Learning to Classify Open Intent via Soft Labeling and Manifold Mixup

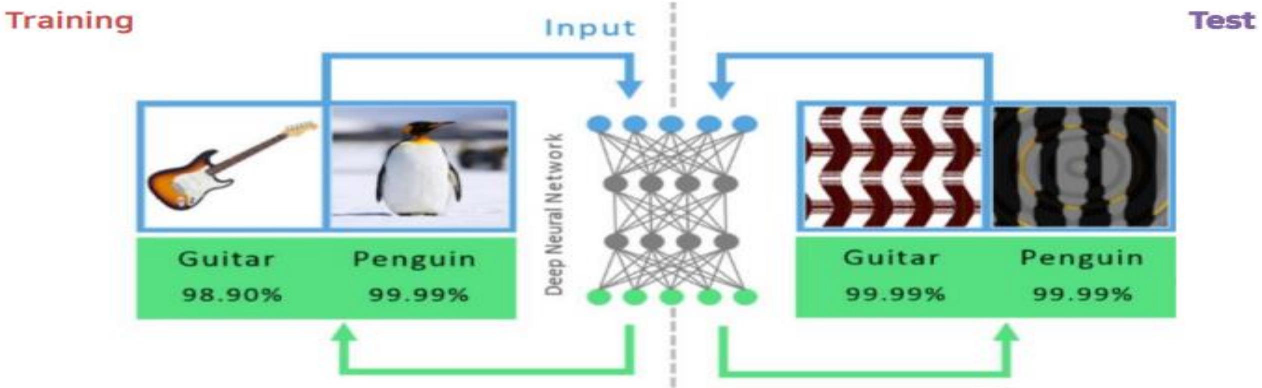
Zifeng Cheng , Zhiwei Jiang , Yafeng Yin , Member, IEEE, Cong Wang , and Qing Gu

EEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, VOL. 30, 2022

전북대학교 IT정보공학과
202018392
박나현

Open intent classification

- 대화시스템에서 아직 난제
  - 기존 심층 신경망에서는 학습한 class에 대한 확률만 출력(closed world 가정)
  - Training 동안 user의 모든 intent를 고려하는 것은 불가능
- **Objective**
  - Open intent 를 찾는 동시에 known intent 를 정확하게 분류하는 것



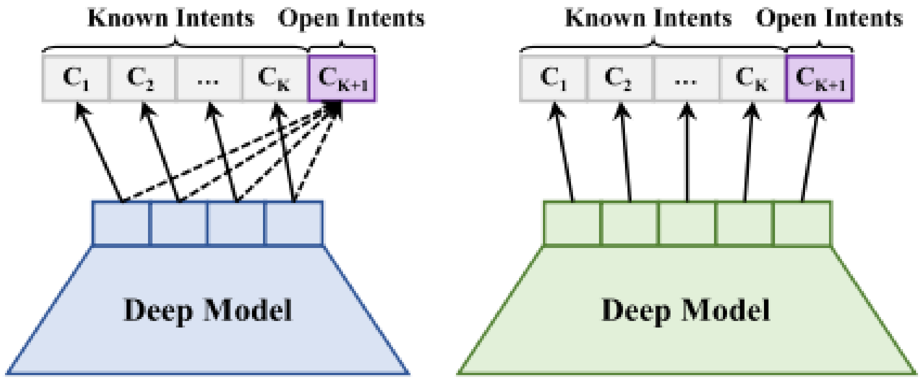
Existing method

- 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

Two main challenge

Two main challenge

- Known class의 boundary 보정 방법
  - Known class의 training sample의 label 분포를 재구성하여 known intent 에 대한 overconfident 줄일수 있을지
- Test 샘플을 k+1 번째 open intent class로 boundary 학습
  - Training sample 기반으로 decision boundary를 최적화하기 위해 open intent 위한 pseudo sample 생성할 수 있을지

제안한 모델에서 해결

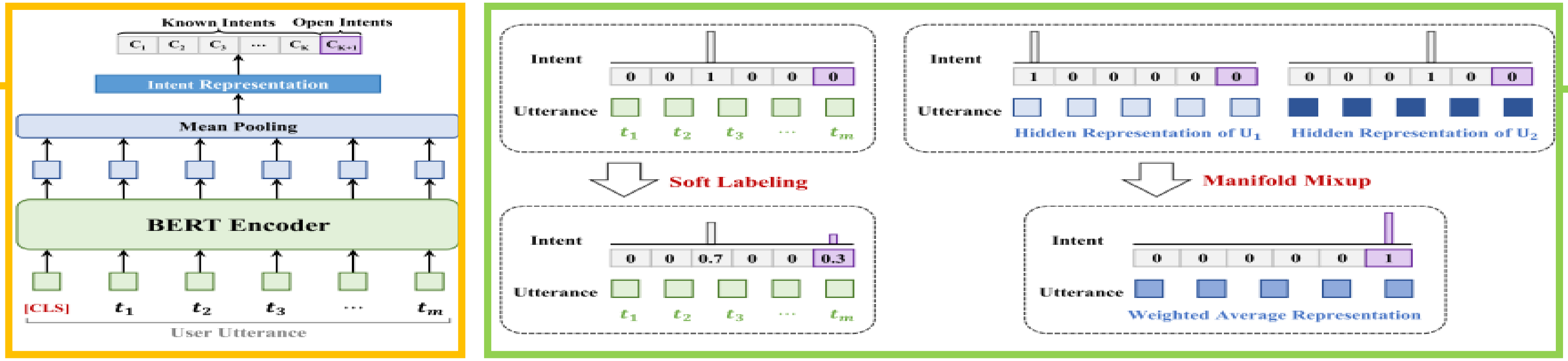
- **Soft Labeling (SL)**
  - 훈련 샘플 intent label 분포 재구성
  - Open intent로 예측될 확률을 줄 일 수 있어 알려진 의도에 대한 overconfident 줄일 수 있음
- **Manifold Mixup(MM)**
  - Open intent 에 대한 pseudo sample 생성

known intent 의 샘플과 open intent의 유사 샘플의 label-reshaped 가 학습되고 open intent classification에서 효과적으로 사용

Framework of the proposed method

- Outlier detection algorithm 사용하지 않고 open intent classification을 위한 (k+1)-class classification framework 제안
- Soft labeling과 manifold mixup 사용
  - Open intent에 대한 추가적인 데이터 없이 decision boundary 학습

Known intent를 위한 pretrainig



Open intent 위한 training -> pseudo data 생성

Fig. 3. Framework of our proposed method. The left part is model backbone in our proposed method. The middle part is soft labeling strategy and the right part is manifold mixup strategy.

Deep intent classification model

- BERT 사용하여 intent representation 추출
- BERT의 마지막 hidden layer에서 token embedding 얻음

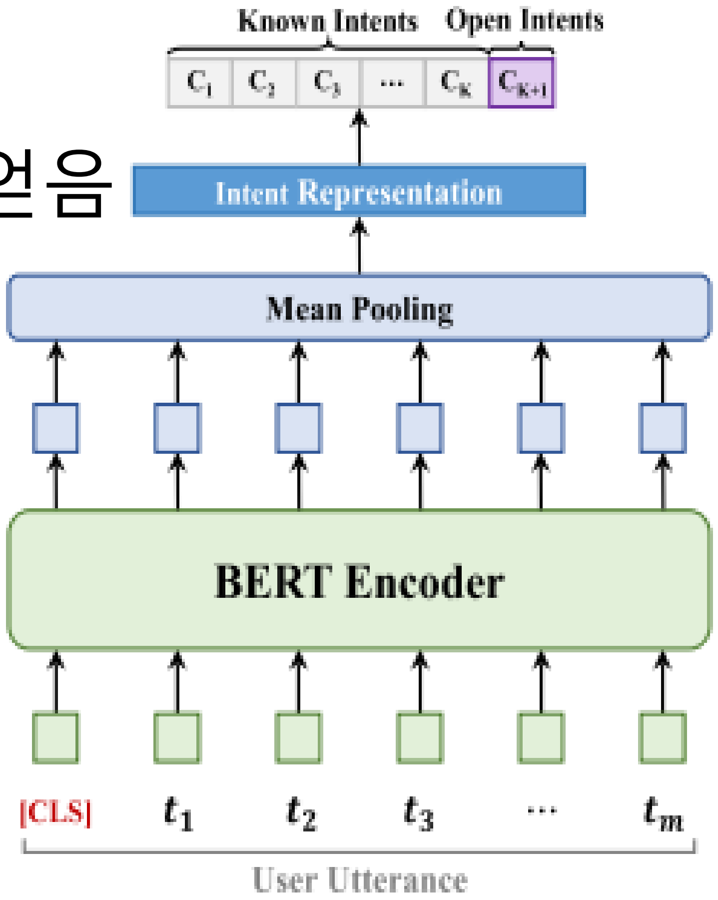
$$[CLS, T_1, \ldots, T_{m_i}] \in \mathbb{R}^{(m_i+1)\times H}$$

- **Mean pooling**
  - Get the average representation

$$\tilde{x}_i = \text{mean-pooling}([CLS, T_1, \ldots, T_{m_i}])$$

- **Dense layer**
  - Get the intent representation

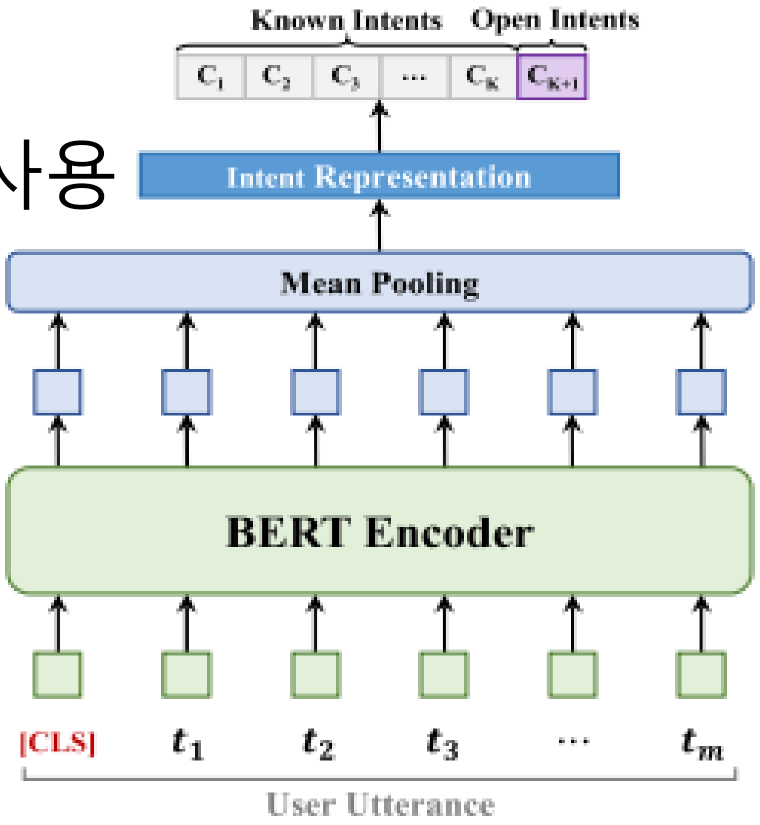
$$z_i = h(x_i) = \text{ReLU}(W_h x_i + b_h)$$



Pre-training for known intents

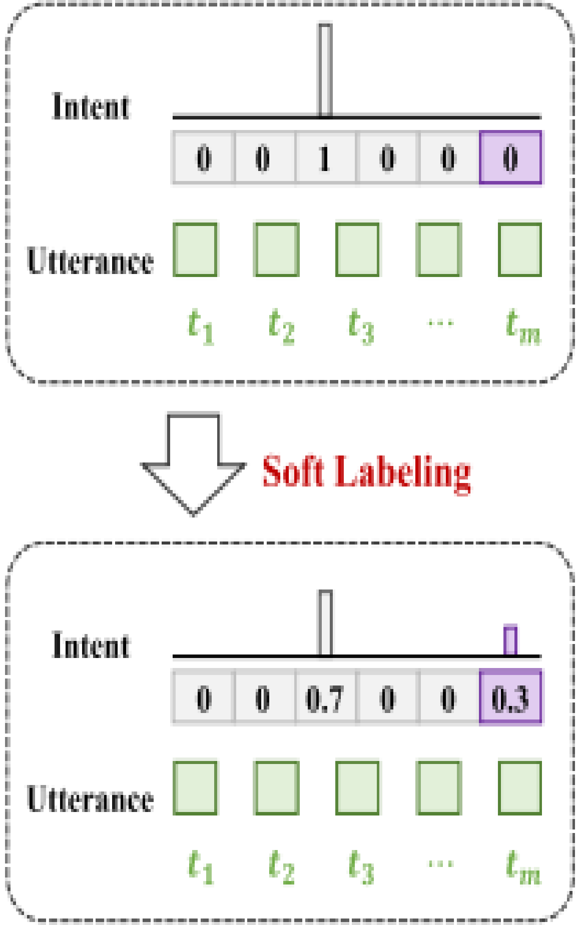
- 더 좋은 intent representation 을 얻기 위해 training set에서 known intent label data를 pretrain
- Intent feature z_i를 학습시키기 위해 softmax loss 사용

$$\mathcal{L}_P = -\frac{1}{L}\sum_{i=1}^{L}\log\frac{\exp(\phi_K(z_i)^{y_i})}{\sum_{j=1}^{K}\exp(\phi_K(z_i)^j)}$$

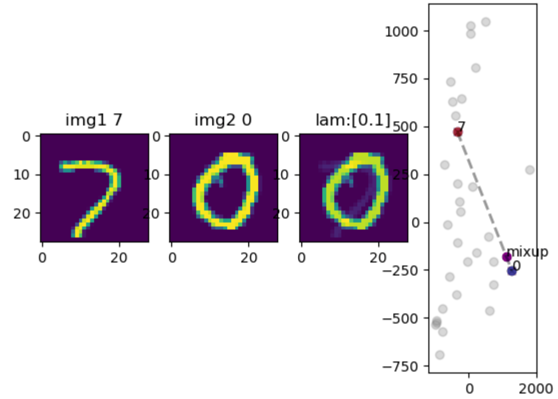


Training for open intent

- **Open intent data 부족**으로 open intent 구별 위한 model train에 어려움 존재
  - Soft labeling과 manifold mixup으로 pseudo data 생성하여 해결
- **Soft labeling**
  - Training set에서 label 분포를 reshape하므로서 known intent에 대한 pseudo data 생성
  - Default 값으로 open intent class로 재할당하므로서 label 분포 soften
    - 작은 확률을 set하여 ground-truth class 의 확률이 open class 확률보다 높도록 set
  - Kl-divergence loss로 학습 $D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right)$
    - P 분포와 Q 분포가 얼마나 다른 지를 측정하는 방법
    - (Cross-Entropy) – (Entropy) $\mathcal{L}_S = \sum_{i=1}^{L} D_{KL}\left(p(x_i) || q(x_i)\right)$



Training for open intent



- **Manifold mixup**
  - Generate open intent samples by interpolating between the representation of two samples of different known intents

sample pair (x_i, x_j): Random shuffling

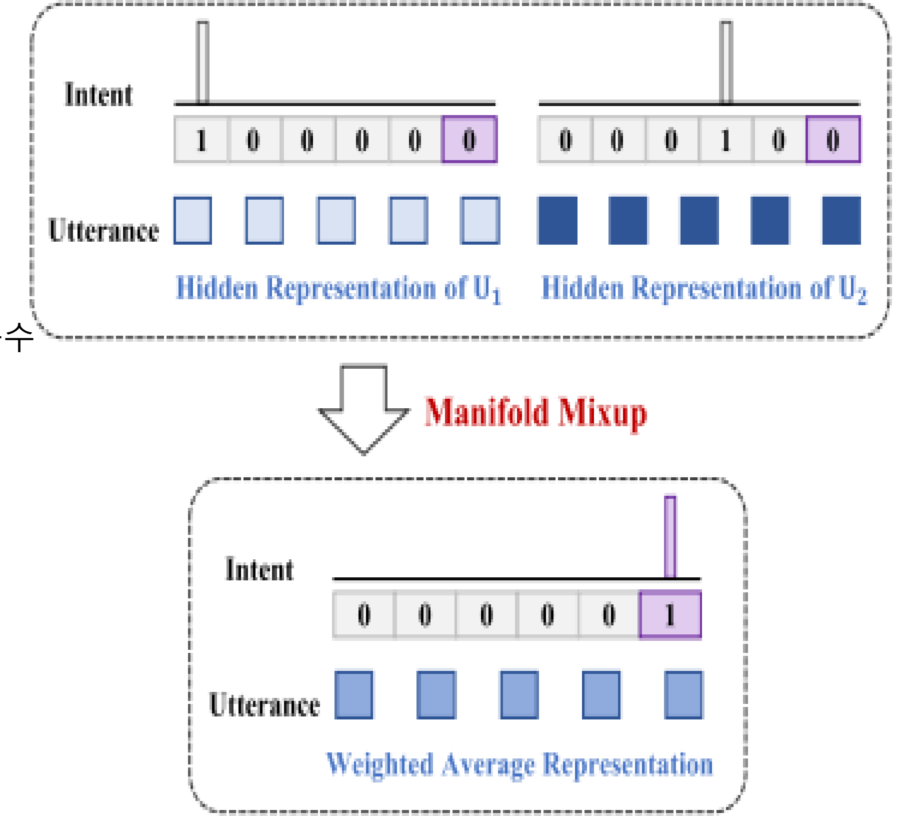
$h_i^l = \mathrm{BERT}(h_i^{l-1}), l \in [1, n]$ $\quad\quad$ $\hat{h}_m^n = \lambda h_i^n + (1-\lambda) h_j^n$ **mixup**

$h_j^l = \mathrm{BERT}(h_j^{l-1}), l \in [1, n]$ $\quad\quad$ $\hat{h}_m^l = \mathrm{BERT}(\hat{h}_m^{l-1}), l \in [n+1, T]$

BERT 전체 층수

Output: $\hat{h}_m^T$ $\quad$ $\hat{z}_m$ $\quad$ $\mathcal{L}_M = -\frac{1}{M}\sum_{m=1}^{M} \log \frac{\exp(\phi_{K+1}(\hat{z}_m)^{K+1})}{\sum_{k=1}^{K+1} \exp(\phi_{K+1}(\hat{z}_m)^k)}$

Meanpooling, dense layer



- **Overall training objective**
  - Soft labeling loss와 manifold mixup loss 합

$$\mathcal{L} = \mu \mathcal{L}_S + (1-\mu)\mathcal{L}_M$$

Experiments

TABLE I
STATISTICS OF DATASET

Dataset	Classes	#Training	#Validation	#Test	Vocabulary Size	Length(mean)
BAKING	77	9003	1000	3080	5028	11.91
CLINC	150	15000	3000	5700	8376	8.31
SNIPS	7	13084	700	700	11971	9.05
ATIS	18	4978	500	893	938	11.37

- Dataset
  - **Banking**
    - Contain 77 intents and 13,083 customer service queries in the banking domain
  - **OOS**
    - Contain 22,550 in-scope queries convering 150 intents and 1200 out-of-scope queries across 10 domains
  - **SNIPS**
    - Contain 7 intents across different domain
  - **ATIS**
    - Contain 18 intents in the airline travel domain

- **Experimental settings**
  - Dataset
    - Train, validation, test
  - **The number of known classes**
    - 25%, 50%, 75%
    - 나머지 class 는 open 으로 여기고 train 때는 제거
    - Test 때는 전체 label 사용
  - Model
    - BERT-Base
  - Soft labeling
    - Open intent class에서 default 확률: 0.3
  - Manifold mixup
    - Interpolate the hidden state before the last transformer layer of BERT
    - Beta distribution에서 알파값 = 2
  - Optimixer: AdamW
  - Learning rate: 2e-5
  - Batch size : 128

Results

TABLE II
PERFORMANCE OF OPEN INTENT CLASSIFICATION WITH DIFFERENT KNOWN CLASS RATIOS (25%, 50%, 75%) ON FOUR BENCHMARK DATASETS

	Methods	BANKING		CLINC		SNIPS		ATIS	
		Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1
25%	MSP	43.67	50.09	47.02	47.62	-	-	-	-
	DOC	56.99	58.03	74.97	66.37	-	-	-	-
	OpenMax	49.94	54.14	68.50	61.99	-	-	-	-
	LMLC	64.21	61.36	81.43	71.16	-	-	-	-
	ADB	78.85	71.62	87.59	77.19	67.41	69.89	63.90	53.21
	SLMM	**80.96±0.83**	**72.58±0.79**	**91.25±0.25**	**80.04±1.22**	**75.30±4.72**	**74.64±3.48**	**69.90±10.81**	**61.18±9.44**
50%	MSP	59.73	71.18	62.96	70.41	-	-	-	-
	DOC	64.81	73.12	77.16	78.26	-	-	-	-
	OpenMax	65.31	74.24	80.11	80.56	-	-	-	-
	LMLC	72.73	77.53	83.35	82.16	-	-	-	-
	ADB	78.86	80.90	86.54	85.05	80.36	82.96	81.27	63.24
	SLMM	**80.28±0.57**	**81.98±0.47**	**89.04±0.27**	**86.72±0.08**	**80.93±0.08**	**83.15±0.15**	**87.37±2.69**	**70.21±2.44**
75%	MSP	75.89	83.60	74.07	82.38	-	-	-	-
	DOC	76.77	83.34	78.73	83.59	-	-	-	-
	OpenMax	77.45	84.07	76.80	73.16	-	-	-	-
	LMLC	78.52	84.31	83.71	86.23	-	-	-	-
	ADB	81.08	85.96	86.32	88.53	82.56	85.13	87.53	71.60
	SLMM	**82.18±1.42**	**86.67±0.75**	**88.88±0.49**	**90.40±0.53**	**85.37±0.14**	**85.57±0.21**	**94.33±0.56**	**81.00±1.03**

"accuracy" and "F1" Denote the accuracy score and macro F1-score over all classes. Performance (mean±std) over 10 runs are reported. The best results are in bold. All results of baselines on BANKING and CLINC from Zhang *et al.* [10]. The results of ADB on SNIPS and ATIS are reproduced from open-source code.

Results

TABLE III
PERFORMANCE OF OPEN INTENT CLASSIFICATION WITH DIFFERENT KNOWN CLASS RATIOS (25%, 50%, 75%) ON FOUR BENCHMARK DATASETS

	Methods	BANKING		CLINC		SNIPS		ATIS	
		Open	Known	Open	Known	Open	Known	Open	Known
25%	MSP	41.43	50.55	50.88	47.53	-	-	-	-
	DOC	61.42	57.85	81.98	65.96	-	-	-	-
	OpenMax	51.32	54.28	75.76	61.62	-	-	-	-
	LMLC	70.44	60.88	87.33	70.73	-	-	-	-
	ADB	84.56	70.94	91.84	76.80	70.99	69.34	64.93	50.33
	SLMM	**86.53±0.51**	**71.85±0.84**	**94.53±0.62**	**79.66±2.81**	**79.68±4.67**	**72.13±1.51**	**67.14±11.79**	**59.84±7.42**
50%	MSP	41.19	71.97	57.62	70.58	-	-	-	-
	DOC	55.14	73.59	79.00	78.25	-	-	-	-
	OpenMax	54.33	74.76	81.89	80.54	-	-	-	-
	LMLC	69.53	77.74	85.85	82.11	-	-	-	-
	ADB	78.44	80.96	88.65	85.00	74.92	84.97	77.27	61.40
	SLMM	**80.14±0.40**	**82.03±0.56**	**91.07±0.05**	**86.67±1.11**	**75.07±2.76**	**85.17±0.08**	**85.66±3.41**	**68.05±1.79**
75%	MSP	39.23	84.36	59.08	82.59	-	-	-	-
	DOC	50.60	83.91	72.87	83.69	-	-	-	-
	OpenMax	50.85	84.64	76.35	73.13	-	-	-	-
	LMLC	58.54	84.75	81.15	86.27	-	-	-	-
	ADB	66.47	86.29	83.92	88.58	69.30	88.29	71.08	71.72
	SLMM	**69.20±0.50**	**86.97±1.08**	**87.15±1.07**	**90.43±0.32**	**69.72±2.68**	**89.00±0.15**	**79.84±2.68**	**81.08±0.92**

"open" and "known" denote the macro F1-score over open class and known classes respectively. APerformance (mean±std) of our method over 10 runs are reported. The best results are in bold. All results of baselines on BANKING and CLINC from Zhang [10]. The results of ADB on SNIPS and ATIS are reproduced from open-source code.

Ablation Study

TABLE IV
ABLATION STUDY OF OPEN INTENT CLASSIFICATION WITH DIFFERENT KNOWN CLASS RATIOS (25%, 50%, 75%) ON CLINC DATASET

	Setting	CLINC			
		Accuracy	**F1**	**Weighed-F1**	**Acc-KoK**
25%	SLMM	**91.25**	80.04	**90.97**	**97.81**
	w/o pre-training	84.01	69.30	88.94	95.18
	w/o SL	84.78	74.82	81.55	95.35
	w/o MM	89.84	**80.26**	90.36	97.54
	w/o SL&MM	19.78	37.31	8.18	97.11
50%	SLMM	**89.04**	86.72	**89.45**	96.93
	w/o pre-training	81.22	76.48	87.86	95.47
	w/o SL	63.84	71.44	72.76	90.98
	w/o MM	88.35	**86.74**	89.00	**97.11**
	w/o SL&MM	39.95	59.92	26.09	96.84
75%	SLMM	**88.88**	**90.40**	**88.70**	**96.40**
	w/o pre-training	81.21	81.35	88.07	95.83
	w/o SL	72.14	81.22	81.01	91.16
	w/o MM	88.25	90.14	88.46	96.25
	w/o SL&MM	59.25	75.05	46.98	96.01

F1 denotes macro-F1 over all classes. Acc-KoK denotes the accuracy of known intents classifier on samples of known intents. Averaged results over 10 runs are reported. The best results are in bold.

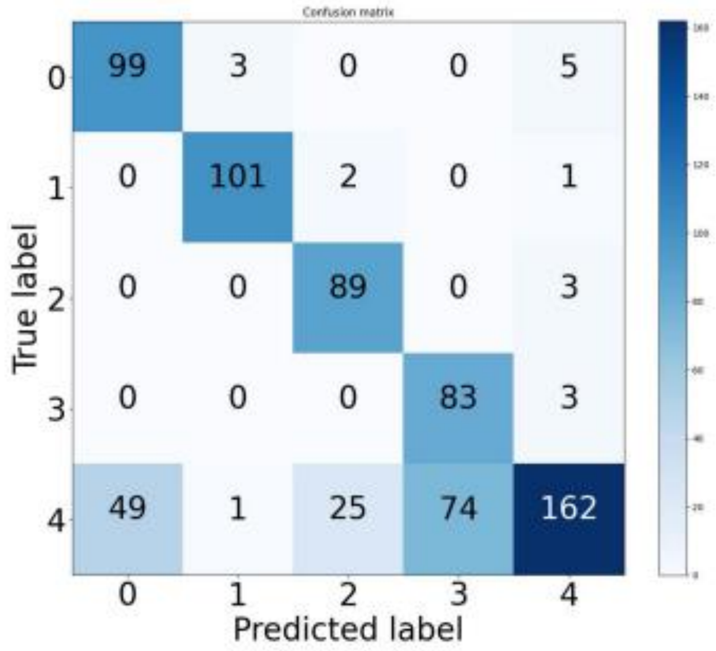
Error Analysis

TABLE VII
TWO EXAMPLES FOR ERROR ANALYSIS

Utterance	**Ground-Truth Intent**	**Predicted Intent**
Add this song to blues roots.	Add To Play List ***Open intents***	Play Music ***Known intents***
Find a movie called living in america.	Search Creative Work ***Open intents***	Search Screening Event ***Known intents***

- 두 종류 오류
  - Open class 관련 오류가 많은 오류를 차지하고 있으며 많은 open class 샘플이 인식되지 않은 것을 알 수 있음
  - There are five samples belonging to known classes error.

- 추가
  - 일부 open intent가 known intent 와 유사하다는 것을 발견



Conclusion

- Soft labeling과 manifold mixup을 기반으로하는 deep open intent classification model 제안
  - Soft labeling
    - 각 sample 이 open intent 로 예측될 확률을 제공
  - Manifold mixup
    - 두 개의 서로 다른 알려진 의도 샘플의 숨겨진 표현 사이의 interpolating을 통해 open intent sample을 생성
- outlier detection algorithms 없이 (k+1)-class classification을 수행

Algorithm 1: Training Flow of SLMM.

Input: The training set
Output: A open intent classification model.

 # Pre-training for Known Intents:
for all iteration $= 1, \cdots,$ MaxIter **do**
  Sample a mini-batch $\{(x_i, y_i)\}$
  Calculate the training loss by (3)
  Obtain derivative and update the model
end for
 # Training for Open Intents:
for all iteration $= 1, \cdots,$ MaxIter **do**
  Sample a mini-batch $\{(x_i, y_i)\}$
  Reshape label distribution via soft labeling
  Calculate soft labeling loss by (4)
  Generate pseudo data via manifold mixup
  Calculate manifold mixup loss by (8)
  Calculate total loss by (9)
  Obtain derivative and update the model
end for
returnThe model converges on validation set.

Thank you

Soft labeling and manifold mixup(slmm)

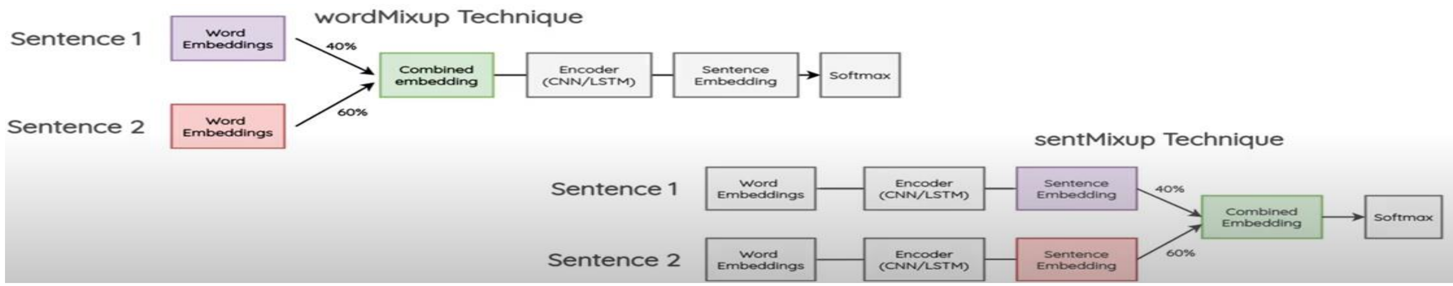
Soft labeling(sl)

- 각 샘플이 공개 의도로 예측될 확률을 줄 수 있으므로 known intent 에 대한 confidence 감소 가능

Manifold mixup(mm)

- Mixup
  - 데이터를 두개씩 쌍을 지어 mixup 한 후 그 데이터를 새로운 학습 데이터로 사용하는 것
- Open intent에 대한 유사샘플을 생성

Manifold mixup



- Mixup
  - Data agumentation method
- Manifold mixup
  - 음냐
- Interpolation
  - 데이터와 데이터 사이의 빈공간을 채워넣는것
- 데이터 증강, 모델 일반화를 위해 Manifold mixup을 사용함으로써 이전 연구들과 차이점을 두고 manifold mixup을 open intent classification task를 위해 선택

Model analysis

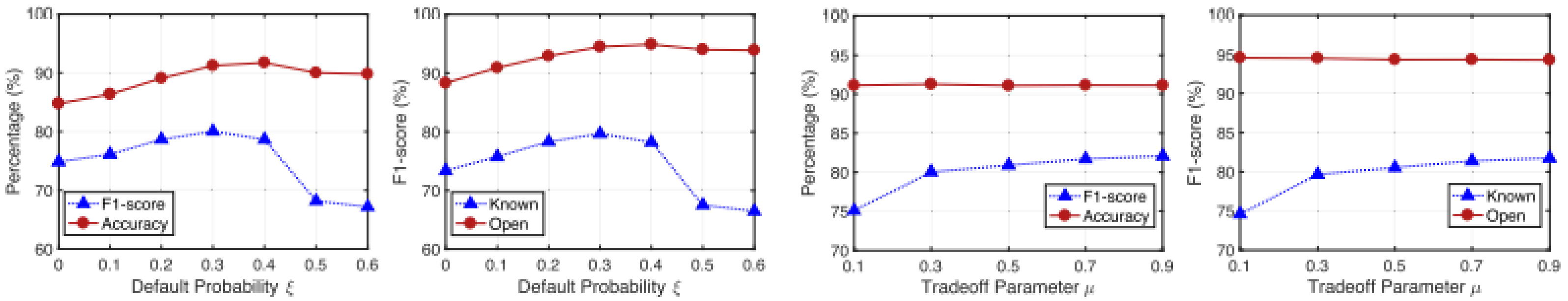
TABLE V
EFFECT OF α OF BETA DISTRIBUTION IN MANIFOLD MIXUP

α	**0.5**	**1**	**2**	**4**
Accuracy	81.50	90.22	91.25	**91.85**
F1	15.08	69.78	**80.04**	79.79

TABLE VI
EFFECT OF INTERPOLATION POSITION n IN MANIFOLD MIXUP

n	**9**	**10**	**11**
Accuracy	87.03	86.85	**91.25**
F1	75.30	75.18	**80.04**

Model analysis



(a) Effects of the default probability ξ on open intent class.

(b) Effects of the tradeoff parameter μ in loss function.

Fig. 4. Effects of ξ and μ on CLINC dataset with 25% known classes. Left part is the effect of ξ and the right part is the effect of μ.