V2W-BERT: A Framework for Effective Multiclass Classification of Software Vulnerabilities

요약

1. 목표

CVE 보고서에 열거된 소프트웨어 취약점들을 CWE 분류 체계에 자동으로 매핑하고자 함

2. 방법론

- 트랜스포머 인코더 적층 구조인 BERT의 응용 모델인 V2W-BERT를 제안
- 이는 샴 네트워크, 자연어 처리, 링크 예측 등 기법을 사용

4. 결과

- 시간 분할(Temporal Partition) 데이터에 대한 취약점 예측에서 정확도(Accuracy) 97% 달성
- 기존 접근법에 비해 Few-shot Learning의 예측 정확도가 약 20% 정도 증가함

5. 중요성

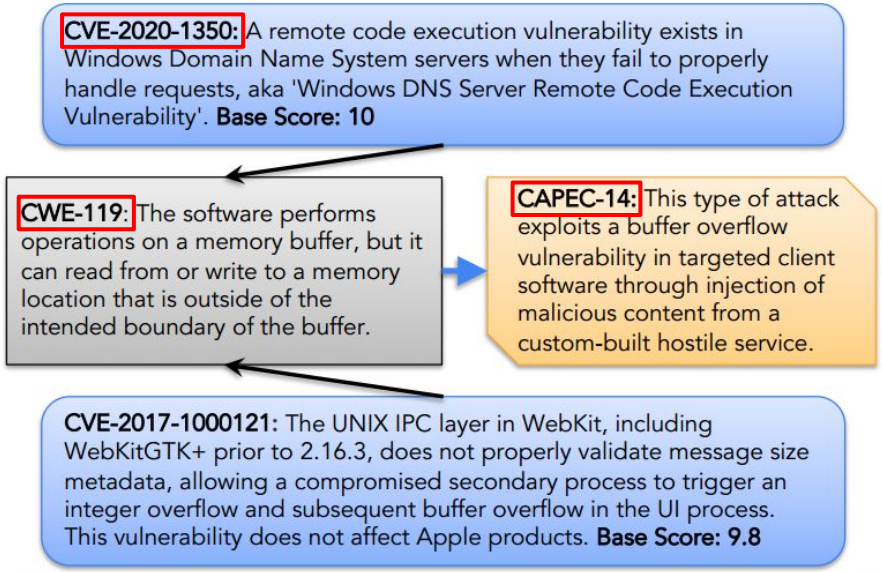
CVE 인스턴스와 CWE 클래스를 자동으로 매핑하여 효과적인 사이버보안 업무의 수행이 가능

Keyword: CVE-CWE, BERT, Link Prediction, Siamese Network

A. 연구 문제: CVEs, CWEs & CAPEC

CVE, CWE, CAPEC

- CVE(Common Vulnerabilities and Exposures)
  - 공개적으로 알려진 사이버 보안 **취약점** 목록
  - **취약점에 대한 표준 식별자** 제공
- CWE(Common Weakness Enumeration)
  - 소프트웨어 및 하드웨어 **약점에 대한 카테고리** 시스템
  - 약점을 구조적으로 정리하여 이해 및 완화 방안 수립에 활용
- CAPEC(Common Attack Pattern Enumeration and Classification)
  - 알려진 **공격 패턴**에 대한 종합적인 목록
  - 공격자의 관점에서 공격을 설명하고 분류하여 방어 전략 수립에 활용



소프트웨어 혹은 하드웨어의 일반적으로 어떤 부분이 약점(CWE)이다.
따라서 각각의 상황에서 이러한 취약점(CVE)들이 속하며,
이러한 약점과 취약점을 기반으로 공격자는 특정 패턴(CAPEC)으로 공격할 수 있다.

B. 연구 배경 & 선행 연구의 한계

“확장 가능(Scalable)하고 신뢰(Reliable)할 수 있는 자동 매핑을 구현하자.”

[연구 배경 & 선행 연구]

기존 수동 매핑 방식은 scalable & reliable 하지 않음
2020년 미국 CERT는 약 17,000개의 새로운 취약점 보고
(4,000개 심각, 10,000개 중간, 3,000개 낮음) [1,2,3]

[도전 과제]

1. 기존 모델들(TF-IDF, SVM, Naive Bayes, NN 등) 사용하기엔 충분한 데이터 부족
(CVE가 100개 미만인 클래스가 70%, 매핑된 CVE가 없는 클래스가 10%) [10 11 12 13]

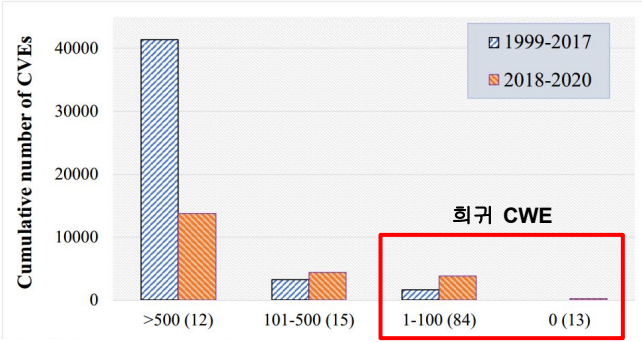
2. CVE와 CWE의 설명에 사용된 용어의 의미적 차이(semantic gap)

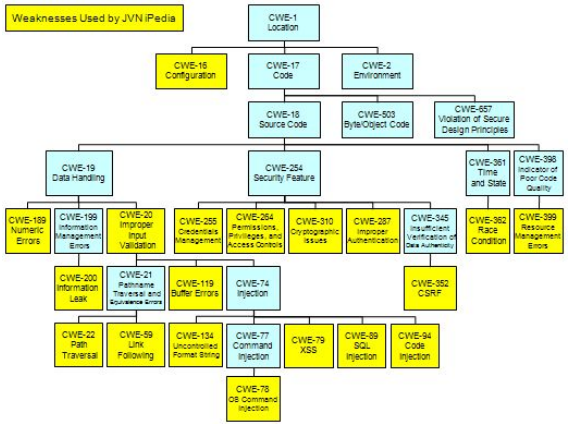
3. CWE 클래스의 비분리 계층 구조 (동일한 취약점에 대하여 계층으로 인한 다중/중복 경로 발생)

[트렌드]

1. 신경망과 워드 임베딩을 도입한 연구가 더 나은 성능을 보였지만, 여전히 희귀 CWE 처리에 한계를 드러냄 [14, 15, 16]

2. 희귀 CWE 출현 빈도 증가





C. BERT 모델 개요

왜 BERT 모델인가?

1. 결합 조건부 모델링(Joint Conditioning) vs. 단순 연결(Concatenation)

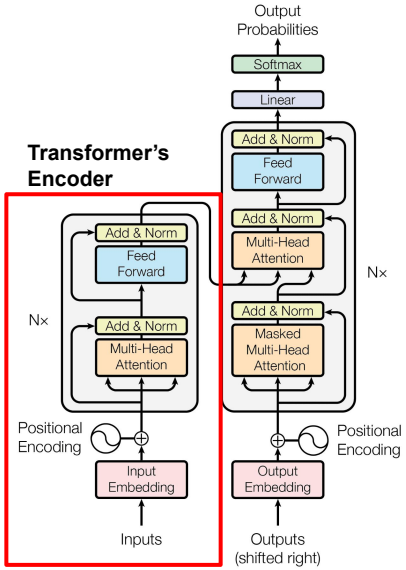
- [ELMo] 단순히 left-to-right, right-to-left 표현을 단순 연결하는 방식
- [BERT] 양방향 문맥을 결합하여 조건부로 문맥 표현 학습
- BERT의 접근 방식은 양방향성을 더 잘 활용하여 문맥을 파악

2. 양방향 문맥(Bidirectional Context) vs. 단방향 문맥(Unidirectional Context)

- [GPT] left-to-right의 단방향 아키텍처 사용
- [BERT] 양방향 아키텍처 사용. 모든 레이어에서 각 단어가 자신을 간접적으로 "보게" 되고, 왼쪽과 오른쪽 문맥을 융합할 수 있어 더 깊이 있는 표현 학습 가능

3. 파인 튜닝 접근법(Fine-Tuning Approach) vs. 특징 튜닝 접근법(Feature-Tuning Approach)

- [BERT] 양방향 표현을 사전 학습하고, 하나의 출력층만 추가하여 파인 튜닝(미세 조정)
- BERT에 직접적인 변형없이 다양한 영역의 NLP 작업(Downstream Task)을 비용 효율적으로 처리



Multi-layer bidirectional Transformer Encoder

- L : number of layers (Transformer Block)
- H : hidden (node) size
- A : number of self-attention heads

$BERT_{Base}$

- L = 12, H = 768, A = 12
- Total parameters = 110M
- Same model size as OpenAI GPT (vs.)

C. BERT 모델 개요

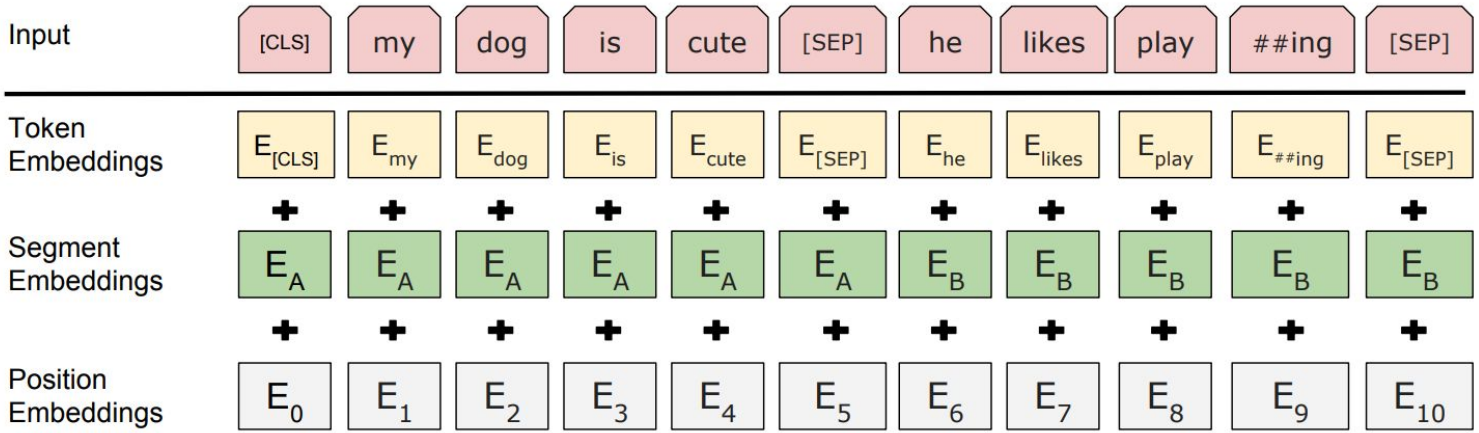
BERT의 입력/출력 표현(Representations)

2. Input representation is the sum of

(1) Token embedding

(2) Segment embedding

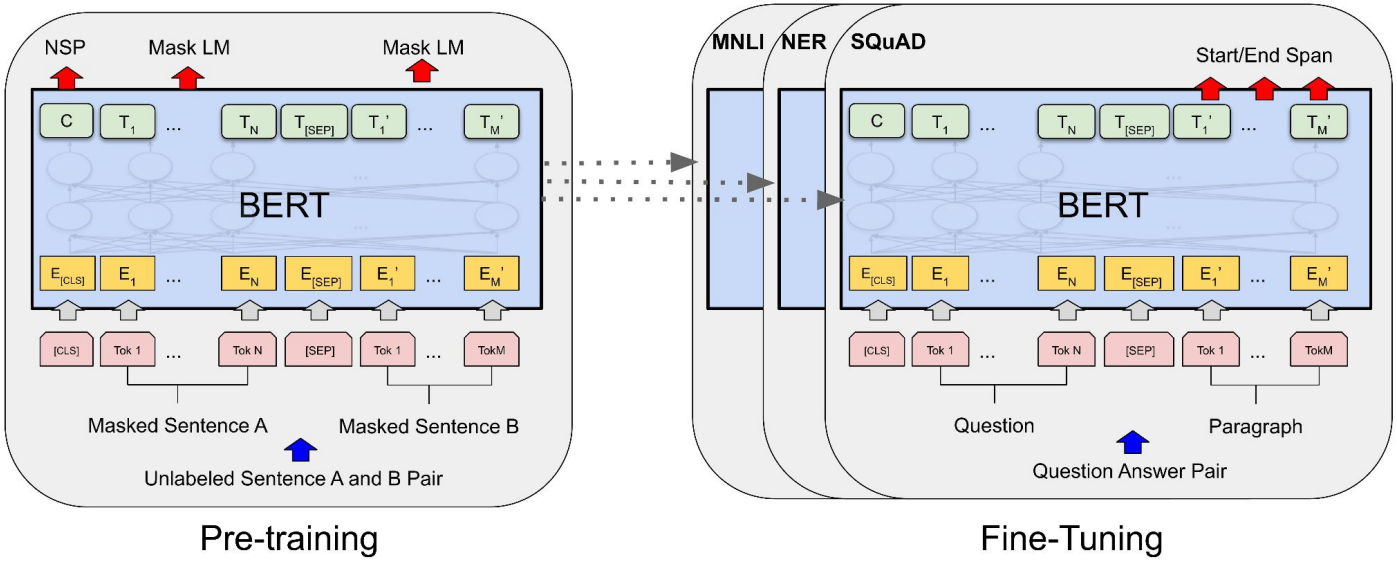
(3) Position embedding



C. BERT 모델 개요

사전 학습

1. Masked Language Model(MLM)
2. Next Sentence Prediction(NSP)

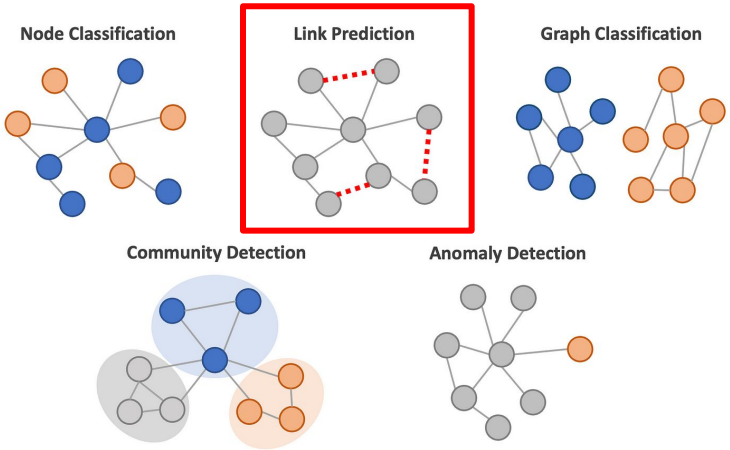


[BERT Transfer Learning Process] https://paperswithcode.com/method/bert

C. 링크 예측

링크 예측(Link Prediction)

- 그래프 구조에서 존재하지 않는 엣지(링크)를 예측하는 문제
- 노드 간 연결 가능성을 학습하여 새로운 링크를 추론
- 추천 시스템, 지식 그래프 완성, 사회 네트워크 분석 등에 활용



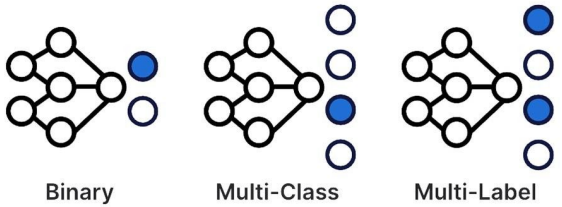
C. 링크 예측

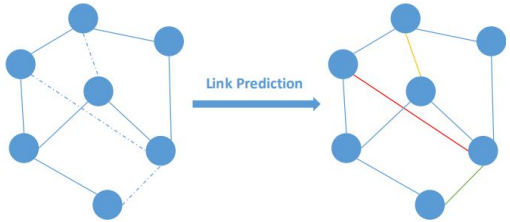
문제 정의

- **CVE와 CWE 데이터 활용**
  - CVE: 입력 텍스트 데이터
  - CWE: 타겟 클래스 (텍스트 세부사항 존재, 기존 분류 기반 방법에서는 미활
- **Multi-class Multi-label 문제를 이진 링크 예측 문제로 변환**
  - CWE 설명을 활용하여 모델의 융통성(flexibility) 향상 목적
  - CVE-CWE description 쌍 (v, w)의 연관성 측정 신뢰도 함수: l = Fθ(v, w)
  - 연관성이 높을수록 1에 가까운 값 반환 (연관성 낮을 경우 0에 가까운 값)
- **학습 가능한 함수(Learnable Function) Fθ**
  - 훈련 과정을 통해 신경망이 학습하거나 근사하는 함
  - 매개변수화, 근사, 최적화, 일반화 수행

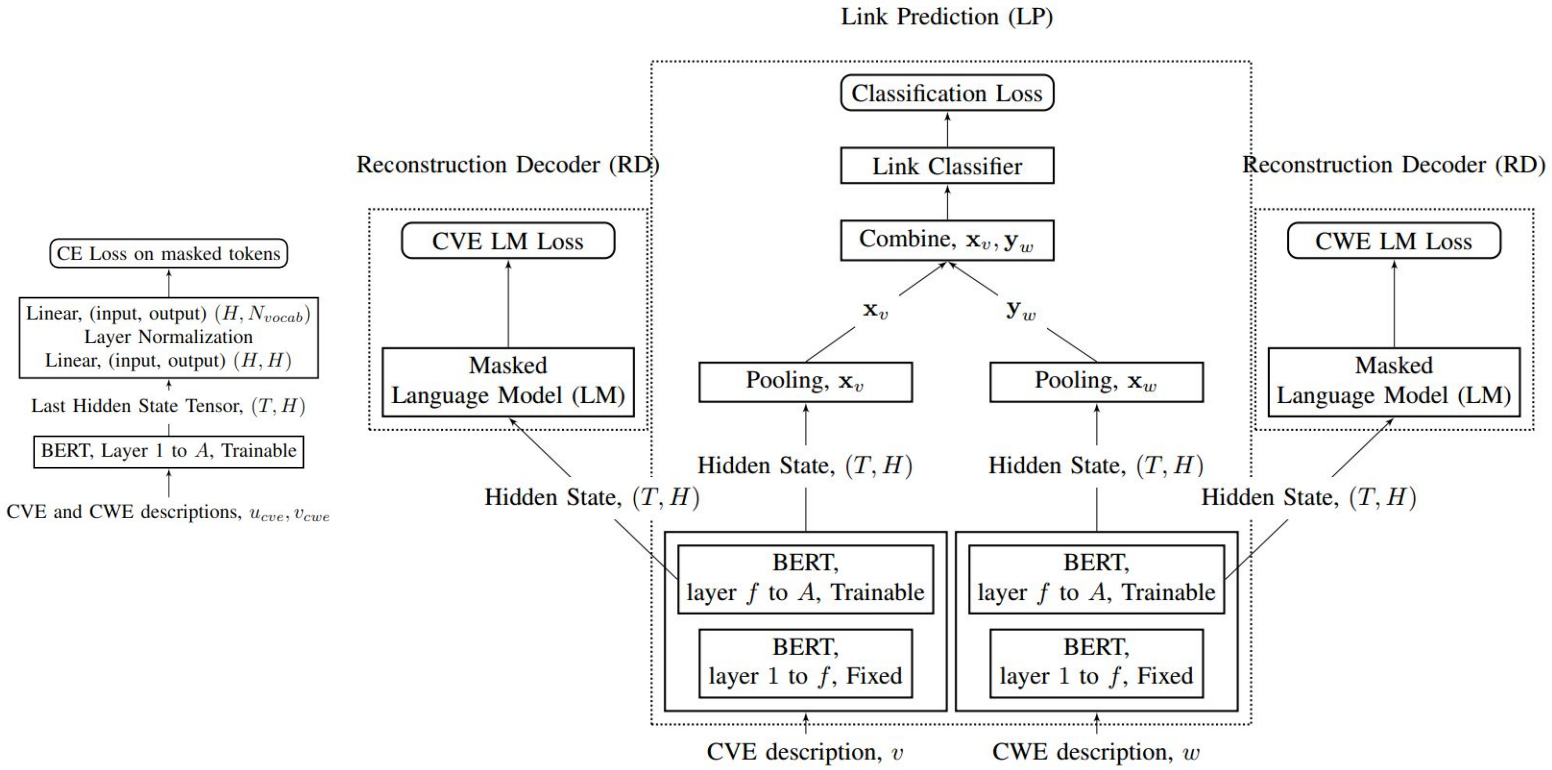
$$l = F_\theta(v, w).$$

- **함수 학습을 위한 조건**
  - 양수 및 음수 링크 사용 (기존에 알려진 CVE-CWE 연결 활용)
  - 이미 알려진 CVE-CWE 매핑의 경우, CVE와 그 조상 간 모든 연결을 양의 링크로 간주
  - 나머지 CVE-CWE 연결은 음의 링크로 간주
  - 계층 구조에서 신뢰도가 가장 높은 링크 탐색 (root → leaf)





V2W 모델 아키텍처

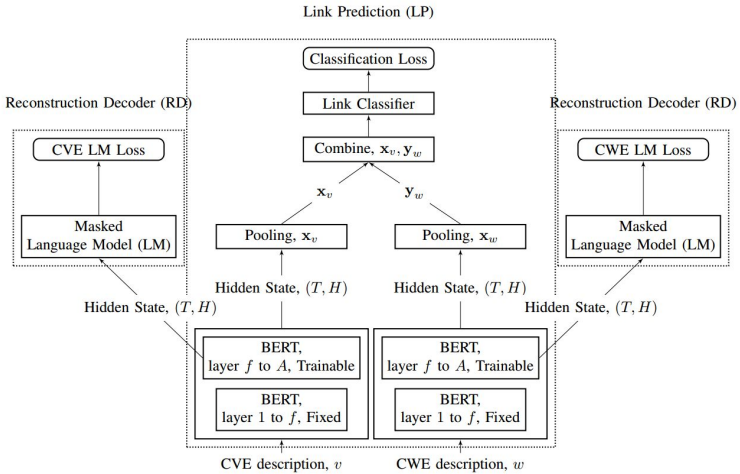

Link Prediction (LP)
Classification Loss
Link Classifier
Combine, $\mathbf{x}_v, \mathbf{y}_w$
$\mathbf{x}_v$
$\mathbf{y}_w$
Pooling, $\mathbf{x}_v$
Pooling, $\mathbf{x}_w$
Reconstruction Decoder (RD)
CVE LM Loss
Masked Language Model (LM)
Reconstruction Decoder (RD)
CWE LM Loss
Masked Language Model (LM)
Hidden State, (T, H)
Hidden State, (T, H)
Hidden State, (T, H)
Hidden State, (T, H)
BERT, layer f to A, Trainable
BERT, layer 1 to f, Fixed
BERT, layer f to A, Trainable
BERT, layer 1 to f, Fixed
CVE description, v
CWE description, w
CE Loss on masked tokens
Linear, (input, output) (H, N_{vocab})
Layer Normalization
Linear, (input, output) (H, H)
Last Hidden State Tensor, (T, H)
BERT, Layer 1 to A, Trainable
CVE and CWE descriptions, u_{cve}, v_{cwe}

링크 예측 (LP)

목적: CVE를 CWE에 매핑

세부 구성 요소:

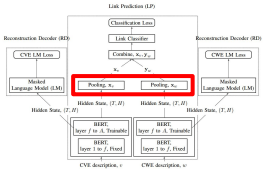
- **Pooling: CVE와 CWE 설명을 고정 크기 벡터로 변환**
  - Mean-Pooling이 가장 좋은 성능을 보임
- **Combination: CVE 벡터와 CWE 벡터를 결합**
  - 절대 차이와 곱셈의 연결 (|xv − yw|, xv × yw)이 최고 성능
- **Link Classification: 결합된 벡터를 Link와 Unlink로 분류**
  - 2개의 뉴런을 가진 출력 레이어와 Softmax 활성화 함수 사용
  - Cross-Entropy Loss 손실함수로 최적화



동작 방식:

1. BERT 인코더를 사용하여 CVE와 CWE 설명을 변환
2. Pooling을 통해 고정 크기 벡터 표현 획득
3. Combination을 통해 CVE 벡터와 CWE 벡터 결합
4. Link Classification을 통해 Link 또는 Unlink로 분류
5. 가장 높은 Link 값을 가진 CVE-CWE 쌍 선택

링크 예측 : Pooling



- **정의:** 입력 시퀀스의 토큰 표현을 집계하여 고정 크기의 벡터로 변환하는 과정

- **필요성 및 효과**

  - 가변 길이의 입력을 **고정 크기**의 벡터로 변환하여 다음 레이어에서 처리할 수 있게 함

  - 입력 시퀀스의 **전역적인 특징을 포착**

  - **계산 효율성을 높이고** 모델의 **복잡성을 줄임**

- **V2W-BERT에서의 적용**

  - 기본값: [CLS] 토큰의 Hidden state가 풀링된 벡터 표현으로 사용됨

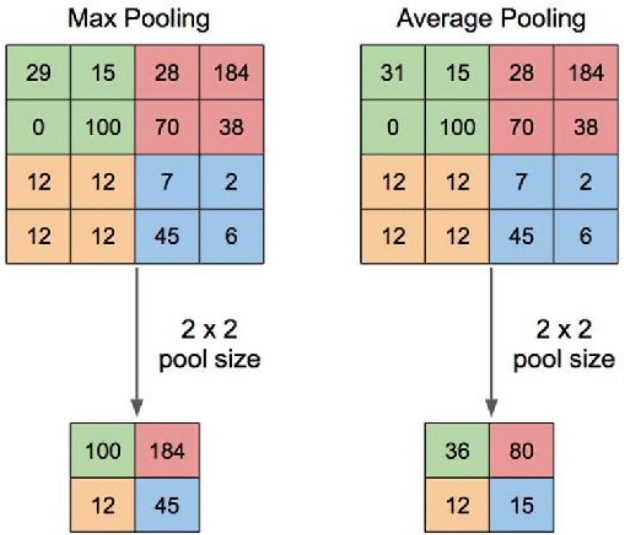
  - V2W-BERT는 두 가지 추가 풀링 방법을 고려함:

    - MAX-풀링: 모든 토큰의 표현 벡터 중 MAX 값을 사용
    - MEAN-풀링: 벡터의 MEAN 값을 사용

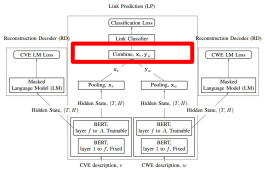
  - 풀링된 표현은 최종 표현을 위해 변환 레이어를 통과함

  - CVE 분류를 위해 MEAN-풀링이 가장 좋은 성능을 보임

  - 풀링된 벡터 표현은 CVE의 경우 xv, CWE의 경우 yw로 표시됨



링크 예측 : Combination



- 정의: 두 개의 풀링된 벡터 표현을 결합하여 입력 쌍 사이의 관계를 나타내는 새로운 벡터를 생성하는 과정

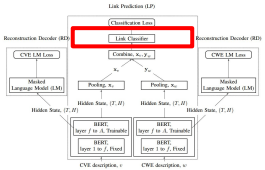
- 필요성 및 효과

- 입력 쌍 사이의 유사성, 차이, 또는 상호작용을 포착할 수 있음
- 결합 방법의 표현력에 따라 모델의 성능이 크게 달라질 수 있음

- V2W-BERT에서의 적용

- 입력 시퀀스 쌍의 풀링된 표현은 다양한 방식으로 결합될 수 있음
- 일반적인 연산: 연결, 곱셈, 덧셈, 집합 연산, 조합
- V2W-BERT에 가장 적합한 연산: 절대 차이와 곱셈의 연결 (|xv − yw|, xv × yw)
- 결합 선택에 따라 성능에 큰 차이가 있음 (부록 VIII-D 참조)

링크 분류 : Classification



- 정의: 결합된 벡터 표현을 사용하여 두 개체 사이의 관계 또는 연결 여부를 예측하는 과정

- 필요성 및 효과

- 개체 사이의 관계를 예측하여 지식 그래프 구축, 추천 시스템, 질의 응답 등의 작업에 활용할 수 있음

- 소프트맥스 활성화 함수를 사용하여 관계의 확률을 출력할 수 있음

- 교차 엔트로피 손실을 사용하여 모델을 최적화 할 수 있음

V2W-BERT에서의 적용

- 결합된 표현은 Link와 Unlink 신뢰도 값으로 분류됨

- 선형 레이어 사용: 두 개의 뉴런 & Softmax 활성화 함수 → CVE-CWE 연결 신뢰도: Softmax 값 [0,1] → Link/Unlink 분류

단일 뉴런을 사용한 링크 예측:

- 결합된 벡터 표현을 단일 뉴런에 전달하여 0과 1 사이의 값 출력
- 출력값이 특정 임계값(예: 0.5) 이상이면 Link로, 그렇지 않으면 Unlink로 분류
- Sigmoid 활성화 함수를 사용하여 출력값을 0과 1 사이로 제한

두 개의 뉴런을 사용한 링크 예측:

- 결합된 벡터 표현을 두 개의 뉴런으로 구성된 출력 레이어에 전달
- 각 뉴런은 Link와 Unlink에 해당하는 신뢰도 값을 출력
- Softmax 활성화 함수를 사용하여 두 뉴런의 출력값을 확률 분포로 변환
- 두 개의 뉴런 중 높은 확률 값을 가진 뉴런에 해당하는 클래스(Link 또는 Unlink)로 분류

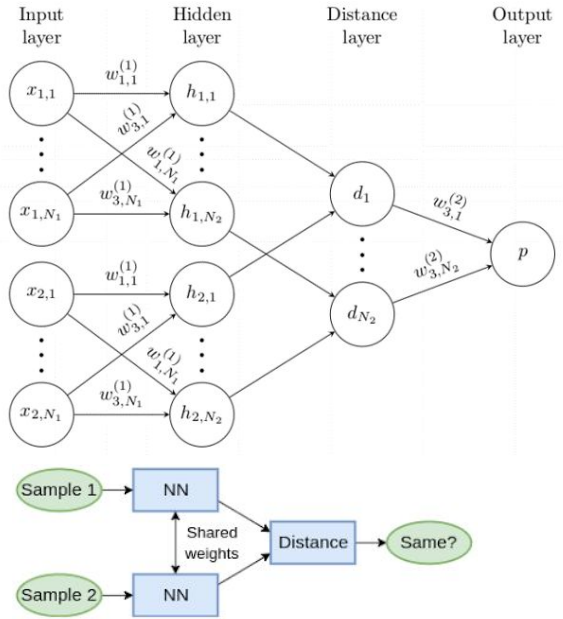
재구성 디코더 (RD)

목적: CVE와 CWE 설명 맥락을 보존하여 희귀 CWE 분류 성능 향상

세부 구성 요소:

- BERT 인코더의 마지막 Hidden State를 Masked LM에 전달
- Masked 토큰에 대해 최적화
- Cross-Entropy Loss를 사용하여 입력과 재구성된 토큰 사이의 차이 최소화

$$H(P,Q) = -\mathbb{E}_{X\sim P}[\log Q(x)] = -\sum P(x)\log Q(x)$$



동작 방식:

1. LP와 RD는 BERT의 Hidden State를 공유(샴 네트워크)
2. BERT 인코더가 CVE/CWE 설명을 변환할 때, 마지막 Hidden State를 Masked LM에 전달
3. Masked LM은 Masked 토큰에 대해 최적화
4. LP와 RD의 학습 가능한 레이어는 Link Classification Loss와 Reconstruction Loss를 동시에 고려하여 업데이트
5. 이를 통해 LP는 링크 분류를 학습하는 동시에 RD는 맥락을 보존

훈련 세부사항

CVE-CWE 링크 정보 수집

• Positive link: CVE가 속한 CWE 및 그 조상 CWE와의 쌍

• Negative link: CVE와 연관되지 않은 나머지 CWE와의 쌍

데이터 준비

• CVE 설명과 CWE 설명 쌍 구성

• CVE-CWE 쌍을 positive 및 negative link로 라벨링

배치 구성

• Mini-batch 내 positive 및 negative link 수를 균등하게 조정

• 학습 편향 방지를 위한 균형 잡힌 배치 구성모델 학습

• BERT 인코더를 사용하여 CVE 및 CWE 설명 임베딩

• Link Prediction (LP) 모듈을 통해 CVE-CWE 링크 예측

• Reconstruction Decoder (RD) 모듈을 통해 CVE 및 CWE 설명의 맥락 보존

손실 함수 계산

• LP 손실: positive 및 negative link 예측 오차 계산

• RD 손실: 입력 설명과 재구성된 설명 간 차이 계산

• 전체 손실: LP 손실과 RD 손실에 가중치를 적용하여 합산 모델 업데이트

• 계산된 손실을 기반으로 모델 매개변수 업데이트

• 경사 하강법을 사용하여 손실 함수 최소화 학습 반복

• 설정된 에포크 수만큼 학습 과정 반복

• 검증 데이터를 사용하여 모델 성능 평가 및 조기 종료 판단

V2W 예측

V2W-BERT는 학습 및 예측 단계에서 동일한 CWE 계층 구조를 고려함

124개의 CWE가 계층의 3단계에 분포되어 있음 (1단계: 34개, 2단계: 78개, 3단계: 16개)

예측은 계층의 루트에서 리프 노드까지 단계별로 수행됨

사용자는 각 단계에서 예측할 CWE의 수(k)를 조절하여 정밀도를 제어할 수 있음:

- 정밀한 예측: 각 단계에서 가장 높은 신뢰도를 가진 하나의 CWE 선택 (k1=1, k2=1, k3=1)
- 중간 정밀도 예측: 1단계에서 상위 k1≤5개, 2단계에서 각 선택된 CWE의 자식 중 상위 k2≤2개, 3단계에서 최선의 k3≤2개 선택
- 보다 느슨한 예측: k1≤5, k2≤2, k3≤2 선택

실험 설계

데이터셋 정보

- CVE 데이터셋 출처: NVD (National Vulnerability Database)
- 데이터셋 규모: 1999년부터 2020년까지 총 137,101개의 CVE 항목
- 분류된 CVE 항목: 82,382개의 CVE가 916개의 MITRE CWE 중 124개로 분류

데이터 분할

- 시간적 분할(Temporal Partition)
  - 실제 시나리오 시뮬레이션
  - 훈련 세트: 1999-2017 (46,003개 항목)
  - 테스트 세트 1 (가까운 미래): 2018 (14,176개 항목)
  - 테스트 세트 2 (먼 미래): 2019-2020 (22,203개 항목)
- 무작위 분할 (k-fold cross validation)
  - 훈련 70%, 검증 10%, 테스트 20%

V2W-BERT 설정

- 사전 훈련: 25 에포크, 미니 배치 크기 32, 모든 BERT 레이어 업데이트
- CVE-CWE 연관 분석: 20 에포크, 미니 배치 크기 32, 마지막 3개의 BERT 레이어 업데이트
- 링크 설정: CVE당 32개의 무작위 부정 링크, 긍정 링크는 부정 링크 수에 맞추어 반복
- 옵티마이저: AdamW, 학습률 2e-5, 총 훈련 인스턴스의 10%에 해당하는 웜업 스텝
- 하드웨어: 두 개의 Tesla P100-PCIE-16GB GPU 및 20개의 CPU에서 훈련

평가 과정

- 분류 계층: 124개의 CWE를 MITRE 계층 구조의 세 레벨로 분류
- 예측 수행: 계층을 따라 각 레벨에서 예측 수행
- 예측 정밀도 설정: 각 레벨에서 다양한 top ki 값 설정 (정밀 예측에서부터 완화된 예측까지)
- F1 점수: 링크 예측 성능 평가를 위해 정확히 분류된 링크의 F1 점수 사용

실험 결과

본문 성능 평가

- 기본 설정: BERTBASE를 사용한 링크 예측 (LP) 모듈
- V2W-BERT 성능: CVE/CWE 설명을 사용하여 Fine-tuning된 BERT가 기본 모델을 능가함
- 재구성 디코더 (RD) 추가: V2W-BERT의 컨텍스트를 보존, 희귀 CWE 클래스의 성능 향상

희귀 CWE 클래스 학습을 위한 재구성 디코더

- Zero-Shot 학습 성능: RD는 보이지 않는 케이스의 정확도를 향상 (테스트 1에서 최대 86%, 테스트 2에서 최대 61%)
- Few-Shot 학습 성능: 과거에 51-100개의 훈련 인스턴스만 가진 상태에서 2018년에 71%-84%의 예측 정확도 달성
- 관련 작업 대비 유의미한 개선: 희귀 경우에서 기존 방법론 대비 상당한 성능 향상

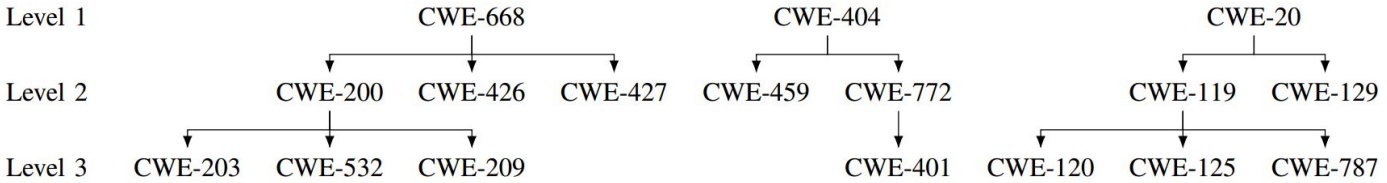


Fig. 4: Partial hierarchy of CWE extracted from MITRE to demonstrate how precise and relaxed prediction is performed.

실험 결과

TABLE III: Performance with randomly partitioned dataset

Model	Test Set (k_1, k_2, k_3)		
	(1,1,1)	(3,2,1)	(5,2,2)
Class, CNN	0.8596	0.9468	0.9645
Class, TF-IDF NN	0.8606	0.9464	0.9668
Link, TF-IDF NN	0.8642	0.9502	0.9693
Class, BERT_{CVE}	0.8812	0.9503	0.9689
Link, V2W-BERT	**0.8916**	**0.9523**	**0.9723**

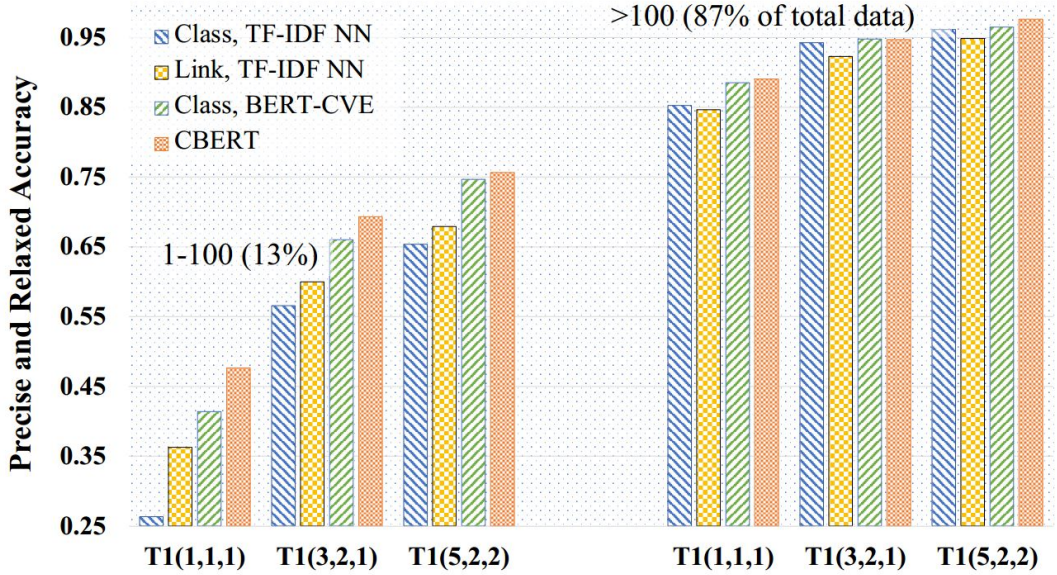


Fig. 6: A summary of the key results for Test 1 (T1) showing superior performance of V2W-BERT with respect to other approaches, especially for rare CWE classes. Details are provided in Table IV.

실험 결과

TABLE IV: Performance comparison of V2W-BERT

	Model	Test 1 (k_1, k_2, k_3)			Test 2 (k_1, k_2, k_3)		
		(1,1,1)	**(3,2,1)**	**(5,2,2)**	**(1,1,1)**	**(3,2,1)**	**(5,2,2)**
1-100	Class, CNN	0.2926	0.5636	0.6373	0.2430	0.4894	0.5823
	Class, TF-IDF NN	0.2631	0.5656	0.6537	0.2519	0.4838	0.5739
	Link, TF-IDF NN	0.3626	0.5998	0.6791	0.3395	0.564	0.659
	Class, BERT_{CVE}	0.4138	0.6602	0.7466	0.2914	0.6105	0.6902
	Link, V2W-BERT	**0.4765**	**0.6933**	**0.7564**	**0.4072**	**0.6293**	**0.7179**
>100	Class, CNN	0.8674	0.9513	0.9721	0.7897	0.9041	0.9430
	Class, TF-IDF NN	0.8524	0.9425	0.9616	0.7815	0.8953	0.9404
	Link, TF-IDF NN	0.8463	0.9227	0.9485	0.7604	0.8738	0.9153
	Class, BERT_{CVE}	0.8852	0.9479	0.9649	0.8067	0.9064	0.9414
	Link, V2W-BERT	**0.8905**	**0.947**	**0.9763**	**0.8113**	**0.9123**	**0.9492**
All	Class, CNN	0.7887	0.8982	0.9263	0.6853	0.8330	0.8822
	Class, TF-IDF NN	0.775	0.893	0.9298	0.6886	0.8231	0.8761
	Link, TF-IDF NN	0.7828	0.8803	0.9132	0.6863	0.8196	0.8706
	Class, BERT_{CVE}	0.8232	0.9101	0.9363	0.7163	0.8578	0.9038
	Link, V2W-BERT	**0.8362**	**0.914**	**0.9442**	**0.7345**	**0.8594**	**0.9151**

실험 결과

Table 2: Zero-shot accuracy with and without RD

Model	Test 1 (k_1, k_2, k_3), 89			Test 2 (k_1, k_2, k_3), 247		
	(1,1,1)	(3,2,1)	(5,2,2)	(1,1,1)	(3,2,1)	(5,2,2)
Random	0.0032	0.0196	0.0653	0.0032	0.0196	0.0653
LP	0.1263	0.5454	0.8483	0.0273	0.2568	0.5902
LP+RD	**0.2809**	**0.6954**	**0.8558**	**0.1012**	**0.3475**	**0.6104**

Table 3: Few-shot accuracy evaluated for rare CWE classes with different training instances between $[n_1, n_2]$

Model	Test 1, n=[1, 50], 1057			Test 2, n=[1, 50], 2632		
(k_1, k_2, k_3)	(1,1,1)	(3,2,1)	(5,2,2)	(1,1,1)	(3,2,1)	(5,2,2)
LP	0.2142	0.4991	0.671	0.2462	0.5151	0.6306
LP+RD	**0.3199**	**0.6176**	**0.705**	**0.2474**	**0.5569**	**0.6736**
	Test 1, n=[51, 100], 800			Test 2, n=[51, 100], 1221		
LP	0.5687	0.8075	**0.8400**	0.5652	0.7771	**0.8054**
LP+RD	**0.7087**	**0.8087**	0.8375	**0.6457**	**0.7870**	0.8035
	Test 1, n=[101, 150], 690			Test 2, n=[101, 150], 1643		
LP	0.6645	0.8373	0.9097	0.4221	0.6605	0.7639
LP+RD	**0.7238**	**0.8475**	**0.9222**	**0.5091**	**0.6648**	**0.7849**

결론

주요 시사점 및 향후 방향성

- [P] CWE 데이터의 계층적 비분리성 문제
  [S] 그래프 링크 예측 문제로 변환

  → 추후 여타 취약점 및 사이버보안 DB의 데이터 처리 시 계층적 비분리성에 대한 대처 가능

- [P] CVE, CWE 간의 semantic gap 문제
  [S] Siamese Net 기반의 hidden state 공유로 해결

  → semantic gap를 해결하기 위한 가중치를 공유하는 여러 방법 추가 조사

- [P] 희귀 CWE 데이터 부족 문제
  [S] One-shot Learning 기법인 Siamese Net 활용 및 Reconstruction Decoder 기반의 맥락 보존

  → 데이터 및 데이터 간 상관성 부족에 대처하는 방법론 획득

  → 다른 Few-Shot Learning 기법에 다른 추가 조사