

대화 데이터를 위한

멘션 탐지 및 상호참조해결 파이프라인 모델

김담린⁰¹, 김홍진¹, 박성식¹, 김학수²

건국대학교 인공지능학과¹, 컴퓨터공학부²

{ekafsl33⁰¹, jin3430¹, a163912¹, nlpdrkim²}@konkuk.ac.kr

Mention Detection and Coreference Resolution Pipeline Model for Dialogue Data

Damrin Kim⁰¹, Hongjin Kim¹, Seongsik Park¹, Harksoo Kim²

Konkuk University Department of Artificial Intelligence¹, Computer Science and Engineering²

요약

상호참조해결은 주어진 문서에서 상호참조해결의 대상이 될 수 있는 멘션을 추출하고, 같은 개체를 의미하는 멘션 쌍 또는 집합을 찾는 자연어처리 작업이다. 하나의 멘션 내에 멘션이 될 수 있는 다른 단어를 포함하는 중첩 멘션은 순차적 레이블링으로 해결할 수 없는 문제가 있다. 본 논문에서는 이러한 문제를 해결하기 위해 멘션의 시작 단어의 위치를 여는 괄호('(', 마지막 위치를 닫는 괄호(')')로 태깅하고 이 괄호들을 예측하는 멘션 탐지 모델과 멘션 탐지 모델에서 예측된 멘션을 바탕으로 포인터 네트워크를 이용하여 같은 개체를 나타내는 멘션을 군집화하는 상호참조해결 모델을 제안한다. 실험 결과, 4개의 영어 대화 데이터셋에서 멘션 탐지 모델은 F1-score (Light) 94.17%, (AMI) 90.86%, (Persuasion) 92.93%, (Switchboard) 91.04%의 성능을 보이고, 상호참조해결 모델에서는 CoNLL F1 (Light) 69.1%, (AMI) 57.6%, (Persuasion) 71.0%, (Switchboard) 65.7%의 성능을 보인다.

주제어: 자연어처리, 딥러닝, 멘션 탐지, 상호참조해결

1. 서론

상호참조해결(coreference resolution)[1]이란 주어진 문서에서 식별한 멘션(mention)을 바탕으로 같은 개체(entity)를 의미하는 멘션들을 찾아 하나의 개체로 군집화(clustering)하는 것을 의미한다. 상호참조해결은 문서요약과 질의응답과 같은 작업을 위한 중요한 전처리 단계의 역할을 한다. 문서나 문장에서 상호참조 관계 가능성이 높은 후보 단어 스패(span)를 멘션이라고 하며, 멘션을 추출하는 작업을 멘션 탐지(mention detection)라고 한다. 일반적으로 멘션 탐지를 먼저 진행한 후에 추출된 멘션을 바탕으로 상호참조해결을 수행하므로 멘션 탐지의 결과가 상호참조해결의 성능에 직접적인 영향을 줄 수 있으며, 하나의 멘션이 T개의 단어로 구성되어 있으면 최대 $N = \frac{T(T+1)}{2} - 1$ 개의 멘션이 중첩되어 등장할 수 있기 때문에 정확한 멘션 탐지를 수행하는 것이 중요하다. 따라서 본 논문에서는 상호참조해결을 위해 1)멘션 탐지와 2)상호참조해결로 나누어 두 단계를 순차적으로 진행하는 파이프라인(pipeline) 모델을 제안한다.

멘션의 시작과 끝 위치를 멘션 경계라고 하며, 이 경계는 괄호로 표현한다. 본 논문에서는 "(길은 (녹색(꽃양배추)))" 와 같이 멘션 경계의 시작은 여는 괄호('(')로 태깅(tagging)하고, 끝은 닫는 괄호(')')로 태깅한다. 또한 각 멘션을 BIO 태그(tag)로 표현하면 [B-I-I]("길은 녹색 꽃양배추"), [O-B-I]("녹색 꽃양배추"), [O-O-B]("꽃양배추")이다. 이와 같이 하나의 멘션에서 2개 이상의 BIO 태그 조합이 나타나므로 기존

의 순차적 레이블링(sequence labeling)으로는 중첩 멘션 탐지 문제를 해결할 수 없다. 따라서 본 논문에서는 BIO 태그를 사용하는 순차적 레이블링이 아닌 여는 괄호와 닫는 괄호를 독립적으로 예측하는 순차적 레이블링을 이용한 멘션 탐지 모델을 제안한다.

멘션 탐지 모델에서 추출된 멘션 집합은 포인터 네트워크(pointer networks)[2]를 이용하여 상호참조해결을 수행한다. 포인터 네트워크는 입력 데이터에 해당되는 위치 인덱스(index)를 출력하는 RNN(Recurrent Neural Network)의 확장된 모델이며, 주어진 입력 데이터 중에서 결과에 영향을 미치는 특정 위치를 중심으로 계산된 어텐션(attention) 가중치를 학습하는 어텐션 메커니즘(mechanism)을 이용하는 모델이다. 포인터 네트워크는 시퀀스 투 시퀀스(sequence-to-sequence) 모델과 다르게 입력 데이터에 대해 어텐션 가중치를 계산하여 정답에 해당하는 위치 인덱스를 출력한다. 본 논문에서는 상호참조해결을 수행하기 위해 포인터 네트워크를 이용한 상호참조해결 모델을 제안한다.

2. 관련 연구

[3, 4]는 스패 순위화(ranking) 방법에 기반하여 멘션 탐지와 상호참조해결을 수행하고 LSTM[5]으로 문맥 표현을 반영하는 end-to-end 방법을 제안한다. 스패 순위화 모델은 문서에서 등장하는 모든 멘션을 추출하여 정답 멘션에 포함된 스패와 marginal log-likelihood를 계산하

여 멘션 탐지 모델을 학습하며, pruning 기법을 사용하여 복잡도가 높은 문제를 보완하는 방법이다. 또한 선행 사 집합에 대해 조건부 확률 분포를 계산하여 상호참조 점수에 따라 상호참조해결 링크 여부를 결정하는 방법을 사용하는 상호참조해결 방법을 제안했다. 또한 [3]에서는 정답 멘션을 가지고 상호참조해결을 수행했을 때, 예측 멘션에 비해 17%의 성능 향상을 보이며 이는 정확한 멘션 탐지의 중요성을 보여준다.

[6]은 중첩 멘션 탐지 문제를 해결하기 위해 시퀀스 태깅 모델(sequence tagging model)을 이용하는 멘션 탐지 모델을 제안한다. 이는 멘션의 시작 위치, 마지막 위치, 멘션에 해당되는 위치, 멘션에 해당되지 않는 위치에 대해 4가지 태그를 사용하여 문장을 태깅한 후에 모델의 입력으로 사용하는 방법을 제안했다.

본 논문에서는 정확한 멘션 탐지 및 상호참조해결을 수행하기 위해 여는 괄호와 닫는 괄호를 독립적으로 예측하는 시퀀스 레이블링을 이용하여 중첩 멘션 문제를 해결한 멘션 탐지 모델과 현재 멘션과 가장 가까이에 위치하는 후행 멘션을 가리키도록 학습하는 포인터 네트워크 방법을 이용한다. 멘션 탐지는 문장 단위, 상호참조해결은 문서 단위이며, [3]과 같이 정확한 멘션 탐지 결과가 상호참조해결의 성능에 직접적인 영향을 주므로 멘션 탐지와 상호참조해결을 순차적으로 진행하는 파이프라인 모델을 제안한다.

3. 상호참조해결

3.1 멘션 탐지

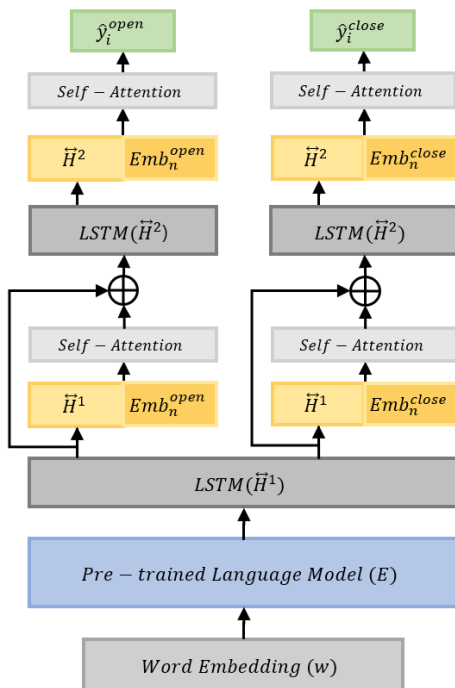


그림 1. 멘션 탐지 모델 구조도

그림 1은 본 논문에서 제안하는 멘션 탐지 모델 구조도이며 크게 사전 학습 언어 모델, 여는 괄호와 닫는 괄호를 예측하는 계층으로 구성되어 있다. 사전 학습 언어 모델은 문장을 입력 받아 단어 표현 벡터(vector)를 생성하고, 여는 괄호와 닫는 괄호를 예측하는 계층에서는 어텐션 연산을 통해 단어 표현 벡터와 괄호 임베딩 벡터와의 연관성을 계산하여 멘션 탐지를 수행한다.

3.1.1 여는 괄호와 닫는 괄호 예측 계층

그림 1의 토큰 표현 E 는 각 발화의 입력 토큰에 대해 사전 훈련된 ELECTRA[7]의 출력이며, 다음 수식과 같이 문맥 표현이 반영된 토큰 표현을 생성한다.

$$\begin{aligned} \vec{h}_i^1 &= LSTM(E_i, \vec{h}_{i-1}^1) \\ \vec{h}_i^1 &= LSTM(E_i, \vec{h}_i^1) \\ \vec{h}_i^1 &= [\vec{h}_i^1, \vec{h}_i^1] \\ \vec{H}^1 &= \{\vec{h}_{CLS}^1, \vec{h}_1^1, \dots, \vec{h}_{SEP}^1\} \end{aligned} \quad (1)$$

수식 (1)에서의 \vec{h}_i^1 는 정방향 은닉 상태(forward hidden state)를 나타내는 벡터이며 \vec{h}_i^1 는 역방향(backward) 은닉 상태를 나타내는 벡터이다. \vec{h}_i^1 는 i 번째 단계에서 정방향 은닉 상태와 역방향 은닉 상태를 연결한 양방향 벡터이다. \vec{H}^1 는 각 단어의 양방향 문맥 정보가 반영된 벡터이다.

괄호 예측 계층에서는 LAN(Label Attention Network)[8]의 구조를 사용하며, 토큰 표현과 여는 괄호 및 닫는 괄호 임베딩 벡터의 연관성을 계산하여 여는 괄호와 닫는 괄호를 독립적으로 학습한다. 수식 (1)에서 계산된 \vec{H}^1 과 그림 1의 사전 임베딩 Emb_n 사이의 연관성을 계산하기 위해 셀프 어텐션(self-attention)을 사용하며 수식은 다음과 같다.

$$\begin{aligned} A^1(C_i) &= softmax\left(\frac{Q * K^T}{\sqrt{d_n}}\right) \cdot V \\ Q &= \vec{H}^1, K = V = Emb_n \end{aligned} \quad (2)$$

수식 (2)의 임베딩 벡터 Emb_n 은 여는 괄호와 닫는 괄호 임베딩 벡터 테이블이 무작위로 초기화되며 Emb_n^{open} 는 여는 괄호 예측에 사용하고, Emb_n^{close} 는 닫는 괄호 예측에 사용한다. 여기서 n 은 괄호 개수를 의미한다. 다음으로, 수식 (1)에서 계산된 \vec{H}^1 과 수식 (2)에서 계산된 괄호 임베딩 분포가 강조된 벡터 $A^1(C_i)$ 를 연결(concatenation)하여 다음과 수식과 같이 양방향 LSTM의 입력으로 사용한다.

$$\begin{aligned} \vec{h}_i^2 &= LSTM([\vec{H}^1; A^1(C_i)], \vec{h}_{i-1}^2) \\ \vec{h}_i^2 &= LSTM([\vec{H}^1; A^1(C_i)], \vec{h}_{i-1}^2) \\ \vec{h}_i^2 &= [\vec{h}_i^2, \vec{h}_i^2] \\ \vec{H}^2 &= \{\vec{h}_{CLS}^2, \vec{h}_1^2, \dots, \vec{h}_{SEP}^2\} \end{aligned} \quad (3)$$

마지막으로 각 토큰의 여는 괄호와 닫는 괄호는 다음과 같이 예측한다.

$$A^2(C_i) = \text{softmax}\left(\frac{Q * K^T}{\sqrt{d_h}}\right)$$

$$Q = \vec{H}^2, K = \text{Emb}_n \quad (4)$$

$$\hat{y}_i^{open} = \text{argmax}(\text{softmax}(A^2(C_i^{open})))$$

$$\hat{y}_i^{close} = \text{argmax}(\text{softmax}(A^2(C_i^{close})))$$

3.1.2 멘션 탐지 학습 방법

본 논문의 멘션 탐지 모델은 여는 괄호와 닫는 괄호를 학습하여 예측된 괄호와 정답 괄호 간의 크로스 엔트로피(cross-entropy)를 최소화하는 방향으로 학습하며, 학습마다 여는 괄호와 닫는 괄호 예측 비용 함수를 각각 0.5 만큼 반영되도록 비율을 설정한다. 각 수식은 다음과 같다.

$$\text{loss}_{open} = - \sum_i \hat{y}_i^{open} \log(y_i^{open})$$

$$\text{loss}_{close} = - \sum_i \hat{y}_i^{close} \log(y_i^{close}) \quad (5)$$

$$\text{loss}_{total} = 0.5 * \text{loss}_{open} + 0.5 * \text{loss}_{close}$$

괄호 예측 계층에서는 여는 괄호와 닫는 괄호 수가 다르게 예측될 수 있으며, 이는 여는 괄호를 기준으로 멘션을 추출한다.

3.2 상호참조해결

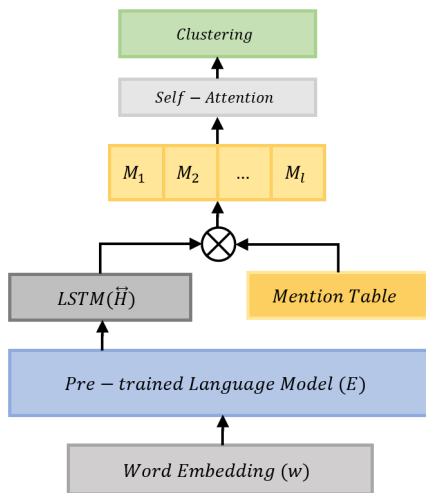


그림 2. 상호참조해결 모델 구조도

그림 2는 본 논문에서 제안하는 상호참조해결을 수행하는 모델의 전체 구조도이며 크게 사전 학습 언어 모델, 멘션 표현 벡터 생성 계층, 상호참조해결 계층으로 구성되어 있다. 사전 학습 언어 모델에서는 문서를 입력받아 단어 표현 벡터를 생성하고, 멘션 표현 생성 계층에서는

생성된 문맥 정보가 반영된 단어 표현 벡터와 멘션 탐지 모델에서 추출된 멘션을 이용하여 멘션 표현 벡터를 생성한다. 마지막으로 상호참조해결 계층에서는 멘션 표현 벡터를 입력으로 셀프 어텐션을 통해 상호참조해결을 수행한다. 상호참조해결을 수행하기 위해서는 전체 발화를 입력으로 사용해야 하지만 사전 학습 언어 모델에는 입력 길이에 제한을 두기 때문에 최대 입력 길이 이상의 발화에 대해서는 겹치지 않도록 나누어 독립적으로 학습한다.

3.2.1 멘션 표현 생성 계층

멘션 표현 벡터를 생성하기 위해 멘션 집합을 오름차순으로 정렬한 후, 다음 수식과 같이 정렬된 멘션을 순차적으로 계산한다.

$$M_i = \frac{1}{N - k + 1} \sum_k^{\leftrightarrow^1} h_k$$

$$M = \{M_1, M_2, \dots, M_l\} \quad (6)$$

수식 (6)에서 \leftrightarrow^1 는 수식 (1)에서 계산된 문맥 정보가 반영된 단어 표현 벡터이다. 또한 수식 (6)에서 볼 수 있듯이 멘션 표현 벡터는 멘션을 이루고 있는 각 토큰 표현 벡터들의 평균이다. 멘션 표현 집합 M 에서 l 은 입력 멘션의 개수이다.

3.2.2 상호참조해결 계층

멘션 표현 벡터 사이의 연관성을 계산하기 위해 셀프 어텐션을 기반으로 한 포인터 네트워크를 이용하여 현재 멘션이 같은 개체로 참조할 가능성이 높은 후행 멘션들 중 가장 가까이에 위치하는 멘션을 가리키거나 같은 개체를 참조하는 멘션이 존재하지 않는 싱글톤(singleton)의 경우에는 자기 자신을 가리키도록 학습한다.

예를 들면, 멘션 M_1, M_2, M_3 이 같은 개체를 참조하는 경우, 제안하는 모델은 $[M_1, M_2, M_3]$ 을 $[M_2, M_3, M_3]$ 처럼 가장 가까이에 있는 후행 멘션을 가리키고, 더 이상 가리킬 멘션이 존재하지 않는 경우에는 자기 자신을 가리키도록 학습하며 수식은 다음과 같다.

$$A(M_i) = \frac{Q * K^T}{\sqrt{d_h}}$$

$$Q = M, K = M \quad (7)$$

$$\hat{y}_i^{coreference} = \text{argmax}(\text{softmax}(A(M_i)))$$

학습이 끝난 후, 같은 개체를 참조하는 멘션들을 순차적으로 군집화하는 후처리를 진행한다. 예를 들면, M_i 가 M_j 를 가리키는 경우에 두 멘션을 $[M_i, M_j]$ 로 군집화한다. 그리고 M_j 가 M_k 를 가리키는 경우에는 $[M_i, M_j, M_k]$ 으로 군집화하고, M_j 가 자기 자신을 가리키고 M_k 가 다른 멘션을 가리키는 경우에는 $[M_i, M_j], [M_k]$ 로 군집화한다.

3.2.3 상호참조해결 학습 방법

각 멘션에 대해 상호참조해결 모델을 학습하여 예측된 후행 멘션과 정답 후행 멘션 간의 크로스 엔트로피 (cross-entropy)를 최소화하도록 학습하며 수식은 다음과 같다.

$$loss_{coref} = - \sum_i \hat{y}_i^{coref} \log(y_i^{coref}) \quad (8)$$

한 문서를 여러 세그먼트로 나누어 학습하기 때문에 각 세그먼트를 독립적으로 군집화한다. 예를 들면, 문서 A 를 a, b, c 로 나누었다면 b 는 a 와 겹치는 엔티티가 없도록 군집화를 진행하고, c 는 a, b 와 겹치는 엔티티가 없도록 군집화를 진행한다.

4. 실험 및 결과

4.1 데이터셋 및 전처리

본 논문에서는 학습을 위해 CRAC 2018/2021 shared-task의 ARRAU 데이터셋을 사용하며 이 중에서 4개의 학습 데이터셋(Trains-93, PEAR, RST, GNOME)과 1개의 개발 데이터셋(Trains-91)을 사용한다. 또한 CODI-CRAC 2021 shared-task의 데이터셋을 사용하며 Light, AMI, Persuasion(PSUA), Switchboard(SWBD)의 개발 데이터셋, 평가 데이터셋을 사용한다. RST와 GNOME을 제외한 모든 학습, 개발 및 평가 데이터셋은 대화 도메인(domain)의 데이터셋이며 Universal Anaphora[9] 주석 형태로 구성되어 있다.

평가 데이터셋으로 사용한 4가지 도메인의 대화 데이터셋은 다음과 같다.

Light: 공상 세계를 가정한 텍스트 모험 게임을 바탕으로 인물 중심에서의 행동, 감정 등을 나타내는 대화를 포함하는 데이터셋[10]이다.

AMI: 여러 대화자가 회의하며 연설하는 사무실 회의록 데이터셋[11]이다. 네 가지 평가 데이터셋 중에 가장 긴 대화를 포함하는 데이터셋이다.

PSUA: 설득자가 자선 단체에 기부하도록 설득하는 대화를 모아 놓은 클라우드소싱 온라인 대화 데이터셋[12]이다.

SWBD: 두 참가자 사이에 아동 교육, 재활용 및 뉴스 미디어와 같이 다양한 주제에 대해 녹음된 전화 통화 데이터셋[13]이다.

표 1의 첫번째 행에서 문서는 총 문서 수이고, 문장은 총 문장 수, 개체는 총 개체 수, 멘션은 총 멘션 수이다. 하나의 문서에 포함되는 평균 문장 수는 총 문장 수를 총 문서 수로 나누면 알 수 있다. 전체 평가 데이터셋 중에서 AMI 평가 데이터셋은 한 문서에 포함되는 문장 수가 가장 많으며, SWBD 평가 데이터셋은 가장 많은 문장 수, 개체 수, 멘션 수를 포함한다.

표 1. 전체 데이터셋 통계

	데이터셋	문서	문장	개체	멘션
	Trains-93	98	7,296	4,523	12,148
학습	PEAR	20	1,973	1,168	3,401
	RST	413	8,735	38,724	62,409
	GNOME	5	982	2,598	5,499
개발	Light	20	908	1,357	3,448
	AMI	7	4,139	3,120	7,441
	PSUA	21	812	1,273	2,437
	SWBD	11	718	1,771	3,421
	Trains-91	16	3,420	828	2,501
평가	Light	21	923	1,359	3,501
	AMI	3	1,967	1,883	4,139
	PSUA	28	1,139	1,857	3,446
	SWBD	22	3,652	3,897	7,847

4.2 평가 방법

멘션 탐지 모델의 평가는 정확률(precision)과 재현율(recall)의 조화평균인 F1-score를 사용하여 성능을 측정하며, 평가 방법의 수식은 다음과 같다.

$$\begin{aligned} \text{정확률} &= \frac{\text{정답 멘션과 일치하는 예측 멘션 수}}{\text{예측 멘션 수}} \\ \text{재현율} &= \frac{\text{정답 멘션과 일치하는 예측 멘션 수}}{\text{정답 멘션 수}} \\ \text{F1-score} &= \frac{2 * \text{정확률} * \text{재현율}}{\text{정확률} + \text{재현율}} \end{aligned} \quad (9)$$

상호 참조 모델의 평가를 위해 *SemEval* 평가 프로그램[14]을 이용하며 링크 기반 방법인 *MUC*[15]와 멘션 기반 방법인 B^3 [16], 개체 기반 방법인 $CEAF_e$ [17]의 성능 지표와 세 가지 성능 지표를 평균으로 한 값인 CoNLL F1 score[18]로 상호참조해결 성능을 측정한다. CoNLL F1 score는 CoNLL-2011에서 공식으로 지정하여 측정하는 성능 지표이다.

4.3 멘션 탐지 실험 및 결과

표 2는 멘션 탐지 모델에서의 4가지 도메인의 평가 데이터셋 성능이다. Light 평가 데이터셋에서 F1-score 94.17%, AMI 평가 데이터셋에서 90.86%, PSUA 평가 데이터셋에서 92.93%, SWBD 평가 데이터셋에서 91.04%의 성능을 보인다.

표 2. 평가 데이터셋에서 멘션 탐지 성능

데이터셋	정확률	재현율	F1
Light	95.4	93.0	94.2
AMI	92.1	89.6	90.9
PSUA	92.9	93.0	92.9
SWBD	92.7	89.4	91.0

표 3의 [19]-1과 [19]-2는 CODICRAC2021 shared-task에서 다른 팀들의 상호참조해결 모델에서 보고된 멘션 탐지 성능이다. [19]-1은 BERT 기반 end-to-end 상호참조해결 모델[3]이며 멘션 순위화 방법을 이용하여 멘션 탐지를 수행하고, [19]-2는 [4,20]의 모델을 개선한 스캔 기반 멘션 탐지 모델이다. 본 논문에서 제안하는 시퀀스 레이블링 기반 멘션 탐지 모델의 정확률과 재현율은 모두 비슷한 성능을 보이며, [19]-1,2와 비교했을 때, 모든 평가 데이터셋에서 F1-score는 가장 높은 성능을 보인다.

4.4 상호참조해결 실험 및 결과

표 4는 상호참조해결 모델에서의 평가 데이터셋 성능이다. Light 평가 데이터셋은 CoNLL-score 69.1%, AMI 평가 데이터셋은 57.6%, PSUA 평가 데이터셋은 71.0%, SWBD 평가 데이터셋은 65.7%의 성능을 보인다.

표 3. 평가 데이터셋에서 멘션 탐지 성능 비교

	Light			AMI			PSUA			SWBD		
	정확률	재현율	F1	정확률	재현율	F1	정확률	재현율	F1	정확률	재현율	F1
[19]-1	90.3	40.6	56.1	97.1	60.0	74.2	94.9	56.8	71.1	92.0	54.0	68.1
[19]-2	92.3	91.6	92.0	86.6	78.6	82.4	91.3	89.7	90.5	89.2	86.1	87.6
ours	95.4	93.0	94.2	92.1	89.6	90.9	92.9	92.9	92.9	92.7	89.4	91.1

표 4. 평가 데이터셋에서 상호참조해결 성능

	MUC			B ³			CEAF _e			CoNLL
	정확률	재현율	F1	정확률	재현율	F1	정확률	재현율	F1	
Light	89.3	76.1	82.1	64.9	56.1	60.2	55.7	78.4	65.1	69.1
AMI	69.0	53.5	60.3	63.6	49.8	55.9	46.5	72.4	56.6	57.6
PSUA	76.5	75.4	77.3	65.6	72.6	68.9	61.9	73.1	67.0	71.0
SWBD	80.9	66.7	73.1	66.2	58.7	62.2	52.0	75.8	61.7	65.7

5. 결론

본 논문에서 제안하는 시퀀스 레이블링을 이용한 멘션 탐지 모델은 높은 성능을 보이며, 포인터 네트워크를 이용한 상호참조해결 모델의 성능은 CODICRAC2021 shared-task에서 최종 3위를 기록했다. 향후 연구로 기존의 멘션 탐지와 상호참조해결 파이프라인 모델을 end-to-end 상호참조해결 모델로 구현할 예정이며, 멘션 탐지와 상호참조해결에 의존 구조와 개체명 인식을 추가하는 실험을 진행할 예정이다.

감사의 글

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2020R1F1A1A1069737). 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학ICT연구센터육성지원사업의 연구결과로 수행되었음 (IITP-2021-2016-0-00465)

참고문헌

- [1] 정석원, 최맹식, 김학수, "랜덤 포레스트를 이용한 한국어 상호참조 해결", 정보처리학회논문지. 소프트웨어 및 데이터 공학, 5.11 pp.535-540, 2016.
- [2] O. Vinyals, et al, Pointer Networks, Advances in Neural Information Processing Systems, pp. 2674-2682, 2015.

- [3] K. Lee, L. He, M. Lewis, L. Zettlemoyer, End-to-end Neural Coreference Resolution, Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 188-197, 2017.
- [4] K. Lee, L. He, L. Zettlemoyer, Higher-Order Coreference Resolution with Coarse-to-Fine Inference, Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pp. 687-692, 2018.
- [5] S. Hochreiter, et al, Long Short-term Memory, Neural computation 9(8), pp. 1735-1780, 1997.
- [6] Lesly Miculicich, James Henderson, Partially-supervised Mention Detection, Proceedings of the 3rd Workshop on Computational Models of Reference, Anaphora and Coreference (CRAC 2020), pages 91-98, 2020.
- [7] K. Clark, et al, ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators, International Conference on Learning Representations (ICLR), 2019.
- [8] C. Leyang and Y. Zhang, Hierarchically-Refined Label Attention Network for Sequence Labeling, Empirical Methods in Natural Language Processing (EMNLP) and International Joint Conference on Natural Language Processing (IJCNLP), pp. 4106-4119, 2019.
- [9] Massimo Poesio, et al, Learning to resolve bridging references, In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, pages 143-150, 2004.
- [10] Jack Urbanek, et al, Learning to speak and act in a fantasy text adventure game, In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, pages 673-683, 2019.
- [11] Jean Carletta, Announcing the ami meeting corpus, The ELRA Newsletter, 11(1):3-5, 2006.
- [12] Xuwei Wang, et al, Persuasion for good: Towards a personalized persuasive dialogue system for social good, In Proceedings of the 57th Conference of the Association for Computational Linguistics, pages 5635-5649, 2019.
- [13] John J, et al, SWITCHBOARD: telephone speech corpus for research and development, In 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing, pages 517-520, 1992.
- [14] Lluís Màrquez, et al, Available: <http://stel.ub.edu/semEval2010-coref/>
- [15] Marc Vilain, et al, A model-theoretic coreference scoring scheme, In: Proceedings of the 6th conference on Message understanding. Association for Computational Linguistics, pp. 45-52, 1995.
- [16] Amit Bagga, et al. Algorithms for scoring coreference chains, In: The first international conference on language resources and evaluation workshop on linguistics coreference. Vol. 1. pp. 563-566, 1998.
- [17] Xiaoqiang Luo, On coreference resolution performance metrics., In: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing. Association for Computational Linguistics, pp. 25-32, 2005.
- [18] Sameer Pradhan, et al, Conll-2011 shared task: Modeling unrestricted coreference in ontonotes, In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task. Association for Computational Linguistics, pp. 1-27, 2011.
- [19] <https://competitions.codalab.org/competitions/30312>
- [20] Liyan Xu, et al, Revealing the myth of higher-order inference in coreference resolution, In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 8527-8533, 2020.