neural units** / transformer encoding 으로 얻는 higher level vector representations 은 all of the same sizes. / multihead self-attention mechanism 의 output 을 받는 feed-forward network 의 size 도 16 이므로 ’[EMBEDDING]’ vector 의 size 가 동일
- dropout = 0.05 learning rate = 0.0001 과 0.001 의 weight decay를 갖는 Adam optimizer 사용
- normal class = 0.5, anomaly class = 1.0
    - 2개의 classes(normal/anomaly) 의 loss function 에 각각의 weights 를 추가해서 normal 과 anomaly samples 개수 의 불균형 문제 해결

Results and Discussion

Baseline 과 비교한 Logsy의 전반적인 성능

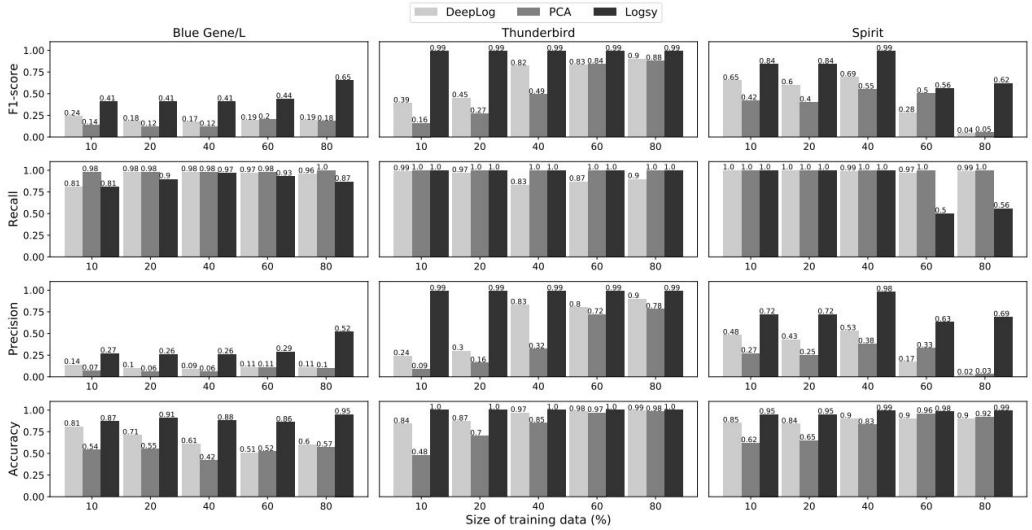


Fig. 4. Comparison of the evaluation scores against the two baselines DeepLog and PCA on three different datasets.

- Blue Gene/L - 0.448, Thunderbird - 0.99, Spirit - 0.77
- DeepLog and PCA 는 lower F1 scores in all experiments performed

Results and Discussion

1) The effect of the auxiliary data on the evaluation scores

다양한 크기의 보조 데이터를 사용할 때 Logsy가 어떻게 수행하는지에 대한 분석을 수행

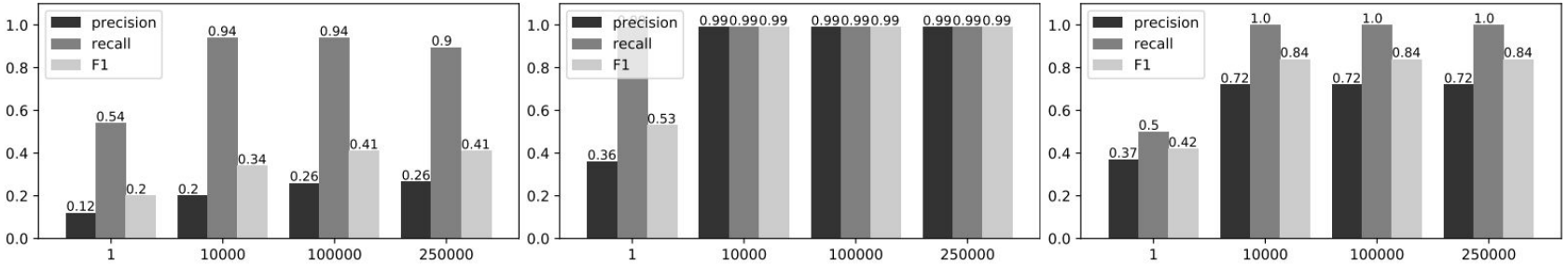


Fig. 5. Increasing the size auxiliary dataset, where the target system are Blue Gene/L, Thunderbird, and Spirit (left, middle, right) on 20% train - 80% test split

- **보조 데이터**가 1에서 250000으로 **증가**하면 모든 **평가 score 증가**
- 100000 ➔ 250000 인 경우, score 변하지 않음

  ➔ 이는 보조 데이터에 존재하는 정보의 양이 비슷하고 **모든 사례가 100000 랜덤 샘플에 이미 존재한다는 것을 보여준다.**

Results and Discussion

2) Including expert labeling

target data set 의 anomaly label 을 증가시켜 실험/ 20%-80% split of the Blue Gene/L dataset.

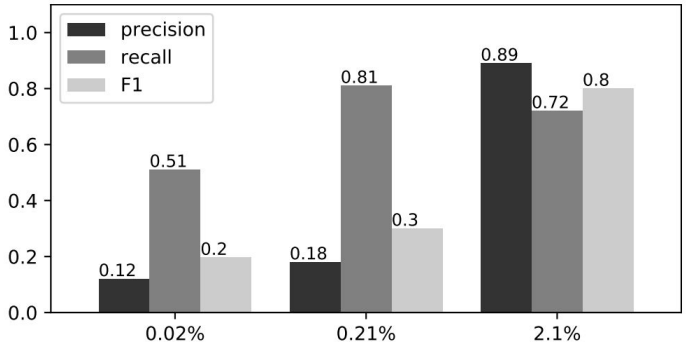


Fig. 6. Increasing the size of the labeled anomaly data in the Blue Gene/L dataset (20% train - 80% test).

- label 된 anomaly sample 의 수를 늘리면 성능이 향상됨.

Results and Discussion

3) Utilization of the learned log embeddings in related approaches

log embeddings 의 중요성을 평가하기 위해, lowest-performing method 인 PCA 의 original TF-IDF log representations 을 Logsy 에서 추출한 embeddings 로 대체 하는 실험 수행

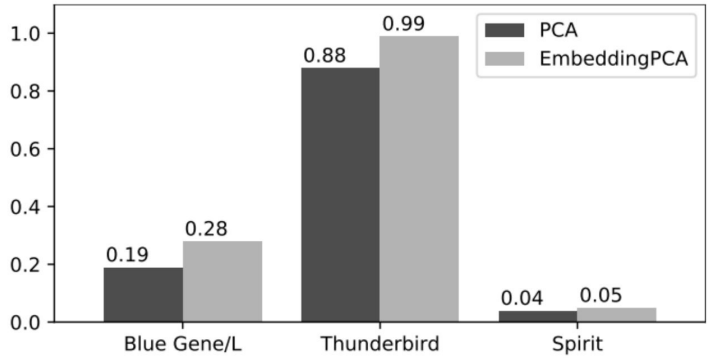


Fig. 8. Comparison in F1 score between the standard PCA [21] and PCA using the embeddings extracted from our method (80%-20% split).

- replacement of the log representation : the performance of PCA 향상 → log representation learning 이 Logsy 뿐만 아니라 new log embeddings 을 사용하도 록 조정할 수 있는 previous approaches 에도 영향을 미친다는 것을 보여줌

6. Conclusion

6. Conclusion

Logsy

- It is based on a **self-attention encoder network** with a **hyperspherical classification objective.**
- **target system 의 정상 training data** 와 **보조 데이터의 비정상 로그 data** 를 구분하도록 formulate.

Logsy 성능 평가

- F1score 0.25
- logsy 에서 생성된 log vector representation 이 다른 방법에도 활용 가능함을 보여줌.
    - PCA 활용 ➜ F1 score 0.07 개선됨

본 논문의 연구진들은

앞으로의 log anomaly detection 연구가 **정상 및 이상 데이터의 다양성을 강조하는 풍부한 domain bias 를 통합하는 대안적 방법을 찾는 데 초점을 맞춰야 한다**고 마무리 한다.