Open World Classification with Adaptive Negative samples

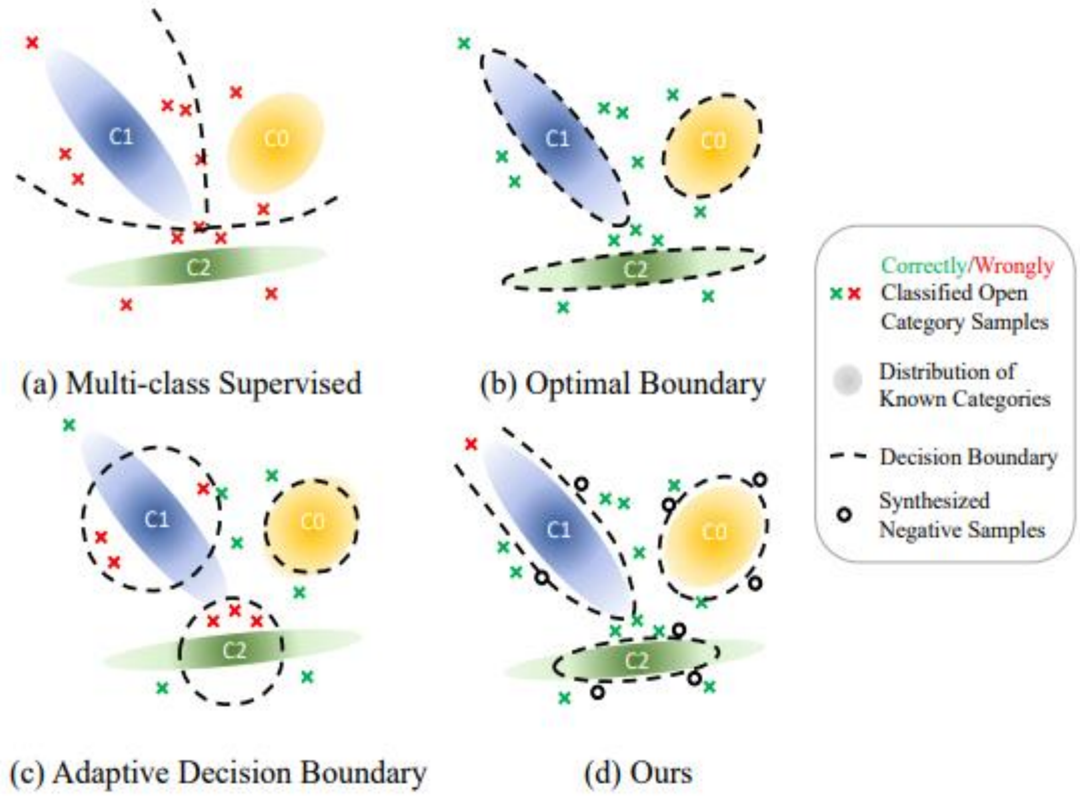
Ke Bai, Guoyin Wang, Jiwei Li, Sunghyun Park, Sungjin Lee,
Puyang Xu, Ricardo Henao, Lawrence Carin

Duke University, Amazon, Zhejiang University, KAUST

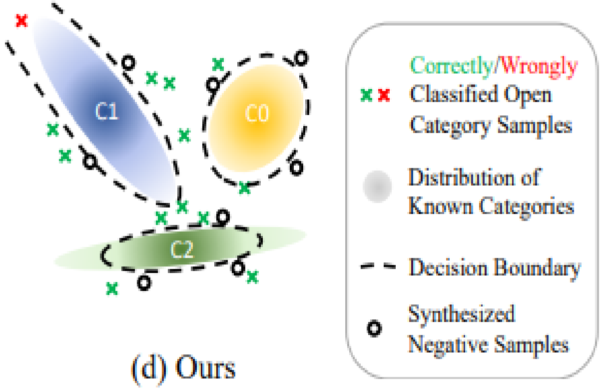
EMNLP 2022

Existing methods

- 연구 초점
  - Known class 사이에 더 tight 한 결정경계 추정
  - 더 좋은 feature representation 학습
- 성능 제한 원인
  - **Training 동안 open class data의 부족**
  - **적절한 decision boundary 를 학습하기에 좋은 mechanism 의 부족**

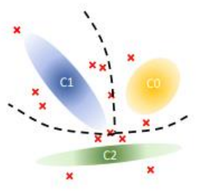


Adaptive Negative Samples (ANS)

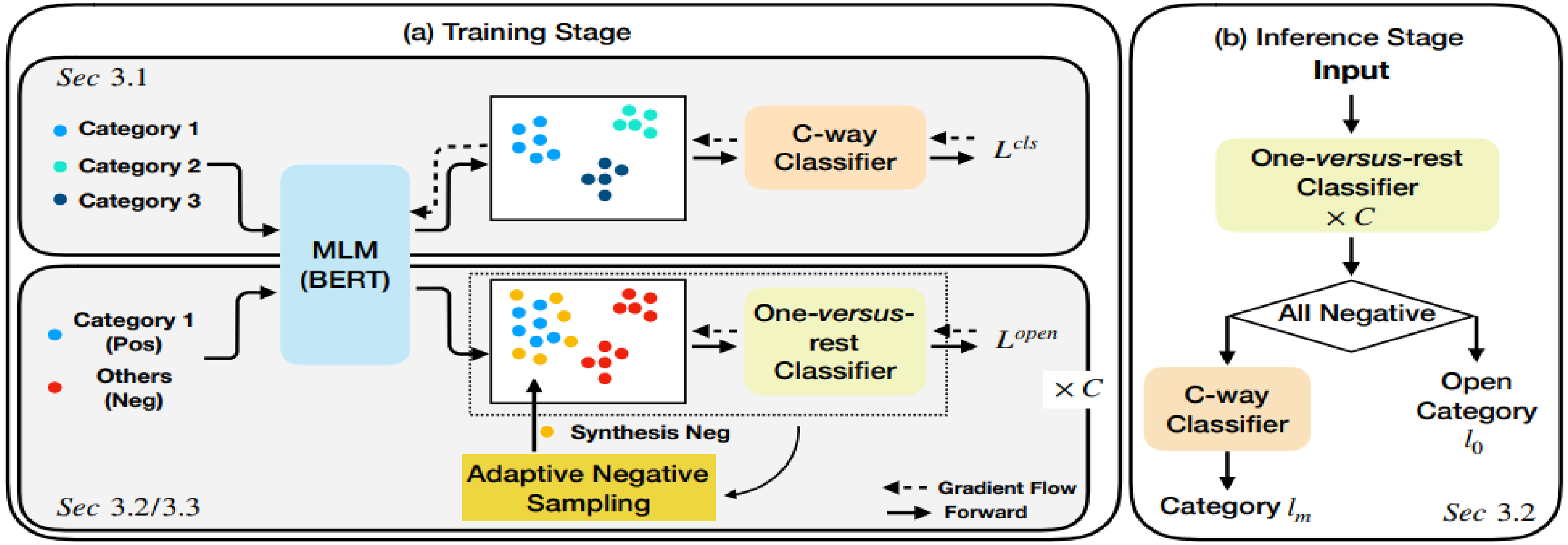


(d) Ours

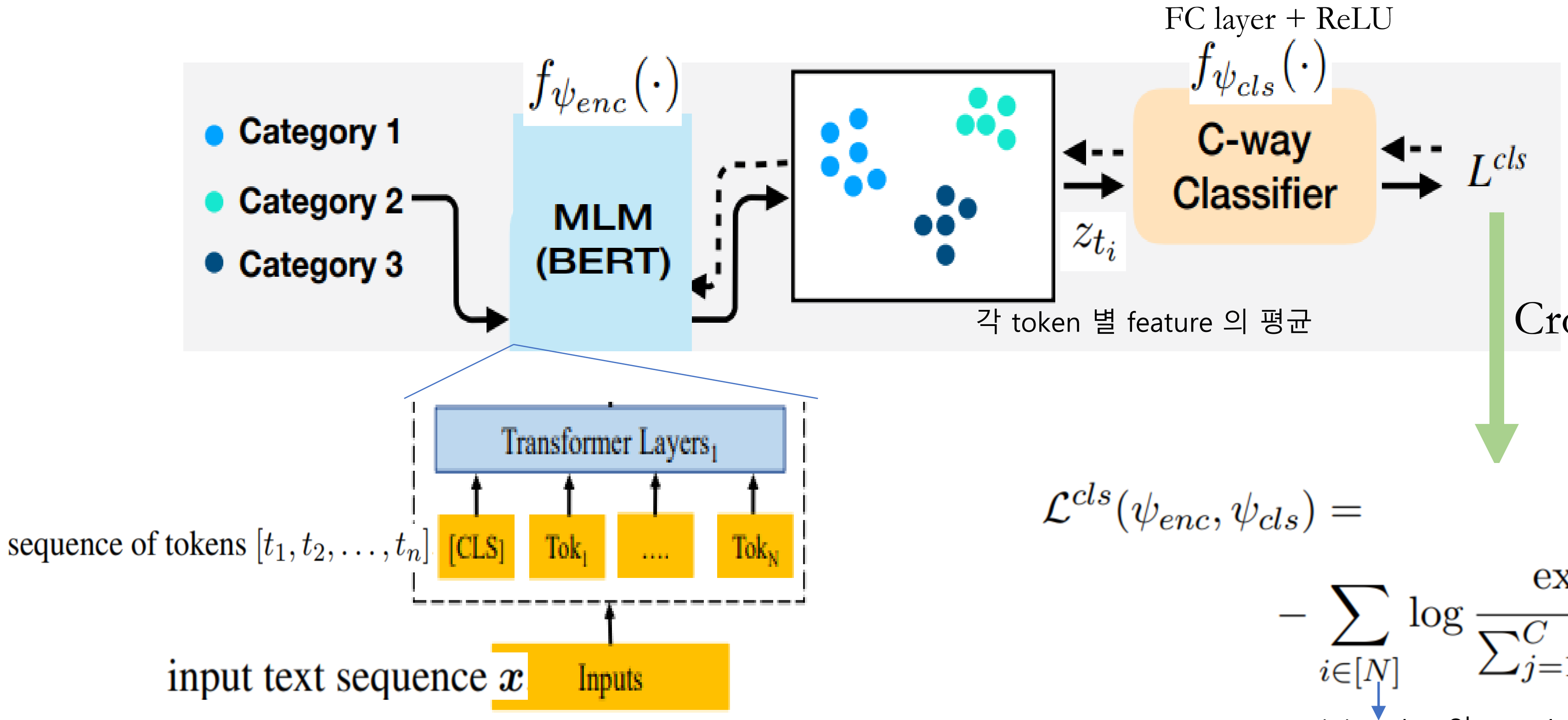
- Adaptive Negative Samples(ANS) 제안
  - 학습 동안 prior knowledge, external datasets **없이** negative sample 생성
- one-versus-rest binary classifiers를 사용하여 더 좋은 결정 경계 학습



(a) Multi-class Supervised



Known Category Classification



Cross-entropy loss

$$\mathcal{L}^{cls}(\psi_{enc}, \psi_{cls}) = \qquad (1)$$

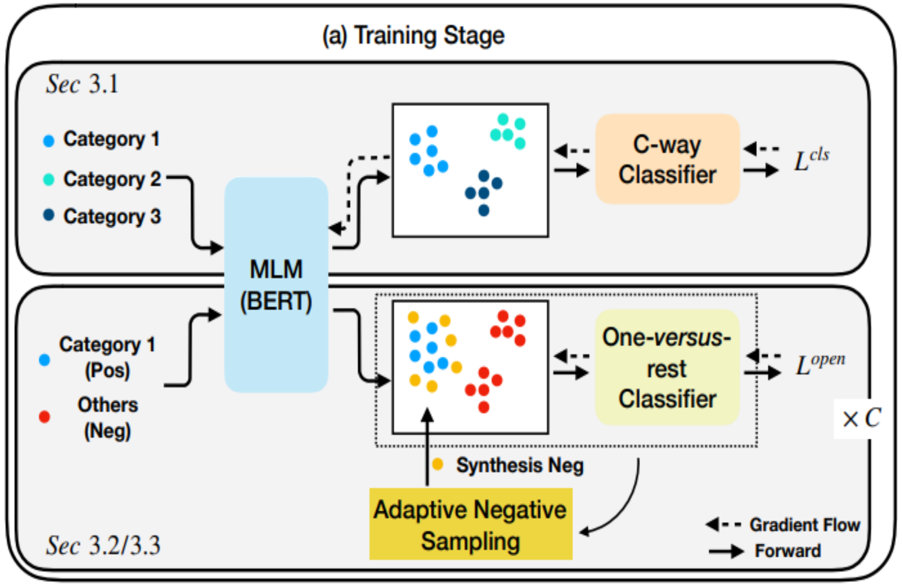
$$-\sum_{i\in[N]} \log \frac{\exp(f^{y_i}_{\psi_{cls}}(\boldsymbol{z}_i))}{\sum_{j=1}^{C} \exp(f^{j}_{\psi_{cls}}(\boldsymbol{z}_i))},$$

Training data의 sample 총 개수

Output logit for the j-th category

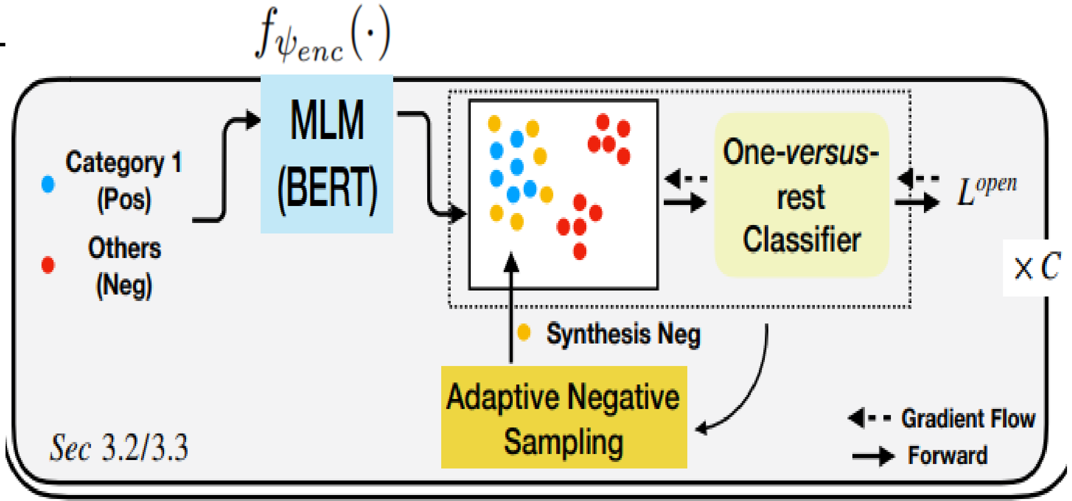
Open Category Recognition

- **Open world classification 목표 : Known 분류하는 동시에 Open 분류**
- Known category classifier $f_{\psi_{cls}}(\cdot)$ (C-way classifier) 단점
  - 단독 사용 시 (C+1) category classifier 에 비해 낮은 성능
- (C+1) category classifier 단점
  - Open category data 부족
- 제안한 방법
  - **One-versus-rest framework 제안**
    - Binary classifier 학습 통해 생성
    - One-versus-rest Classifier 학습시키기 위해 효과적인 open-sample generation 방법 제시

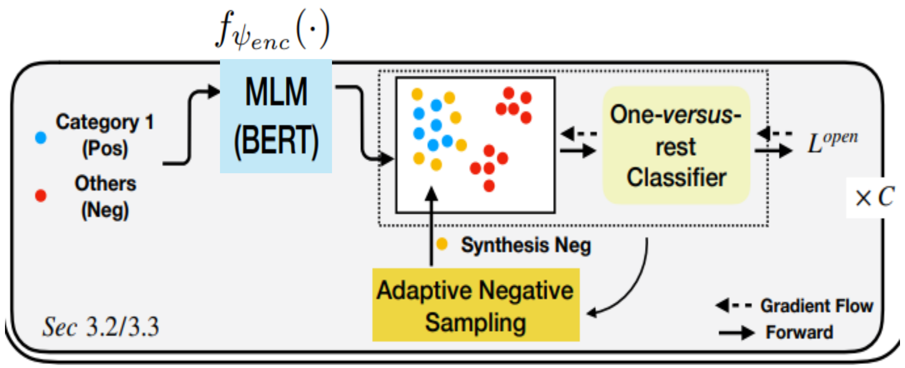


Open Category Recognition

- **One-versus-rest binary Classifier**
  - FC three-layer neural network with ReLU
  - known 분류위해 학습시킨 feature extractor $f_{\psi_{enc}}(\cdot)$사용하여 feature $z \in \mathbb{R}^d$ 추출
  - Binary classifier $g_{\theta_m^{cls}}(z) : \mathbb{R}^d \to \mathbb{R}$
    - $g_{\theta_m^{cls}}(z) > 0$ 이면 input x는 m-th category 로 분류
    - 0보다 작으면 다른 category 로 분류



Open Category Recognition



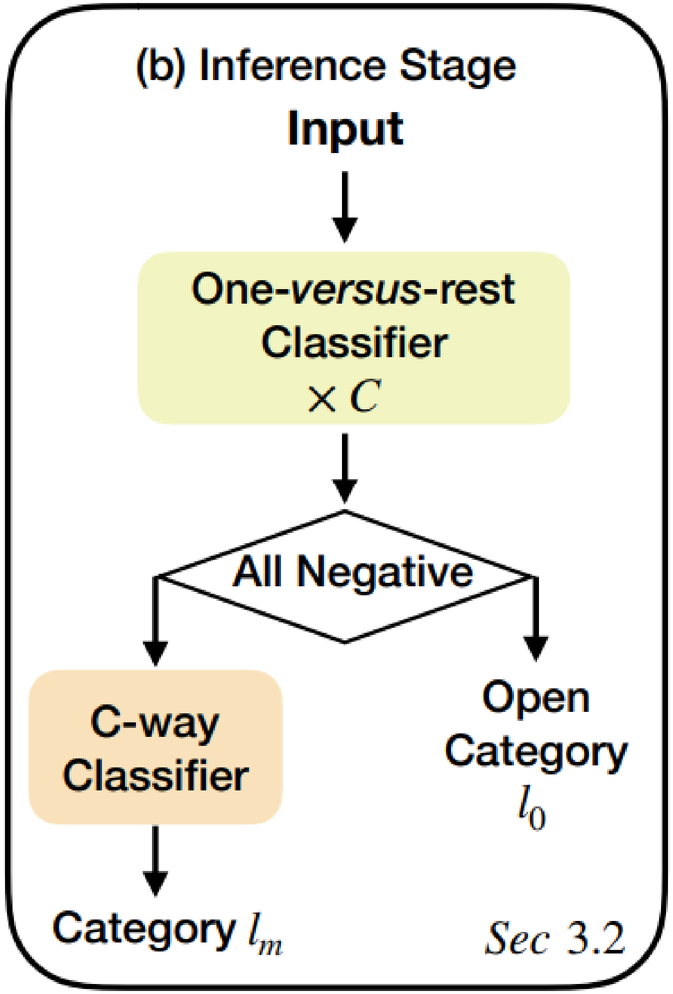
- **One-versus-rest binary Classifier**
  - m-th category 에 대한 전체 binary classification framework
    - 학습 위한 데이터 **training data** $\mathcal{D}$
      - Positive set $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{N_m}\}$ D에 있는 라벨 l_m의 data point 사용
      - Negative set $\{\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \ldots, \hat{\boldsymbol{x}}_{N-N_m}\}$ D에 없는 라벨 l_m 의 data point 사용
    - Binary cross-entropy loss function $\mathcal{L}^{rest}$

$$\mathcal{L}^{rest}(\theta_m^{cls}) = \sum_{i \in [N_m]} \log(1 + \exp(-g_{\theta_m}(\boldsymbol{x}_i))) + \sum_{i \in [N-N_m]} \log(1 + \exp(g_{\theta_m}(\hat{\boldsymbol{x}}_i))). \qquad (2)$$

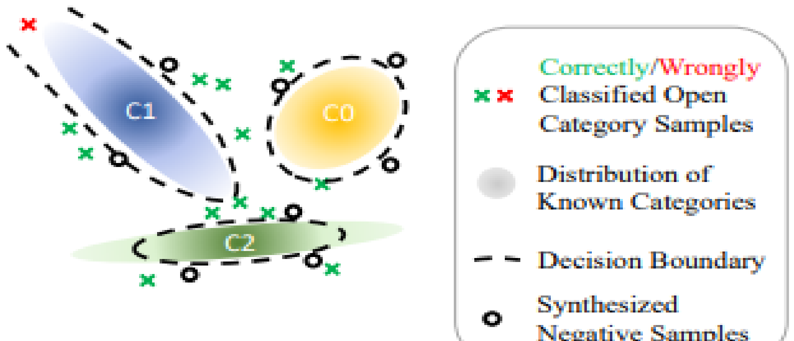
Open Category Recognition

- inference

$$\hat{y} = \begin{cases} \text{open}, & \text{if } g_{\theta_m}(\boldsymbol{x}_i) < 0, \forall m \in [C]; \\ \text{known}, & \text{otherwise}. \end{cases}$$

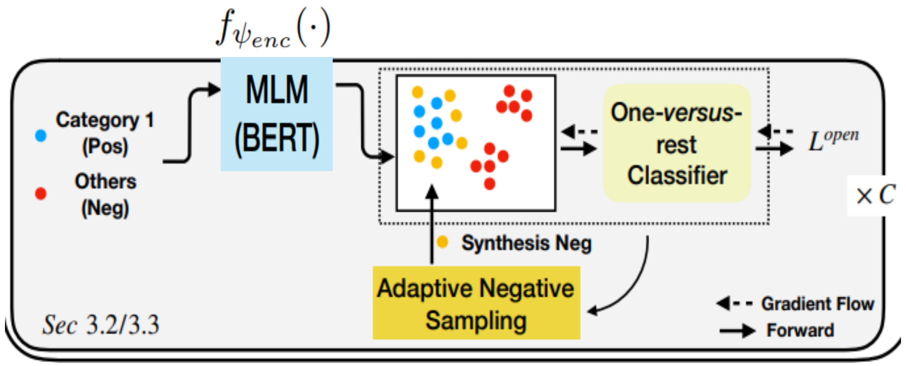


Adaptive Negative Sampling (ANS)



(d) Ours

- 실제 negative sample 부족과 관련된 문제
  - negative sample $\tilde{\boldsymbol{x}}$ 생성 제안
  - 실제 open text example 생성 어려움 고려하여 feature 공간에 virtual text example $\tilde{\boldsymbol{z}}$ 표시
- feature 공간에 known samples 와 가까이 있는 포인트들을 보는 것이 decision boundary 형성에 도움이 됨



Adaptive Negative Sampling (ANS)

- Data 가 Low-dimensional manifold 에 놓인다라고 가정
  - Euclidean 으로 거리 계산 가능
  - **The set of pseudo negative samples (adaptive synthesized open set)**

$$\mathcal{N}_i(r) =: \left\{ \tilde{z} : r \le \|\tilde{z} - z_j\|_2; \|\tilde{z} - \boxed{z_i}\|_2 \le \gamma \cdot r, \forall j : y_j = m \right\}$$

Distance radiance

hyperparameter $\gamma > 1$

- **2가지 조건**
  - 모든 known sample 로부터 synthesized sample 떨어져 있도록
  - Synthesized sample 과 known 이 너무 멀지 않도록

Adaptive Negative Sampling (ANS)

$\sqrt{2\,\mathrm{Tr}(\Sigma)}$: Covariance matrix의 분포로부터 나온 random samples 사이의 유클리드 거리의 평균

- Radius r 구하기
  - Category m의 known samples 을 사용하여 z 의 Covariance matrix Σ 계산
  - 아래 두 조건을 만족하는 r 선택

    $r \leq \underline{\sqrt{2\,\mathrm{Tr}(\Sigma)}} \qquad \gamma r \geq \underline{\sqrt{2\,\mathrm{Tr}(\Sigma)}}$

    - 명제 따름

Proposition 1 *The expectation of the euclidean distance between random points sampled from distribution with covariance matrix* Σ *is smaller than* $\sqrt{2\,\mathrm{Tr}(\Sigma)}$, *i.e.*

$$\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim\mathcal{D}(\boldsymbol{\mu},\Sigma)}\sqrt{\|\boldsymbol{x}-\boldsymbol{y}\|^2} \leq \sqrt{2\,\mathrm{Tr}\,\Sigma} \qquad (4)$$

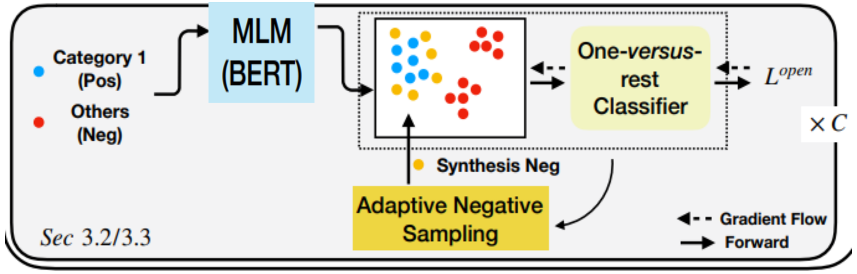
$$\begin{aligned}\mathbb{E}(\|\boldsymbol{x}-\boldsymbol{y}\|_2^2) &= \sum_i \mathbb{E}(\boldsymbol{x}_i^2 - 2\boldsymbol{x}_i\boldsymbol{y}_i + \boldsymbol{y}_i^2) \\ &= 2\sum_i (\mathbb{E}(\boldsymbol{x}_i^2) - \mathbb{E}(\boldsymbol{x}_i)\mathbb{E}(\boldsymbol{y}_i)) \\ &= 2\sum_i (\mathbb{E}(\boldsymbol{x}_i^2) \\ &= 2\,\mathrm{Tr}(\Sigma)\end{aligned}$$

$$\mathbb{E}(\sqrt{Z}) \leq \sqrt{\mathbb{E}[Z]}$$

Random variable

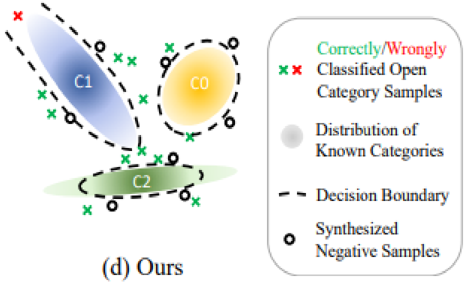
combination $\qquad \mathbb{E}(\|\boldsymbol{x}-\boldsymbol{y}\|_2) \leq \sqrt{2\,\mathrm{Tr}(\Sigma)}$

Adaptive Negative Sampling (ANS)



- **Binary Classification with ANS**
  - Known category m의 각 sample 은 open sample set 의 adaptive synthesized 제공

    $\mathcal{N}_i(r)$ $\tilde{z}$
  - Binary cross-entropy loss



(d) Ours

$$\mathcal{L}^{syn}(\theta_m^{cls}) = \sum_{i \in [N_m]} \log(1 + \exp(g_{\theta_m^{cls}}(\tilde{z}_i)))$$

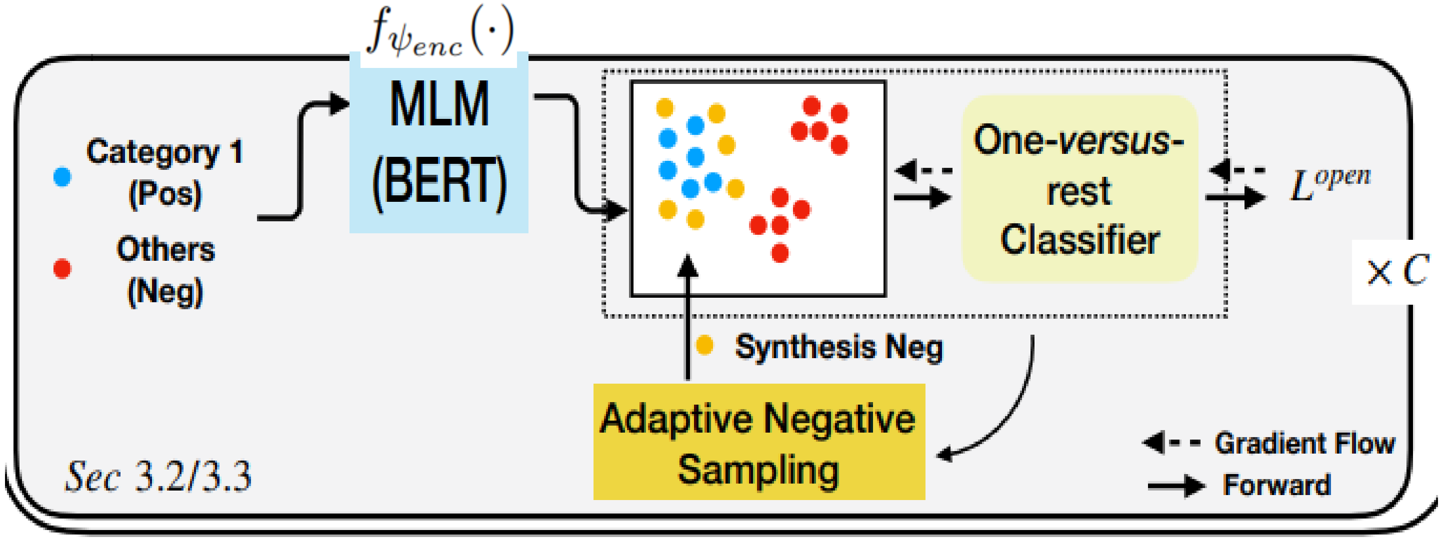
Open set 에 수많은 point가 존재하여 랜덤보다 negative라고 분류하기 어려운 sample 고르는 것이 더 효과적

$$\mathcal{L}^{syn}(\theta_m^{cls}) = \sum_{i \in [N_m]} \boxed{\max_{\tilde{z}_i \in \mathcal{N}_i(r)}} \log(1 + \exp(g_{\theta_m^{cls}}(\tilde{z}_i)))$$

N_m : l_m 에 있는 known samples 전체 수

Adaptive Negative Sampling (ANS)

- Complete loss $\mathcal{L}^{open} = \mathcal{L}^{rest} + \lambda \mathcal{L}^{syn},$



$$\mathcal{L}^{rest}(\theta_m^{cls}) = \sum_{i \in [N_m]} \log(1 + \exp(-g_{\theta_m}(\boldsymbol{x}_i))) + \sum_{i \in [N - N_m]} \log(1 + \exp(g_{\theta_m}(\hat{\boldsymbol{x}}_i))). \quad (2)$$

$$\mathcal{L}^{syn}(\theta_m^{cls}) = \sum_{i \in [N_m]} \max_{\tilde{\boldsymbol{z}}_i \in \mathcal{N}_i(r)} \log(1 + \exp(g_{\theta_m^{cls}}(\tilde{\boldsymbol{z}}_i)))$$

Adaptive Negative Sampling (ANS)

- Projected Gradient Descend-Ascent
  - Gradient descent
    - $\mathcal{L}^{open}$ minimize 위해 사용
  - Gradient ascent
    - 가장 어려운 synthetic negative samples 찾기 위해 사용

Algorithm 1 Adaptive Negative Sampling.

Input: Training data $\mathcal{D} = \boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n, \hat{\boldsymbol{x}}_1, \cdots, \hat{\boldsymbol{x}}_n$. Parameters of current binary classifier θ_m.
Hyper-Parameters: Radius r, step-size η, number of gradient steps k
for Batch number $B = 1, \cdots, n_0$ **do**
    $X_B(\hat{X}_B)$: Collect a batch of positive (negative) samples.
    Calculate loss $\mathcal{L}_1 = \mathcal{L}^{real}(X_B, \hat{X}_B)$ using Eq. 2
    Calculate the feature $\boldsymbol{z}_B$ over the positive samples.
    Adaptive Negative Sampling:
    sample $\boldsymbol{\epsilon} \sim \mathbb{N}(0, 4 \cdot diag(\Sigma_{\boldsymbol{z}}))$
    for $i = 1, \cdots, k$ **do**
        Calculate loss $\ell(\boldsymbol{z}_B + \boldsymbol{\epsilon})$ using Eq. 5
        $\boldsymbol{\epsilon} = \boldsymbol{\epsilon} + \eta \frac{\nabla_{\boldsymbol{\epsilon}} \ell(\boldsymbol{z}_B + \boldsymbol{\epsilon})}{\|\nabla_{\boldsymbol{\epsilon}} \ell(\boldsymbol{z}_B + \boldsymbol{\epsilon})\|}$ ▷ Gradient Ascend
    end for
    Calculate α using Eq.7
    $\boldsymbol{\epsilon} = \frac{\alpha}{\|\boldsymbol{\epsilon}\|} \cdot \boldsymbol{\epsilon}$
    Calculate loss $\mathcal{L}_2 = \mathcal{L}^{syn}_{\theta'_m}(\boldsymbol{z}_B + \boldsymbol{\epsilon})$ using Eq. 5
    $\theta_m = \theta_m - \nabla_{\theta_m}(\mathcal{L}_1 + \mathcal{L}_2)$ ▷ Gradient Descend
end for

Adaptive Negative Sampling (ANS)

- **Projected Gradient Descend-Ascent**
  - $\tilde{z}_i{}' = \tilde{z}_i + \epsilon$ 직접 gradient ascent 진행

    $\longrightarrow \mathcal{N}_i(r)$ 공간 벗어날 가능성 존재

    $\longrightarrow \tilde{z}_i = \arg\min_{\boldsymbol{u}} \|\tilde{z}_i{}' - \boldsymbol{u}\|^2, \forall\ \boldsymbol{u} \in \mathcal{N}_i(r)$

    위의 제약조건에 따른 가장 가까운 $\tilde{\boldsymbol{z}}_i$ 찾기
  - 앞에서 정의한 $\mathcal{N}_i(r)$은 계산 복잡

    $\longrightarrow$ 가정과 관찰: training sample이 저차원에 있고 합성된 sample이 known sample 과 가까이 존재

    $\longrightarrow \boldsymbol{z}_i$: $\mathcal{N}_i(r) = \{\tilde{\boldsymbol{z}} : r \leq \|\tilde{\boldsymbol{z}} - \boldsymbol{z}_i\|_2 \leq \gamma \cdot r\}$
- **Projection** $\tilde{z}_i = \tilde{z}_i{}' + \alpha \frac{\tilde{z}_i{}' - z_i}{\|\tilde{z}_i{}' - z_i\|}$

$$\alpha = \begin{cases} 1, & \text{if } r \leq \|\tilde{\boldsymbol{z}}_i - \boldsymbol{z}_i\|_2 \leq \gamma \cdot r \\ \frac{r\gamma}{\|\tilde{\boldsymbol{z}}_i - \boldsymbol{z}_i\|_2}, & \text{if } \gamma \cdot r \leq \|\tilde{\boldsymbol{z}}_i - \boldsymbol{z}_i\|_2 \\ \frac{r}{\|\tilde{\boldsymbol{z}}_i - \boldsymbol{z}_i\|_2}, & \text{if } \|\tilde{\boldsymbol{z}}_i - \boldsymbol{z}_i\|_2 \leq r \end{cases}$$

Algorithm 1 Adaptive Negative Sampling.

```
Input: Training data D = x_1, x_2, ..., x_n, x̂_1, ..., x̂_n.
    Parameters of current binary classifier θ_m.
Hyper-Parameters: Radius r, step-size η, number of gradient steps k
for Batch number B = 1, ..., n_0 do
    X_B(X̂_B): Collect a batch of positive (negative) samples.
    Calculate loss L_1 = L^real(X_B, X̂_B) using Eq. 2
    Calculate the feature z_B over the positive samples.
    Adaptive Negative Sampling:
    sample ε ~ N(0, 4 · diag(Σ_z))
    for i = 1, ..., k do
        Calculate loss ℓ(z_B + ε) using Eq. 5
        ε = ε + η ∇_ε ℓ(z_B+ε) / ||∇_ε ℓ(z_B+ε)||      ▷ Gradient Ascend
    end for
    Calculate α using Eq.7
    ε = α / ||ε|| · ε
    Calculate loss L_2 = L^syn_{θ'_m}(z_B + ε) using Eq. 5
    θ_m = θ_m − ∇_{θ_m}(L_1 + L_2)      ▷ Gradient Descend
end for
```

Experiments

- Dataset
  - **BANKING**
    - 은행 관련 dataset
  - **CLINC**
    - OOD detection 위한 dataset
  - **Stackoverflow**
    - Technical question dataset

Dataset	Avg. Samples per category	Avg. Length	Classes
BANKING	117	11.91	77
CLINIC	100	8.31	150
STACKOVERFLOW	600	9.18	20

Table A.2: Statistics of benchmark datasets

Dataset	Category	Examples
Banking	exchange_via_app	What currences are available for exchange? Does your app allow currency exchange from USD to GBP? I want to exchange USD and GBP with the app
	wrong_exchange_rate_for_cash_withdrawal	I was given the wrong exchange rate when getting cash I think I was charged a different exchange rate than what was posted at the time. I need an accurate exchange rate, when I make my withdrawals.
	card_payment_wrong_exchange_rate	Why didn't I receive the correct exchange rate for an item that I purchased? The fee charged when I changed rubles into British pounds was too much. I am being charged the wrong amount on my card.
CLINC	flight_status	what time is this flight supposed to land what time will i be able to board the plane so when is my flight landing
	time	what time is it in adelaide, australia right now please tell me the time how late is it now in ourense
	how_busy	how long will i wait for a table at red lobster can i expect chili's to be busy at 4:30 so how busy is the outback steakhouse at 5 pm
Stackoverflow	wordpress	How Display Recent Posts in all 3 languages at once in Wordpress How secure is Wordpress? Need MySQL Queries to delete WordPress Posts and Post Meta more than X Days Old
	apache	Apache / PHP Disable Cookies for Subdomain ? Increase PHP Memory limit (Apache, Drupal6) is setting the uploads folder 777 permision secure
	excel	Condition to check whether cell is readonly in EXCEL using C# EXCEL XOR multiple bits How I can export a datatable to excel 2007 and pdf from asp.net?

Table A.1: Extracted samples from three main datasets.

Experiments

- **Results**

%	Methods	BANKING Accuracy	BANKING F1-score	CLINC Accuracy	CLINC F1-score	StackOverflow Accuracy	StackOverflow F1-score
25	MSP	43.67	50.09	47.02	47.62	28.67	37.85
	DOC	56.99	58.03	74.97	66.37	42.74	47.73
	OpenMax	49.94	54.14	68.50	61.99	40.28	45.98
	DeepUnk	64.21	61.36	81.43	71.16	47.84	52.05
	ADB	78.85	71.62	87.59	77.19	86.72	80.83
	SelfSup*	74.11	69.93	88.44	80.73	68.74	65.64
	Ours	**83.93**	**76.15**	**92.64**	**84.81**	**90.88**	**84.52**
50	MSP	59.73	71.18	62.96	70.41	52.42	63.01
	DOC	64.81	73.12	77.16	78.26	52.53	62.84
	OpenMax	65.31	74.24	80.11	80.56	60.35	68.18
	DeepUnk	72.73	77.53	83.35	82.16	58.98	68.01
	ADB	78.86	80.90	86.54	85.05	**86.40**	85.83
	SelfSup*	72.69	79.21	88.33	86.67	75.08	78.55
	Ours	**81.97**	**83.29**	**90.23**	**88.01**	86.08	**85.90**
75	MSP	75.89	83.60	74.07	82.38	72.17	77.95
	DOC	76.77	83.34	78.73	83.59	68.91	75.06
	OpenMax	77.45	84.07	76.80	73.16	74.42	79.78
	DeepUnk	78.52	84.31	83.71	86.23	72.33	78.28
	ADB	81.08	85.96	86.32	88.53	82.78	85.99
	SelfSup*	81.07	**86.98**	88.08	89.43	81.71	85.85
	Ours	**82.49**	86.92	**88.96**	**89.97**	**84.40**	**87.49**

Experiments

- **Results**

%	Methods	Banking Open	Banking Known	Clinic Open	Clinic Known	StackOverflow Open	StackOverflow Known
	MSP	41.43	50.55	50.88	47.53	13.03	42.82
	DOC	61.42	57.85	81.98	65.96	41.25	49.02
	OpenMax	51.32	54.28	75.76	61.62	36.41	47.89
25	DeepUnk	70.44	60.88	87.33	70.73	49.29	52.60
	ADB	84.56	70.94	91.84	76.80	90.88	78.82
	SelfSup*	80.12	69.39	92.35	80.43	74.86	63.80
	Ours	**86.39**	**72.29**	**95.33**	**84.53**	**93.96**	**82.63**
	MSP	41.19	71.97	57.62	70.58	23.99	66.91
	DOC	55.14	73.59	79.00	78.25	25.44	66.58
	OpenMax	54.33	74.76	81.89	80.54	45.00	70.49
50	DeepUnk	69.53	77.74	85.85	82.11	43.01	70.51
	ADB	78.44	80.96	88.65	85.00	**87.34**	85.68
	SelfSup*	67.26	79.52	90.30	86.54	71.88	79.22
	Ours	**81.06**	**83.52**	**92.16**	**87.95**	86.83	**85.81**
	MSP	39.23	84.36	59.08	82.59	33.96	80.88
	DOC	50.60	83.91	72.87	83.69	16.76	78.95
	OpenMax	50.85	84.64	76.35	73.13	44.87	82.11
75	DeepUnk	58.54	84.75	81.15	86.27	37.59	81.00
	ADB	66.47	86.29	83.92	88.58	73.86	86.80
	SelfSup*	60.71	87.47	86.28	**89.46**	65.44	87.22
	Ours	**70.54**	**88.13**	**87.20**	89.18	**74.82**	**88.34**

Discussion

- **Synthesized Negative is Beneficial for a Variety of Structures**
  - Synthesized samples 의 기여도와 one-versus-rest의 구조를 확인하기 위해, MSP와 ADB 에 추가하여 실험 진행
    - MSP : softmax 의 max 값을 confidence score로 사용하여 분류하는 방식

Banking 25%

Methods	Add Neg.	Acc	F1	F1-open	F1-known
MSP		44.46	51.95	43.22	52.41
	✓	57.07	58.92	61.40	58.79
ADB		78.39	71.53	84.18	70.86
	✓	79.71	73.01	85.18	72.12
One-vs -Rest		51.33	41.40	50.83	43.54
	✓	80.11	73.35	85.57	72.71

$$\hat{y} = \begin{cases} \text{open}, & \text{if } \max(p(\boldsymbol{x}_i)) < 0.5 \\ \arg\max p(\boldsymbol{x}_i), & \text{otherwise.} \end{cases}$$

$$\hat{y} = \begin{cases} \text{open}, & \text{if } \max(p(\boldsymbol{x}_i)) < 0.5 \\ \text{open}, & \text{if } \arg\max(p(\boldsymbol{x}_i)) == l_0 \\ \arg\max p(\boldsymbol{x}_i), & \text{otherwise} \end{cases}$$

Discussion

- **Adaptive Negative Samples Generation**
  - Negative 로 noise 있는 samples 을 추가
    - Classifier 의 overconfidence 문제 완화
    - 성능 향상
  - Synthesized samples 을 $\mathcal{N}(r)$ 로 제한
    - Synthesized samples 이 known sample과 너무 가깝거나 멀지 않도록 하여 성능 향상
  - Gradient ascent step 추가
    - 성능 강화

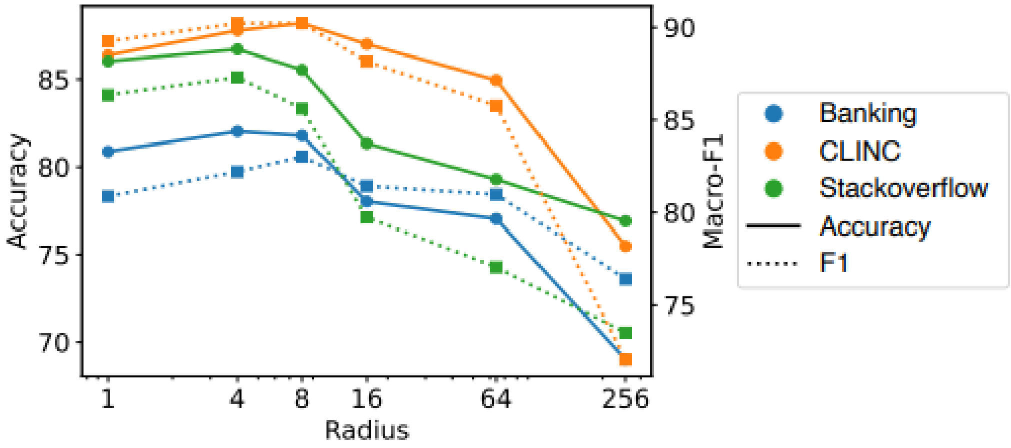
Algorithm 1 Adaptive Negative Sampling.

Input: Training data $\mathcal{D} = \boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n, \hat{\boldsymbol{x}}_1, \cdots, \hat{\boldsymbol{x}}_n$. Parameters of current binary classifier θ_m.
Hyper-Parameters: Radius r, step-size η, number of gradient steps k
for Batch number $B = 1, \cdots, n_0$ **do**
    $X_B(\hat{X}_B)$: Collect a batch of positive (negative) samples.
    Calculate loss $\mathcal{L}_1 = \mathcal{L}^{real}(X_B, \hat{X}_B)$ using Eq. 2
    Calculate the feature $\boldsymbol{z}_B$ over the positive samples.
    Adaptive Negative Sampling:
    sample $\boldsymbol{\epsilon} \sim \mathbb{N}(0, 4 \cdot diag(\Sigma_{\boldsymbol{z}}))$ → Original sample 에 Gaussian noise 추가
    for $i = 1, \cdots, k$ **do**
        Calculate loss $\ell(\boldsymbol{z}_B + \boldsymbol{\epsilon})$ using Eq. 5
        $\boldsymbol{\epsilon} = \boldsymbol{\epsilon} + \eta \frac{\nabla_{\boldsymbol{\epsilon}} \ell(\boldsymbol{z}_B + \boldsymbol{\epsilon})}{\|\nabla_{\boldsymbol{\epsilon}} \ell(\boldsymbol{z}_B + \boldsymbol{\epsilon})\|}$ ▷ Gradient Ascend
    end for
    Calculate α using Eq.7
    $\boldsymbol{\epsilon} = \frac{\alpha}{\|\boldsymbol{\epsilon}\|} \cdot \boldsymbol{\epsilon}$ Projection
    Calculate loss $\mathcal{L}_2 = \mathcal{L}^{syn}_{\theta'_m}(\boldsymbol{z}_B + \boldsymbol{\epsilon})$ using Eq. 5
    $\theta_m = \theta_m - \nabla_{\theta_m}(\mathcal{L}_1 + \mathcal{L}_2)$ ▷ Gradient Descend
end for

%	METHODS	ACC	F1	F1-OPEN	F1-KNOWN
25	BASELINE ($\lambda = 0$)	57.05	53.12	63.83	52.84
	+ GAUSSIAN NOISE	90.37	82.09	93.75	81.78
	+ PROJECTION	92.02	83.99	94.91	83.71
	+ ASCEND (OURS)	92.32	84.34	95.11	84.05
50	BASELINE ($\lambda = 0$)	64.60	71.65	60.28	71.80
	+ GAUSSIAN NOISE	88.01	86.90	89.82	86.86
	+ PROJECTION	90.22	88.18	92.01	88.12
	+ ASCEND (OURS)	90.23	88.22	92.02	88.17
75	BASELINE ($\lambda = 0$)	76.17	83.63	63.23	83.81
	+ GAUSSIAN NOISE	88.67	90.45	86.71	90.48
	+ PROJECTION	88.89	89.95	87.52	89.97
	+ ASCEND (OURS)	88.96	89.97	87.62	90.00

CLINC
Baseline : vanilllia one-versus-rest framework
without the use of synthesized negative samples

Discussion

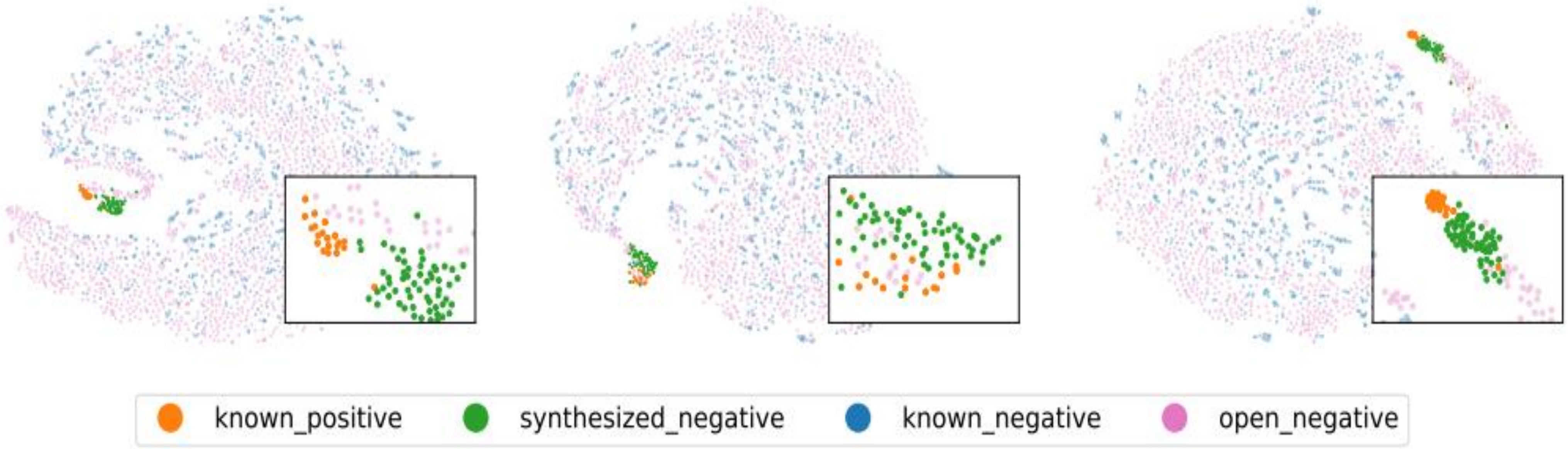


- **Radius r analysis**
  - Radius, gamma : synthesized sample 과 known sample 사이 거리의 상한과 하한 결정하는 hyperparameter
  - 반지름이 커질수록 synthesized negative examples 의 영향이 줄어 성능 감소
    - Synthesized negative examples 와 positive sample 사이의 거리가 늘어나서
  - 반지름이 감소하면 synthesized negative samples이 known samples 에 가깝기 때문에 positive 로 분류 가능성 증가하고 성능 감소
    - Banking, CLINC
    - Stackoverflow 성능은 증가 ⟶ knn 등 사용하여 성능향상 고려
  - **Radius 가 성능에 영향주지만 범위가 넓어야만 확인 가능**

CLINC with 50% known categories
Category 3개 선택
one-*versus*-rest classifier g_m^{cls} 두번째 layer에서 추출된 feature

Discussion

- Visualization
  - **Synthesized negative 는 known positive 와 가깝고 positive 와 다른 known category 사이 이어주는 역할 보여줌**



Conclusions

- Open-world classification 문제를 위한 pseudo open category sample 생성 접근법인 ANS 소개
- 이전 지식 없이 Pseudo sample 생성
- Synthesized samples 효과적
    - One-versus-rest framework 에 합치면 성능 향상
- Gradient-based negative sample generation 을 다른 NLP task에서도 사용 가능할 것

Limitations

- 사용한 dataset 이 training sample의 수 적고 문장 길이도 짧다
- 복잡한 데이터를 다룰 수 있는 model 능력을 만들어야한다
    - 많은 input 문장들이 카테고리 설명이 포함

Thank you

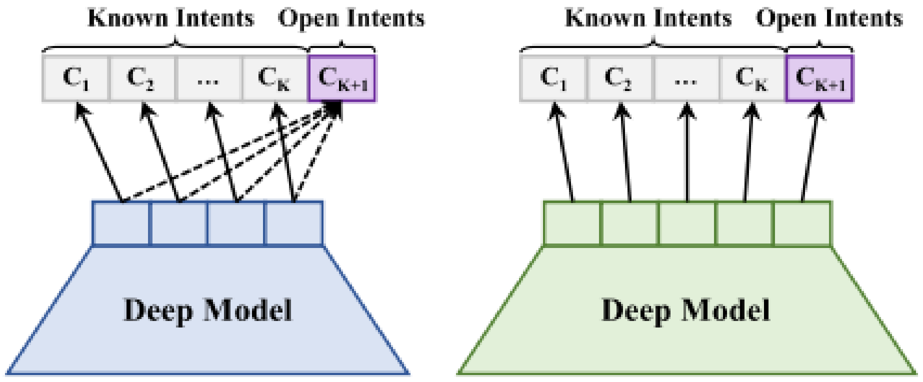
Existing method

- 대부분 k-class classifier와 합쳐진 outlier detection algorithm 사용
  - 현실은 open world 인데 closed world에서만 잘 작용
- Open intent classification 위해 (k+1)-classifier 직접 학습
  - k-class 샘플을 (k+1)-class classifier 로 training 시키기 어려움 존재

k개의 class로 분류

K+1번째 새로운 class = open intent

Utterance	Intent Label
I am still waiting on my card?	Card_arrival
Delete my account please.	Terminate_account
What ATM accepts Mastercard?	ATM_support
What ATMs can I use this card?	ATM_support
……	……
My card isn't working at all.	**Open intents**
Where can I use this card?	**Open intents**



(a) outlier detection **(b) (K+1)-class classification**

SelfSup

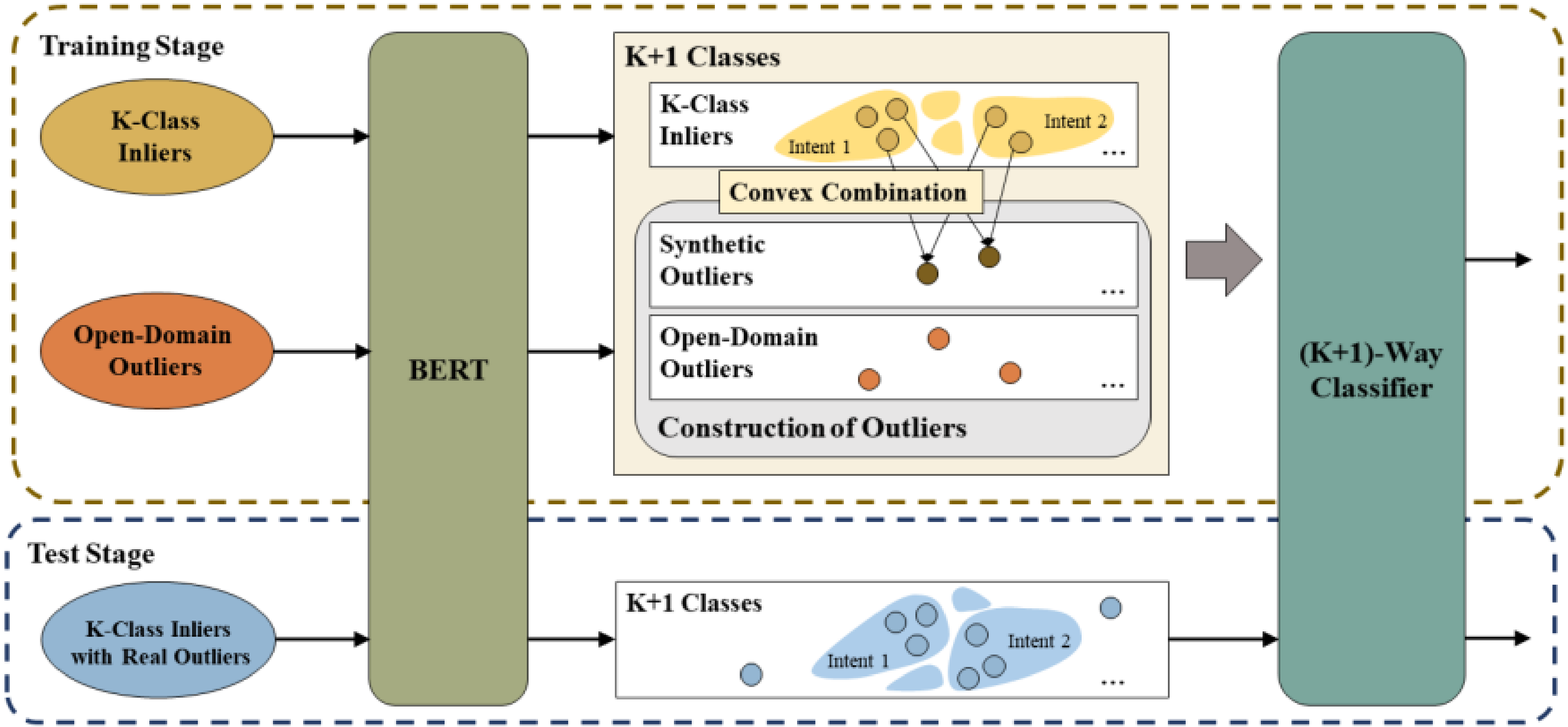
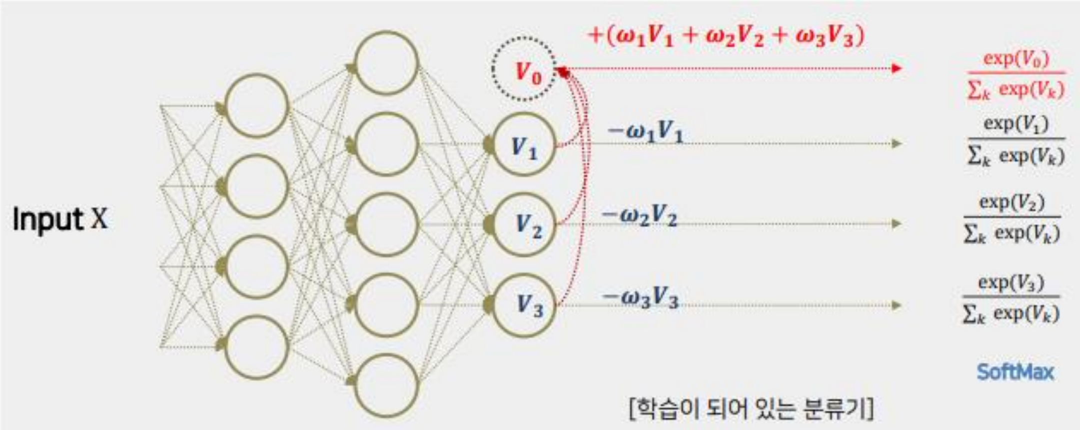


Figure 2: An illustration of our proposed method. We use BERT as the utterance encoder. At training stage, we train a (K+1)-way classifier by constructing two types of pseudo outliers. The open-domain outliers are collected from an auxiliary dataset disjoint from both the training and test data. The synthetic self-supervised outliers are generated during training by random convex combinations of features of inliers from different known classes.

Out-of-Scope Intent Detection with Self-Supervision and Discriminative Training 2021

Baselines

- **OpenMax**
  - 신경망의 학습이 모두 끝난 후, 후처리를 통한 open set recognition
  - Open intent부족으로 threshold 0.5를 default하여 사용
- DOC
- MSO
- DeepUnk



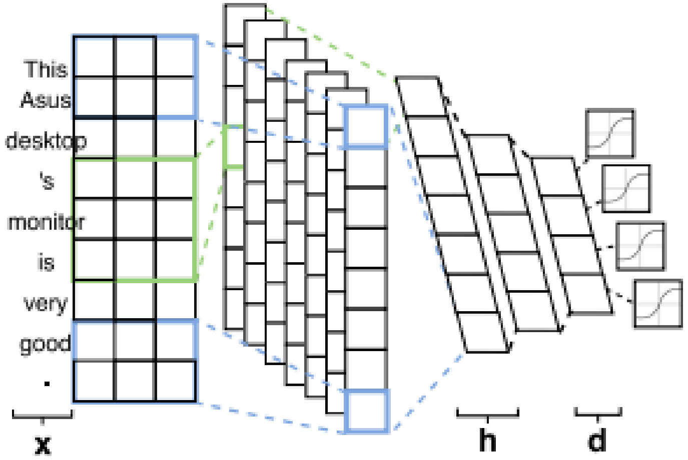


Figure 1: Overall Network of DOC

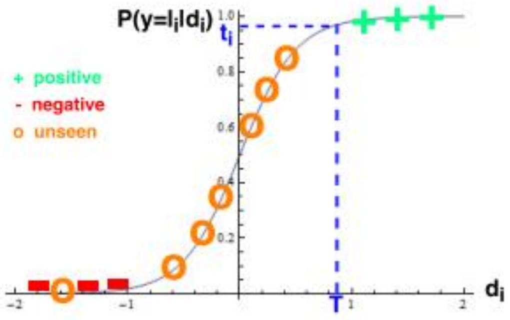


Figure 2: Open space risk of sigmoid function and desired decision boundary $d_i = T$ and probability threshold t_i.