

언급 특질을 이용한 Bi-LSTM 기반

한국어 상호참조해결 종단간 학습

신기연[○], 한기종, 이민호, 김건태, 최기선

한국과학기술원

{nuclear852, han0ah, pathmaker, kuntaek, kschoi}@kaist.ac.kr

Korean Co-reference Resolution End-to-End Learning using Bi-LSTM with Mention Features

Giyeon Shin[○], Kijong Han, Minho Lee, Kuntaek Kim, Key-Sun Choi
KAIST

요 약

상호참조해결은 자연언어 문서 내에서 등장하는 명사구 언급(mention)과 이에 선행하는 명사구 언급을 찾아 같은 개체인지 정의하는 문제이다. 특히, 지식베이스 확장에 있어 상호참조해결은 언급 후보에 대해 선행하는 개체의 언급이 있는지 판단해 지식트리플 획득에 도움을 준다. 영어권 상호참조해결에서는 F1 score 73%를 웃도는 좋은 성능을 내고 있으나, 평균 정밀도가 80%로 지식트리플 추출에 적용하기에는 우리가 있다. 따라서 본 논문에서는 한국어 문서에 대해 영어권 상호참조해결 모델에서 사용되었던 최신 모델인 Bi-LSTM 기반의 딥 러닝 기술을 구현하고 이에 더해 언급 후보 목록을 만들어 개체명 유형과 경계를 적용하였으며 품사형태를 붙인 토큰을 사용하였다. 실험 결과, 문자 임베딩(Character Embedding) 값을 사용한 경우 CoNLL F1-Score 63.25%를 기록하였고, 85.67%의 정밀도를 보였으며, 같은 모델에 문자 임베딩을 사용하지 않은 경우 CoNLL F1-Score 67.92%와 평균 정밀도 77.71%를 보였다.

주제어: 상호참조해결, Bi-LSTM, 언급(mention)

1. 서론

상호참조해결이란 자연언어로 이루어진 문서에서 나타나는 명사구들에 대해 서로 같은 개체를 뜻하는 언급(mention)들을 찾는 작업을 말한다. 자연언어 문서에서 명사구들이 같은 개체를 나타내지만 축약어나 대명사, 지시 관형사등과 같이 다른 형태로 등장하는 경우가 있기 때문에 명사구 간의 상호참조 관계를 밝혀 문서요약, 질의응답과 같은 자연언어처리 문제에 사용할 수 있다.

또한, 상호참조해결로 언급의 개체명이 지식베이스에서 명확해져 지식추출 성능 향상에 기여한다[1]. 예를 들어 ‘버락 오바마는 미국의 대통령이다. 그의 출생지는 미국 하와이주 호놀룰루이다.’ 라는 문장에서, 개체명 인식을 통해 <버락_오바마, isA, 미국의_대통령>과 같은 지식 삼항관계만을 획득할 수 있지만, ‘그’ 라는 대명사가 [버락_오바마]라는 개체로서 참조 문제가 해결되면 <버락_오바마, birth_place, 미국_하와이주_호놀룰루>에 대한 추가적인 지식트리플을 획득할 수 있다.

지식베이스 상에서 이러한 지식트리플 추출[1]을 위해 먼저, 개체명 인식(Named Entity Recognition)을 통해 자연 언어 문서에서 나타날 수 있는 개체들을 탐지한다. 이 단계를 거치면 주어진 자연 언어 문서에서 개체명을 가질 수 있는 언급 후보와 이 후보에 대한 개체명의 유형이 주어진다. 이 후, 대명사 검출기를 통해 나온 대명사와 지시 관형사로 시작하는 명사구들과 개체명 후보를 포함하는 언급 후보 목록에 대하여 상호참조해결을 실시

한다.

기존의 영어권 상호참조해결 모델[2,3]에서는 모든 언급들을 대상으로 상호참조해결을 진행하여 CoNLL F1-score 73%의 좋은 성능을 보여주었다.[3] 하지만 이는 정밀도가 80%로 매우 높은 수준이 아니기 때문에 지식트리플 추출 시에 노이즈 데이터가 생성될 수 있어 바로 적용하기에 무리가 있다.

본 논문에서는 입력 데이터가 개체명 인식이 이루어진 데이터로 가정하여 상호참조해결을 하고자 한다. 즉, 데이터 집합에 개체명을 가질 수 있는 개체 후보 목록과 개체명 유형이 주어져있다. 여기에 재현율 96%의 규칙 기반 대명사 검출기를 이용하여 추출된 대명사와 지시 관형사로 시작하는 명사구를 추가해 정답 언급 후보 목록을 만들었다.

위와 같이 정의된 문제 범위에서 영어 상호참조해결 모델을 한국어에 적용하였으며, 다음의 추가 작업을 통해 성능을 확인하였다.

- 1) 개체 후보 목록에 대명사 검출기를 통해 추출된 대명사와 지시 관형사가 포함된 명사구 추가해 정답 언급 후보 목록(Gold Mention Candidate List) 생성
- 2) 형태소 분석기를 이용하여 (단어/품사) 토큰 사용
- 3) 해당 모델에 해당 정답 언급 후보 목록의 경계 정보 추가
- 4) 특질 벡터에 언급의 개체명 유형 추가

2. 관련 연구

기존의 한국어 상호참조해결 문제에서는 규칙 기반 모델로 좋은 성능을 보여준 연구가 있다. 의존구문트리를 이용하여 언급(mention) 추출을 진행하고 다 단계 체 (Multi-pass Sieve) 시스템을 적용하여 CoNLL F1-Score 약 60%의 좋은 성능을 보여주었다. 좋은 재현율을 보여 주지만 정밀도가 약 60%로 낮다는 단점이 있다.[4]

최근 자연언어처리 연구에서는 규칙 기반의 어려움을 해결하기 위해 딥러닝 기술을 도입하고 있고 실제로 좋은 성능을 보여준다. 특히, 단어를 임의의 차원으로 연결한 Word2Vec나 ELMo와 같은 단어 임베딩 개발로 자연어처리를 위한 딥러닝 아키텍처의 성능을 높이고 있다.[5,6]

특히 상호참조해결 문제에서 Bi-RNN (Bidirectional Recurrent Neural Network) 아키텍처가 효과적이다.[7] Bi-RNN은 입력 문자열을 RNN을 통해 양방향, 순차적으로 처리하여 문맥상에서 나타나는 입력 토큰의 성질을 파악할 수 있는 딥 뉴럴 아키텍처이다. 특히 RNN 중에서 GRU (Gated Recurrent Unit)[8]나 LSTM (Long-Short Term Memory)[9]은 RNN의 기본 구조에서 망각 게이트(forget gate)를 추가하여, 긴 문서 단위의 입력 데이터를 처리할 시에 발생할 수 있는 Vanishing Gradient 문제를 해결하였다.

딥러닝을 이용한 한국어 상호참조해결에서 이러한 특징을 가지는 Bi-GRU 기반의 연구[10]가 이루어졌으며, Bi-GRU를 통해 압축된 벡터값에 어텐션 기반 포인터 네트워크를 더해 중심어 기반 성능에서 CoNLL F1-Score 약 72%를 기록하였다. 이 때 입력 데이터에 형태소 분석을 통해 명사만 골라 명사 목록을 추가하였으며 명사 목록에 대해 어텐션 메커니즘 기법을 사용하였다. 하지만 언급의 중심어에 대해 선행 여부를 판단하기 때문에 언급의 경계를 알 수 없어 지식트리플을 추출하는 데 어려움이 있다. Bi-LSTM을 기반으로 영어권에서도 상호참조해결의 연구[2][3]가 이루어졌으며, 종단간 딥뉴럴 구조로 최고 성능인 F1-score 약 73%를 기록하였다.[3] 이는 언급 탐지와 상호참조해결을 동시에 해결한 아키텍처로 문서의 모든 단어들을 대상으로 진행하였으며, 약 73.3%의 정밀도를 보여주었다. 하지만 모든 단어들을 대상으로 상호참조해결이 진행되기 때문에 발생하는 낮은 정밀도로 지식트리플 추출에 적용하기에는 어려움이 있다.

따라서 본 논문에서는 입력데이터의 개체 후보 목록에 대명사 검출기를 통해 추출된 대명사와 지시 관형사로 시작하는 명사구 후보군을 추가하여 정답 언급 후보 목록을 만들고 해당 집합의 경계를 이용하여 앞서 설명한 Bi-LSTM을 이용한 모델[3]에 적용하였으며, 언급의 개체명 유형을 비교하는 특질의 추가와 (단어)/(품사) 형태의 토큰을 사용하였다.

3. 접근 방법

그림 1은 본 논문의 상호참조해결 모델의 전체적인 흐름도를 보여준다.

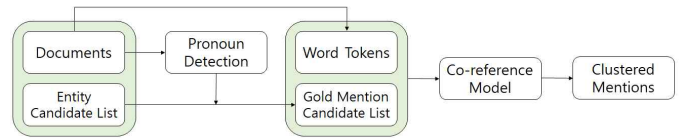


그림 1. 상호참조모델의 전체적인 흐름도

본 논문의 입력 데이터는 다음을 가정한다: 모든 단어들의 품사 주석, 개체 후보 목록 (명사구로 언급된), 개체 후보의 개체명 유형. 이 때, 개체의 기준은 지식베이스에 등록할 수 있는 고유 명사구 개체와 시간 개체이다.

3.1 대명사 추출

개체 후보 목록이 주어진 입력 텍스트의 예시는 그림 2와 같다.

[1413년]에 [마리]는 [샤를 6세]와 [바이에른의 이자보]의 다섯 번째 아들인 [샤를 7세]와 약혼을 했다. [[1422년] [4월]]에 그녀는 [부르주]에서 그녀의 사촌이었던 [샤를]과 혼인하였고 이후 [프랑스]의 왕비가 되었다.

그림 2. 개체 후보가 주석된 입력 문장의 예시

해당 개체 후보들은 개체명을 가질 수 있는 명사구에 대하여 태깅되어 있다. 하지만 대명사나 지시 관형사가 포함된 명사구가 태깅되어 있지 않은 상태이다.

이후 규칙기반의 대명사 검출기를 이용하여 대명사와 지시 관형사가 포함된 명사구를 추출하여 그림 3과 같이 언급의 경계가 포함된 형태로 변환하였으며, 해당 명사구들을 개체 후보 목록에 포함시켜 상호참조용 언급 후보 목록을 생성하였다.

[1413년]에 [마리]는 [샤를 6세]와 [바이에른의 이자보]의 다섯 번째 아들인 [샤를 7세]와 약혼을 했다. [[1422년] [4월]]에 [그녀]는 [부르주]에서 [[그녀]의 사촌]이었던 [샤를]과 혼인하였고 이후 [프랑스]의 왕비가 되었다.

그림 3. 대명사 검출기를 통해 변환된 입력 문장의 예시

본 논문에서 사용된 규칙기반의 대명사 검출기에서는 대명사 및 지시 관형사로 시작하는 한정사구 추출을 위해 [4]의 언급 탐지 기법을 이용하여 후보 명사구들을 추출한 후 세종전자사전에 명시된 대명사 및 지시 관형사로 시작하는 명사구만을 선택하는 방식으로 진행되었다.

3.2 벡터 임베딩

본 논문에서는 문서의 토큰화 과정에서 형태소 분석기를 이용하여 각 토큰의 결과값을 (과천시/NNP)와 같이 (단어/품사)의 형태로 만들었으며, 미리 학습된 Word2Vec[5], ELMo[6] 임베딩 값 그리고 문자 임베딩 값을 묶어 사용하였다.

Word2Vec 임베딩 값은 한국어 위키백과 덤프 파일을 사용하여 차원은 300차원, 창 크기는 10으로 설정하여 생성되었다. 상호참조해결 문제에서는 명사구만을 대상으로 삼기 때문에 품사의 형태가 중요하다. 따라서 본 논문에서는 형태소 분석기를 통해 나온 품사의 형태를 같이 넣어준 토큰의 값을 사용하였으며, 이를 Word2Vec으로 임베딩 값을 생성하였다. ELMo 임베딩 값은 본 논문의 실험에서 사용된 학습 데이터와 개발 데이터를 이용해 생성하였으며 이를 Word2Vec 임베딩값과 묶어 문서의 토큰 임베딩값으로 사용하였다. 또한 문자 임베딩(Character Embedding)을 사용하기 위해 (단어/품사) 형태에서 단어 부분을 따로 추출하여 문자 임베딩을 학습하였으며, 문자 임베딩 값이 사용되지 않은 모델과 함께 비교해보았다.

3.3 정답 언급 후보 목록 경계 적용

상호참조해결 모델에서의 언급 탐지는 문서상에서 나타나는 연속적인 단어가 실제 언급인지에 대해 학습한다. 해당 언급 탐지의 기능이 본 논문에서 정의한 상호참조해결 문제에 필요한 이유는 대명사 추출시에 나오는 명사구들의 후보군들 중 언급 대상이 아닌 후보들을 거르기 위함이다. [2,3]

여 정답 언급 후보 목록에 있지 않은 언급 후보들의 언급 점수는 모두 0으로 설정한다. 위와 같이 정답 언급 후보 집합 경계 정보를 이용한 까닭은 그림 4의 [예/JKB 마리/NNP]와 같이 연속적인 단어가 언급 대상이 아니거나 일반명사와 같이 상호참조의 대상이 아닐 경우를 제외시켜주기 위함이다.

이 후, 기존 Bi-LSTM 기반의 모델과 동일하게 다음과 같은 형태의 언급 후보 임베딩 벡터값을 생성한다. [2,3]

$$span_i = [x_{START(i)}^*, \hat{x}_i, x_{END(i)}^*, \theta(i)]$$

i번째 생성된 언급 후보의 벡터 값을 $span_i$ 라 하고, 언급 후보의 시작과 끝을 이루는 토큰의 Bi-LSTM 임베딩 값을 각각 $x_{START(i)}^*, x_{END(i)}^*$ 라 한다. 해당 언급 후보에서의 언급 크기에 대한 특질 값은 $\theta(i)$ 이며, 어텐션 메커니즘을 이용한 값은 \hat{x}_i 이다. 이는 기존 Bi-LSTM 기반 논문에서 아래의 수식에 의해 유도된다. [2]

$$\begin{aligned} \alpha_t &= w_\alpha \cdot FFNN_\alpha(x_t^*) \\ a_{i,t} &= \frac{\exp(\alpha_t)}{\sum_{k=START(i)}^{END(i)} \exp(\alpha_k)} \\ \hat{x}_i &= \sum_{t=START(i)}^{END(i)} a_{i,t} \cdot x_t \end{aligned}$$

이 때, \hat{x}_i 는 $span_i$ 에 포함된 토큰들의 Bi-LSTM 임베딩 값을 일정 비율로 합친 값이다. [2]

3.4 개체명 유형 특질 사용

Bi-LSTM 기반 상호참조해결에서 i번째 생성된 언급 후보와 j번째 생성된 언급 후보간의 상호참조 점수 함수 $s(i,j)$ 는 다음과 같이 정의된다. [3]

$$s(i,j) = s_m(i) + s_m(j) + s_a(i,j) + s_c(i,j)$$

이 때, 언급 후보 점수 $s_m(i)$ 와 $s_m(j)$ 는 $span_i$ 와 $span_j$ 값에 대해 다음과 같이 정의된다. [3]

$$s_m(i) = w_m \cdot FFNN_M(span_i)$$

$s_c(i,j)$ 와 $s_a(i,j)$ 는 j번째 언급이 i번째 언급의 선행사인지에 대해 계산하는 점수값이다. $s_c(i,j)$ 는 다음의 수식과 같이 간단하고 빠르게 계산되는 점수 값이다. [3]

$$s_c(i,j) = span_i^T W_c span_j$$

이는 빠르게 계산된 $s_m(i) + s_m(j) + s_c(i,j)$ 점수를 통해 i번째 언급 후보와 가장 높은 점수를 가진 선행사 후보 K개를 뽑아 계산량을 줄이기 위함이다. 이 후, 선행사 후보 K개에 대해 정확한 상호참조 점수를 구하기 위해 다음과 같은 $s_a(i,j)$ 의 값을 구한다. [3]

$$s_a(i,j) = w_a^T FFNN_a([span_i, span_j, span_i \circ span_j, \sigma(i,j)])$$

$\sigma(i,j)$ 는 i번째 언급 후보와 j번째 언급 후보간의 특질 벡터값을 의미한다. 본 논문에서는 이 특질 벡터값에 개체명 유형의 정보를 추가하였다.

개체명 유형의 정보는 다음과 같은 종류가 있다: ① 인물, ② 시간, ③ 장소, ④ 기관, ⑤ 사건, ⑥ 기타.

위와 같이 개체명의 유형에 따라 원-핫 벡터 값을 사용하였고, i번째 언급 후보와 j번째 언급 후보의 개체명 유형을 다음과 같은 경우로 비교하였다.

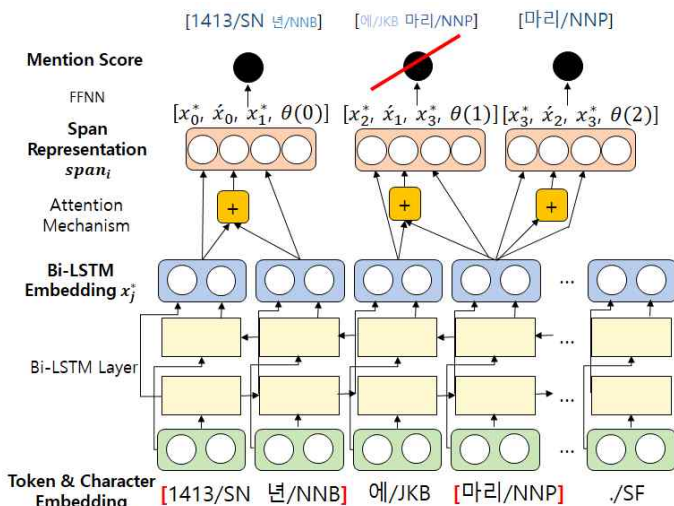


그림 4. Bi-LSTM 기반 언급 점수 계산 [2]

기존 Bi-LSTM 기반 상호참조해결 모델[3]에서의 언급 탐지 첫 번째 과정은 모든 언급 후보들에 대하여 Bi-LSTM의 임베딩 표현을 학습시키는 것이다. Bi-LSTM은 문서에서의 문맥을 고려한 임베딩 벡터 값을 생성하기 위해 사용되었다. 이 때, 본 논문에서는 3.1에서 앞서 만들어 놓은 정답 언급 후보 목록에서의 경계를 이용한

표 1. 정답 언급 후보 목록 특질 추가의 성능 변화 (문자 임베딩 값 포함)

단위 (%)	MUC F1	B^3 F1	CEAF-e F1	Average		
특질 사용				정밀도	재현율	F1 (CoNLL)
베이스라인 [3]	54.19	47.09	44.21	71.52	36.58	48.50
+ 정답 언급 후보 경계	64.67	59.91	56.75	76.18	50.22	60.44
+ 정답 언급 후보 경계, 개체 유형	69.21	62.27	58.26	85.67	50.19	63.25

(1) i 번째 언급 후보와 j 번째 언급 후보의 개체명 유형이 같은 경우

(2) i 번째 언급 후보와 j 번째 언급 후보의 개체명 유형이 다른 경우

(3) i 번째 언급 후보나 j 번째 언급 후보의 개체명 유형 정보가 없는 경우 (언급 후보가 대명사나 지시관형사와 같은 경우)

위의 경우에 따라 i 번째 언급 후보와 j 번째 언급 후보간 개체명 유형 특질 벡터값을 원-핫 벡터로 만들고 해당 원-핫 벡터를 특질 벡터 크기에 맞추어 $\sigma(i, j)$ 에 추가하였다. 위와 같이 개체명 유형 정보를 사용한 이유는, 다른 개체명 유형 정보를 갖고 있을 경우 두 언급 후보가 서로 상호참조할 수 없기 때문이다. 대명사나 지시관형사의 경우 유형 정보를 갖고 있지 않기 때문에 (3)의 경우로 분류하였다.

4. 성능 평가

4.1 데이터집합

데이터 집합은 위키피디아의 140개 문서를 대상으로 클라우드소싱을 통해 개체 후보 목록을 만든다. 또한, 각 개체 후보마다 지식베이스의 개체와 개체명 유형을 주석하였다. 그 이후 전문가 4명이 수동으로 대명사와 지시 관형사로 시작하는 명사구가 어떠한 개체를 참조하는지 주석하였다. 이 때 너무 긴 문서는 모델 학습 시에 자원의 소모가 너무 커서 반으로 잘라서 140개 위키피디아 문서를 총 207개의 문서 데이터 집합으로 나누었다. 해당하는 문서 데이터집합을 8:1:1로 나누어 실험의 학습 데이터 / 개발 데이터 / 테스트 데이터로 나누어 성능을 평가하였다. 단, 개체 경계 정보에서 대명사나 지시관형사로 시작하는 명사구의 정답 경계 정보는 제외하였다. 표 2는 실험 문서 데이터집합의 평균 구성에 대한

표이다.

	평균 토큰 갯수	평균 정답 언급 후보 갯수	평균 상호참조대상 언급 갯수
학습 데이터	433	156	19
개발 데이터	398	166	19
테스트 데이터	347	153	16

표 2. 실험 문서 데이터집합의 평균 구성

4.2 성능 평가 방식

본 논문의 상호참조해결 성능 평가를 위해 SemEval 평가 프로그램을 사용하여, MUC, B^3 , CEAF-e의 성능 지표로 성능을 평가하고 해당 성능 지표 값을 평균내어 CoNLL F1-Score값과 정밀도, 재현율을 측정하였다. 이는 CoNLL-2011에서 공식 지정한 성능 지표 값이다.[11] 앞서 언급한 데이터집합에 대해 성능 지표 값을 측정하는 테스트 작업을 진행하였다.

4.3 상호참조해결 성능평가 결과

표 1은 문자 임베딩(Character Embedding)을 사용하고, 정답 언급 후보 목록에서 나타나는 언급 경계 정보와 개체명 유형 특질 추가에 따른 결과값이다. 정답 언급 후보 목록에서 나타나지 않은 언급에 대해서 언급 점수를 0으로 준 결과, 전반적으로 베이스라인 모델[3]에 비해 F1-Score가 12%가 상승하며 좋아진 것을 확인할 수 있었다. 특히 재현율 부분에 있어 성능이 약 14%로 상승한 것을 확인할 수 있었다.

정밀도 향상을 위해 개체명 유형에 따른 특질 벡터를 추가한 결과, 정밀도가 약 9.5% 상승하였으며, CoNLL F1-Score 값이 약 2.8% 상승하였다. 재현율은 올라가지 않고 오히려 0.03% 가량 하락하는 결과를 보여주었다.

표 3. 정답 언급 후보 목록 특질 추가의 성능 변화 (문자 임베딩 값 제거)

단위 (%)	MUC F1	B^3 F1	CEAF-e F1	Average		
특질 사용				정밀도	재현율	F1 (CoNLL)
베이스라인 [3]	63.95	57.06	53.94	81.21	45.67	58.32
+ 정답 언급 후보 경계	69.01	64.32	63.45	83.7	54.03	65.59
+ 정답 언급 후보 경계, 개체 유형	71.14	67.01	65.62	77.71	60.44	67.92

표 3은 문자 임베딩(Character Embedding)을 사용하지 않고, 정답 언급 후보 목록에서 나타나는 언급 경계 정보와 개체명 유형 특질 추가에 따른 결과값이다. 문자 임베딩 값을 사용하지 않았을 때, 문자 임베딩 값을 사용했을 때보다 F1-Score가 약 4.6% 상승하였다. 특히 재현율이 10.3% 가량 상승하여 더 좋은 성능을 보여주었다. 정밀도에서는 문자 임베딩 값을 사용했을 때, 문자 임베딩을 사용하지 않았을 때보다 약 2% 더 좋은 성능을 보여주었다.

단위 (%)	Average		
토큰 형태	정밀도	재현율	F1 (CoNLL)
(단어/품사)	85.67	50.19	63.25
(단어)	77.08	41.99	54.31

표 4. 토큰 임베딩의 형태에 따른 성능 변화

표 4는 토큰 임베딩의 형태에 따른 성능 변화를 측정한 결과이다. 문자 임베딩 값을 포함하고 정답 언급 후보의 경계와 개체 유형이 포함되었을 때의 모델을 기준으로 성능을 측정하였으며, (단어/품사) 기반의 토큰 형태가 약 F1-Score 8% 정도 성능이 더 좋은 것을 확인하였다.

5. 결론

본 논문에서는 영어 상호참조 모델 [2,3]을 한국어에 적용할 경우, 고려하여야 할 점을 실험적으로 분석하였다. 본 논문에서 추가한 점은 (단어/품사) 형태의 토큰을 사용했으며, 언급 후보 목록의 경계 정보와 개체명 유형 특질 벡터를 추가하였다. 기존 모든 단어들을 대상으로 하는 규칙 기반의 한국어 상호참조해결[4]보다 좋은 성능을 보였으며, 기존 언급의 중심어 기반으로 학습하는 한국어 상호참조해결 모델[10]과는 달리 선행하는 언급의 경계에 대해 학습하였다.

문자 임베딩 벡터를 사용할 경우 정밀도가 더 좋았으나, 사용하지 않을 경우 F1-Score가 약 4.6% 높았다. 이는 문자 임베딩 값이 학습 데이터와 검증 데이터를 통해 생성되는데 “가” 부터 “힉” 까지 모든 한국어 문자에 대해 학습되지 않기 때문이다. 따라서 한국어 문자 임베딩 값을 사용했을 때, 사용하지 않을 때보다 정밀도에서 좋은 성능을 보여주었으나, 한국어 문자 임베딩 값을 사용하지 않았을 때 재현율과 F1-Score가 더 좋은 성능을 보여주었다.

사사

이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (2013-0-00109, WiseKB: 빅데이터 이해 기반 자가학습형 지식베이스 및 추론 기술 개발)

참고문헌

- [1] 김지호, et al., 지식베이스 확장을 위한 행렬 분해 모델, 제 29회 한글 및 한국어 정보처리 학술대회 논문집, pp. 3-7, 2017.
- [2] Lee, K., et al., End-to-End Neural Coreference Resolution, Proceedings of the 2017 Coreference on Empirical Methods in Natural Language Processing, pp. 188-197, 2017.
- [3] Lee, K., He, L., & Zettlemoyer, L., Higher-order Coreference Resolution with Coarse-to-fine Inference, Proceedings of NAACL-HLT 2018, pp. 687-692, 2018.
- [4] 박천음, 최경호, 이창기, “Multi-pass Sieve를 이용한 한국어 상호참조해결”, 정보과학회논문지, 41(11), pp. 992-1005, 2014.
- [5] Tomas Mikolov, et al., Distributed Representations of Words and Phrases and their Compositionality, Advances in Neural Information Processing Systems, 26, 2013.
- [6] Peters, Matthew E., et al., Deep contextualized word representations, NAACL, 2018.
- [7] M. Schuster and K. K. Paliwal, Bidirectional recurrent neural networks, IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673-2681, 1997.
- [8] Junyoung Chung, et al., Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling, NIPS 2014 Deep Learning and Representation Learning Workshop, 2014.
- [9] S. Hochreiter, et al., Long short-term memory, Neural Computation, 1997.
- [10] 박천음, 이창기, “포지션 인코딩 기반 스택 포인터 네트워크를 이용한 한국어 상호참조해결”, 정보과학회 컴퓨팅의 실제 논문지, 24.(3), pp.113-121, 2018.
- [11] S. Pradhan, et al., CoNLL-2011 shared task: Modeling unrestricted coreference in ontologies, Proc. of the Fifteenth Conference on Computational Natural Language Learning: Shared Task, Association for Computational Linguistics, pp. 1-27, 2011.