

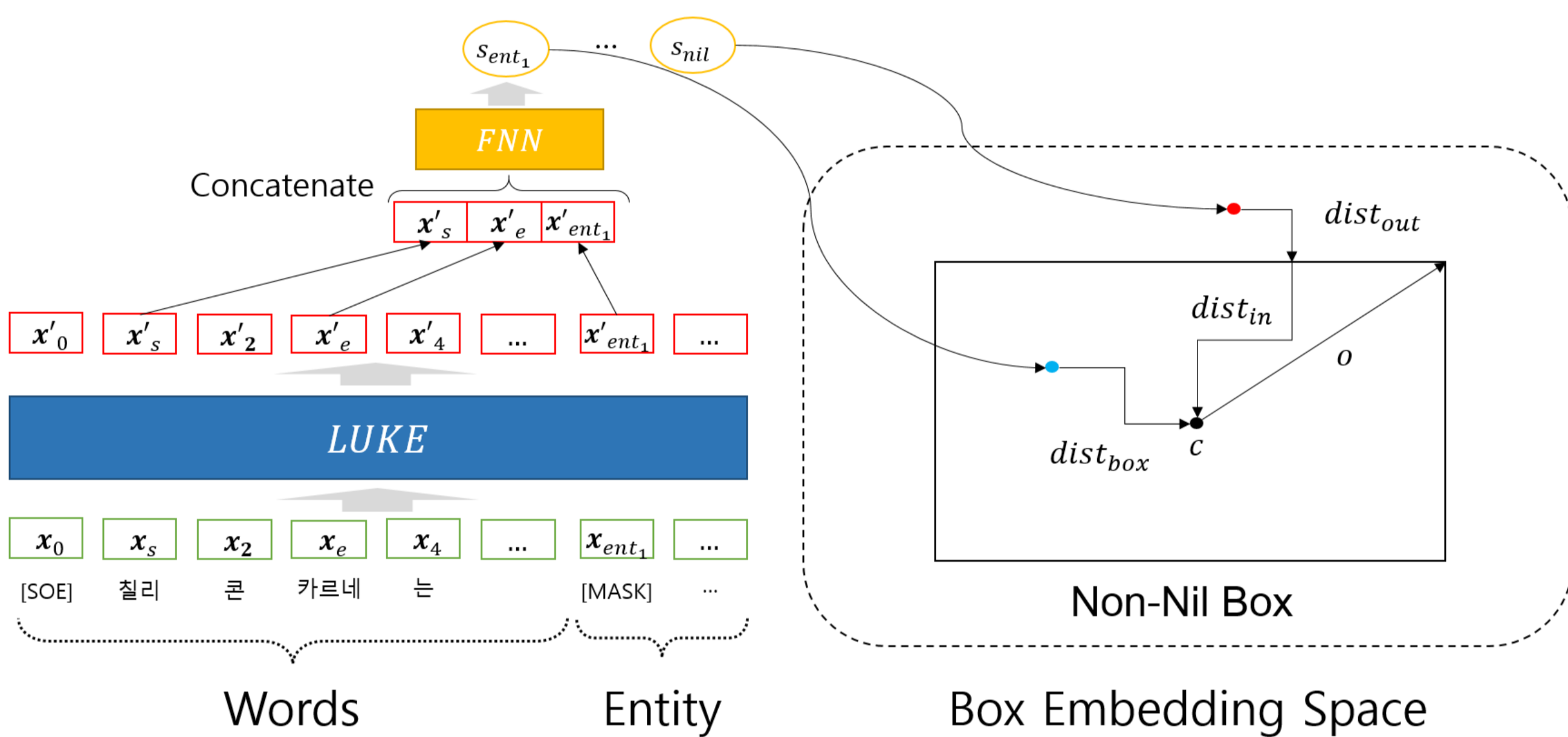
## I. 서론

Nil 멘션 탐지는 문장 내에서 특별히 의미가 있는 부분과 일반적인 부분을 구별하는 것을 의미한다. 이는 엔티티 링킹 등의 작업을 하기 위한 전단계로, 해당 작업을 수행하기 위한 부분을 판별하기 위함이기 때문이다. 따라서 높은 성능의 엔티티 링킹을 위해서는 정확한 멘션 탐지가 요구된다.

칠리 콘 카르네는 간 소고기에 강낭콩,  
칠리 파우더를 넣고 끓인 매운 스투이다.

위의 예제 문장에서 칠리 콘 카르네, 소고기, 강낭콩, 칠리 파우더는 각각 요리나 식재료를 나타내는 문서로 연결 될 수 있다. 이러한 멘션을 Non-nil 멘션 이라고 하고, 그렇지 않은 경우를 Nil 멘션이라고 한다.

## II. Box One-class Nil 탐지 모델



Nil 탐지 모델은 Luke모델을 기반으로 구성되며, 멘션의 표상은 입력 문장  $x$ 을 Luke를 통해 Encoding 한 후, 멘션의 첫 부분에 해당하는 토큰  $x_s$ 와 마지막에 해당하는 토큰  $x_e$ 의 출력  $x'_s, x'_e$ 과 엔티티의 표상  $x'_{ent}$ 을 연결하여 멘션에 대한 표상으로 사용했다.

$$x' = \text{LUKE}(x) = [x'_1, x'_2, \dots, x'_s, \dots, x'_e, \dots, x'_{ent}]$$

$$s = \text{Linear}(\text{Cat}(x'_s, x'_e, x'_{ent}))$$

Box 임베딩은 기존의 TransE 등의 임베딩 방법과 달리 임베딩 공간상의 초-직육면체 형태로 임베딩을 구성한다. Box  $\mathbf{B}$ 는 중심점  $\mathbf{c}$ 와 오프셋  $\mathbf{o}$ 로 구성된다. 따라서,  $\mathbf{B}$ 는 다음의 영역 내의 공간을 의미하게 된다.

$$\mathbf{B} \equiv \{v \in \mathbb{R}^d : \mathbf{c} - \mathbf{o} \leq v \leq \mathbf{c} + \mathbf{o}\}$$

$\mathbf{B}$ 와 점 사이의 거리는  $dist_{in}$ 과  $dist_{out}$ 로 계산 된다. 이때,  $dist_{in}$ 는 Box 내부에서의 거리,  $dist_{out}$ 는  $\mathbf{B}$  외부에서의 거리를 의미한다. 따라서, 멘션의 표상과  $\mathbf{B}$  간의 거리  $dist_{box}(s, \mathbf{B})$ 는 다음과 같이 계산 된다.

$$dist_{box}(s; \mathbf{B}) = dist_{out}(s, \mathbf{B}) + \alpha \cdot dist_{in}(s, \mathbf{B})$$

$$dist_{out}(s; \mathbf{B}) = |\text{Max}(s - \mathbf{B}_{max}, 0) + \text{Max}(\mathbf{B}_{min} - s, 0)|_1$$

$$dist_{in}(s; \mathbf{B}) = |\mathbf{c} - \text{Min}(\mathbf{B}_{max}, \text{Max}(\mathbf{B}_{min}, s))|_1$$

$$B_{max} = \mathbf{c} + \mathbf{o}$$

$$B_{min} = \mathbf{c} - \mathbf{o}$$

최종적인 Nil 탐지는 멘션의 표상  $s$ 와  $\mathbf{B}$  사이의 거리를 통해 다음과 같이 계산되도록 했다.

$$p(s|nonnil)_{box} = \sigma(\gamma - dist_{box}(s; \mathbf{B}))$$

## III. 베이스 라인 모델과의 결합

베이스라인 모델은 LUKE에 기반한 End-to-End 엔티티 링킹 모델로, Nil 탐지와 엔티티 링킹을 동시에 수행한다. 이는 후보 엔티티에 Nil 엔티티를 추가하여, Nil 엔티티를 선택 하는 경우 멘션이 Nil이라고 예측 하는 것이다. 이 모델에서 각각의 엔티티  $e$ 에 대한 스코어는 Biaffine 연산을 통해 다음과 같이 계산된다.

$$p(s|e)_{base} = \text{Softmax}(\text{score}(s, e); \text{score}(s, e'))$$

$$\text{score}(s|e) = s \cdot U \cdot e^T + W_s \cdot s + W_e \cdot e + \mathbf{b}$$

Box One-class Nil 탐지 모델과 베이스 라인 모델의 결합은 다음과 같이 시행했다. 이때,  $e_{nil}$ 은 베이스라인 모델의 Nil 엔티티이고, 멘션이 Nil일 확률을 계산하기 위해 Non-nil 확률을 뒤집었다.

$$p(s|nil)_{comb} = \alpha \cdot p(s|e_{nil})_{base} + (1 - \alpha)(1 - p(s|nonnil)_{box})$$

Nil 엔티티가 아닌 엔티티에 대한 예측값은 베이스 라인 모델의 것을 사용했다.

## IV. 실험 결과

### • $\alpha$ 에 따른 결합 멘션 탐지 성능(%)

$\alpha$	Precision	Recall	F1
0.0	86.94	91.31	89.07
0.1	87.08	91.37	89.17
0.3	87.44	91.51	89.43
0.5	88.23	91.30	89.74
0.7	88.27	90.97	89.60
0.9	88.22	90.60	89.39
1.0	88.20	90.43	89.30

### • $\alpha$ 에 따른 엔티티 링킹 성능(%)

$\alpha$	Precision	Recall	F1
0.0	85.22	89.50	87.30
0.1	85.35	89.56	87.41
0.3	85.71	89.70	87.66
0.5	86.54	89.55	88.02
0.7	86.60	89.25	87.91
0.9	86.57	88.91	87.72
1.0	86.55	88.74	87.63

## V. 결론 및 향후 연구

이번 연구에서는 Box 임베딩에 기반한 One-class 멘션 탐지 모델을 구성, 베이스 라인 모델과 결합하여 멘션 탐지 성능과 엔티티 링킹 성능을 향상시켰다. 하지만, 일괄적으로  $\alpha$ 를 적용 하기 보다 각 멘션에 맞춰 적합한  $\alpha$ 를 적용 할 경우 더 높은 성능 향상을 볼 수 있을 것으로 보인다.

또한, Box 임베딩은 논리곱 연산에 대응되는 Intersection, 논리합 연산에 대응되는 Union 등의 연산을 통해 Box 간 관계를 학습 시킬 수 있다. 따라서, 엔티티간에 존재하는 관계를 학습시키도록 Box 임베딩을 학습시켜 엔티티 링킹 전체로 확장 할 수 있을 것이라고 본다.