

데이터 중요도를 반영한 적응 결정경계 기반 오픈 의도 분류 (Open Intent Classification with Data Weighted Adaptive Decision Boundary)

박나현, 송현제
전북대학교 IT 정보공학과

오픈 의도 분류

- 기존 의도 분류에서 분류 시 달한 세계 가정
- 오픈 의도 분류
 - 의도 분류에서 발화 텍스트가 미리 정의된 의도 클래스들에 속하지 않았을 때, 'open' 으로 분류하는 문제
- 오픈 의도 분류 목표
 - open intent 를 찾는 동시에 known intent 정확히 분류
 - k-class known intent 만을 이용하여 (k+1)번째 class 인 open intent 를 식별

□ 오픈 의도 분류 예시

User utterances	Intent Label
Book a flight from LA to Madrid.	Booking flight
Can you get me a table at Steve's?	Restaurant reservation
Book Delta ticket Madison to Atlanta.	Booking flight
Schedule me a table at Red Lobster.	Restaurant reservation
.....
What time is it now?	Asking time (Open)
Where is the nearest school?	Asking place (Open)

K개의 클래스로 분류

K+1번째 새로운 class = open intent

적응 결정경계 기반 오픈 의도 분류

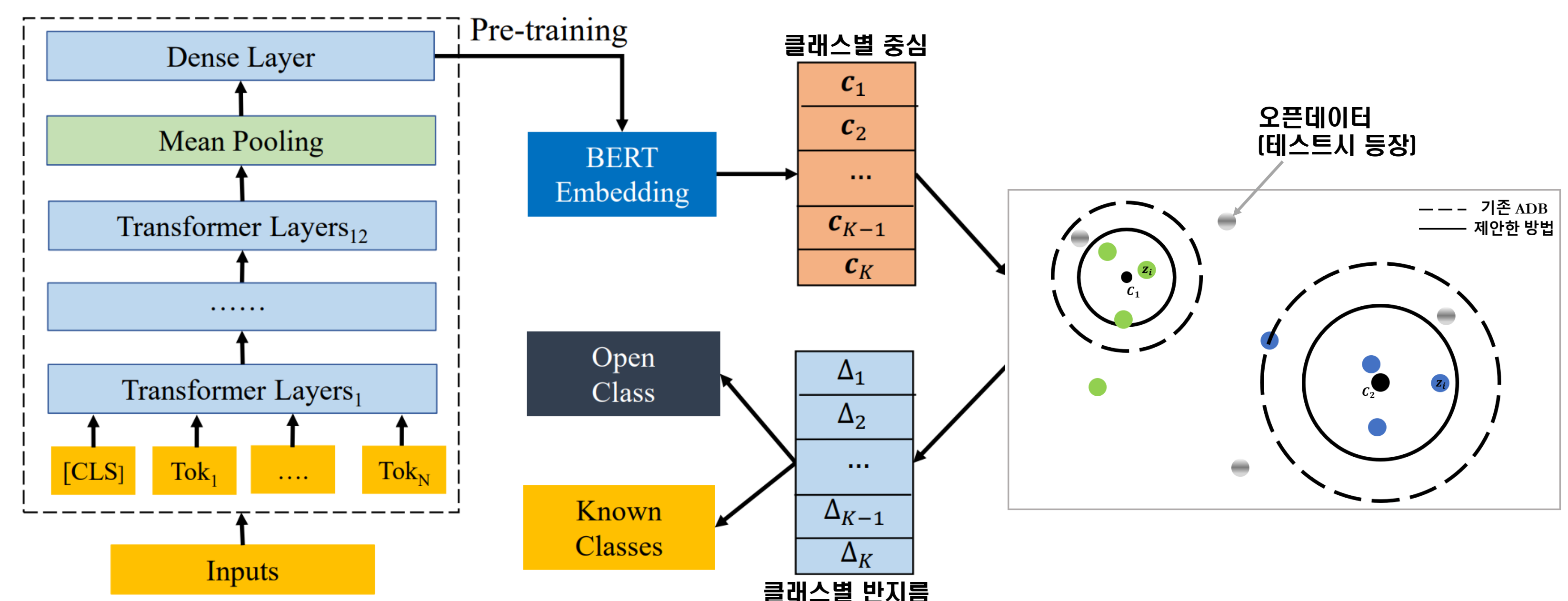
- 사전학습 언어모델을 통해 문장별 텍스트를 의도자질로 표현
- 클래스별 의도자질들의 평균벡터를 중심으로 간주하고
반경으로 클래스별 결정경계를 표현
- 결정경계의 조건에 부합하는 손실함수 사용하여 결정경계 학습
 - 경험적 리스크와 open space risk 동시에 최소화하도록 학습
 - 결정경계의 조건
 - 알려진 의도를 최대한 둘러싸기에 충분한 넓이 (경험적 리스크)
 - 오픈을 알려진 의도로 식별하지 않도록 촘촘한 넓이 (open space risk)
- 결정경계 조건에 맞는 결정경계 학습을 위한 손실함수

$$\mathcal{L}_b = \frac{1}{N} \sum_{i=1}^N [\delta_i (\|z_i - c_{y_i}\|_2 - \Delta_{y_i}) + (1 - \delta_i) (\Delta_{y_i} - \|z_i - c_{y_i}\|_2)]$$

클래스별 중심
문장별 의도자질
클래스별 반지름

$$\delta_i = \begin{cases} 1, & \text{if } \|z_i - c_{y_i}\|_2 > \Delta_{y_i} \\ 0, & \text{if } \|z_i - c_{y_i}\|_2 \leq \Delta_{y_i} \end{cases}$$

- 분류
 - 새로운 데이터가 미리 정의된 모든 의도 클래스의 결정경계 밖에 존재하면 그 데이터를 'open' 으로 분류하고 그렇지 않은 경우 클래스의 중심과 가장 가까운 클래스로 분류
- 문제점
 - 모든 데이터의 중요도가 같다고 간주하기 때문에 반경이 큰 결정경계가 학습
 - 다수의 오픈 텍스트들이 미리 정의된 의도 클래스로 분류



데이터 중요도를 반영한 적응 결정경계

- 데이터 중요도
 - 중심에 가까운 데이터일수록 클래스를 대표하는 데이터
 - 중심에서 먼 데이터는 클래스를 대표하지 않을 확률이 높고 덜 중요하다고 여길 수 있는 데이터
- 문제 해결을 위해 중요도를 추가한 손실함수 제안
 - 각 데이터의 중요도를 차등 반영하여 적응 결정경계 구축
 - 중요도
 - 의도 자질과 해당 클래스 결정경계의 중심 간 유클리디언 거리의 역수로 정의

$$\mathcal{L}_{wb} = \frac{1}{N} \sum_{i=1}^N [w_i (\delta_i (\|z_i - c_{y_i}\|_2 - \Delta_{y_i}) + (1 - \delta_i) (\Delta_{y_i} - \|z_i - c_{y_i}\|_2))]$$

중요도

$$w_i = \frac{k}{\|z_i - c_{y_i}\|_2}$$

실험

- 오픈 의도 분류 실험
 - 데이터 전체의 accuracy, macro F1-score(F1-score) 와 각 데이터의 오픈과 이미 알고 있는 데이터 각각 macro F1-score (F1-score)
 - 데이터셋: BANKING, OOS, StackOverflow

표 1. 미리 정의된 의도 클래스 비율을 25%, 50%, 75%로 설정하여 학습한 후 'open' 분류에 대한 정확도 및 F1-score.

		BANKING		OOS		StackOverflow	
	Methods	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
25%	ADB	76.43	70.9857	88.18	77.1444	89.73	83.2968
	Proposed Method	77.92	71.6319	88.63	77.6504	90.10	83.5063
50%	ADB	79.38	81.0024	86.72	85.1147	88.97	87.9562
	Proposed Method	80.10	81.2485	87.26	85.4592	88.90	87.8335
75%	ADB	81.98	86.8214	86.61	88.6013	84.80	87.7931
	Proposed Method	82.14	86.8577	86.74	88.5513	84.58	87.5654

표 2. 미리 정의된 의도 클래스 비율을 25%, 50%, 75%로 설정하여 학습한 후, 'open' 및 정의된 클래스 분류에 대한 성능.

		BANKING		OOS		StackOverflow	
	Methods	Open	Known	Open	Known	Open	Known
25%	ADB	82.9256	70.3572	92.3680	76.7438	93.0754	81.3411
	Proposed Method	84.2464	70.9680	92.6980	77.2544	93.3633	81.5349
50%	ADB	79.4754	81.0426	88.8450	85.0649	89.9379	87.7580
	Proposed Method	80.5023	81.2682	89.3951	85.4067	89.8918	87.6277
75%	ADB	67.7220	87.1507	84.4884	88.6381	76.3578	88.5555
	Proposed Method	68.4246	87.1755	84.7258	88.5854	76.2582	88.3192

- 기존 방법에 비해 대부분의 실험에서 성능 향상
 - 오픈 의도 분류 목표에 맞는 오픈 뿐만 아니라 미리 정의된 클래스에 대해서도 잘 분류
- 모든 데이터셋에 대해 25% 비율에서 성능 향상
- StackOverflow 의 50% 이상에서 성능이 하락
 - StackOverflow 의 경우 전체 클래스 개수는 적고 학습 데이터 수가 많아 오픈 의도 분류 실험에서 중요도가 큰 비중을 차지 하지 않음