By
MEDIAZEN

On
2024.01.26

Personalization for BERT-based Discriminative Speech Recognition Rescoring

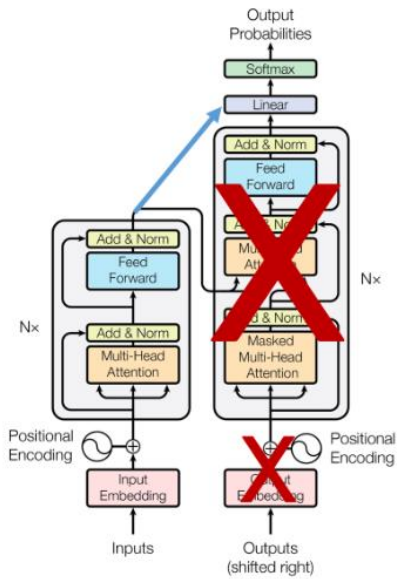
Amazon Alexa(Jari et al)

AI 모델연구팀 황수림

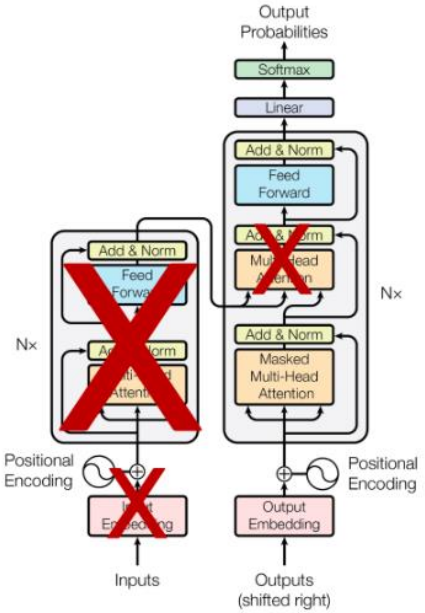
I. RescoreBERT

(1) BERT VS GPT, RescoreBERT

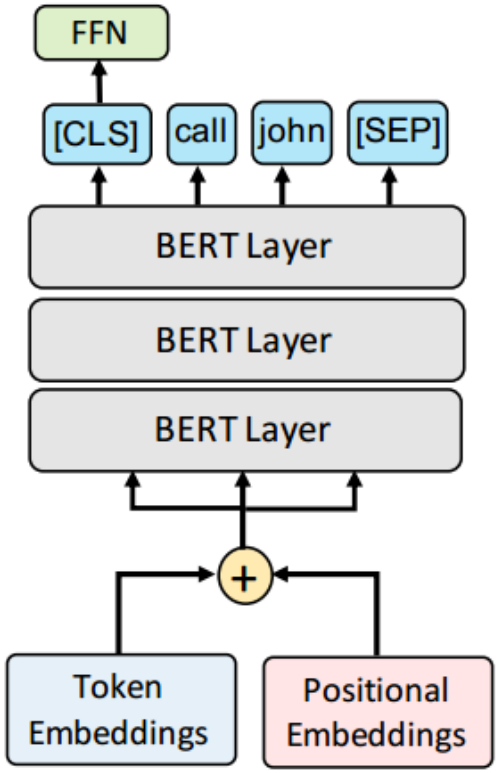
$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_\mathbf{K}}})\mathbf{V}$$



BERT
(Bidirectional Encoder Representations from **Transformer,** masked LM)



GPT
(Generative Pretrained **Transformer**, unidirectional)



RescoreBERT

Transformer

- Encoder: 소스 시퀀스 압축 및 전송
- Decoder: 타깃 시퀀스 생성
- Self Attention: Q, K, V 로 문맥적 관계성 학습

  -> Multi-head attention

Rescoring

- 1st pass in ASR: 음성 정보로 예측값 리스트 생성
- 2nd : 가능한 표기 확인하여 재집계
- 최종 점수: 20*1st + 2nd $v_i = \alpha u_i + \beta s_i$

RescoreBERT

- Bidirectional Transformer Rescoring Model
- ASR loss로 학습하여 WER 최소화 위해 예측값으로부터 단일 점수 예측
- FFN + CLS token

II. ER in ASR

(1)CSID, Edit Distance, WER, CER, PER, WERR

- **CSID: Correct(일치), Substitution(대체), Insertion(삽입), Deletion(삭제)**
- **ER = Edit Distance / N(정답지 단어 수) = min(S + D + I) / (S + D + C)**

  = {WER(Word), CER(Character), PER(Phoneme), TER(Token), SER(Sentence)…}

  *** WERR: WER Reduction (+ : WER ↑ ←→ - : WER ↓ (성능↑))**

Model	Personalized	General
Oracle	-57%	-58%
Tiny RescoreBERT	+3.9%	-5.3%
Big RescoreBERT	+4.8%	-7.1%
Tiny RescoreBERT (fine-tuned)	+2.5%	-5.3%
Big RescoreBERT (fine-tuned)	+1.7%	-6.8%

MEDIAZEN

II. ER in ASR

(2) CSID(Levenshtein) Algorithm, (1) snow VS sunny

- 방향

(1) ↓ +1 : Deletion
(2) → +1 : Insertion
(3) ↘ +0 : Correct
(3) ↘ +1 : Substitution

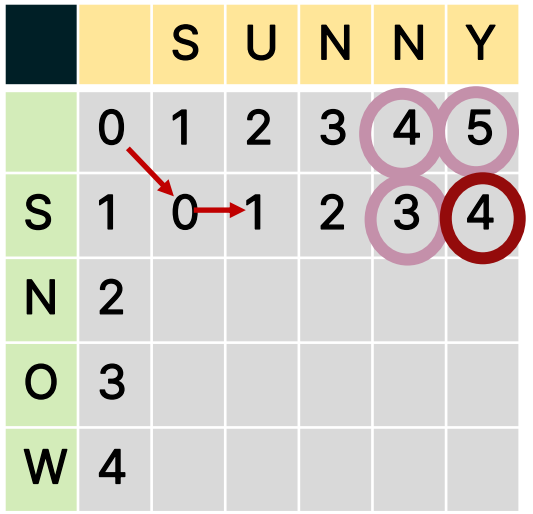
편집 거리 계산

- 알고리즘 일부

```
arr[i][j] = min(
        arr[i-1][j]+1, # D(1) ↓
        arr[i][j 1]+1, # I(2) →
        arr[i-1][j-1] + cost)
        # S(cost=1)|C(0)(3) ↘
```

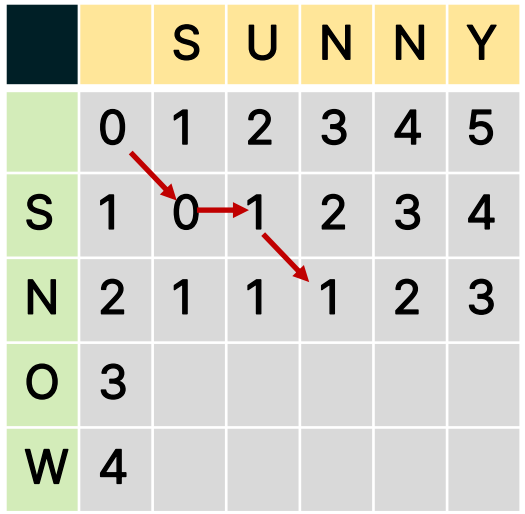
		S	U	N	N	Y
	0	1	2	3	4	5
S	1					
N	2					
O	3					
W	4					

(2,2): min(1+1(D), 1+1(I), 0+0(C)) =0

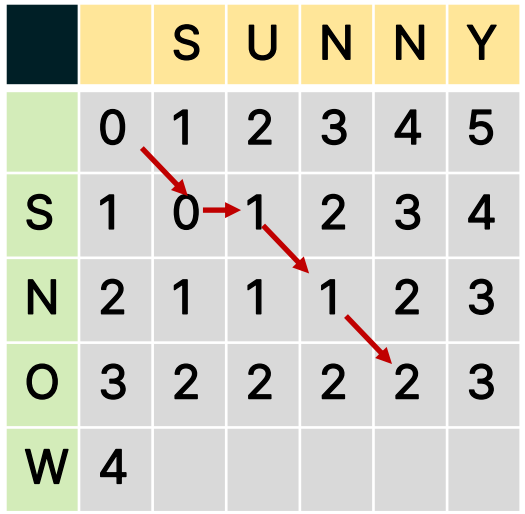


C I

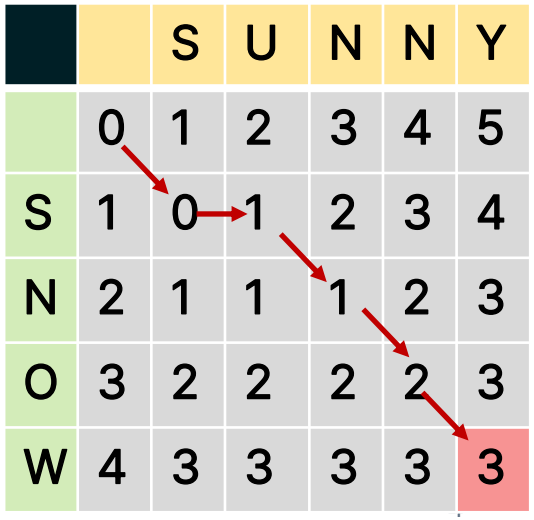
(2,6): min(5+1(D), 3+1(I), 4+1(S)) =4



C I C



C I C S

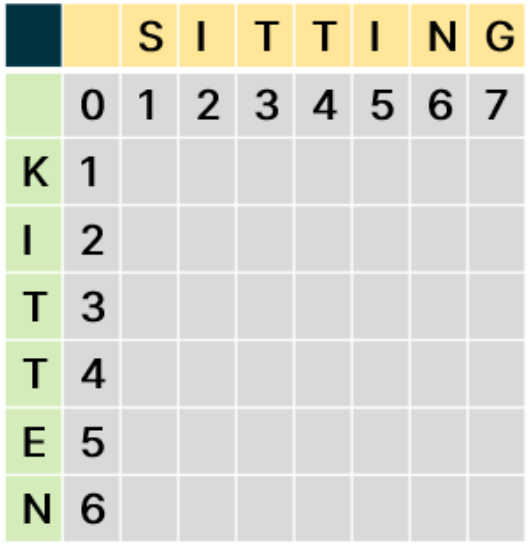


CICSS → 3(not C)

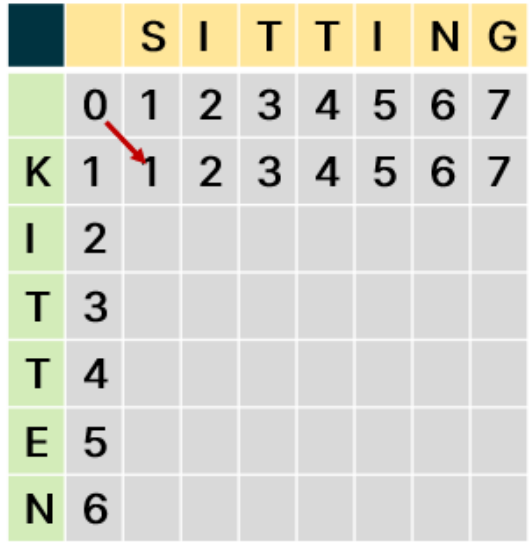
MEDIAZEN

II. ER in ASR

(2) kitten VS sitting

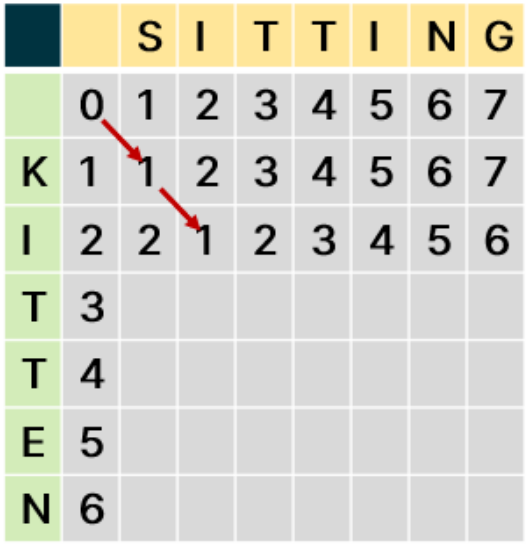


		S	I	T	T	I	N	G
	0	1	2	3	4	5	6	7
K	1							
I	2							
T	3							
T	4							
E	5							
N	6							



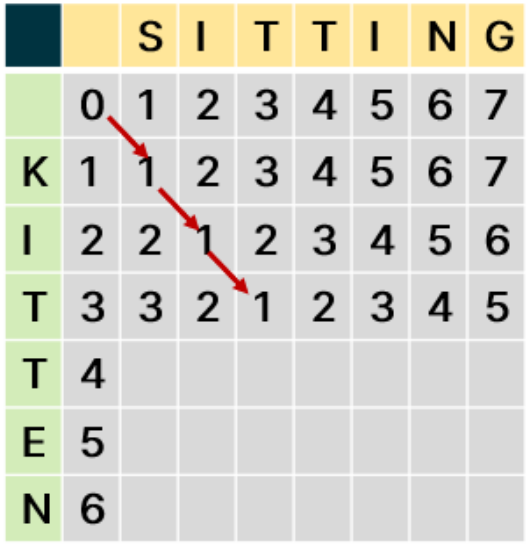
		S	I	T	T	I	N	G
	0	1	2	3	4	5	6	7
K	1	1	2	3	4	5	6	7
I	2							
T	3							
T	4							
E	5							
N	6							

S



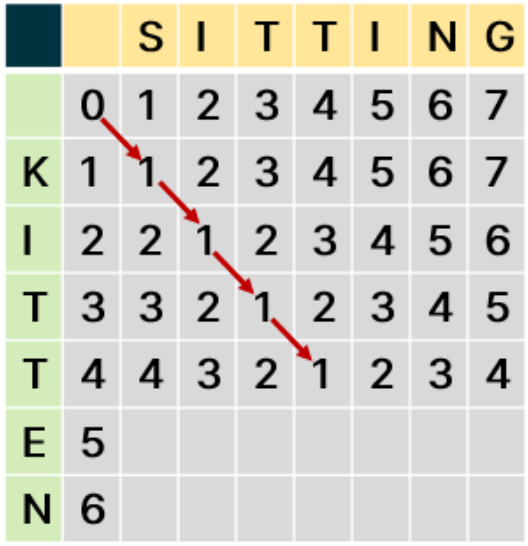
		S	I	T	T	I	N	G
	0	1	2	3	4	5	6	7
K	1	1	2	3	4	5	6	7
I	2	2	1	2	3	4	5	6
T	3							
T	4							
E	5							
N	6							

S C



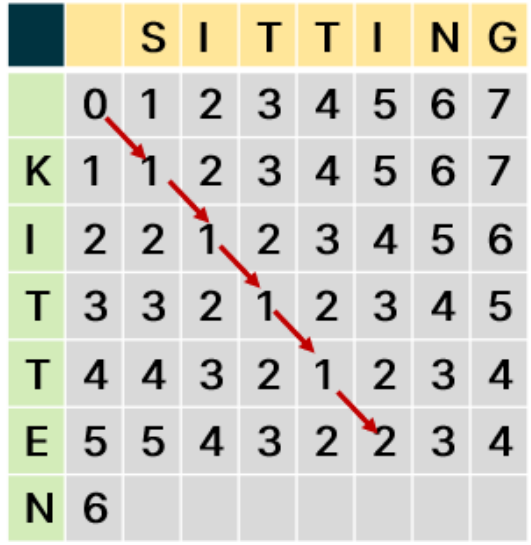
		S	I	T	T	I	N	G
	0	1	2	3	4	5	6	7
K	1	1	2	3	4	5	6	7
I	2	2	1	2	3	4	5	6
T	3	3	2	1	2	3	4	5
T	4							
E	5							
N	6							

S C C



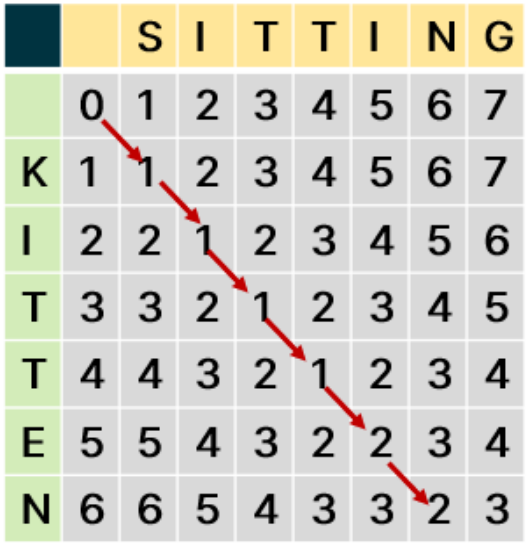
		S	I	T	T	I	N	G
	0	1	2	3	4	5	6	7
K	1	1	2	3	4	5	6	7
I	2	2	1	2	3	4	5	6
T	3	3	2	1	2	3	4	5
T	4	4	3	2	1	2	3	4
E	5							
N	6							

S C C C



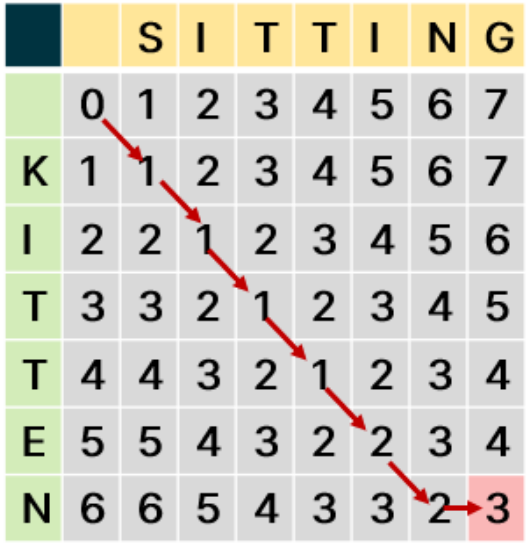
		S	I	T	T	I	N	G
	0	1	2	3	4	5	6	7
K	1	1	2	3	4	5	6	7
I	2	2	1	2	3	4	5	6
T	3	3	2	1	2	3	4	5
T	4	4	3	2	1	2	3	4
E	5	5	4	3	2	2	3	4
N	6							

S C C C S



		S	I	T	T	I	N	G
	0	1	2	3	4	5	6	7
K	1	1	2	3	4	5	6	7
I	2	2	1	2	3	4	5	6
T	3	3	2	1	2	3	4	5
T	4	4	3	2	1	2	3	4
E	5	5	4	3	2	2	3	4
N	6	6	5	4	3	3	2	3

S C C C S C

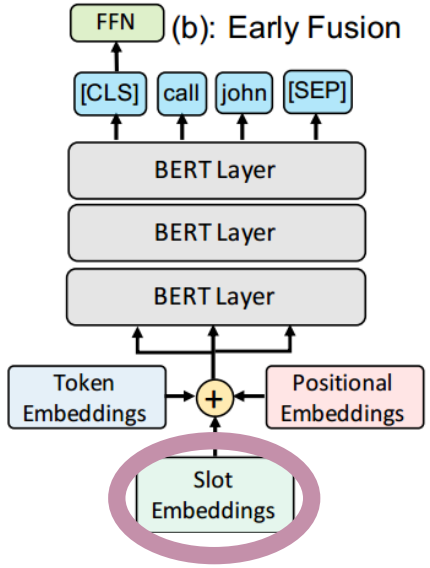


		S	I	T	T	I	N	G
	0	1	2	3	4	5	6	7
K	1	1	2	3	4	5	6	7
I	2	2	1	2	3	4	5	6
T	3	3	2	1	2	3	4	5
T	4	4	3	2	1	2	3	4
E	5	5	4	3	2	2	3	4
N	6	6	5	4	3	3	2	3

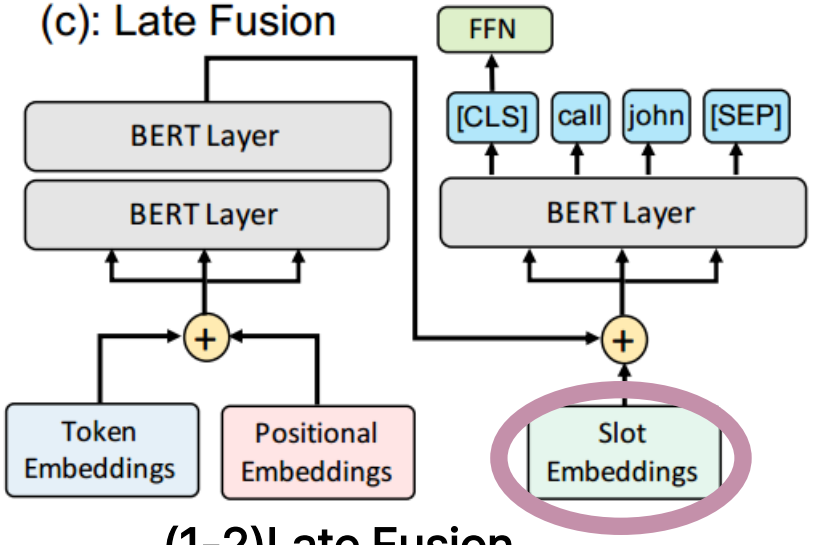
S C C C S C I → 3(not C)

III. Personalized Approaches

(1) Gazetteers 적용 2가지 방식; early fusion & late fusion



(1-1)Early Fusion



(1-2)Late Fusion

- Original: $s_i = f \circ g(E_t(x_i) + E_p).$

  ➔ f(FFN), g(BERT layer), x_i = hypothesis token, $E_t(x_i)$=TE, E_p=PE

* Ex of slot: 영화 제목, 노래 제목

- Gazetteers: 토큰 임베딩 + 위치 임베딩 + 슬롯 임베딩
  - Early: $s_i = f \circ g(E_t(x_i) + E_p + E_s(y_i)),$
  - Late : $s_i = f \circ g_n\left(E_s + g_{n-1} \circ \cdots \circ g_1(E_t + E_p)\right).$

III. Personalized Approaches

(2) Natural Language prompting

(2) Natural Language Prompting

Match Condition: 문장 일부 ⊂ 집합 D{ = 토큰화된 문자열, ...}

Ex) I want you to call Felix(call [entity])

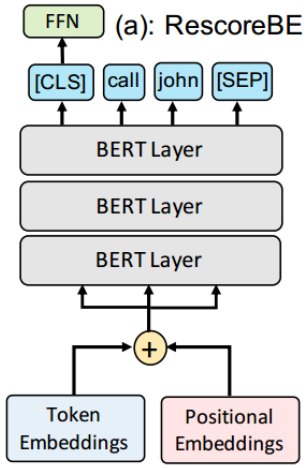
Prompt: 원문+ 구(증강 프롬프트)

Ex) I want you call to Felix as I need to contact Felix

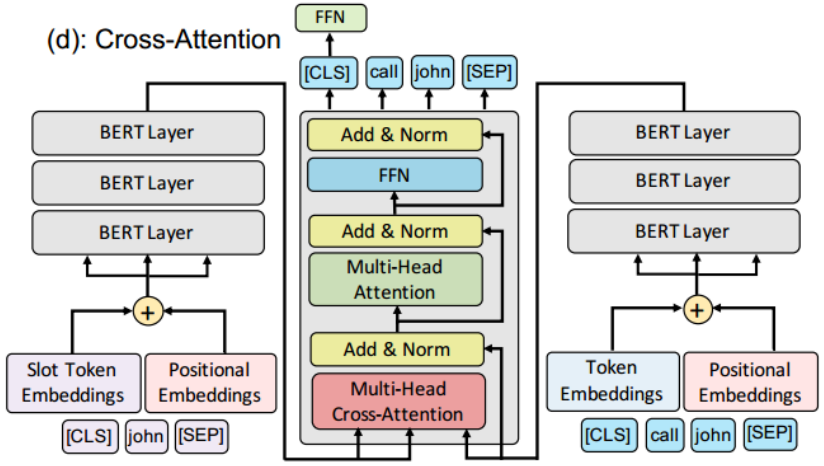
(as I need to contact [Entity])

III. Personalized Approaches

(3)Cross Attention based encoder-decoder model



(0)RescoreBERT(baseline)



(3)Cross Attention based encoder-decoder model

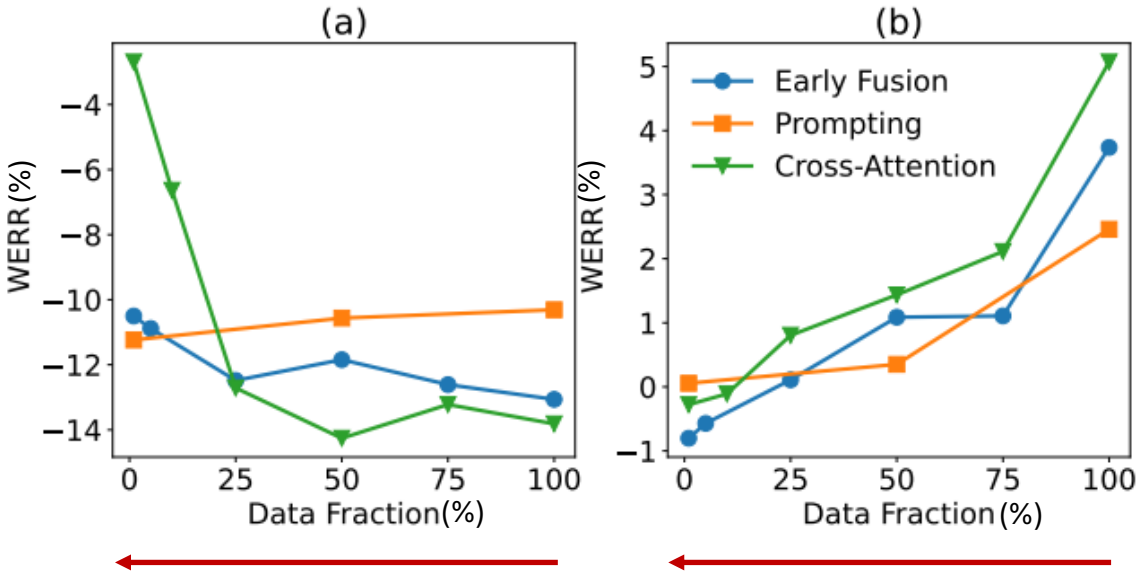
(3) Cross Attention based encoder-decoder model

z_i: 슬롯 토큰 시퀀스

ex) [CLS] [entity #1] [SEP] [entity #2] [SEP] ··· [SEP].

$$X_i = g(E_t(x_i) + E_p)$$
$$Z_i = g(E_t(z_i) + E_p)$$
$$s_i = f \circ m(Z_i, X_i),$$

IV. Result of Experiments



Explanation

- 데이터: 개인화된 개체명 포함하는 Alexa 데이터
- 목적: 개인화 내용에서의 WER ↓
- X축: 개인화된 개체 데이터 비율
- 실험진행방향: 일반화 데이터 추가되는 방향(반대)
- L/R: 개인화 / 일반화 테스트 집합

(1) NL Prompting

- (2) & (3) 성능의 중간
- 無학습 WERR -7% & 일반화데이터 성능 매우 약간 ↓
  (학습 데이터 ↓ 시 유용)

(2) Cross Attention

- 데이터 비율에 민감(L 왼쪽방향 급격한 상승)
- 일반화에서 성능 ↓

(3) Gazetteers(Early Fusion)

- L: <= -10 % WERR
- R: WER 유일 감소(< 0)

MEDIAZEN

Thank you

surim31 @mediazen.co.kr

2024.01.26