DyLex를 이용한 한국어 개체명 인식

DyLex for Korean Named Entity Recognition

강대욱, 나승훈, 김태형, 류휘정, 장두성

전북대학교, KT

서론

- 딥러닝 기반 신경망들은 개체명 인식에 적용되어 준수한 성능을 보임
- 이 중 신경망에 사전을 추가하여 성능을 향상하려는 시도 또한 존재
- DyLex 또한 사전을 통해 얻은 단어 독립적 정보를 모델의 출력과 융합해 성능 향상시킴
- 본 논문에서는 DyLex를 한국어 개체명 인식에 적용해 기존 모델 대비 성능이 향상됨을 보임

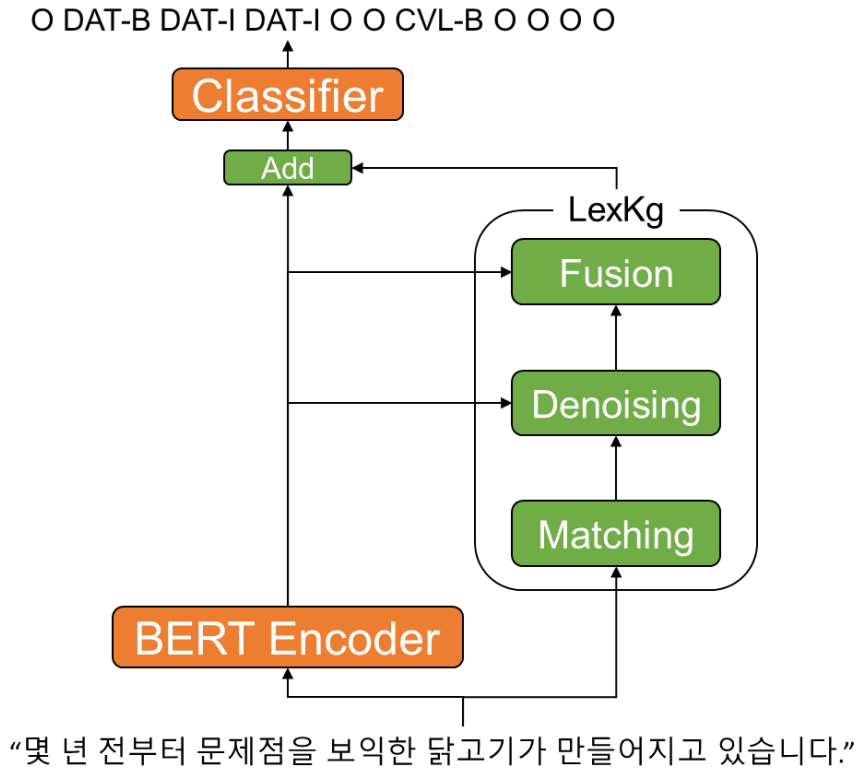
관련 연구

- 기존의 사전을 이용한 연구들은 사전의 단어를 그대로 임베딩에 입력
  - 사전이 수정될 경우 모델의 재학습 필요
  - 분류 태그는 학습에 직접 활용하지 못함

- DyLex는 입력을 사전과 대조해 추정한 토큰들의 태그를 언어 모델의 임베딩과 융합해 분류

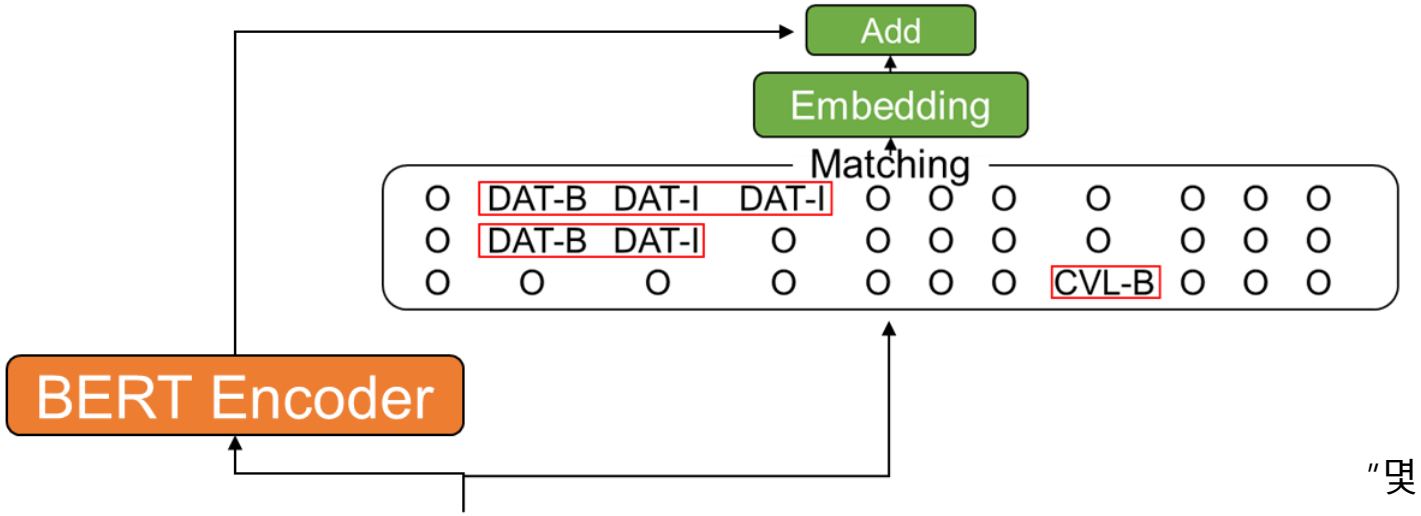
- 위와 같은 방법은 사전의 용어를 직접 사용하지 않아 상기 문제들을 해소

모델 구성

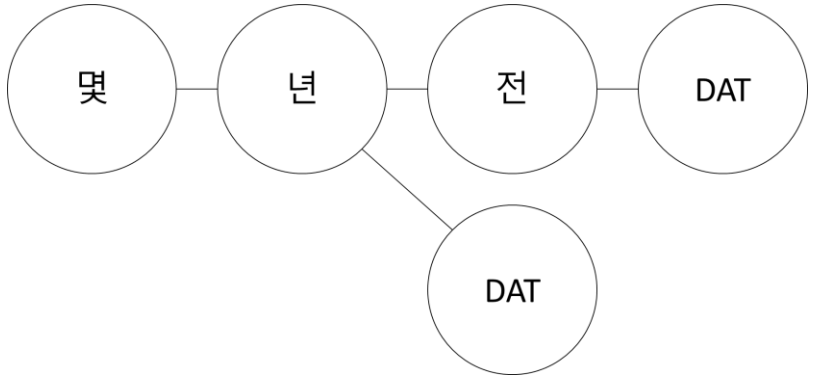


- 기존의 DyLex의 모델을 구조변경 없이 그대로 사용
- 모델은 BERT 인코더와 LexKg 모듈로 구성
- LexKg 모듈은 *Matching*, *Denoising*, *Fusion*의 세 단계로 세분화됨

모델 구성



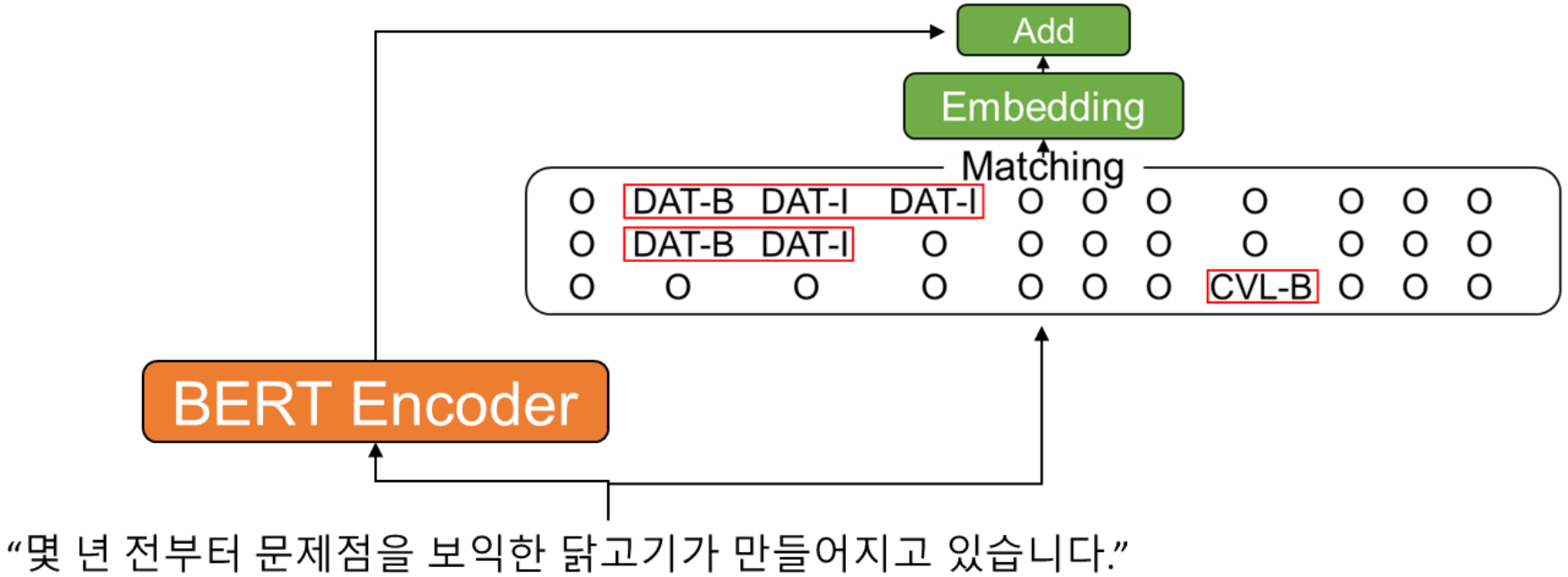
"몇 년 전부터 문제점을 보익한 닭고기가 만들어지고 있습니다."



"몇 년"과 "몇 년 전"이 트라이 구조로 저장된 사전의 예시

- 토큰화된 문장 X가 입력되면 *Matching* 단계에서 입력 문장을 사전과 비교해 탐색
- 탐색으로 얻은 후보 용어의 위치에 태그를, 이외에는 'O' 태그를 채워 i번째 검색 결과 $T^{(i)}$ 생성

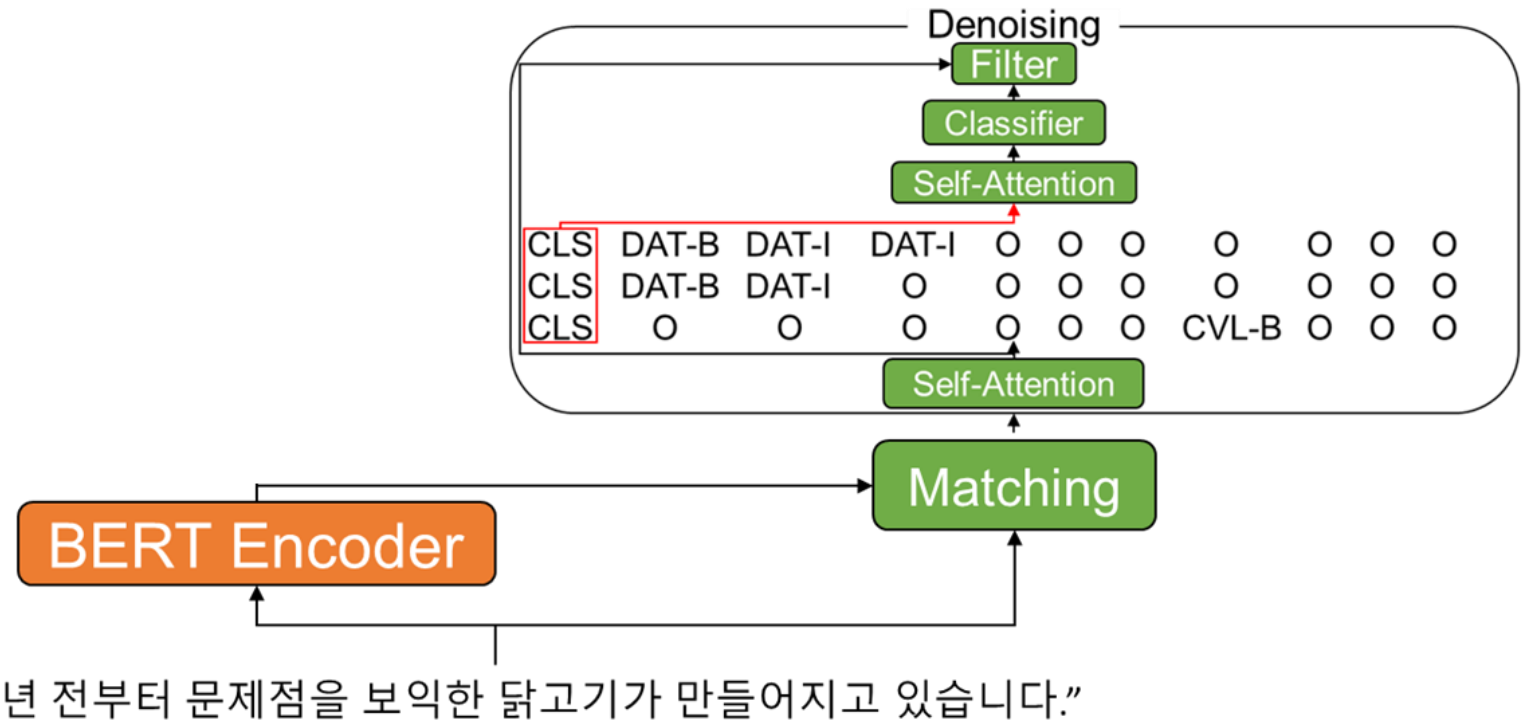
모델 구성



- $T^{(i)}$의 첫 부분에 [CLS] 토큰을 붙인 뒤 임베딩 층을 거쳐 $E_t^{(i)}$=Embedding($T^{(i)}$)와 같이 임베딩 생성

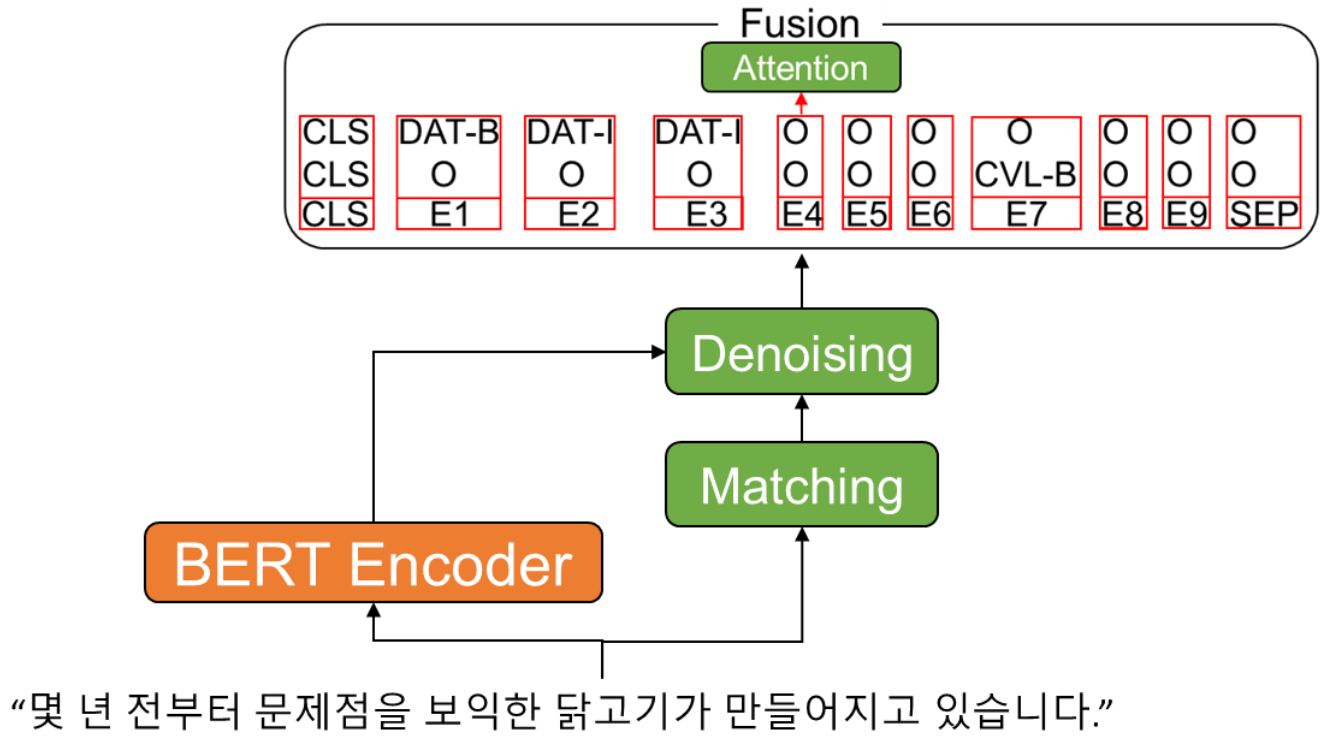
- 생성된 임베딩을 BERT의 임베딩 E_u = BERT(X)와 더해 $E_d^{(i)} = E_t^{(i)} + E_u$ 생성

모델 구성



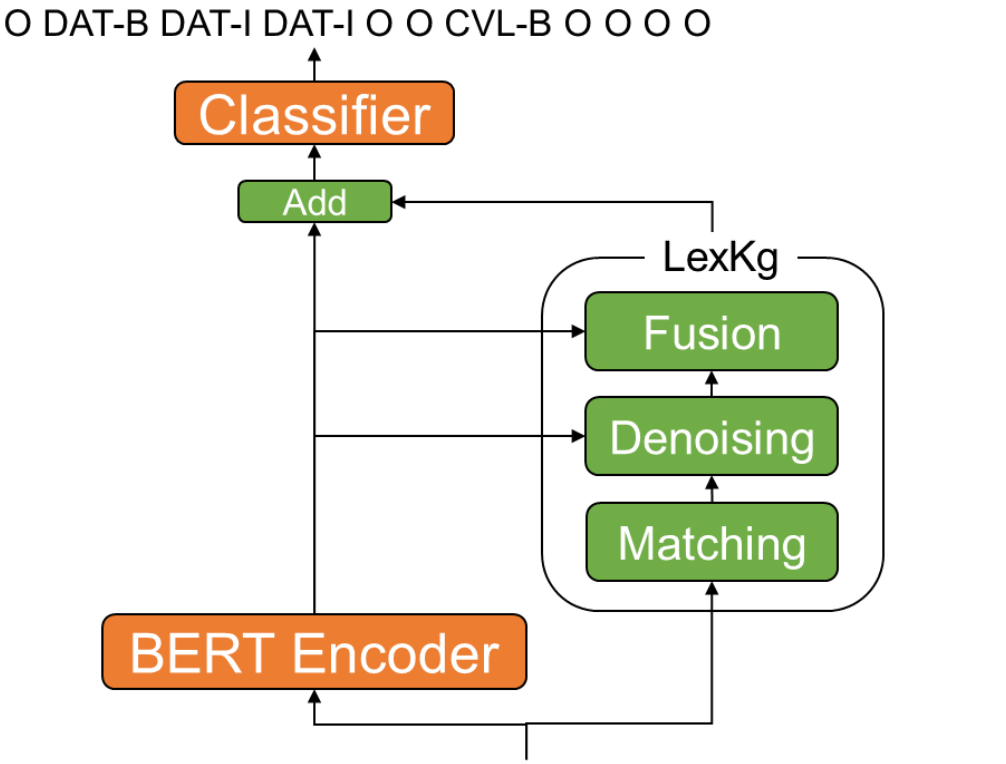
- *Denoising* 단계에서는 *Matching* 단계에서 얻은 결과에서 노이즈를 제거
- E_d를 이용해 $R_d^{(i)}$=Self-Attention($E_d^{(i)}$)를 얻은 뒤 R_d들의 [CLS] 토큰들 R_{cls}를 사용해 Y=Self-Attention(R_{cls})를 얻음
- 이후 P=Linear(Y)와 같이 선형층을 통해 분류해 참으로 분류된 $R_d^{(i)+}$를 얻음

모델 구성



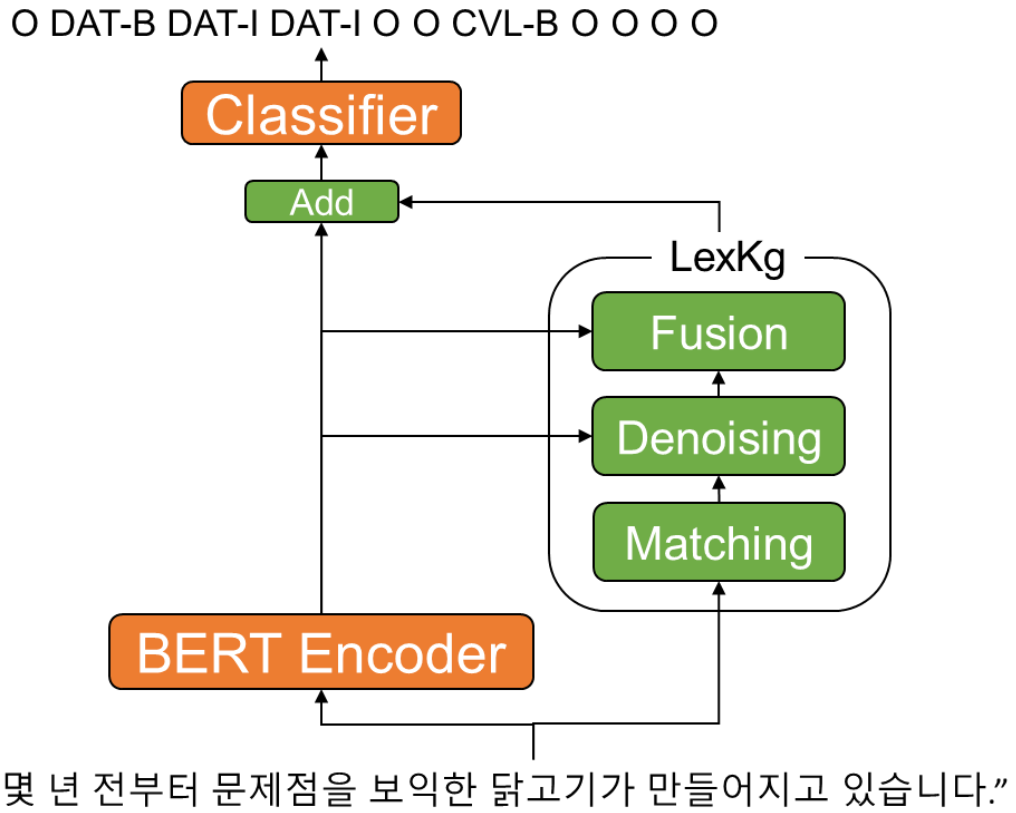
- *Fusion* 단계에서는 BERT와 Denoising으로 걸러낸 정보를 융합
- 문장에서 동일 위치에 있는 토큰끼리 어텐션을 수행해 $E_k = \text{Attention}(Q, K, V)$ 를 얻음
- 이 때 *Query, Key, Value*는 각각 $Q = E_u^{(j)}$, $K = V = \left[R_u^{(1,j)}; \cdots; R_d^{(m,j)}\right]$

모델 구성



- 융합된 출력을 BERT의 출력과 더해 $E_O = E_u + E_k$를 얻음
- 이후 선형층을 거쳐 최종 결과 $O = \text{Linear}(E_O)$를 얻음
- 이때 $O \in \mathbb{R}^{|X| \times nd}$, nd는 정답 클래스의 개수

실험 구성



- 한국어 적용을 위해 기존 DyLex의 BERT를 KoBERT로 교체
- DyLex에서 LexKg 모듈을 제거해 Baseline으로 사용
- 최대 5토큰 길이의 용어를 비교
- 같은 위치에서 여러 길이의 결과가 검색된 경우 가장 긴 결과 1개를 선택

실험 구성

Hyperparameters	
Batch Size	64
Learning Rate	5e-5
Optimizer	AdamW
Weight Decay	0.01
Dropout	0.1
Warmup Proportion	10%
Epochs	15

- 2018년 공개된 네이버/창원대 개체명 인식 데이터 사용
- 정답 데이터를 토큰화 한 뒤 트라이에 저장해 사전을 구성

실험 결과 및 결론

Model	Precision	Recall	F1 Score
KoBERT	83.94	86.80	85.37
DyLex	**85.97**	**89.06**	**87.49**

네이버 개체명 인식데이터 실험 결과

감사합니다