

# Dual Decomposition을 이용한 전이 기반 및 그래프 기반 의존 파서 통합 모델

민진우<sup>아</sup>, 나승훈<sup>†</sup>, 신종훈<sup>††</sup>, 김영길<sup>††</sup>

전북대학교<sup>†</sup>, ETRI<sup>††</sup>

Jinwoomin4488@gmail.com, nash@jbnu.ac.kr, jhsin82@etri.re.kr, kimyk@etri.re.kr

## Integrating Transition-based and Graph-based Dependency Parsers using Dual Decomposition

Jin-Woo Min<sup>아</sup>, Seung-Hoon Na<sup>†</sup>, Jong-Hun Sin<sup>††</sup>, Young-Kil Kim<sup>††</sup>  
Jeonbuk National University<sup>†</sup>, ETRI<sup>††</sup>

### 요약

딥러닝을 이용한 한국어 의존 파싱은 전이 기반 방식과 그래프 기반 방식으로 나누어 연구되어 왔다. 전이 기반 방식은 입력 버퍼와 스택으로부터 자질을 추출하여 모델을 통해 액션을 결정하고 액션에 따라 파스트리를 생성해 나가는 상황식(Bottom-Up)의 지역적 모델이고 그래프 기반 방식은 문장 내의 모든 단어에 대해 지배소, 의존소가 될 수 있는 점수를 딥러닝 모델을 통해 점수화하여 트리를 생성하는 전역적 모델이다. 본 논문에서는 Dual Decomposition을 이용하여 하이브리드 방식으로 전이 기반 파서와 그래프 기반 파서를 결합하는 방법을 제안하고 BERT 언어 모델을 반영하여 세종 데이터 셋에서 UAS 94.47%, LAS 92.58% 그리고 SPMRL '14 데이터 셋에서 UAS 94.74%, UAS 94.20%의 성능을 보여 기존 그래프 기반 파서의 성능을 더욱 개선하였다.

**주제어:** 의존 파싱, 전이 기반, 그래프 기반, 딥러닝, Dual Decomposition

### 1. 서론

딥러닝을 이용한 한국어 의존 파싱은 전이 기반 방식과 그래프 기반 방식의 두 패러다임으로 나누어 연구되고 있다. 전이 기반 방식은 입력 버퍼와 스택으로부터 자질을 추출하여 추출된 자질을 딥러닝 모델을 통해 다음 액션을 결정하고 결정된 액션에 따라 파스트리를 생성해 나가는 지역적 모델이고 그래프 기반 방식은 문장 내의 모든 단어에 대해 지배소(head), 의존소(modifier)가 될 수 있는 점수를 계산하기 위해 딥러닝 모델을 도입한다[9, 10].

전이 기반 방식은 스택과 버퍼의 지역적 정보에 의존하여 문장 내에서 서로 먼 거리의 단어 사이의 의존 관계를 분석하기 어려우며 초기의 잘못된 전이 액션이 전체 구문 분석 성능에 영향을 주어 성능이 하락하는 문제가 존재한다[11].

본 연구에서는 Dual Decomposition[4, 10, 16] 알고리즘을 이용하여 전이 기반 모델과 그래프 기반 모델을 하이브리드 방식으로 결합하는 방법을 제안하고 BERT 언어 모델을 반영하여 세종 구문 셋에서 UAS 94.47%, LAS 92.58%와 SPMRL '14 데이터 셋에서 UAS 94.74%, LAS 94.20%를 성능을 보여 최고 성능을 더욱 향상시켰다.

### 2. 관련 연구

그래프 기반 의존 파싱의 대표적인 모델은 Biaffine

어텐션 모델[3]로 인코딩된 문장에 대해 각각 지배소와 의존소에 대해 MLP를 적용하여 지배소에 대한 표상과 의존소에 대한 표상을 얻고 이를 Biaffine 어텐션을 통해 모든 단어 쌍의 의존소에 대한 지배소의 점수를 얻는 모델로 영문과 중국어 의존 파싱에 적용하여 높은 성능을 보였다. 한국어에서도 한국어에 맞게 단어 표상을 단어를 구성하는 형태소 혹은 음절들로 합성한 후 위와 동일하게 Biaffine 모델을 적용하여 한국어 의존 파싱에서 최고 성능을 보이고 있다[8, 13-14].

전이 기반 의존 파싱은 Arc-Eager[1], Arc-Standard[2] 방식이 대표적이며 전자는 버퍼의 Top 노드와 Top의 Top 노드 사이에서 의존성을 결정하고 후자는 스택의 Top 노드 사이에서 의존성을 결정하는 방식이다. Arc-Hybrid[4]는 Arc-Eager 방식과 Arc-Standard을 혼합한 방식이다. 전이 기반 한국어 의존 파싱은 초기에 [8]에서 Arc-Eager 방식으로 연구 수행되었으며 버퍼와 스택으로부터 자질을 얻고 딥러닝 모델인 Feed Forward Neural Networks(FNNs)으로 다음 액션을 결정하는 연구를 진행하였으며 [17]은 Arc-Standard 방식으로 버퍼, 스택, 액션 버퍼를 Stack LSTM을 통해 인코딩한 후 인코딩된 표상으로부터 자질을 얻어 LSTM을 통해 다음 액션을 결정하는 Stack LSTM 모델을 한국어에 적용하였다.

위에서 제시한 전이 기반 파싱은 모두 부분 트리로부터 최종 트리를 만들어나가는 Bottom-up 방식으로 [6]에서는 ROOT로부터 자식들을 포인터 네트워크를 이용하여 포인팅하며 트리를 생성하는 “Stack Pointer-Networks”

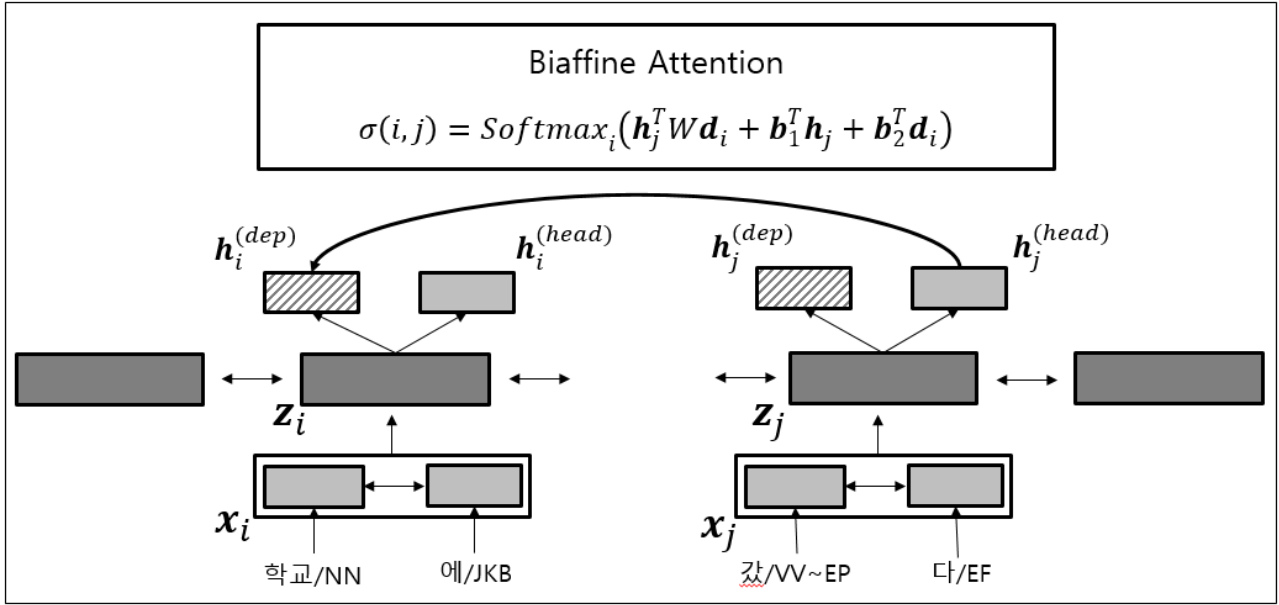


그림 1. Biaffine 한국어 의존 파싱 모델의 구조

라 불리는 새로운 Top-Down 방식의 전이 기반 모델을 제안하여 영문권에서 Biaffine 모델에 비하는 성능을 보였으며 한국어 의존 파싱에도 적용되어 기존의 성능을 더욱 향상시켰다[11]. 문장 내의 단어에 대하여 문맥을 고려한 대용량 코퍼스로부터 사전 학습된 언어 모델인 ELMo[12], BERT[7]를 이용하여 응용 태스크에서 미세조정 (fine-tuning)하는 최근 자연어 처리 분야에서 각광받고 있다. 대부분의 태스크에 높은 성능 향상을 보였으며 한국어 의존 파싱에서도 [12-13]에서는 ELMo, [13-14]에서는 한국어 BERT 모델 각각 적용하여 기존의 의존 파싱의 성능을 높이는 성과를 보였다.

Dual Decomposition은 주어진 최적화 문제를 풀기 쉬운 부분 문제들의 디코딩을 반복적으로 수행하면서 해를 구하는 방법으로 [17]과 [10]에서는 각각 기계학습 기반 의존 파싱과 딥러닝 기반 한국어 의존 파싱에서 그래프 기반 기존 모델과 2nd Order 모델을 Dual Decomposition으로 결합하여 성능 향상을 보였다. [4]는 양방향 전이 기반 파서를 통합 디코딩하는 연구를 진행하였는데 여기서 제안한 디코딩 방법은 1) Joint Scoring, 2) Dual Decomposition, 3) Dynamic Oracle 등을 3가지로 각각 방식이 단일 모델을 사용한 결과보다 높은 성능을 보였다.

### 3. 전이 · 그래프 기반 결합 모델

#### 3.1 인코더

한국어 의존 파싱은 문장의 단어(어절)단위로 수행되며 단어 임베딩을 직접 얻는 것은 UNK 문제에 취약하여 단어를 구성하고 있는 형태소를 Bi-LSTM 기반으로 합성하는 방식[9]을 사용하였다.

$$\mathbf{x}_i = [\mathbf{w}_i; \mathbf{t}_i; \mathbf{b}_i] \quad (1)$$

$$\{\mathbf{z}_1, \dots, \mathbf{z}_n\} = LSTM(\{\mathbf{x}_1, \dots, \mathbf{x}_n\}) \quad (2)$$

식 (1)에서 보듯이 얻어진 형태소 기반 단어 표상과 어절의 시작 형태소와 끝 형태소의 태그를 결합한 어절 태그 그리고 입력 토큰 단위로 토큰화된 입력을 BERT 기반 인코더를 통해 어절의 끝 토큰에 해당하는 출력 표상을 결합하며 최종 단어 임베딩을 구성한다. 구성된 단어 입력열을 식(2)과 같이 다층의 Bi-LSTM을 인코더를 통해 은닉 표상을 얻는다.

#### 3.2 전이 기반 파서

본 논문에서 사용하는 전이 기반 파서는 Arc-Hybrid[4] 방식을 사용하며 SHIFT, LEFT, RIGHT의 3가지 액션으로 구성되며 LEFT는 Arc-Eager와 같이 수행되며 RIGHT는 Arc-Standard와 같이 수행되어 하이브리드 방식으로 부른다. 각 전이 액션 별 스택 및 버퍼 정보의 갱신 과정을 표 1로 제시한다.

표 1. 전이 액션별 스택 버퍼의 갱신 과정

Action	$S_t$	$S_{t+1}$
SHIFT	$(\sigma, b_0   \beta, T)$	$(\sigma   b_0, \beta, T)$
LEFT( $l$ )	$(\sigma   s_1   s_0, b_0   \beta, T)$	$(\sigma   s_1, b_0   \beta, T \cup \{b_0, s_0, l\})$
Right( $l$ )	$(\sigma   s_1   s_0, \beta, T)$	$(\sigma   s_1, \beta, T \cup \{s_1, s_0, l\})$

표 1에서 보듯이  $\sigma, \beta, T$ 를 각각 스택, 버퍼, 아크(arc) 셋이라 하고  $b_0$ 를 버퍼의 Top 원소,  $s_0, s_1$ 를 버퍼의 Top 원소로 정의하며  $l$ 를 의존 관계 레이블로 정의한다. SHIFT 액션이 수행되면 단순히 버퍼의 Top 원소  $b_0$ 를 스택으로 이동하며 LEFT가 수행되면 스택의 Top 원소  $s_0$ 를 POP한 후  $b_0$ 를 지배소  $s_0$ 를 의존소로 하는 의존 관계를 생성하고 아크 셋에 추가한다. RIGHT가 수행되면 스택의 Top 원소  $s_0, s_1$ 를 POP 한 후  $s_1$ 을 지배소,  $s_1$ 를 의존소로 하는 의존 관계를 생성하고 아크 셋에 추가한 후  $s_1$ 를 다시 스택  $\sigma$ 에 PUSH한다.

$$P_t = [z_{s_2}; z_{s_1}; z_{s_0}; z_{b_0}] \quad (3)$$

전이 액션을 결정하기 위해서는 Bi-LSTM을 통해 인코딩된 은닉열  $\{z_1, \dots, z_n\}$ 로부터 자질을 추출하게 되는데 본 논문에서는 스택의 Top3에 해당하는 은닉상태  $z_{s_2}, z_{s_1}, z_{s_0}$ 와 버퍼의 Top1의 은닉 상태  $z_{b_0}$ 를 식(3)으로 결합하여 파서 상태 표상을 구성한 후 MLP층으로 연결되어 다음 전이 액션을 결정하게 된다.

### 3.3 그래프 기반 파서

그래프 기반 파서는 현재 한국어 의존 파싱에서 가장 높은 성능을 보이고 있는 Biaffine 어텐션 모델을 적용하였으며 위 쪽의 그림 1은 Biaffine 한국어 의존 파싱 모델의 구조를 보여준다.

$$h_i = MLP^{head}(z_i), d_i = MLP^{dep}(z_i) \quad (4)$$

$$\sigma(i, j) = Softmax_i(h_i^T W d_j + b_1^T h_i + b_2^T d_j) \quad (5)$$

그림 1에서 보듯이 3.1절서 설명한 Bi-LSTM을 통해 인코딩된 은닉열  $\{z_1, \dots, z_n\}$ 에 각각  $MLP^{head}$ ,  $MLP^{dep}$ 를 적용하여 지배소에 대한 표상과 의존소에 대한 표상을 얻고 얻어진 표상에 Biaffine 어텐션을 적용하여 모든 지배소와 의존소에 대한 스코어를 얻은 후 가장 스코어가 최대가 되는 파스 트리를 선택하여 구문 분석을 완료한다.

### 3.4 Dual Decomposition

Dual Decomposition은 서로 다른 두 모델에 대하여 Disagreement 가중치  $u$ 를 반영하여 두 모델을 디코딩한 후 결과를 비교하여 Disagreement 가중치가 서로 같아질 때까지 반복하여 조정하는 방법이다[10]. 본 논문에서는 Dual Decomposition이 그래프 기반 모델 사이에서 적용되었던 연구[10,16]와 전이 기반 모델 사이에서 적용되었던 연구[4]를 혼합하여 그래프 기반 모델과 전이 기반 모델 사이에 적용하여 결합하는 방법을 제안한다.

**Input** a sentence  $x$ , graph score matrix  $\gamma$

**Output** a dependency parse tree  $y$

$u \leftarrow 0$

$y^0 \leftarrow 0, z^0 \leftarrow 0$

**for**  $k = 1$  to  $K$  **do**

$y^{(k)} = GraphDecode(\gamma, u, y^{(k-1)}, 1)$

$z^{(k)} = TransDecode(x, u, z^{(k-1)}, -1)$

**if**  $y^{(k)} = z^{(k)}$  **then**

**return**  $y^{(k)}$

**else**

$u \leftarrow u - a(y^{(k)} - z^{(k)})$

**return**  $y^{(K)}$

그림 2. Dual Decomposition 알고리즘

입력 문장  $x$ 와 Biaffine Attention 모델로부터 얻은 스코어  $\gamma$ 가 입력으로 들어오게 되고 Disagreement 가중치  $u$ 가 0으로 초기화된다.  $\gamma$ 와  $z$ 는 단어(의존소)에 대한 지배소가 존재하면 해당 좌표  $(i, j)$ 에 1로 표시하는 각각 그래프, 전이 파스 트리를 나타내는 이진 행렬이며  $\gamma, z$  모두 0으로 초기화 된다. 미리 설정한  $K$ 만큼 반복하면서 가중치 Disagreement 가중치  $u$ 를 반영하면서 두 파서를 디코딩하고 두 파서가 동일한 파스 트리로 수렴하면 해당 시점  $k$ 에서 파스 트리  $y$ 를 반환하고 일치하지 않으면 하이퍼파라미터  $a$ 만큼 가중치  $u$ 를 갱신한다. 만약, 두 파서가 최대 반복 횟수  $K$ 만큼 반복해도 일치하지 않으면  $K$ 시점의 트리  $y$ 를 반환한다. 전이 기반 그래프 기반 파서에서  $u$ 를 반영하며 디코딩하는 함수는 다음 그림에 3에서 설명한다.

```

function GraphDecode( $\gamma, u, y, sgn$ )
   $\gamma'(i, j) = \gamma(i, j) + sgn \cdot u(i, j) \cdot \gamma(i, j)$ 
   $y \leftarrow MST(\gamma')$ 
  return  $y$ 

function TransDecode( $x, u, y, sgn$ )
   $c \leftarrow Initial(x)$ 
  while not Terminal( $c$ ) do
     $v \leftarrow 0$ 
    for  $a$  in Legal( $c$ ) do
      if  $a$  is not SHIFT then
         $arc \leftarrow a(c).T \setminus c.T$ 
         $v(a) \leftarrow sgn \cdot u(arc) \cdot y(arc)$ 
     $\hat{a} \leftarrow argmax_{a \in Legal(c)} f(c, a) + v(a)$ 
     $c \leftarrow \hat{a}(c)$ 
   $y \leftarrow c.T$ 
  return  $y$ 

```

그림 3. 파서 디코딩 함수

그래프 기반 파서의 디코딩 알고리즘은 Biaffine 어텐션 모델로부터 얻은 스코어  $\gamma$ 에  $u$ 를 반영하여 스코어를 업데이트하고 업데이트 된 스코어  $\gamma'$ 의 그래프로부터 최대 신장 트리(Maximum Spanning Tree) 알고리즘을 통해 새로운 트리를 생성한 후 리턴하는 간단한 과정으로 디코딩 된다.

반면, 전이 기반 파서의 디코딩 알고리즘은 입력 문장으로부터 초기 상태를 설정하고 초기 상태에서 파스 트리가 생성되는 최종 상태에 도달할 때까지 각 전이 시점마다 가중치  $u$ 를 반영할 조건을 만족하면 반영하며 진행한다. 각 전이 시점에서  $u$ 를 반영하는 조건은 상태  $c$ 에서 허용되는 액션 중에서 SHIFT가 아닌 의존성을 생성하는 LEFT, RIGHT 액션일 때 해당 액션을 통해 의존소  $i$ 와 지배소  $j$ 의 좌표  $(i, j)$  해당하는  $u$ 의 가중치를 전이 스코어에 반영함으로써 선택되는 전이 액션이 달라지도록 하여 최종 상태에 도달할 때까지 반복하며 생성된 트리를 반환한다.

## 4. 실험

### 4.1 실험 세팅

실험 및 평가를 위해 세종 구문 분석 데이터 셋과 SPMRL '14[19] 데이터 셋을 사용하였다. 세종 구문 분석 데이터 셋은 총 53,842 문장의 학습 셋과 5,817 문장의 평가 셋으로 구성되어 있으며 학습 셋에서 1,000 문장을 별도의 개발 셋으로 나누어 학습하였다. SPMRL '14 셋은 총 27,363문장으로 학습 셋 23,010문장, 개발 셋 2,066문장 그리고 평가 셋 2,287문장으로 구성된다.

두 데이터 셋은 모두 CRF 기반(F1:97.60%)으로 자동 형태소 분석을 수행하였으며 “형태소/태그” 단위로 사전 학습한 glove 형태소 임베딩을 사용하였다. 또한, ETRI BERT 모델[20] 사용을 위해 함께 공개된 ETRI 형태소 분석기로 별도로 형태소 분석을 수행하였으며 실험에 사용된 하이퍼 파라미터의 세팅은 다음과 표 2와 같다.

표 2. 모델의 하이퍼파라미터

	Hyper Parameter	value
BERT	인코더 블록 개수	12
	은닉 차원수	768
	어텐션 헤드 수	12
	최대 문장 길이	512
	활성함수	gelu
	Optimizer	Adam
Model	학습률	$1.5e^{-5}$
	Optimizer	Adam
	학습률	0.001
Encoder	Dropout rate	0.33
	형태소 임베딩	100
	어절 태그 임베딩	50
	BERT 임베딩	768
	Char RNN 은닉 차원 수	512
	RNN 은닉 차원 수	512
Transition	RNN Layers	3
	자질 차원 수	64
Graph	Arc MLP	512
	Label MLP	128

### 4.2 실험 결과

BERT 인코더를 적용하지 않은 버전과 적용한 버전으로 나누어 실험하였으며 Dual Decomposition 방법의 성능 비교를 위해 베이스라인으로는 대상이 되는 전이 기반 모델과 그래프 기반 모델과 비교하며 각 데이터 셋에서의 기존 연구들의 성능을 함께 제시한다.

정확한 평가를 위해 Dual Decomposition에 사용되는 하이퍼 파라미터  $\alpha$ 를 개발 셋에서 0.001~0.01까지 0.001씩 증가하며 성능 향상 폭을 측정하였으며 평가 셋에서 Dual Decomposition을 적용 시 개발 셋에서 가장 성능 향상 폭이 높은  $\alpha$ 를 설정하였다. 표 3은 BERT 기반 모델의

SPMRL '14 개발 셋에서  $\alpha$ 에 변화에 따른 성능 향상 변화를 보여준다.

표 3. SPMRL '14 개발 셋에서의  $\alpha$ 값에 따른 성능 변화

value $\alpha$	UAS	LAS
0.000 (base)	95.61%	95.12%
0.001	95.62%	95.13%
0.002	95.66%	95.17%
0.003	95.63%	95.14%
0.004	95.64%	95.15%
0.005	95.68%	95.19%
0.006	95.68%	95.19%
0.007	<b>95.71%</b>	<b>95.22%</b>
0.008	95.68%	95.19%
0.009	95.61%	95.12%
0.01	95.61%	95.12%

표 4와 5는 각각 세종 셋과 SPMRL '14 셋에서의 의존 파싱 성능을 비교하며 이중선을 경계로 하여 위 쪽은 BERT 인코더를 사용하지 않은 모델들과 아래 쪽은 BERT 인코더를 사용한 모델들로 나누어 실험 결과를 제시한다.

표 4. 세종 셋에서의 의존 파싱 성능 비교

	UAS	LAS
Stack LSTM [17]	90.83%	88.49%
Deep Biaffine [9]	91.78%	89.76%
멀티헤드 어텐션 +ELMo [12]	<b>92.85%</b>	<b>90.65%</b>
Arc-Hybrid 전이기반	91.32%	89.17%
Deep Biaffine(impl)	91.72%	89.55%
Dual Decomposition	91.78%	89.59%
Bert+ELMo + ML Biaffine [13]	93.06%	90.97%
Bert+Biaffine [14]	94.06%	92.00%
[Bert] Arc-Hybrid 전이기반	94.27%	94.36%
[Bert] Deep Biaffine(impl)	94.42%	92.52%
[Bert] Dual Decomposition	<b>94.47%</b>	<b>92.58%</b>

표 4에서 보듯이 Dual Decomposition을 통해 BERT를 적용하지 않은 모델에서는 UAS 0.06%, LAS 0.04%의 성능향상을 보였다. 또한, BERT 전이기반, 그래프 기반 모델 모두 기존 연구에서의 최고 성능을 개선하였으며 Dual Decomposition을 통해 추가로 UAS 0.05%, LAS 0.06%의 개선된 성능을 보였다.

아래의 표 5는 SPMRL '14셋에서의 실험 결과이며 BERT 인코더를 사용하지 않은 모델과 사용한 모델 모두 성능 향상을 보였으며 성능 향상의 폭이 미미했던 세종 셋에 비해 높음을 알 수 있다. BERT를 적용하지 않은 모델에 대해서는 UAS 0.18%, LAS 0.16%의 성능 향상을 보였으며 세종 셋과 마찬가지로 BERT를 적용한 모델에서는 전이 기반, 그래프 기반 모두 기존의 최고 성능보다

높은 성능을 보였으며 Dual Decomposition을 적용하여 그래프 기반 모델보다 UAS 0.16%, LAS 0.15% 높은 성능의 결과를 보인다.

표 5. SPMRL '14 셋에서의 의존 파싱 성능 비교

	UAS	LAS
Stack LSTM [17]	89.94%	88.36%
Deep Biaffine [9]	90.85%	89.31%
Deep Biaffine + Dual [10]	91.07%	N/A
Arc-Hybrid 전이기반	91.69%	91.08%
Deep Biaffine(impl)	92.20%	91.68%
Dual Decomposition	92.38%	91.84%
Bert+ELMo+ML Biaffine [13]	93.46%	92.85%
Bert+Biaffine [14]	93.87%	93.06%
[Bert] Arc-Hybrid 전이기반	94.36%	93.39%
[Bert] Deep Biaffine(impl)	94.58%	94.05%
[Bert] Dual Decomposition	<b>94.74%</b>	<b>94.20%</b>

## 5. 결론

본 논문에서는 Dual Decomposition 알고리즘을 이용하여 전이 기반 파서와 그래프 기반 파서를 하이브리드 방식으로 결합하는 모델을 제안하고 한국어 의존 파싱에서 최고 성능을 달성하였다. 향후 연구로는 BERT에 나아가 더 개선된 언어모델인 RoBERTa, XLNet 등의 언어 모델을 한국어 코퍼스에 학습하여 의존 파싱에 반영하는 연구를 진행할 예정이다.

### 감사의 글

이 논문은 2019년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (R7119-16-1001, 지식증강형 실시간 동시통역 원천기술 개발)

### 참고문헌

- [1] Chen, D., & Manning, C. (2014, October). A fast and accurate dependency parser using neural networks. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 740-750).
- [2] Dyer, C., Ballesteros, M., Ling, W., Matthews, A., & Smith, N. A. (2015). Transition-based dependency parsing with stack long short-term memory. arXiv preprint arXiv:1505.08075.
- [3] Dozat, Timothy, and Christopher D. Manning. "Deep biaffine attention for neural dependency parsing." arXiv preprint arXiv:1611.01734 (2016).
- [4] Yuan, Y., Jiang, Y., & Tu, K. (2019). Bidirectional Transition-Based Dependency Parsing.
- [5] Petar Velikovi, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Li, and Yoshua Bengio. 2018. Graph attention networks. In International Conference on Learning Representations.
- [6] Ma, X., Hu, Z., Liu, J., Peng, N., Neubig, G., & Hovy, E. (2018). Stack-pointer networks for dependency parsing. arXiv preprint arXiv:1805.01087.
- [7] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
- [8] 이창기, 딥러닝을 이용한 한국어 의존 구문 분석, HCLT, pp 87-91, 2014.
- [9] 나승훈, 이진일, 신중훈, 김강일, "Deep Biaffine Attention을 이용한 한국어 의존 파싱", KCC 2017
- [10] 민진우, 나승훈, 신중훈, 김영길. (2018). Dual Decomposition을 이용한 뉴럴 그래프 기반 한국어 의존 파싱. 한국정보과학회 학술발표논문집, (), 643-645.
- [11] 차다운, 이동엽, 임희석, Stack-Pointer Network를 이용한 한국어 의존 구문 분석, HCLT 2018.
- [12] 박성식, 오신혁, 김홍진, 김시형, 김학수, "ELMo와 멀티헤드 어텐션을 이용한 한국어 의존 구문 분석", HCLT 2018
- [13] 홍승연, 나승훈, 신중훈, 김영길, "BERT와 ELMo 문맥화 단어 임베딩을 이용한 한국어 의존 파싱", KCC 2019
- [14] 박천음, 이창기, 임준호, 김현기, "BERT를 이용한 한국어 의존 구문 분석", KCC 2019
- [15] Kiperwasser, Eliyahu, and Yoav Goldberg. "Simple and accurate dependency parsing using bidirectional LSTM feature representations." Transactions of the Association for Computational Linguistics 4 (2016): 313-327.
- [16] Koo, Terry, et al. "Dual decomposition for parsing with non-projective head automata." Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2010.
- [17] 나승훈, 김강일, 김영길, "Stack LSTM을 이용한 전이기반 한국어 의존 파싱, KCC 2016
- [18] Seddah, D., Kübler, S., & Tsarfaty, R. (2014, August). Introducing the spmrl 2014 shared task on parsing morphologically-rich languages. In Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages (pp. 103-109)
- [19] [http://aiopen.etri.re.kr/service\\_dataset.php](http://aiopen.etri.re.kr/service_dataset.php)