

# 형태소 수준의 한국어 상호참조해결

조경빈\*<sup>0</sup>, 최요한\*, 이창기\*, 류지희\*\*, 임준호\*\*

\*강원대학교 빅데이터메디컬융합학과, \*\*한국전자통신연구원  
{jkb5570, choiyohan, leeck}@kangwon.ac.kr, {chrisjihee, joonho.lim}@etri.re.kr

## Korean Coreference Resolution at the Morpheme Level

Kyeongbin Jo\*<sup>0</sup> Yohan Choi\* Changki Lee\* Jihee Ryu\*\* Joonho Lim\*\*

\*Department of Big Data Medical Convergence, Kangwon National University

\*\*Electronics and Telecommunications Research Institute

### 요약

상호참조해결은 주어진 문서에서 상호참조해결 대상이 되는 멘션(mention)을 식별하고, 동일한 개체(entity)를 의미하는 멘션들을 찾아 그룹화하는 자연어처리 태스크이다. 최근 상호참조해결에서는 BERT를 이용하여 단어의 문맥 표현을 얻은 후, 멘션 탐지와 상호참조해결을 동시에 진행하는 End-to-End 모델이 주로 연구가 되었다. 그러나 End-to-End 방식으로 모델을 수행하기 위해서는 모든 스패를 잠재적인 멘션으로 간주해야 되기 때문에 많은 메모리가 필요하고 시간 복잡도가 상승하는 문제가 있다. 본 논문에서는 서브 토큰을 다시 단어 단위로 매핑하여 상호참조해결을 수행하는 워드 레벨 상호참조해결 모델을 한국어에 적용하며, 한국어 상호참조해결의 특징을 반영하기 위해 워드 레벨 상호참조해결 모델의 토큰 표현에 개체명 자질과 의존 구문 분석 자질을 추가하였다. 실험 결과, ETRI 질의응답 도메인 평가 셋에서 F1 69.55%로, 기존 End-to-End 방식의 상호참조해결 모델 대비 0.54% 성능 향상을 보이면서 메모리 사용량은 2.4배 줄아졌고, 속도는 1.82배 빨라졌다.

주제어: 상호참조해결, BERT, end-to-end

### 1. 서론

상호참조해결(coreference resolution)은 명사, 대명사, 명사구 등의 멘션(mention) 후보를 식별하고, 동일한 개체(entity)를 의미하는 멘션들을 찾아 그룹화(clustering)하는 자연어처리 태스크이며, 기계 번역, 질의 응답, 문서 요약 등의 여러 자연어처리 태스크에 활용될 수 있다. 과거에는 문서 내에서 멘션을 찾는 멘션 탐지 태스크와 찾은 멘션과 그에 해당하는 선행사를 그룹화하는 상호참조해결 태스크로 나누어서 수행을 하였으나, 최근 멘션 탐지 태스크와 상호참조해결 태스크를 동시에 수행하는 End-to-End 방식의 상호참조해결 모델[1,2]이 주로 연구되었다.

최근 한국어 상호참조해결은 BERT[3]를 이용한 End-to-End 방식의 모델[1,2]이 주로 연구가 되고 있다. 이러한 모델들은 End-to-End 방식으로 상호참조해결을 수행하기 위해 문서 내의 모든 스패(span)를 잠재적인 멘션으로 간주하고, 이로 인해 계산 복잡도가 높아지며, 이를 해결하기 위해 멘션 스코어를 계산하여 상위 k개의 멘션만을 다루어 상호참조해결을 수행하고 있으나 여전히 많은 메모리가 필요하고 학습 및 실행 속도가 느리다는 문제점이 있다.

지금까지의 End-to-End 방식의 한국어 상호참조해결[1]은 이러한 멘션 스코어를 계산하기 위해 BERT 출력을 그대로 사용하는 것이 아닌 하나의 스패 표현을 생성하였으며, 문서 내의 모든 스패에 대해 이러한 표현을 계산하는 데에 있어서 상당한 비용이 소요가 되었다. 그러나 최근에는 단순히 BERT 출력에 Linear 함수를 적용하여

시작과 끝 표현을 계산하여 별도의 스패 표현 없이 메모리가 효율적이면서 속도가 빠른 모델[2]이 제안되었다. 그러나 모든 멘션을 고려하였을 경우 계산량이 높기 때문에 멘션 스코어 기준으로 일부 멘션 후보들에 대해서만 모델을 수행한다는 단점이 존재한다.

본 논문에서는 이와 같은 단점을 개선하여 BPE(Byte-pair encoding)단위의 서브 토큰들을 다시 단어 수준으로 매핑한 토큰 표현으로 모든 멘션들에 대해 상호참조해결을 수행하는 워드 레벨 상호참조해결 모델[4]을 한국어에 적용하였고, 한국어 상호참조해결의 특징을 반영하기 위해 기존 워드 레벨 상호참조해결 모델의 토큰 표현에 개체명 자질과 의존 구문 분석 자질을 추가하였다.

### 2. 관련 연구

최근 BERT 기반의 End-to-End 상호참조해결 모델[1,2]이 주로 연구가 되었다. 이러한 모델은 멘션 탐지 태스크와 상호참조해결 태스크를 동시에 진행하기 위해서 모든 스패를 잠재적인 멘션으로 간주하기 때문에 멘션 후보들이 많아진다. 문서의 입력 토큰 수를  $T$ 라 할 때 스패 후보 수는  $T(T+1)/2$ 가 된다. 모든 스패 후보를 고려하여 모델을 수행하면  $O(T^4)$ 의 높은 계산 복잡도가 소요가 되기 때문에 스코어를 기준으로 가지치기(Pruning)를 수행하여 일부 멘션만을 가지고 모델을 수행하는 특징을 가진다. 초기에는 이러한 스코어를 위해 스패 표현을 생성하여 계산하였지만, 최근에는 스패 표현을 사용하지 않아 효율적으로 모델을 수행하는 상호참조해결 모델[2]이 연구가 되고 있다.

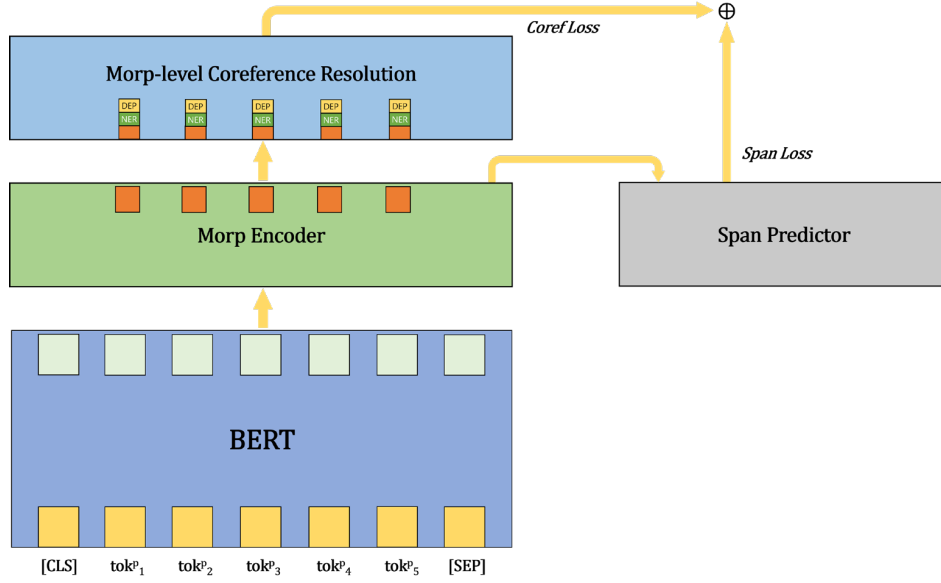


그림 1 형태소 수준의 한국어 상호참조해결 모델, 상호참조해결 모듈과 스캔 추출 모듈을 분리하여 학습을 수행하며, 한국어 상호참조해결의 특징을 반영하기 위해 개체명 자질과 의존 구문 분석 자질을 추가함

### 2.1 스캔 표현을 사용하는 상호참조해결

스캔 표현을 사용하는 상호참조해결 End-to-End 모델 [1]은 문맥 표현을 반영한 BERT의 출력에 해당하는 토큰들의 개체명과 의존 구문 분석 자질을 연결(concatenation)한 후 bi-LSTM 레이어를 통해 인코딩하여 히든 스테이트  $h_t$ 를 생성하였는데 이는 상호참조해결의 대상이 되는 멘션이 대명사(구)와 명사(구)로 구성되어 있어 개체명과 의존 구문 분석 자질을 추가하였을 때 성능 향상을 보였기 때문이다. 이후, 각 스캔에 해당하는  $h_t$ 의 시작과 끝 표현과 어텐션 매커니즘[5]을 적용한 중심어 표현과 스캔 사이의 거리 자질  $\phi$ 를 연결하여 스캔 표현  $g_i$ 를 생성한다. 스캔 표현  $g_i$ 에 FFNN(Feed-forward Neural Network)을 적용한 멘션 스코어  $s_m$ 를 통해 상위  $k$ 개의 멘션 후보를 추출한다.

$$w_t = ([BERT(x_t); ner(x_t); dep(x_t)]) \quad (1)$$

$$h_t = biLSTM(w_t) \quad (2)$$

$$a_t = w_a \cdot FFNN_a(h_t) \quad (3)$$

$$a_{i,t} = \frac{\exp(a_k)}{\sum_{k=START(i)}^{END(i)} \exp(a_k)} \quad (4)$$

$$\hat{h}_t = \sum_{t=START(i)}^{END(i)} a_{i,t} \cdot x_t \quad (5)$$

$$g_i = [h_{START(i)}, h_{END(i)}, \hat{h}_t, \phi(i)] \quad (6)$$

$$s_m(i) = w_m \cdot FFNN_m(g_i) \quad (7)$$

이후  $k$ 개의 스캔 표현에 대해서 bilinear 연산  $s_c$ 을 적용하여 상위  $c$ 개의 선행사를 추려내고,  $c$ 개의 후보들에 대해 각 멘션과 선행사의 스캔 표현을 사용하여 선행사 스코어  $s_a$ 를 계산한다. 식 9에서  $i$ 와  $j$  ( $1 \leq j \leq i-1$ )는

현재 멘션과 선행사의 인덱스이며,  $g_i$ 와  $g_j$ 는  $i$ 와  $j$  멘션의 스캔 표현을 나타낸다. 마지막으로, 식 10과 같이 멘션 스코어와 선행사 스코어를 합하여 상호참조해결 스코어를 계산한다. 모델의 출력은 상호참조해결 스코어가 가장 높은  $k$ 개의 (멘션, 선행사) 쌍이다.

$$s_c(i, j) = g_i^T W_c g_j \quad (8)$$

$$s_a(i, j) = w_a \cdot FFNN_a([g_i, g_j, g_i \circ g_j, \phi(i, j)]) \quad (9)$$

$$s(i, j) = \begin{cases} 0 & j = \epsilon \\ s_m(i) + s_m(j) + s_a(i, j) + s_c(i, j) & j \neq \epsilon \end{cases} \quad (10)$$

### 2.2 스캔 표현을 사용하지 않는 상호참조해결

[2]의 연구에서는 스캔 표현을 적용하지 않고 상호참조해결 스코어를 계산하는 모델[6]을 한국어 상호참조해결에 적용하였으며, 성능을 개선하기 위해 식 1과 같이 개체명 자질과 의존 구문 분석 자질을 추가하였고, 한국어가 중심어 후행 언어(head-final language)라는 특성을 반영하여 목적 함수를 수정하였고, 중심어에 특화된 모델을 구현하기 위해 시작과 관련된 표현식을 제거하여 한국어에 최적화된 모델을 구현하였다. 멘션 스코어  $f_m$ 를 통해 상위  $k$ 개의 멘션 후보를 추출하며, 스캔 표현을 사용하지 않아 계산량이 적기 때문에 이후  $k$ 개의 멘션 후보들에 대해 선행사 스코어를 계산한다.

$$GLU(x) = x \otimes \sigma(FFNN(x)) \quad (11)$$

$$h_t = GLU([BERT(x_t), ner(x_t), dep(x_t)]) \quad (12)$$

$$m^s = \text{GeLU}(w_m^s h_t) \quad m^e = \text{GeLU}(w_m^e h_t) \quad (13)$$

$$f_m(q) = v_s \cdot m_{q_s}^s + v_e \cdot m_{q_e}^e + m_{q_s}^s \cdot B_m \cdot m_{q_e}^e \quad (14)$$

멘션 스코어를 계산할 때, 별도의 스캔 표현을 생성하지 않고, 문맥 표현  $h_t$ 에 linear 연산을 적용한 후

GeLU 함수를 적용하여 시작과 끝 표현을 계산하고, 식 14에서 시작과 끝 표현에 bilinear 연산을 합하여 멘션 스코어를 계산한다.  $v_s$  와  $v_e$  가중치와  $B_m$  행렬은 학습 가능한 파라미터이다. 식 14의 첫 번째와 두 번째 항을 통해 해당 스캔이 정답 멘션의 시작과 끝에 해당하는 지 여부를 판단할 수 있고, 세 번째 항을 통해 해당 멘션이 정답 멘션과 같은 경계를 가지고 있는 지 판단할 수 있게 된다.

$$a^s = \text{GeLU}(w_a^s h_t) \quad a^e = \text{GeLU}(w_a^e h_t) \quad (15)$$

$$f_a(c, q) = a_{c_s}^s \cdot B_a^{ss} \cdot a_{q_s}^s + a_{c_s}^e \cdot B_a^{se} \cdot a_{q_e}^e + a_{c_e}^e \cdot B_a^{es} \cdot a_{q_s}^s + a_{c_e}^s \cdot B_a^{ee} \cdot a_{q_e}^e \quad (16)$$

또한, 선행사 스코어도 멘션 스코어와 동일하게 계산하기 위한 시작과 끝 표현식을 생성한다. 식 16에서는 쿼리 스캔과 후보 스캔 간 각 경계에 대해 시작과 시작, 시작과 끝, 끝과 시작, 끝과 끝의 총 4가지 연산을 수행한다. 해당 모델[6]에 한국어가 중심어 후행 언어라는 특성을 반영하기 위해 시작 표현에 해당하는 식 14의 첫 번째 항과 식 16의 첫 번째 항을 제거하여 목적 함수를 재구성하였다.[2]

### 3. 형태소 수준의 한국어 상호참조해결

본 논문에서는 서브 토큰을 다시 단어 수준으로 매핑하여 수행하는 상호참조해결 모델[4]을 한국어에 적용하였으며, 한국어의 경우 영어와 다르게 단어 단위로 매핑할 경우 조사도 함께 고려되므로 형태소 단위로 적용하였다. 모델의 가장 큰 특징 2가지는 첫 번째로 스캔 추출 모듈을 모델부터 분리하여 가벼운 모델로 구현하였고, 두 번째로는 서브 토큰을 단어 단위로 매핑하는 토큰 표현을 생성하여 계산 복잡도를 낮추었다. 이로 인해 모든 멘션 후보를 다룰 수 있다는 이점을 가지고 있다.

#### 3.1 토큰 표현

BERT 출력  $X$ 에 linear 함수와 서브 토큰에서 단어 단위의 마스킹(masking) 정보를 이용하여 [단어, 서브 토큰] 크기의 행렬  $Z$  를 생성한다. 이후, 소프트맥스(softmax)를 적용하여 어텐션 스코어를 계산해주고, 어텐션 스코어와 BERT 출력  $X$ 를 행렬 곱 연산(matrix multiplication)을 통해 최종적인 토큰 표현  $T$ 를 계산한다. 여기서  $W_a$ 는 학습 가능한 파라미터이며, 모든 멘션을 다루기 때문에 멘션 스코어는 계산하지 않는다. 그림 1의 Morp Encoder 부분에서 토큰 표현을 계산한다.

$$Z = W_a \cdot X \quad (17)$$

$$T = \text{softmax}(Z) \cdot X \quad (18)$$

#### 3.2 Coarse-to-fine 선행사 가지치기

모든 멘션에 대해 선행사 스코어를 계산하면 계산량이 커지기 때문에 계산량을 줄이기 위해 1차적으로 거친(Coarse) 가지치기를 진행한다. 2.1절에서의 식 8과 같이 bilinear 연산을 통해  $k$ 개의 선행사 후보를 추려낸 후,

$k$ 개의 선행사 후보들에 대해서 토큰  $i$ 와 토큰  $j$ , 토큰  $i$ 와  $j$ 의 유사도(similarity)를 의미하는 원소별 곱셈(element-wise multiplication), 마지막으로 두 토큰 사이의 거리 자질  $\phi$ 과 본 논문에서 새로 추가한 개체명 자질(ner)과 의존 구문 분석 자질(dep)를 연결하여 FFNN을 통해 선행사 스코어  $s_a$ 를 계산한다. 개체명과 의존 구문 분석 자질은  $k$ 개의 선행사 후보들의 토큰들에 해당되는 개체명 태그 및 의존 구문 분석 태그 임베딩이다. 해당 임베딩을 추가함으로써 한국어 상호참조해결의 특성을 반영하여 좋은 성능을 얻을 수 있었다. 각 최종적인 상호참조해결 스코어는  $s_c$ 와  $s_a$ 의 합으로 계산이 된다.

$$s_c = T \cdot W_c \cdot T^T \quad (19)$$

$$s_a(i, j) = \text{FFNN}_a([T_i, T_j, T_i \odot T_j, \phi, \text{ner}, \text{dep}]) \quad (20)$$

$$s(i, j) = s_c(i, j) + s_a(i, j) \quad (21)$$

### 3.3 스캔 추출

단어 수준의 상호참조해결 모델[4]의 경우 모델의 경량화와 추후 확장성을 위해 스캔 추출 모듈을 따로 분리하였다. 스캔 추출 모듈이 추가되면서 생기는 계산량을 줄이기 위해 상호참조가 되어진 토큰들에 대해서만 모듈이 수행된다. 각 토큰에 대해서 모듈은 같은 문장에서 가장 가능성이 있는 시작과 끝 토큰을 예상하여 스캔을 재구성하며 학습을 한다. 이와 같이 재구성하기 위해서 같은 문장 내에 있는 모든 토큰들에 목적 스캔에 해당하는 중심어 토큰을 연결한 후 FFNN을 거치며, 시작과 끝 스코어를 계산하기 위해 2개의 출력 채널을 가진 Convolution Network를 거치게 된다. 커널 사이즈(kernel size)는 3이다. 따라서, 해당 모듈을 학습하는 데에 있어서 각 스캔에 해당하는 중심어를 학습 데이터로 주어 야 하는데 영어 데이터의 경우 의존 구문 분석 결과를 통해 중심어 토큰을 결정하였으며, 한국어의 경우 중심어 후행 언어이므로, 스캔의 끝 토큰에서 조사를 제외한 토큰으로 결정하였다. 이와 같이 시작과 끝 토큰을 중심어 토큰을 통해 결정하게 된 이유는 가장 최적의 스캔 경계는 스캔에 해당될 것 같은 토큰과 그렇지 않을 것 같은 토큰 사이에 있기 때문이다.[6]

### 4. 실험 및 결과

본 논문에서 사용한 데이터는 ETRI 질의응답 도메인 상호참조해결 데이터 셋이며, 성능 측정을 위해 중심어 경계(head boundary)를 기준으로 MUC, B<sup>3</sup>, CEAF-e, CoNLL-F1[7]을 사용하였다. 데이터 셋은 학습 데이터 2,819 문서, 개발 데이터 645 문서, 평가 데이터 571 문서로 구성된다. 형태소 수준의 상호참조해결 모델에 개체명과 의존 구문 분석 자질 추가에 따른 성능 변화 실험을 하였고, 다른 모델과의 비교를 통해 형태소 수준의 상호참조해결 모델의 성능과 속도 및 메모리를 비교하였다.

본 논문에서 실험한 2.2절의 모델에는 ETRI의 어절기반 KorBERT(BERT-base)를 사용하였으며, 2.1절 모델과 본

논문에서 개체명과 의존 구문 분석 자질을 추가한 3장의 모델은 형태소 기반의 KorBERT(BERT-base)를 사용하였다. 상호참조해결 모델 학습을 위한 하이퍼 파라미터는 다음과 같다. 태스크 학습률(learning rate)은  $3e-04$ , BERT 학습률은  $1e-05$ , FFNN은 드랍 아웃 0.3, FFNN의 히든 레이어 차원 수 1024, FFNN의 히든 레이어 수 1, 최대 스펠 길이 64, 학습을 위해 옵티마이저(optimizer)는 Adam을 사용하였다. 실험에 사용된 GPU는 RTX TITAN 24GB이다.

표 1은 자질 추가에 따른 모델의 성능 변화이다. 개체명 자질을 추가하였을 때, 평가 셋에서 0.33%의 성능 향상이 있었으며, 개체명과 의존 구문 분석 자질을 모두 추가하였을 때, 기존 모델 대비 평가 셋에서 0.74%의 성능 향상이 되었다.

표 2는 기존에 연구되었던 상호참조해결 모델과의 성능 비교이다. 본 논문에서 제안한 모델이 평가 셋 F1 69.55%의 성능으로 가장 높은 성능을 보였으며, 그 다음으로 스펠 표현을 사용하는 모델[1] 69.01%, 스펠 표현을 사용하지 않는 시작과 끝 경계 중심의 모델[2] 67.42%의 성능을 보였다. 이는 스펠 표현을 사용하는 모델이 정보가 더 많다는 장점으로 보일 수도 있으나 형태소 수준의 상호참조해결 모델처럼 스펠 표현을 사용하지 않고 더욱 효과적인 방식으로 더 높은 성능의 상호참조해결을 수행할 수 있음을 보여준다. 이는 모델의 매커니즘도 영향을 주지만 멘션 가지치기없이 모든 멘션을 후보로 모델을 수행한 영향도 크다고 생각된다.

표 3은 ETRI 질의응답 상호참조해결 평가 셋 571 문서에 대한 각 모델에 대한 속도 및 메모리 비교이다. 스펠을 사용하지 않는 모델들이 속도와 메모리에서 우수한 성능을 보이는 것을 알 수 있으며, 그 중 경계 중심의 모델[2]이 가장 좋은 결과를 보였다. 이는 스펠 표현을 모든 스펠에 대해 생성하는 데에 있어서 드는 계산량으로 보인다. 형태소 수준의 상호참조해결 모델이 가장 좋은 성능을 보이지 못한 이유는 스펠 추출 모듈을 사용하는 것과 토큰 표현의 생성, 모든 스펠을 고려하는 데에 있어서 드는 비용으로 생각되며, 단순히 BERT 출력에 linear 연산을 수행하는 모델[2]이 속도 및 메모리에서는 우세한 것으로 보인다.

## 5. 결론

본 논문에서는 서브 토큰을 형태소 단위로 매핑하여 상호참조해결을 수행하는 형태소 단위의 상호참조해결 모델을 제안하였으며, 모델의 성능 개선을 위해 개체명과 의존 구문 분석 자질을 추가하였다. 실험 결과, ETRI 질의응답 도메인 상호참조해결 평가 셋에서 CoNLL F1 69.55%로 최신 모델 대비 가장 높은 성능을 보였다. 이는 모델의 매커니즘도 영향을 주겠지만 별도의 멘션 가지치기 없이 모든 멘션을 고려하여 상호참조해결을 수행하였기 때문에 더욱 높은 성능을 얻은 것으로 생각된다. 또한, 스펠 표현을 사용하지 않기 때문에 End-to-End 방

식의 상호참조해결 모델[1]보다는 속도와 메모리면에서 크게 향상 되었지만, 경계 중심으로 상호참조해결을 수행하는 모델[2]보다는 속도와 메모리 면에서 부족한 부분을 보였다. 따라서, 추후 연구로는 형태소 수준의 상호참조해결 모델의 속도와 메모리 사용량을 개선해볼 예정이다.

표 1 자질 추가에 따른 성능

TEST				
Model	MUC	B <sup>3</sup>	CEAF <sub>fe</sub>	F1
Morp_level_coref	70.51	67.3	68.62	68.81
+NER	70.85	67.82	68.75	69.14
<b>+DEP</b>	<b>71.16</b>	<b>68.39</b>	<b>69.11</b>	<b>69.55</b>

표 2 모델 성능 비교

TEST				
Model	MUC	B <sup>3</sup>	CEAF <sub>fe</sub>	F1
e2e_coref[1]	70.07	67.67	69.29	69.01
s2e_coref_eojeol [2]	69.21	66.29	66.76	67.42
<b>Our_Model</b>	<b>71.16</b>	<b>68.39</b>	<b>69.11</b>	<b>69.55</b>

표 3 모델 속도 및 메모리 사용량 비교

Model(Test)	메모리 사용량	소요 시간
e2e_coref[1]	5.2GB	18.2초
<b>s2e_coref_eojeol[2]</b>	<b>1.6GB</b>	<b>5.6초</b>
Our Model	2.1GB	10초

## 감사의 글

이 논문은 2022년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2013-2-00131, 휴먼 지식증강 서비스를 위한 지능 진화형 Wise QA 플랫폼 기술 개발).

## 참고문헌

- [1] 김기훈, 박천음, 이창기, 김현기, "BERT 기반 End-to-end 신경망을 이용한 한국어 상호참조해결," *정보과학회논문지*, 제47권, 제10호, 2020.
- [2] 조경빈, 정영준, 최요한, 이창기, 류지희, 임준호, "스펠 표현을 사용하지 않는 어절 단위의 한국어 End-to-End 상호참조해결," *한국정보과학회 학술발표 논문집*, 2022.
- [3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, 2018.

- [4] Vladimir Dobrovolskii, "Word-Level Coreference Resolution," *EMNLP*, 2021
- [5] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," *ICLR*, 2015
- [6] Yuval Kirstain, Ori Ram, Omer Levy, "Coreference Resolution without Span Representations," *ACL*, 2021.
- [7] Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, "Deterministic Coreference Resolution Based on Entity-Centric, Precision-Ranked Rules," *Computational Linguistics*, Volume 39, Issue 4, 2013.