한국ITS학회논문지 J. Korea Inst. Intell. Transp. Syst.

양방향 GPT 네트워크를 이용한 VMS 메시지 이상 탐지

Detection of Anomaly VMS Messages Using Bi-Directional GPT Networks

최 효 림* · 박 승 영**

* 주저자 : 강원대학교 전기전자공학과 학사과정
** 교신저자 : 강원대학교 전기전자공학과 교수

Hyo Rim Choi* · Seungyoung Park**

* Dept. of Electrical and Electronics Engineering., Kangwon National Univ.
** Dept. of Electrical and Electronics Engineering., Kangwon National Univ.

† Corresponding author : Seungyoung Park, s.young.park@kangwon.ac.kr

Vol. 21 No.4(2022)
August, 2022
pp.125~144

pISSN 1738-0774
eISSN 2384-1729
https://doi.org/10.12815/kits.2022.21.4.125

Received 15 June 2022
Revised 16 July 2022
Accepted 18 August 2022

요 약

VMS (variable message signs) 시스템이 악의적인 공격에 노출되어 교통안전과 관련된 거짓 정보를 출력하게 된다면 운전자에게 심각한 위험을 초래할 수 있다. 이러한 경우를 방지하기 위해 VMS 시스템에 사용되는 메시지들을 수집하여 평상시의 패턴을 학습한다면 VMS 시스템에 출력될 수 있는 이상 메시지를 빠르게 감지하고 이에 대한 대응을 할 수 있을 것이다. 본 논문에서는 양방향 GPT (generative pre-trained transformer) 모델을 이용하여 VMS 메시지의 평상 시 패턴을 학습한 후 이상 메시지를 탐지하는 기법을 제안한다. 구체적으로, 제안된 기법에 VMS 메시지 및 시스템 파라미터를 입력 하고 이에 대한 NLL (negative log likelihood) 값을 최소화하도록 학습한다. 학습이 완료되면 판정해야 할 대상의 NLL 값을 계산한 후, 문턱치 값 이상일 경우 이를 이상으로 판정한다. 실험 결과를 통해, 공격에 의한 악의적인 메시지 탐지뿐만 아니라 시스템의 오류가 발생하는 상황에 대한 탐지도 가능함을 보였다.

핵심어 : VMS, GPT, NLL, 딥러닝, 이상탐지

ABSTRACT

When a variable message signs (VMS) system displays false information related to traffic safety caused by malicious attacks, it could pose a serious risk to drivers. If the normal message patterns displayed on the VMS system are learned, it would be possible to detect and respond to the anomalous messages quickly. This paper proposes a method for detecting anomalous messages by learning the normal patterns of messages using a bi-directional generative pre-trained transformer (GPT) network. In particular, the proposed method was trained using the normal messages and their system parameters to minimize the corresponding negative log-likelihood (NLL) values. After adequate training, the proposed method could detect an anomalous message when its NLL value was larger than a pre-specified threshold value. The experiment results showed that the proposed method could detect malicious messages and cases when the system error occurs.

Key words : VMS, GPT, NLL, Deep Learning, Anomaly Detection

Ⅰ. 서 론

VMS (variable message signs) 시스템 (이하, 시스템)은 도로 교통 상황에 대한 정보를 운전자에게 실시간으로 제공하여 사고 예방에 도움을 준다 (Kelarestaghi et al., 2018). 예를 들면, 과속 금지 구간에 대한 정보를 시스템을 통해 전달하게 되면 단순 표지판만을 이용하는 경우에 비하여 차량 과속 비율을 낮출 수 있다. 또한, 공사 중인 도로 주변에서 마주 오는 차량의 속도에 대한 정보를 전달하게 되면 작업자와 운전자의 안전을 효과적으로 향상시킬 수 있다. 그러나 시스템이 악의적인 공격에 노출되거나 오류가 발생하여 교통 상황과 관련된 이상 VMS 메시지 (이하, 메시지) 를 출력하게 된다면 사고를 유발할 수 있을 것이다.

국외의 경우, 여러 시스템 해킹 사례가 존재한다. 2009년 2월 텍사스주 오스틴에 있는 시스템이 'Anonymous' 해커들에 의해 해킹 되었다 (Wired, 2022). 도로전광판에 공공 안전 메시지 대신 "CAUTION!! ZOMBIES! AHEAD!" 라는 메시지가 띄워져 운전자들을 당황케 했다고 한다. 유사한 사건으로 2009년 12월 플로리다주 게인스빌에서도 시스템 해킹 사건이 있었다 (The Gainesville Sun, 2022). 출근 인파가 몰리는 시간대에 시스템에 공공 안전 메시지 대신 "ZOMBIE ATTACK! EVACUATE" 라는 해킹 메시지가 출력되었으며, 일부 운전자들은 이 메시지가 실제일 수 있다고 믿었다고 한다. 비교적 최근에 일어난 사건으로는 2016년 5월 30일 텍사스 댈러스의 코크렐 힐 로드 근처 주간고속도로에 있는 시스템에 출력되었어야 할 공사를 주의하라는 메시지 대신 "DONALD TRUMP IS A SHAPE SHIFTING LIZARD" 라는 해킹 메시지가 출력되었다고 한다. 국내의 경우, 실제 도로 상에 설치된 시스템에 대한 해킹 사건이 발생하지는 않았지만, 2021년도 11월 19일 충북 충주 인근 중부내륙고속도로 하행선 인근 시험 도로 전광판에 "!?$좀비출현$?!"이라는 메시지가 출력되는 사건이 있었다 (Korean Broadcasting System, 2022). 조사 결과, 해킹 상황 대비 훈련을 한 것이었지만 주변의 실제 차로를 지나는 몇몇 운전자들에게 혼란을 야기했다고 한다.

해킹이외에도 시스템 오류로 인해 메시지 출력과 관련된 시스템 파라미터가 변경된다면 예측하지 못한 현상 (예를 들면, 해당 시스템이 설치된 도로 관련 정보가 다른 시스템에서 출력된다던가, 문자 인코딩 문제로 문자가 정상적으로 출력이 되지 않는 현상) 이 발생될 수 있으므로 이에 대한 빠른 탐지와 조치가 필요하다. 따라서, 시스템에 사용된 메시지 및 관련 시스템 파라미터를 수집하고 자연어 처리 분야에서 활용되고 있는 딥러닝 기법을 이용하여 이들에 대한 패턴을 학습한다면 악의적인 메시지 혹은 시스템 오류를 빠르게 탐지할 수 있을 것으로 기대된다.

자연어 문장은 단어들로 구성된 시퀀스 데이터로 볼 수 있으므로 자연어 처리 분야에서는 주로 RNN (recurrent neural network) 계열의 모델을 사용해 왔다 (Yin et al., 2017). 그러나, 통상적으로 시퀀스 길이가 증가할수록 RNN 모델의 성능이 저하되는 문제가 발생한다. 이러한 문제점을 해결하기 위해 transformer 기법이 제안되었다 (Vaswani et al., 2017). Transformer 기법은 문장을 구성하는 단어와 문장에서 해당 단어의 위치를 각각 word token embedding과 positional embedding을 통해 벡터로 변환하고, 이들을 multi-head attention 구조를 이용해 처리함으로써 긴 문장에서의 성능 저하 문제를 해결하였다. Transformer 기법에서 파생된 GPT (generative pre-trained transformer) 기법은 비영리 인공지능 연구기관인 OpenAI가 개발한 문장 생성 기법이다 (Radford et al., 2019). GPT 기법은 이전 시점에 발생된 단어를 기반으로 현시점에 등장할 단어를 예측하고, 이를 다시 입력하여 다음 시점에 등장할 단어를 예측하는 auto-regressive 방식으로 문장을 생성한다. Bena and Kalita(2020) 는 GPT 모델을 사용하여 영어로 시를 생성하고 읽는 이에게 정서적 반응을 이끌어낼 수 있음을 보였고, Otter et al.(2021) 은 GPT 모델이 여러 언어로 시를 창작할 수 있음을 보였다.

GPT 기법과 같은 딥러닝 기법을 이용하여 메시지 패턴을 학습하기 위해서는 메시지를 구성하는 단어에 대한 vocabulary가 필요하다(Vajjala et al., 2020). 그러나, 메시지는 한글, 영문, 숫자, 특수 기호로 구성되어 있

으므로 이에 적합한 vocabulary를 구성하는 것은 쉽지 않다. 또한, vocabulary에 포함되지 않은 새로운 단어가 메시지에 포함되는 경우 이러한 단어를 제대로 처리할 수 없으며 이를 OOV (out-of-vocabulary) 문제라고 한다. 메시지는 통상적으로 매우 짧은 문장으로 구성되어 있음을 고려한다면 OOV 문제는 메시지 패턴 학습과정에서 어려움을 가중시킬 것이다.

이러한 문제를 해결하기 위해 본 논문에서는 메시지를 문자 단위로 분할하고 시스템 파라미터 정보와 결합하여 시스템 이상을 탐지하는 GPT 기반 기법을 제안하고자 한다. GPT를 이상탐지에 적용하게 된다면, 이전 시점에 발생된 문자를 기반으로 현시점에 등장할 문자를 예측하고 실제 값과 예측 값의 차이를 이용하여 이상을 탐지하게 될 것이다. 이때, 메시지의 앞부분에는 관측할 수 있는 문자의 수가 적기 때문에 문자 예측 성능이 상대적으로 낮아지는 문제점이 발생한다. 이를 해결하기 위해 메시지를 각각 정방향과 역방향으로 GPT 기법에 적용하여 문자를 예측한 뒤 각각의 예측 결과를 결합하여, 메시지의 앞부분 예측에 발생하는 성능저하를 해결하고자 한다(Schuster and Paliwal, 1997; Nam et al., 2021).

본 논문은 다음과 같이 구성된다. 2장에서는 관련 연구를 소개한다. 3장에서는 GPT 기법을 소개한다. 4장에서는 제안하는 이상탐지를 위한 양방향 GPT 기법에 대해 설명한다. 5장에서는 실험 결과에 대해서 서술하며, 6장에서 결론을 제시한다.

Ⅱ. 관련 연구

국내에 설치된 VMS 시스템은 공중망 등의 네트워크를 통해 관제 센터와 연결되어 있으며, 보안을 위해 VPN (virtual private network) 이 적용되어 있다고 알려져 있다 (CNS-LINK, 2022). VPN은 공중망 등의 네트워크를 전용망처럼 구성한 사설 네트워크이며, 이를 이용하면 모든 트래픽을 IP (internet protocol) 수준에서 암호화하므로 보안 통신이 가능하다. 그러나, VPN은 여러 취약점이 지속적으로 발견되어 왔다. 특정 VPN 제품의 경우, 알려진 취약점에 대한 패치가 배포되었음에도 불구하고 사용자들이 이를 제때 설치하지 않아 공격에 노출된 사례가 2021년 1사분기동안에만 약 20배 증가했다고 한다 (Nuspire, 2021). 이는 VPN 만으로는 시스템 보안에 한계가 있음을 의미한다.

Kelarestaghi et al.(2018)은 시스템 보안을 위해 VPN 이외에도 아래의 보안 기술을 추가로 도입해야 한다고 제안하였다.

- 불필요한 텔넷, 웹페이지 및 시스템의 물리 인터페이스 비활성화
- 복잡한 암호의 사용
- 공용망으로의 노출 최소화
- 공용망으로부터의 분리
- 권한 있는 사용자의 원격 접근에 대한 보안
- 물리적 공격을 방어하기 위한 인증 메커니즘을 구현
- SNMP (simple network management protocol)의 최신 버전 업그레이드
- 기본 SNMP community string의 변경
- 원격 로깅 활성화 및 로그 모니터링
- 메시지 키워드에 대한 화이트리스트 및 블랙리스트 적용

- 예정되지 않은 시간에 표출되는 메시지에 대한 필터링
- Network security zone 도입

그러나, 갈수록 지능화되는 해킹 공격을 방어하기 위해서는 이보다 좀 더 발전된 보안 기술이 도입될 필요가 있다. 예를 들어, 악의적인 공격자에 의해 메시지 내용이 변경되는 경우, 기존의 제안들로는 이를 검출하기 어렵다. 따라서, 자연어 처리 기법을 이용하여 메시지 내용에 대한 진실성 (integrity) 을 검증할 필요가 있다 (Kelarestaghi, 2019).

자연어 문장에 대한 이상을 탐지하기 위해 딥러닝 기반의 다양한 자연어 처리 기법들이 제안되어 왔다. Ruff et al.(2019)는 transformer 기반 언어 모델인 BERT (bidirectional encoder representations from transformer) 구조를 이용한 문장 이상 탐지 기법을 제안하였다. 이 기법의 동작 과정은 다음과 같다. 정상 문장을 구성하는 단어를 토큰으로 변환하여 토큰 (token) 시퀀스를 생성하고 BERT 네트워크를 통해 일정한 크기의 표현 (representation) 벡터로 변환한다. 이를 정상 문장들에 대한 여러 주제를 표현하는 맥락 (context) 벡터들과의 코사인 유사도를 최소화 하도록 훈련을 수행한다. 이때, 맥락 벡터들과 표현 벡터를 생성하는 BERT 네트워크가 동시에 최적화된다. 학습이 완료되면, 판정 대상 문장에 대한 표현 벡터와 맥락 벡터들과의 코사인 유사도를 측정하여 이상여부를 탐지한다.

Manolache et al.(2021)은 자기지도학습 (self-supervision) 기반의 transformer 구조를 활용한 DATE (detecting anomalies in text via self-supervision of transformers) 기법을 제안하였다. 해당기법은 정상 문장을 토큰 시퀀스로 변환하고, 미리 정의된 masking 패턴들 중 하나를 임의로 선택하여 일부 토큰들을 mask 토큰으로 치환한다. 이렇게 변환된 토큰 시퀀스를 transformer 기반의 생성자 (generator) 와 구분자 (discriminator) 로 적층된 네트워크에 입력한다. 생성자는 입력된 mask 토큰을 임의의 토큰으로 치환하여 구분자에 전달하고, 구분자는 각 토큰의 치환여부와 어떤 masking 패턴이 사용되었는지를 동시에 추정하도록 학습을 진행한다. 학습이 완료되면 구분자만을 사용하여 입력된 판정 대상 문장의 이상여부를 탐지한다.

Mai et al.(2022)은 BERT 구조를 이용한 다양한 문장 이상 탐지 기법의 성능을 비교하였다. 구체적으로, 입력되는 토큰의 일부를 mask 토큰으로 치환하고 이들의 원래 토큰을 추정하는 MLM (masked language modeling), 입력되는 토큰 시퀀스의 다음에 등장할 토큰을 추정하는 CLM (casual language modeling), 동일한 토큰 시퀀스에 서로 다른 dropout 패턴을 적용하고 이들의 코사인 유사도를 최대화하도록 학습하는 contrastive learning에 대한 성능을 비교하였다.

Transformer 기반의 문장 이상 탐지 기법들은 문장을 단어단위로 토큰 변환을 해야 하므로 문장을 구성하는 단어에 대한 vocabulary가 필요하며, 이는 OOV 문제를 야기한다. 주어진 문장을 문자단위로 분할하여 이상 탐지를 수행한다면, 이러한 문제없이 문장 이상탐지를 수행할 수 있을 것이다. 문장을 문자단위로 분할하여 transformer 기반의 네트워크에 적용하여 문장 이상 탐지를 수행하는 기법은 없으나, 문장을 문자단위로 분할하여 이미지로 변환하고 CNN (convolutional neural network)을 이용하여 이상 탐지를 수행한 기법은 존재한다. Park et al.(2018)은 CNN (convolutional neural network) 기반의 오토인코더 (autoencoder)를 이용한 HTTP 메시지 이상탐지 기법을 제안하였다. 이 기법에서는 정상 HTTP 메시지를 문자단위로 분할하고, 각각의 문자를 one-hot 벡터로 변환한다. 즉, HTTP 메시지는 0과 1로 구성된 이미지로 변환된다. 이를 CNN 기반의 오토인코더에 입력하여 BCE (binary cross entropy) 를 최소화하도록 학습한다. 학습이 완료되면 판정 대상 HTTP 메시지에 대한 BCE를 측정하여 이상여부를 판정한다. Mohaghegh and Abdurakhmanov(2021)은 주어진 문장을 문자 단위 분할하여 one-hot 벡터로 변환한 후 CNN을 이용하여 특성 (feature) 벡터로 변환하고 이에 대해 PCA (principal component analysis), HBOS (histogram-based outlier detection), isolation forest, CBLOF (cluster-

based local outlier factor), LODA (lightweight on-line detector for anomaloes) 기법을 적용하여 문장 이상 탐지를 수행하는 기법을 제안하였다.

기존의 문장 이상 탐지 기법들은 일반적으로 인터넷을 통해 수집되거나 공개된 말뭉치 데이터들에 대하여 적용되었지만, VMS 메시지를 대상으로 적용된 바가 없다. 본 논문에서는 VMS 메시지를 문자단위 토큰 시퀀스로 변환하고 이를 양방향 GPT 기법에 입력하여 이상 탐지를 수행하는 기법을 제안한다. 양방향 GPT 기법은 CAN (controller area network) 환경에서 발생하는 CAN 신호의 ID (identifier) 를 시간 순으로 정렬한 시퀀스에 대하여 높은 이상 탐지 성능을 얻을 수 있음이 알려져 있다 (Nam et al., 2021). 문자 토큰 시퀀스와 CAN ID 시퀀스는 모두 제한된 값을 가진 정수만으로 이루어진 시퀀스임을 고려한다면, 양방향 GPT 기법이 VMS 메시지 이상 탐지에서도 높은 성능을 얻을 수 있을 것으로 기대된다.

Ⅲ. GPT 기법 소개

<Fig. 1>은 GPT 네트워크의 구조를 보여준다. 이 그림으로부터 GPT 네트워크는 G개의 GPT 모듈이 적층되어 있으며, 각 GPT 모듈은 masked multi-head self-attention module, layer normalization module, feed-forward module, layer normalization module로 구성되어 있음을 알 수 있다. 본 논문에서는 메시지 패턴을 학습하기 위해 문자 단위로 변환된 토큰 시퀀스를 입력 받는 방식을 고려한다. 따라서, 본 장에서는 메시지를 문자 단위로 토큰으로 변환한 후, 이를 GPT 네트워크에 입력하여 학습하는 과정을 설명한다.

1. 문자 토큰 임베딩 및 위치 임베딩 계층

우선, L개의 문자 토큰으로 구성된 시퀀스

$$\mathbf{x} = [x_0 \ \cdots \ x_{(L-1)}]^{\top} \quad (1)$$

을 생성하는 GPT 네트워크를 고려하자. 이때, 발생 가능한 총 문자 토큰 개수를 K라 가정하면, x_t은 $0 \le x_t \le (K-1)$을 만족하는 정수가 된다. GPT 네트워크는 입력되는 문자 토큰에 대해 다음에 등장할 문자 토큰을 예측하므로, 입력 시퀀스 $\mathbf{x}$에 대한 출력 시퀀스

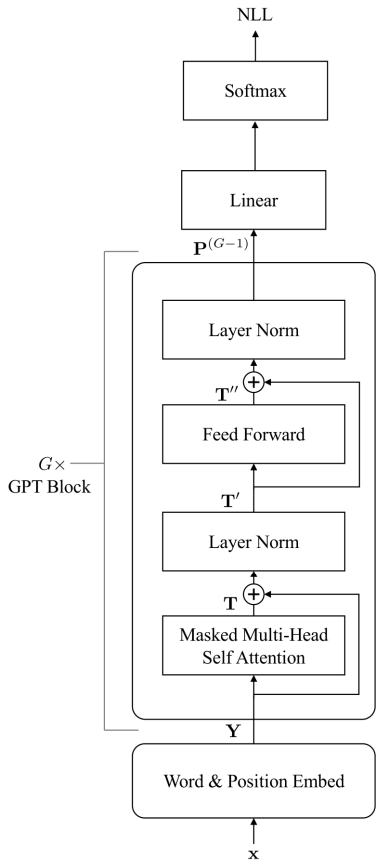
$$\tilde{\mathbf{x}} = [x_1 \ \cdots \ x_L]^{\top} \quad (2)$$

의 확률을 예측한다고 볼 수 있다 (Radford et al., 2019). 여기서, $\tilde{\mathbf{x}}$는 $\mathbf{x}$를 한 문자만큼 지연한 동일한 시퀀스이다.

이러한 구성에서, 문자 토큰 x_t은 E차원 word embedding 벡터로 변환된 후, 메시지 내의 해당 문자의 위치에 따라 E차원 positional embedding 벡터가 더해져 $\mathbf{y}_t$로 변환된다 (Vaswani et al., 2017). 이때, positional embedding 벡터는 메시지의 길이인 L개만큼의 서로 다른 벡터들로 구성된 벡터 집합으로부터 해당 문자의 위치에 따라 그에 해당되는 벡터를 선택하여 사용한다 (Radford et al., 2019). 결과적으로 시퀀스 $\mathbf{x}$는

$$\mathbf{Y} = [\mathbf{y}_0 \cdots \mathbf{y}_{(L-1)}] \quad (3)$$

로 변환된다. $\mathbf{Y}$를 구성하는 L개의 벡터 $\{\mathbf{y}_l\}$은 H개의 head들로 구성된 masked multi-head self-attention 계층에 동시에 입력된다.



<Fig. 1> Block Diagram of GPT Network

2. Masked Multi-Head Self-Attention 계층

Masked multi-head self-attention 계층의 h번째 head는 $\mathbf{Y}$를 각각 query 행렬

$$\mathbf{Q}^{(h)} = \left(\mathbf{W}_Q^{(h)}\right)^{\top}\mathbf{Y}, \quad (4)$$

key 행렬

$$\mathbf{K}^{(h)} = \left(\mathbf{W}_K^{(h)}\right)^{\top}\mathbf{Y}, \quad (5)$$

value 행렬

$$\mathbf{V}^{(h)} = \left(\mathbf{W}_V^{(h)}\right)^{\mathsf{T}} \mathbf{Y} \quad (6)$$

로 변환한다. 여기서, $\mathbf{W}_Q^{(h)}$, $\mathbf{W}_K^{(h)}$, $\mathbf{W}_K^{(h)}$는 각각 $E \times p$ 크기의 행렬이며, p와 H는 $H \times p = E$ 를 만족하는 정수이다. 이러한 행렬들을 사용하여 $p \times L$차원의 attention value 행렬

$$\begin{aligned}\mathbf{Z}^{(h)} &= \left[\mathbf{z}_0^{(h)} \ \cdots \ \mathbf{z}_{(L-1)}^{(h)}\right] \\ &= \mathbf{V}^{(h)} \cdot \mathbf{softmax}\left(\frac{1}{\sqrt{p}}\left(\mathbf{Q}^{(h)}\right)^{\top} \mathbf{K}^{(h)} + \mathbf{M}\right)\end{aligned} \quad (7)$$

를 구성한다. 수식 (7)에서, $\mathbf{M}$은 $L \times L$ 크기의 mask 행렬로서 (i,j) 번째 성분 $m_{i,j}$는

$$m_{i,j} = \begin{cases} 0 & \text{if } j \le i, \\ -\infty & \text{if } j > i \end{cases} \quad (8)$$

를 만족한다. 또한, $\mathbf{softmax}(\cdot)$는 $L \times L$ 크기의 행렬 $\mathbf{A}$에 대하여

$$\mathbf{softmax}(\mathbf{A}) = \begin{bmatrix} \frac{\exp(a_{0,0})}{\sum_{l=0}^{L-1}\exp(a_{l,0})} & \cdots & \frac{\exp(a_{0,(L-1)})}{\sum_{l=0}^{L-1}\exp(a_{l,(L-1)})} \\ \vdots & \ddots & \vdots \\ \frac{\exp(a_{(L-1),0})}{\sum_{l=0}^{L-1}\exp(a_{l,0})} & \cdots & \frac{\exp(a_{(L-1),(L-1)})}{\sum_{l=0}^{L-1}\exp(a_{l,(L-1)})} \end{bmatrix} \quad (9)$$

를 출력한다. 여기서,

$$\mathbf{A} = \left(\frac{1}{\sqrt{p}}\left(\mathbf{Q}^{(h)}\right)^{\mathsf{T}} \mathbf{K}^{(h)} + \mathbf{M}\right) \quad (10)$$

이며, $a_{i,j}$는 행렬 $\mathbf{A}$의 (i,j)번째 성분이다.

수식 (7)에서, mask 행렬 $\mathbf{M}$으로 인해 l번째 문자 토큰의 attention value 벡터 $\mathbf{z}_l^{(h)}$는 해당 시점까지의 입력 문자 토큰들 $\{x_{l'}\}_{l'=0}^{l}$만을 사용하여 계산되며, 미래의 입력 문자 토큰들 $\{x_{l'}\}_{l'=(l+1)}^{(L-1)}$은 사용되지 않는다. 따라서, GPT 네트워크는 l번째 까지 입력된 문자 토큰들만을 사용하여 다음 시점 $(l+1)$에 등장할 문자 토큰에 대한 확률

$$\mathbb{P}\left(x_{(l+1)} \,|\, x_0, \ \cdots, x_l\right) \quad (11)$$

를 추정함을 알 수 있다 (Radford et al., 2018).

각 head들로부터 출력된 H개의 attention value 행렬 $\left\{\mathbf{Z}^{(h)}\right\}_{h=0}^{H-1}$을 행 방향으로 연접하여 $E \times L$ 크기의 행렬을 생성하고 이를 linear 계층에 입력하여, masked multi-head self-attention 계층의 최종출력인 multi-head attention 행렬

$$\mathbf{T} = \mathbf{W}_O^{\top} \begin{bmatrix} \mathbf{Z}^{(0)} \\ \vdots \\ \mathbf{Z}^{(H-1)} \end{bmatrix} \quad \cdots\cdots (12)$$

를 생성한다. 여기서, $\mathbf{W}_O$는 linear 계층을 구성하는 $E \times E$ 크기의 행렬이다.

3. Feed-Forward 계층

수식 (12)의 multi-head attention 행렬 $\mathbf{T}$에 수식 (3)의 residual input $\mathbf{Y}$을 더하고 (He et al., 2016), layer normalization을 적용하여 (Ba et al., 2016)

$$\mathbf{T}' = \mathrm{layernorm}\left(\mathbf{T} + \mathbf{Y}\right) \quad \cdots\cdots (13)$$

을 생성한다. 이어서, 행렬 $\mathbf{T}'$를 linear 계층, activation function, 그리고 linear 계층로 구성된 feed-forward 계층에 입력하여

$$\mathbf{T}'' = \mathbf{W}_{F_1}^{\top}\,\mathrm{GELU}\left(\mathbf{W}_{F_0}^{\top}\mathbf{T}'\right) \quad \cdots\cdots (14)$$

을 생성한다. 여기서, $\mathbf{W}_{F_0}$와 $\mathbf{W}_{F_1}$는 각각 $E \times 4E$와 $4E \times E$ 크기의 행렬이며, $\mathrm{GELU}(\cdot)$는 Gaussian error linear unit 이다 (Hendrycks and Gimpel, 2016).

Feed-forward 계층으로부터 출력된 행렬 $\mathbf{T}''$에 수식 (13)의 residual input $\mathbf{T}'$을 더한 후, layer normalization을 적용하면 첫번째 GPT 모듈은 $E \times L$ 크기의 행렬

$$\mathbf{P}^{(0)} = \mathrm{layernorm}\left(\mathbf{T}'' + \mathbf{T}'\right) \quad \cdots\cdots (15)$$

를 최종적으로 출력하게 된다.

4. 메시지 생성 확률

GPT 네트워크는 G개의 GPT 모듈이 적층되어 있으므로 위의 과정이 G회 반복되어 최종적으로 $E \times L$ 크기의 multi-head attention 행렬 $\mathbf{P}^{(G-1)}$를 출력하게 된다. 최상위 linear 계층은 이를 입력 받아 $E \times K$ 크기의 행렬 $\mathbf{W}_D$를 이용하여 $K \times L$ 크기의 행렬

$$\mathbf{U} = \mathbf{W}_D^{\top}\mathbf{P}^{(G-1)} = [\mathbf{u}_0 \ \cdots \ \mathbf{u}_{(L-1)}] \quad (16)$$

를 출력한다.

Softmax 계층은 최상위 linear 계층에서 출력되는 행렬 $\mathbf{U}$를 구성하는 l번째 열벡터

$$\mathbf{u}_l = [u_{0,l} \ \cdots \ u_{(K-1),l}]^{\top} \quad (17)$$

을 이용하여 문자 토큰 x_l 다음에 등장할 문자 토큰 $x_{(l+1)}$에 대한 확률 값

$$\mathrm{softmax}(\mathbf{u}_l) = \left[\frac{\exp(u_{0,l})}{\sum_{k=0}^{K-1}\exp(u_{k,l})} \ \cdots \ \frac{\exp(u_{(K-1),l})}{\sum_{k=0}^{K-1}\exp(u_{k,l})} \right]^{\top} \quad (18)$$

을 추정한다. 이때, $\mathbf{u}_l$은 수식 (7)의 mask 행렬 $\mathbf{M}$에 의해 입력 문자 토큰들 중 $\{x_{l'}\}_{l'=0}^{l}$ 만을 사용해서 생성된다. 따라서, 수식 (18)은 l 시점까지의 입력 토큰이 주어졌을 때, 다음 문자 토큰 $x_{(l+1)}$에 대한 조건부 확률에 대한 추정치

$$\hat{\mathbb{P}}\left(x_{(l+1)} \mid \{x_{l'}\}_{l'=0}^{l}\right) \quad (19)$$

로 표현할 수 있다. 최종적으로 chain rule에 의해 수식 (2)의 추정해야 할 메시지 시퀀스 $\tilde{\mathbf{x}}$ 에 대한 확률에 대한 추정치는

$$\hat{\mathbb{P}}(\tilde{\mathbf{x}}) = \prod_{l=0}^{(L-1)} \hat{\mathbb{P}}\left(x_{(l+1)} = s_{(l+1)} \mid \{x_{l'}\}_{l'=0}^{l}\right) \quad (20)$$

로 표현할 수 있다. 여기서, $s_{(l+1)}$는 $x_{(l+1)}$에 대한 ground truth 이다 (Radford et al., 2018).

Ⅳ. 제안 기법

본 장에서는 양방향 GPT 기반의 메시지 이상 탐지 기법에 대해 설명한다. 이를 위해, 먼저 메시지에 대하여 문자 단위로 분할하고 이에 대한 토큰 변환을 수행한 후, 앞부분에 시스템 파라미터를 표현하는 토큰들을 추가하여 토큰 시퀀스를 생성한다. 생성된 시퀀스를 양방향 GPT 네트워크에 입력하여 NLL (negative log likelihood) 값을 계산하고, 문턱치 값과 비교하여 이상 여부를 탐지한다.

1. 문자 단위 토큰 변환 과정

국가교통정보센터의 교통정보공개서비스에서 제공되는 VMS 관련 데이터는 '생성시간', '기관코드', 'VMSID', 'VMSTYPE', 'VMSSEQNO', 'VMSMSGCODE', '표출시간', '메시지내용' 으로 구성되어 있다 (National Transport Information Center, 2022). 본 논문에서는 메시지 내용과 함께 시스템과 관련된 기관코드, VMSTYPE, VMSMSGCODE를 입력 데이터로 사용하였다. 여기서, 기관코드는 시스템이 설치된 지역을 관할하는 총 6개의 기관 (서울지방국토관리청, 원주지방국토관리청, 대전지방국토관리청, 익산지방국토관리청, 부산지방국토관리청, 한국도로공사)을 나타낸다. VMSTYPE과 VMSMSGCODE는 각각 0~1, 0~9까지의 숫자로 구성되어 있다. 메시지내용 데이터는 CP949로 인코딩되어 있다. 해당 방식은 영문, 숫자, 기호에 대해서는 문자 당 1바이트로 한글에 대해서는 문자 당 2바이트로 인코딩 한다.

임의의 기관이 관리하는 시스템에 출력된 '정체 30분 예상' 이라는 메시지 변환과정을 고려해보자. 메시지의 공백을 제거한 총 7개의 문자에 대하여 문자 단위로 토큰 변환을 수행한다. 본 논문에서는 문자 단위 토큰 변환 방식 A, B, C를 고려하며, <Table 1>은 문자 변환 방식에 따른 변환 결과를 보여준다.

<Table 1> Example of Character Conversion

Character	1	2	3	4	5	6	7
Scheme A (1 or 2 tokens per character)	193/164	195/188	51	48	186/208	191/185	187/243
Scheme B (2 tokens per character)	193/164	195/188	0/51	0/48	186/208	191/185	187/243
Scheme C (proposed) (1 tokens per character)	11,057	11,437	6,131	6,128	9,855	10,722	10,068

이 표에서 첫 번째 행에 표시된 1부터 7까지의 숫자는 메시지에 해당되는 7개의 문자를 의미하며, 각각 '정', '체', '3', '0', '분', '예', '상'에 해당한다. 문자 변환 방식 A는 바이트 당 1개의 토큰을 사용하는 방식이다. 구체적으로, 문자 당 2바이트 크기로 인코딩 되는 한글 문자는 2개의 토큰을 사용하여 표현하고 1바이트로 표현되는 문자는 1개의 토큰을 사용하여 표현한다. 이때, 토큰 번호는 각 바이트를 10진수로 변환하여 사용한다. 따라서, 발생 가능한 토큰의 총 개수는 256 (= 2^8) 개가 된다. <Table 1>에서 보여준 바와 같이 세 번째와 네 번째 문자에 해당하는 '3'과 '0'은 각각 1바이트 크기의 '0x33'과 '0x30'으로 인코딩 되므로 해당 바이트를 10진수로 변환한 51번과 48번 토큰으로 표현된다. 세 번째와 네 번째 문자를 제외한 문자들은 모두 한글이므로 2바이트 크기로 인코딩 된다. 따라서, 해당 문자들은 각각 2개의 토큰으로 표현된다. 예를 들어, 첫 번째 문자 인 '정'은 '0xC1'과 '0xA4'로 인코딩 되므로 이들을 10진수로 변환한 193번과 164번 토큰으로 표현된다. 이 방식은 문자의 종류 별로 사용되는 토큰의 개수가 다르므로 한글과 그 외 문자가 혼용된 메시지가 딥러닝 네트워크에 입력되는 경우, 문자간의 구분이 어려우므로 성능 열화가 발생할 수 있다.

이러한 문제를 해결하기 위해, 문자 변환 방식 B는 한글 외 문자에 대하여 임의로 0을 의미하는 1바이트를 추가하는 방식으로 모든 문자를 2개의 토큰을 사용하여 표현한다. 따라서, 방식 A와는 달리 세 번째와 네 번째 문자인 '3'과 '0'은 각각 0번 51번 그리고 0번 48번 토큰으로 표현된다. 모든 문자를 2개의 토큰을 사용하여 표현하므로 딥러닝 네트워크에서는 2개의 토큰을 하나의 문자로 인식할 수 있게 되어 방식 A에서 발생하는 문제를 완화할 수 있을 것이다.

본 논문에서 제안하는 문자 변환 방식 C는 다음과 같다. CP949 인코딩은 총 17,048개의 문자를 지원하므로 해당 문자들을 순차적으로 0번부터 17,047번 토큰까지 대응 시킨다. 예를 들어, 첫 번째 문자인 '정'은 CP949 인코딩에서 11,057번째 문자에 해당되므로 11,507번 토큰으로 표현된다. 이 방식은 하나의 문자를 하나의 토큰으로 표현하므로 문장을 토큰 시퀀스로 변환할 때 그 길이가 방식 A와 B에 비해 짧아진다. Transformer를 구성하는 매개변수의 수와 계산량은 시퀀스의 길이의 제곱에 비례하여 증가한다는 사실을 고려한다면 이는 분명 장점이 될 수 있다 (Zaheer et al., 2021). 그러나, 방식 A와 B에 비해 방식 C는 추정해야 하는 토큰의 종류가 크게 증가한다. 수식 (18)에 의하면 네트워크의 최종 출력층의 출력 개수는 추정해야 하는 토큰의 종류와 동일하므로, 방식 C가 적용되는 네트워크의 최종 출력층의 매개변수의 수는 크게 증가하게 된다. 변환 방식 별 매개변수와 계산량에 대해서는 6장에서 자세히 논한다.

메시지와 함께 시스템 파라미터를 학습하기 위해, 메시지 토큰 시퀀스 앞에 기관코드, VMSTYPE, VMSMSGCODE에 해당되는 각 1개씩의 토큰을 추가한다. 따라서, 발생 가능한 토큰 종류는 방식 A와 B의 경우 문자를 표현하는 토큰과 기관코드 6종류, VMSTYPE 2종류, VMSMSGCODE 10종류의 토큰을 합산하여 총 278개가 되고, 방식 C의 경우 총 17,070개가 된다.

2. 양방향 GPT 구조

앞서 설명했던 바와 같이 GPT 기법은 예측된 토큰이 다음 토큰을 예측하기 위한 입력으로 사용되는 구조이기 때문에 초기 구간에서는 예측에 이용할 토큰의 개수가 작아 예측 성능이 감소 될 수 있다. 이를 해결하기 위해 본 논문에서는 토큰 시퀀스를 각각 정방향 GPT 네트워크와 역방향 GPT 네트워크에 입력한다. 이 경우, 정방향 GPT 네트워크는 상대적으로 뒤쪽 시퀀스에 대한 예측 성능이 높을 것이며 역방향 GPT 네트워크는 상대적으로 앞쪽 시퀀스에 대한 예측 성능이 높을 것이다. 따라서, 두 예측 결과를 결합한다면 전체적인 예측 성능이 향상될 수 있다 (Nam et al., 2021).

구체적인 과정은 다음과 같다. 입력 메시지는 L개의 토큰으로 구성되며, 발생 가능한 토큰 수는 K개라고 가정하자. 즉, 메시지는 토큰 시퀀스

$$\mathbf{x} = [x_0 \ \cdots \ x_{(L-1)}]^{\mathsf{T}} \quad (21)$$

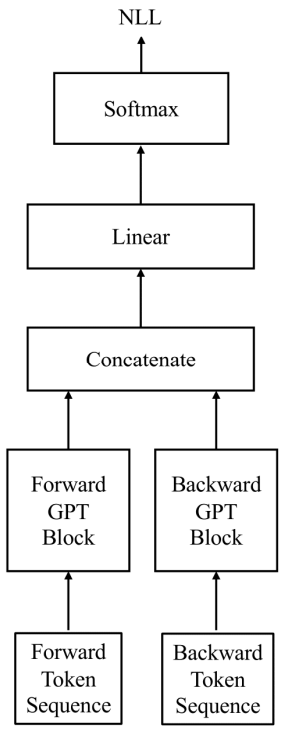
로 표현된다. <Fig. 2>는 정방향과 역방향 GPT 모듈이 결합된 양방향 GPT 네트워크 구조를 보여준다. 그림으로부터, 입력 토큰 시퀀스 $\mathbf{x}$는 각각 정방향 시퀀스

$$\mathbf{f} = [x_0 \ \cdots \ x_{(L-2)}]^{\mathsf{T}} \quad (22)$$

와 역방향 시퀀스

$$\mathbf{b} = [x_{(L-1)} \ \cdots \ x_1]^{\mathsf{T}} \quad (23)$$

로 변환되어 각각 정방향과 역방향 GPT 모듈에 입력됨을 알 수 있다. 이때, 3장에서 설명한 바와 같이 GPT 네트워크는 현 시점까지 입력된 문자들을 이용하여 다음 문자를 예측한다는 사실을 고려한다면, 정방향 GPT 모듈은 $\mathbf{f}$를 입력받아 $E \times (L-1)$차원의 행렬



<Fig. 2> Block Diagram of Bi-Directional GPT Network

$$\mathbf{F} = \left[\tilde{\mathbf{f}}_0 \ \cdots \ \tilde{\mathbf{f}}_{(L-2)}\right] \quad \cdots\cdots (24)$$

를 출력함을 알 수 있다. 여기서, $\tilde{\mathbf{f}}_t$은 3장에서 설명한 바와 같이 입력 $\{x_{t'}\}_{t'=0}^{t}$을 이용한 $x_{(t+1)}$ 에 대한 예측, 즉

$$\mathbb{P}\left(x_{(t+1)} \mid x_0, \ \cdots \ x_t\right)$$

와 관련된 정보를 가지고 있는 E차원 벡터이다. 이와 유사하게, 역방향 GPT 모듈은 $\mathbf{b}$를 입력 받아 $E \times (L-1)$차원의 행렬

$$\mathbf{B} = \left[\tilde{\mathbf{b}}_0 \ \cdots \ \tilde{\mathbf{b}}_{(L-2)}\right] \quad \cdots\cdots (25)$$

를 출력한다. 여기서, $\tilde{\mathbf{b}}_t$은 입력 $\{x_{t'}\}_{t'=(L-t-1)}^{(L-1)}$을 이용한 $x_{(L-t-2)}$ 에 대한 예측, 즉,

$$\hat{\mathbb{P}}\left(x_{(L-t-2)} \mid x_{(L-t-1)}, \ \cdots \ x_{(L-1)}\right) \quad \cdots\cdots (26)$$

와 관련된 정보를 가지고 있는 벡터이다.

앞서 언급한 바와 같이 초기 예측 시점에서는 이용할 수 있는 관측 문자의 개수가 적다는 문제가 발생한다. 이러한 문제를 완화하기 위해 $\mathbf{F}$와 $\mathbf{B}$를 결합하여 $2E \times L$ 차원 행렬

$$\mathbf{C} = \begin{bmatrix} \mathbf{0}_E & \mathbf{F} \\ \mathbf{BG} & \mathbf{0}_E \end{bmatrix} = [\mathbf{c}_0 \ \cdots \ \mathbf{c}_{(L-1)}] \quad (27)$$

를 생성한다. 여기서, 행렬 $\mathbf{G}$는 $\mathbf{B}$를 구성하는 각 열벡터의 순서를 $\mathbf{F}$의 열벡터 순서와 일치시키기 위한 exchange 행렬이고 (Horn and Johnson, 2012), $\mathbf{0}_E$은 E차원 0벡터이다. <Fig. 2>의 상단 linear 계층은 수식 (27)의 $\mathbf{C}$를 입력 받아 $K \times L$ 차원의 행렬

$$\mathbf{U} = \mathbf{W}^{\mathrm{T}}\mathbf{C} = [\mathbf{u}_0 \ \cdots \ \mathbf{u}_{(L-1)}] \quad (28)$$

를 출력한다. 여기서, $\mathbf{W}$는 상단 linear 계층을 구성하는 $2E \times K$의 크기를 갖는 행렬이다. 수식 (28)의 행렬 $\mathbf{U}$를 구성하는 각각의 열벡터 $\mathbf{u}_l$은 softmax 층으로 전달된다. 수식 (27)에 의하면 정방향과 역방향 GPT 모듈의 출력 행렬이 결합되어 있으므로, softmax 층의 출력은 다음과 같은 x_l에 관한 조건부 확률 추정치

$$\hat{\mathbb{P}}\left(x_l \,\middle|\, \{x_{l'}\}_{l'=0, l' \neq l}^{(L-1)}\right) \quad (29)$$

로 볼 수 있다. 따라서, 최종적으로 수식 (21)에서 제시된 입력 토큰 시퀀스 $\mathbf{x}$에 대한 추정 확률은 수식 (29)를 이용하여

$$\hat{\mathbb{P}}(\mathbf{x}) = \prod_{l=0}^{(L-1)} \hat{\mathbb{P}}\left(x_l = s_l \,\middle|\, \{x_{l'}\}_{l'=0, l' \neq l}^{(L-1)}\right) \quad (30)$$

로 표현할 수 있다.

3. VMS 이상 메시지 탐지 과정

본 절에서는 양방향 GPT 네트워크를 이용한 이상 메시지 탐지 과정을 설명한다. 훈련을 위한 N개의 정상 메시지가 존재한다고 가정하자. 이를 이용하여 양방향 GPT 네트워크의 출력에 대해 다음과 같이 정의되는 NLL 손실 함수

$$\mathrm{NLL} = -\frac{1}{NL}\sum_{n=0}^{N-1}\sum_{l=0}^{L-1}\log\hat{\mathbb{P}}\left(x_l^{(n)} = s_l^{(n)} \,\middle|\, \left\{x_{l'}^{(n)}\right\}_{l'=0, l' \neq l}^{(L-1)}\right) \quad (31)$$

를 최소화하도록 훈련을 수행한다. 여기서, $x_l^{(n)}$은 훈련에 사용되는 n번째 정상 메시지의 l번째 토큰이며,

$s_t^{(n)}$은 $x_t^{(n)}$에 대한 ground truth 값이다.

훈련이 완료된 후, 주어진 시퀀스 $\mathbf{x}$의 이상 여부를 탐지하기 위해 이를 양방향 GPT 네트워크에 입력하여

$$\mathrm{NLL}_{\mathbf{x}} = -\frac{1}{L}\sum_{t=0}^{L-1}\log \hat{\mathbb{P}}\left(x_t = s_t \,\middle|\, \{x_{t'}\}_{t'=0, t'\neq t}^{(L-1)}\right) \quad (32)$$

를 계산한다. 최종적으로, $\mathrm{NLL}_{\mathbf{x}}$의 값이 문턱치 Γ보다 더 크다면, 즉,

$$\mathrm{NLL}_{\mathbf{x}} > \Gamma \quad (33)$$

를 만족하면 이상 메시지라 판정한다.

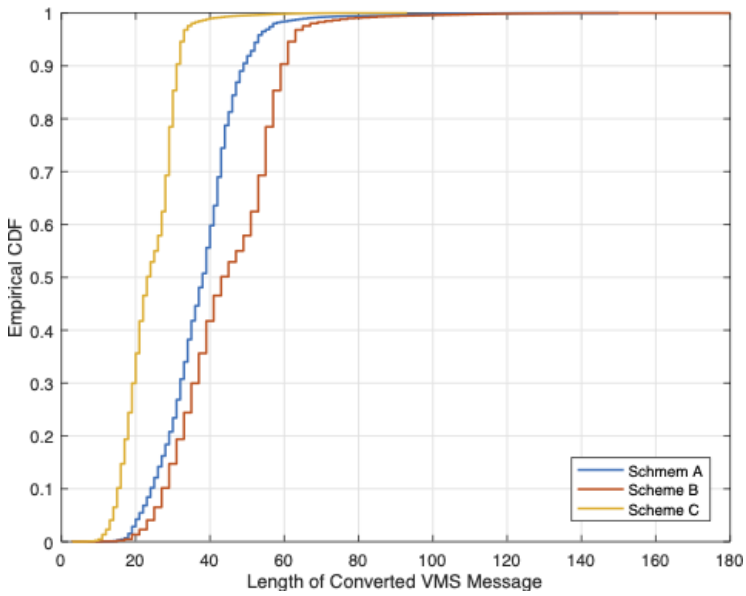
V. 실험 결과

1. 실험 환경 및 성능 평가지표

본 논문에서는 교통정보공개서비스에서 제공하는 2015년 01월 01일부터 2021년 07월 15일까지 약 5년간의 VMS 메시지 관련 데이터 중 기관코드, VMSTYPE, VMSMSGCODE, 메시지내용이 중복되는 데이터를 제거한 약 150만개의 데이터를 사용하여 정상 메시지 패턴을 학습하였다 (National Transport Information Center, 2022). 이상 탐지 성능 평가를 위한 데이터는 2021년 7월 16일부터 7월 31일까지 수집된 데이터 중, 우선 훈련에 사용된 데이터와 동일한 데이터를 제거한 다음 기관코드, VMSTYPE, VMSMSGCODE, 메시지내용이 중복되는 데이터를 제거하여 약 10,000개의 평가용 정상 데이터를 선정하였다.

본 논문에서는 두가지 이상유형을 고려한다. 이상 유형 1 (anomaly type 1)은 메시지는 정상이지만 시스템 오류가 발생하는 경우에 대한 탐지 성능을 측정하기 위해 평가용 정상 데이터를 대상으로 기관코드, VMSTYPE, VMSMSGCODE를 임의로 변경하여 생성하였다. 이상 유형 2 (anomaly type 2)는 정상적인 메시지와 관계없는 악의적인 메시지에 대한 탐지 성능을 측정하기 위해 평가용 정상 데이터의 메시지를 뉴스 기사 제목으로 변경하여 생성하였다.

성능 평가를 위한 지표로서, TPR (true positive rate), FNR (false negative rate), FPR (false positive rate)을 사용한다. 여기서, TPR은 이상 데이터를 이상 (즉, true positives) 으로 판정한 비율이며, FNR은 이상 데이터를 정상 (즉, false negatives) 으로 오판한 비율이고, FPR은 정상 데이터를 이상 (즉, false positive) 으로 오판한 비율이다. 또한, 이러한 성능지표를 기반으로 이상 탐지 시스템의 성능을 시각적으로 보여주기 위해 ROC (receiver operating characteristics) 곡선을 사용한다. ROC 곡선은 임의의 판정 문턱치 값에 해당하는 FPR과 TPR 값을 각각 가로축과 세로축에 출력한 2차원 그래프이다. 서로 다른 기법의 ROC 성능을 비교하기 위해 AUC (area under the curve) 값을 이용한다. AUC 값은 ROC 곡선의 아래 면적을 정규화하여 계산하므로 이상적인 탐지 기법의 성능에 대한 AUC 값은 1이 된다. 따라서, 일반적인 탐지 기법의 AUC 값은 0에서 1사이로 나타나며, 1에 가까울수록 높은 성능을 가진 것으로 판단할 수 있다.



<Fig. 3> Empirical CDF of the Token Sequence Length

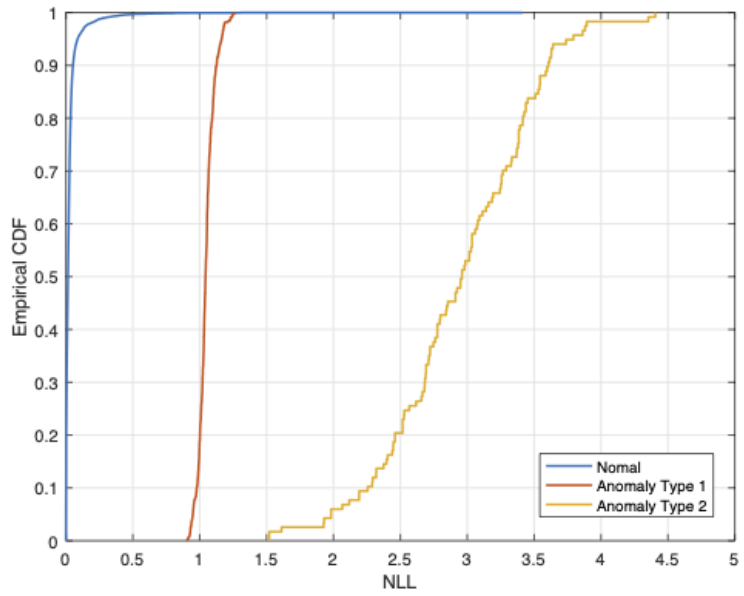
제안하는 양방향 GPT 네트워크를 구성하는 정방향과 역방향 GPT 모듈은 토큰 임베딩 차원 E = 768, head 개수 H = 8, dropout 확률 = 0.1, GPT 모듈 계층 수 G = 6으로 구성되어 있다. 훈련은 CentOS 7, Intel® Xeon® Gold 5218R, Nvidia Tesla P100으로 구성된 시스템에서 PyTorch v1.11.0를 사용하여 mini-batch 크기 16, 학습율 10^{-4}의 ADAM 최적화 알고리즘을 적용 후 50 epoch 동안 진행하였다.

2. 제안 기법의 성능

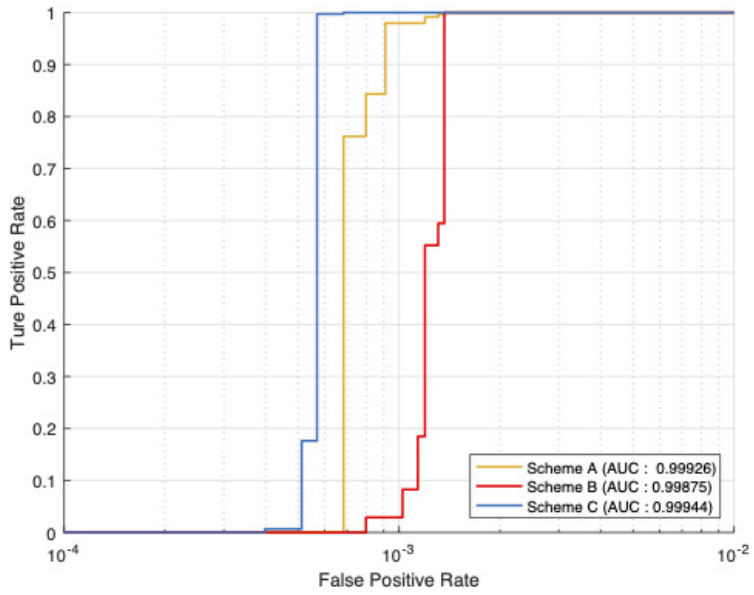
<Fig. 3>은 3가지 문자 변환 방식에 의해 변환된 메시지를 구성하는 토큰 시퀀스 길이에 대한 실험 누적 분포 함수를 보여준다. 그림의 범례에서 'Scheme A', 'Scheme B', 'Scheme C'는 각각 문자 변환 방식 A, B, C를 의미한다. 문자 당 2개의 토큰을 사용하여 표현하는 방식 B의 변환이 가장 길고, 문자 당 1개의 토큰을 사용하여 표현하는 방식 C (제안 기법) 가 가장 짧음을 알 수 있다. 변환 방식 A, B, C의 최대 길이는 각각 150, 183, 93이며, 이를 각 방식별 양방향 GPT 네트워크에 입력 가능한 최대 길이 L로 결정하였다.

<Fig. 4>는 이상 유형 1과 2에 대한 제안 기법 (즉, 문자 변환 방식 C를 양방향 GPT 네트워크에 적용) 의 NLL 값에 대한 실험 누적 분포 함수를 보여준다. 그림의 범례에서, 'Normal', 'Anomaly Type 1', 'Anomaly Type 2'는 각각 정상, 이상 유형 1, 이상 유형 2 메시지를 의미한다. 그림으로부터, 이상 유형 2에 대한 NLL 값이 이상 유형 1에 대한 NLL 값보다 더 높게 나오는 것을 확인할 수 있다. 그 이유는 이상 유형 1의 경우 메시지 자체는 정상이므로 전체적인 정상 메시지 패턴과 비교 시 차이가 작은 반면, 이상 유형 2의 경우 메시지 자체가 변경되었으므로 차이가 매우 크기 때문이다. 향후 제시되는 결과들은 탐지 성능이 상대적으로 낮은 이상 유형 1에 대한 성능만을 고려한다.

<Fig. 5>는 문자 변환 방식 별 이상 유형 1에 대한 ROC 성능을 보여준다. 그림의 범례에서 각 방식 별 AUC 값을 표시하였다. 그림으로부터, 제안하는 문자 변환 방식 C가 가장 우수한 탐지 성능을 보임을 확인할 수 있다.



<Fig. 4> Empirical CDF of the NLL Values of the Proposed Scheme (Character Conversion Scheme C)



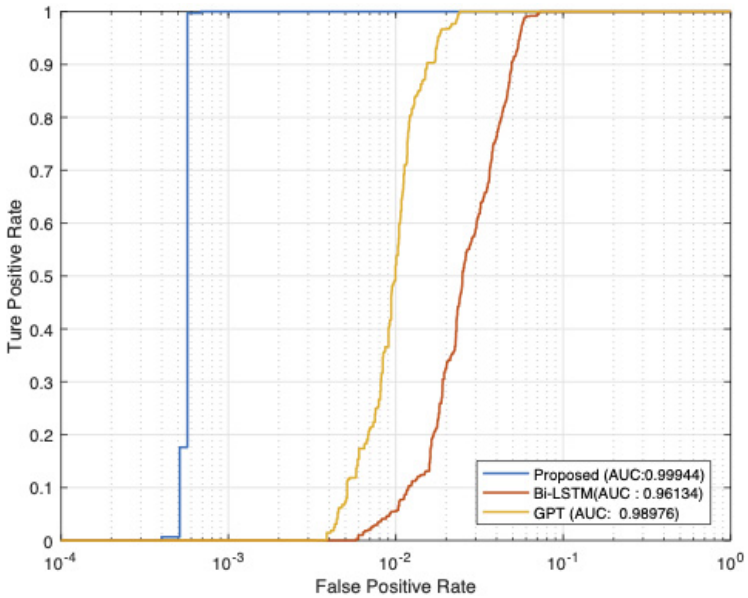
<Fig. 5> ROC Performance of the Proposed Method with Character Conversions for Anomaly Type 1

3. 탐지 기법 별 성능 비교

본 절에서는 문자변환 방식 C를 사용하는 양방향 GPT 기법과 다른 딥러닝 기법들과의 성능을 비교한다. 첫 번째로 고려하는 비교 기법은 3장에서 설명한 단방향 GPT 기법이며, 이는 정방향으로만 토큰 시퀀스를 입력하고 순차적으로 다음에 등장할 토큰을 예측하여 이상을 탐지하는 기법이다. 단방향 GPT 기법은 정방

향 GPT 모듈로만 구성되어 있으므로 제안하는 기법과 유사한 복잡도를 유지하도록 하기 위해 GPT 모듈 계층 수는 G = 12로 설정하였다. 두 번째로 고려하는 비교 기법은 GPT 모듈 대신 LSTM 모듈을 사용하는 양방향 LSTM 기법이다. 이 기법을 구성하는 정방향과 역방향 LSTM 모듈은 제안기법의 모듈과 동일한 계층 수 G = 6와 E = 768차원의 hidden 및 cell state를 사용하였다.

<Fig. 6>은 이상 유형 1에 대한 ROC 성능을 보여준다. 그림의 범례에서, 'Proposed', 'bi-LSTM', 'GPT'는 각각 제안기법, 양방향 LSTM 기법, 단방향 GPT 기법을 의미한다. AUC 값을 기준으로 성능을 비교한 결과 유사한 복잡도를 유지함에도 제안 기법, 단방향 GPT 기법, 양방향 LSTM 기법 순으로 우수한 성능을 보임을 확인할 수 있다.



<Fig. 6> ROC Performance Comparison of Different Detection Methods for Anomaly Type 1

이상탐지 시스템에서 통상적으로 문턱치를 낮추면 FNR이 감소하고 FPR이 증가하는 상충관계가 존재한다. 즉, 대부분의 이상이 탐지되도록 시스템을 운영하면 정상을 이상으로 오판하는 경우(FPR)가 증가하게 되고 운영자는 이를 확인하기 위해 많은 노력을 필요로 한다. <Table 2>는 모든 이상을 탐지하는 조건 (즉, FNR = 0) 하에 이상 유형 1에 대한 FPR 성능을 보여준다. 해당 표로부터 제안기법의 성능이 가장 우수한 것을 확인하였다. 즉, 제안기법은 FP의 발생 비율이 가장 낮기때문에 운영자의 부담을 줄일 수 있다.

<Table 2> FPR Performance of Different Anomaly Detection Schemes for Anomaly Type 1 (FNR = 0)

Proposed	Bi-Directional LSTM	GPT
5.05×10^{-4}	7.42×10^{-2}	2.41×10^{-2}

4. 복잡도 및 추론 시간 비교

<Table 3>은 네트워크를 구성하는 매개변수 개수와 메시지 당 평균 추론 시간을 보여준다. 평균 추론 시

간을 측정하기 위해, 1000개의 메시지에 대한 추론을 수행하고 해당 시간의 평균을 구하였다. 양방향 GPT 네트워크 구조에 문자 변환 방식 C를 적용하는 경우, 방식 A와 B에 비하여 매개변수의 개수가 약 1.6배 증가하지만 오히려 평균 추론 시간은 감소함을 확인할 수 있다. 제안기법이 매개변수의 개수와 평균 추론시간 간의 비례관계가 성립되지 않는 이유는 다음과 같다. 우선, 4장에서 설명한 바와 같이 GPT 모듈의 매개변수 개수와 추론시간은 토큰 시퀀스의 길이의 제곱에 비례한다. 방식 C의 토큰 시퀀스 길이는 A와 B에 비해 짧으므로 GPT 모듈의 매개변수 개수와 추론시간은 감소되었다. 하지만 방식 C의 최종출력단의 개수가 274개에서 17,066개로 약 62배 증가함에 따라 전체 네트워크 매개변수 개수가 증가하였다.

<Table 3> Comparison of Number of Paramters and Average Inference Time per VMS Message

		No. of Parameters	Avg. Inference Time (Sec.)
Bidirectional GPT	Scheme A	8.62×10^{7}	5.28×10^{-2}
	Scheme B	8.63×10^{7}	5.62×10^{-2}
	Scheme C (Proposed)	1.38×10^{8}	4.73×10^{-2}
Unidirectional GPT (Scheme C)		1.11×10^{8}	3.89×10^{-2}
Bidirectional LSTM (Scheme C)		1.09×10^{8}	1.23×10^{-1}

방식 C에서 단방향 GPT 네트워크는 제안하는 양방향 GPT 기법 (정방향과 역방향의 각 계층 수 G = 6)과 유사한 매개변수 개수로 구성하기 위해 계층 수를 2배 (G = 12)로 증가시켰음에도 불구하고, 제안기법의 매개변수 수가 여전히 1.2배 더 많음을 확인할 수 있다. 이는 정방향 GPT 모듈과 역방향 GPT 모듈을 결합을 위해 추가적인 매개변수가 필요하기 때문이다. 이로 인해 평균 추론 시간이 단방향 GPT에 비해 약 1.2배 증가함을 확인할 수 있다. 양방향 LSTM 네트워크의 경우, 매개변수 개수가 감소함에도 불구하고 평균 추론 시간이 급격하게 증가함을 확인할 수 있다. 이는 GPT 모듈의 경우 입력되는 토큰 시퀀스를 동시에 처리하지만 LSTM 모듈의 경우 이를 순차적으로 처리하기 때문이다.

Ⅵ. 결 론

본 논문에서는 양방향 GPT 기법을 이용한 VMS 메시지 이상 탐지 기법을 제안하였다. 기존 단방향 GPT 기법은 이전 시점에 발생된 문자를 기반으로 현시점에 등장할 문자를 예측하기 때문에 초기 구간에서는 예측에 이용할 문자의 개수가 적어서 탐지 성능이 감소 될 수 있다. 이를 해결하기 위해 VMS 메시지의 시퀀스를 각각 정방향과 역방향 GPT 모듈에 입력하고, 각 출력들을 결합하여 탐지를 수행하는 양방향 GPT 기법을 적용하였다. 실험 결과로부터, 공격에 의한 악의적인 메시지 탐지뿐만 아니라 VMS 시스템 오류에 대한 탐지도 가능함을 보였다. 일반적으로 대부분의 이상이 탐지되도록 시스템을 운영하면 정상을 이상으로 오판하는 경우 (즉, false positive) 가 증가하게 되며 운영자는 이를 확인하기 위해 많은 노력을 필요로 한다. 제안방식은 모든 이상을 탐지하는 조건 (즉, FNR = 0) 하에서 우수한 FPR 성능을 보임을 확인하였다. 따라서 제안 기법을 VMS 메시지 이상 탐지에 적용한다면 운영자의 부담을 경감시킬 수 있을 것이다.

ACKNOWLEDGEMENTS

본 결과물은 농림축산식품부의 재원으로 농림식품기술기획평가원의 스마트팜다부처패키기혁신기술개발사업 (421040-04)과 아우토크립트(주)의 지원을 받아 연구되었음

REFERENCES

Ba, J. L., Kiros, J. R. and Hinton, G. E.(2016), *Layer normalization* [Online], Available at https://arxiv.org/abs/1607.06450

Bena, B. and Kalita, J.(2020), *Introducing aspects of creativity in automatic poetry generation* [Online], Available at https://arxiv.org/abs/2002.02511

CNS-LINK, https://new-m2m.tistory.com/21, 2022.08.04.

He, K., Zhang, X., Ren, S. and Sun, J.(2016), *Identity mappings in deep residual networks* [Online], Available at https://arxiv.org/abs/1603.05027

Hendrycks, D. and Gimpel, K.(2016), *Gaussian error linear units (GELUs)* [Online], Available at https://arxiv.org/abs/1606.08415

Horn, R. A. and Johnson, C. R.(2012), *Matrix Analysis* (2nd ed.), Cambridge, UK, Cambridge University Press.

Kelarestaghi, K. B.(2019), *A risk based approach to intelligent transportation systems security*, Doctoral Dissertation, Virginia Polytechnic Institute and State University.

Kelarestaghi, K. B., Heaslip, K., Khalilikhah, M., Fuentes, A. and Fessmann, V.(2018), "Intelligent transportation system security: Hacked message signs", *Society of Automotive Engineers International Journal of Transportation Cybersecurity and Privacy*, vol. 1, no. 2, pp.1-15.

Korean Broadcasting System, https://news.kbs.co.kr/news/view.do?ncd=5329562, 2022.05.12.

Mai, K. T., Davies, T. and Griffin, L. D.(2022), *Self-supervised losses for one-class textual anomaly detection* [Online], Available at https://arxiv.org/abs/2204.05695.

Manolache, A., Brad, B. and Burceanu, E.(2021), *DATE: Detecting anomalies in text via self-supervision of transformers* [Online], Available at https://arxiv.org/abs/2104.05591

Mohaghegh, M. and Abdurakhmanov, A.(2021), "Anomaly detection in text data sets using character-level representation", *Proc. International Conference on Machine Vision and Information Technology*, Auckland, New Zealand, pp.1-6.

Nam, M., Park, S. and Kim, D.(2021), "Intrusion detection method using bi-directional GPT for in-vehicle controller area networks", *IEEE Access*, vol. 9, pp.124931-124944.

National Transport Information Center, https://openapi.its.go.kr:8090, 2022.04.25.

Nuspire(2021), *Nuspire Threat Report Q1 2021* [Online], Available at https://www.nuspire.com/resources/q1-2021-threat-report

Otter, D. W., Medina, J. R. and Kalita, J. K.(2021), "A Survey of the usages of deep learning for natural language processing", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp.604-624.

Park, S., Kim, M. and Lee, S.(2018), "Anomaly detection for HTTP using convolutional autoencoders",

IEEE Access, vol. 6, pp.70884-70901.

Radford, A., Narasimhan, K., Salimans, T. and Sutskever, I.(2018), *Improving language understanding by generative pre-training* [Online], Available at https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. and Sutskever, I.(2019), *Language models are unsupervised multitask learners* [Online], Available at http://www.persagen.com/files/misc/radford2019language.pdf.

Ruff, L., Zemlyanskiy, Y., Vandermeulen, R., Schnake, T. and Kloft, M.(2019), "Self-attentive, multi-context one-class classification for unsupervised anomaly detection on text", *Proc. Annual Meetings of the Association for Computational Linguistics*, Florence, Italy, pp.4061-4071.

Schuster, M. and Paliwal, K. K.(1997), "Bi-directional recurrent neural networks", *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp.2673-2681.

The Gainesville Sun, http://www.gainesville.com/article/20091002/articles/910021006, 2022.06.04.

Vajjala, S., Majumder, B., Gupta, A. and Surana, H.(2020), *Practical natural language processing: A Comprehensive guide to building real-world NLP systems*, O'ReillyMedia, USA.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I.(2017), "Attention is all you need", *Proc. International Conference on Neural Information Processing Systems*, Long Beach, CA, USA, pp.6000-6010.

Wired, https://www.wired.com/2009/02/austin-road-sig, 2022.06.04.

Yin, W., Kann, K., Yu, M. and Schütze, H.(2017), *Comparative study of CNN and RNN for natural language processing* [Online], Available at https://arxiv.org/abs/1702.01923

Zaheer, M., Guruganesh, G., Dubey, A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L. and Ahmed, A.(2021), *Big Bird: Transformers for Longer Sequences* [Online], Available at https://arxiv.org/abs/2007.14062