의 존 구 문 분 석
(Dependency Parsing)

남궁영
한국해양대학교 컴퓨터공학과
young_ng@kmou.ac.kr

2019. 04. 15.

국립 한국해양대학교
KOREA MARITIME AND OCEAN UNIVERSITY

Content

- What is Graph-based Dependency Parsing
  - Arc-factored model
- Chu-Liu-Edmond's Algorithm
- Neural Models for Graph-based Dependency Parsing
  - Phrase Embeddings
  - BiLSTM Feature Extractors
  - BiAffine Classifier

I Graph-based Dependency Parsing

❖ **Define a space of candidate dependency trees for a sentence**

- Learning: induce a model for **scoring** an entire tree
- Parsing: find a tree with the **highest score**, given the induced model

Factor the weight/score graphs by subgraphs

$$Y^* = \underset{Y \in \Phi(X)}{argmax} \sum_{(x_i \to x_j) \in Y} score(x_i \to x_j)$$

| http://clear.colorado.edu/dependency/Dependency-Parsing-Workshop-2009/mcdonald.pdf

I Graph-based Dependency Parsing

❖ **Arc-factored scoring model**

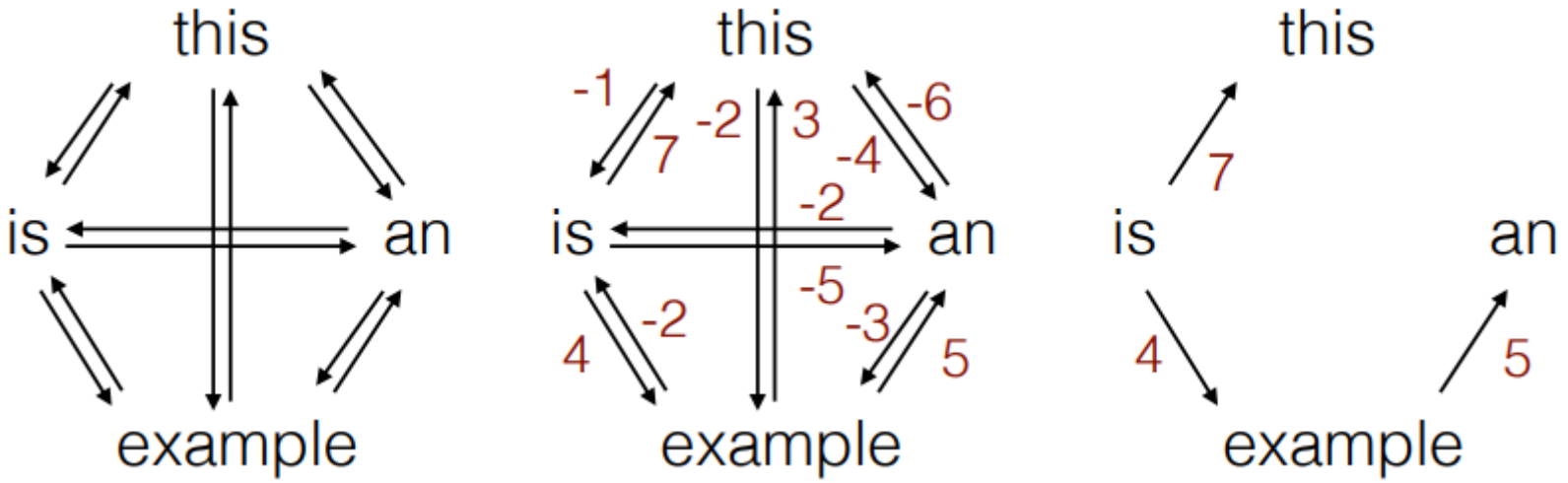
- **Chart-based parsing**
  - Creates projective dependency trees
  - Inspired by the CKY algorithm
    - Collins' algorithm
    - Eisner's algorithm

- **Minimum Spanning Tree**
  - Creates non-projective dependency trees
    - Edmond's algorithm

Arc-factored scoring model

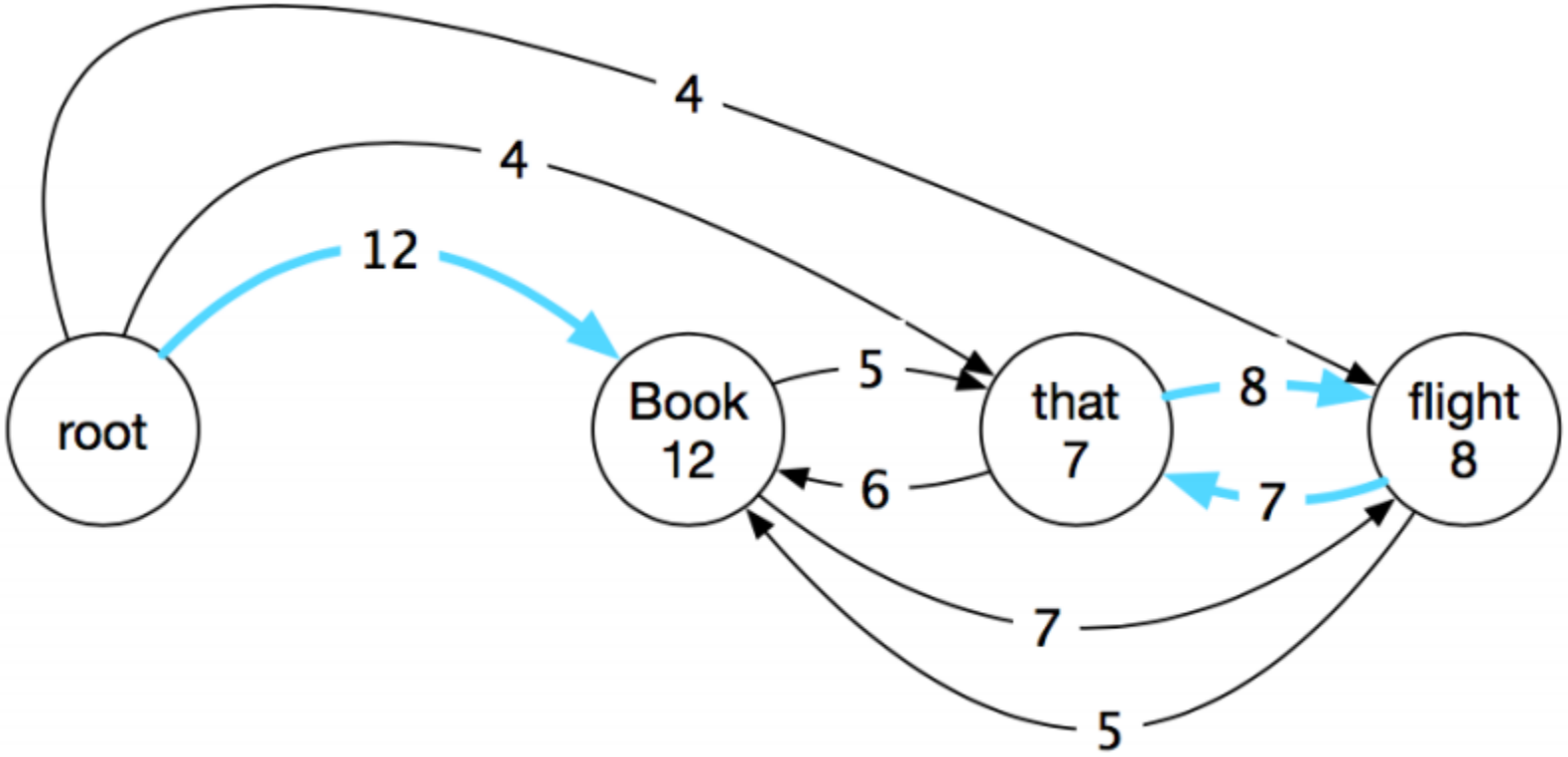
Chu-Liu-Edmond's Algorithm

- Express sentence as fully connected directed graph
- Score each edge independently
- Find maximum spanning tree



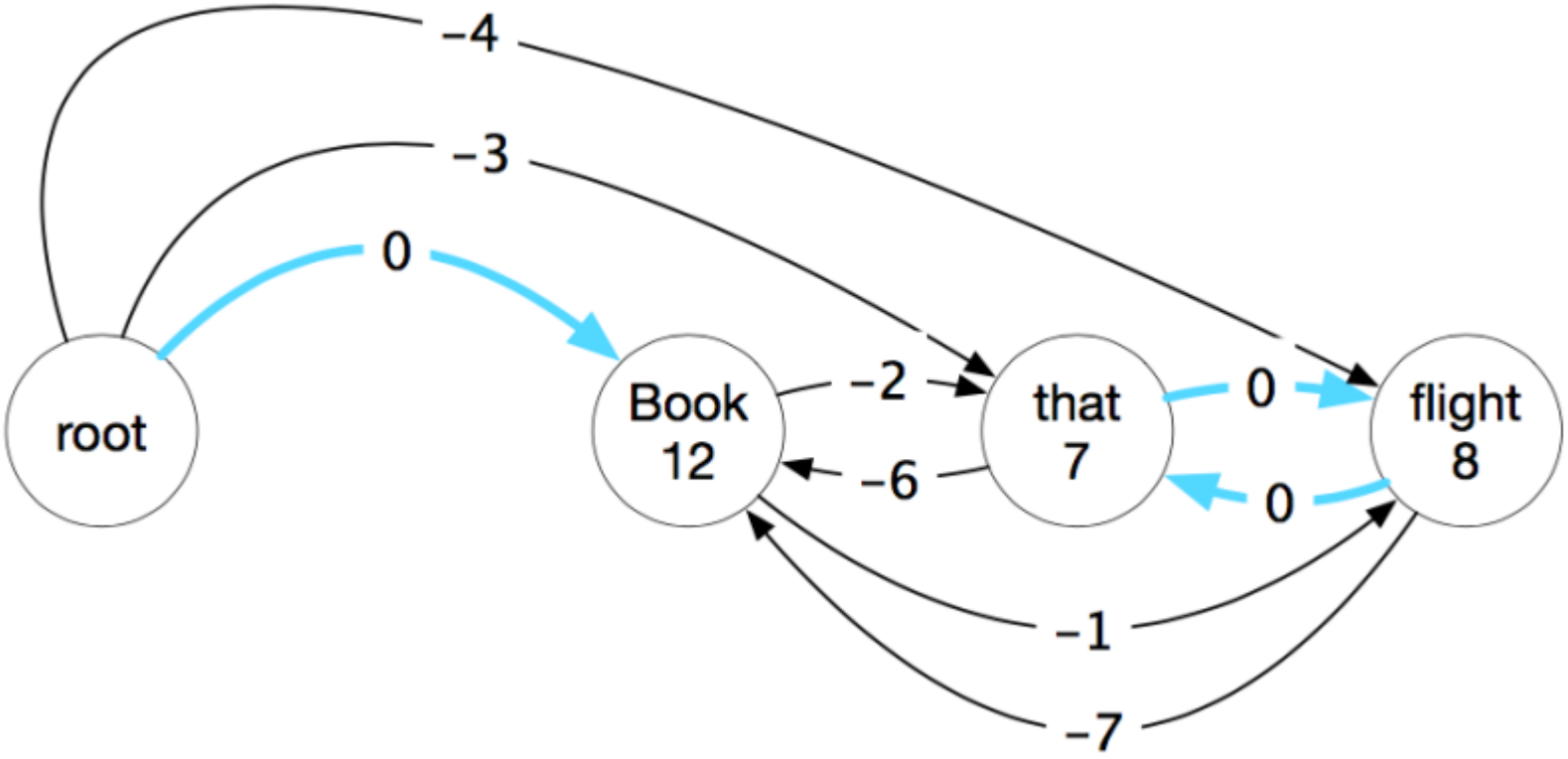
Chu-Liu-Edmond's Algorithm

❖ **Find the Best Incoming**



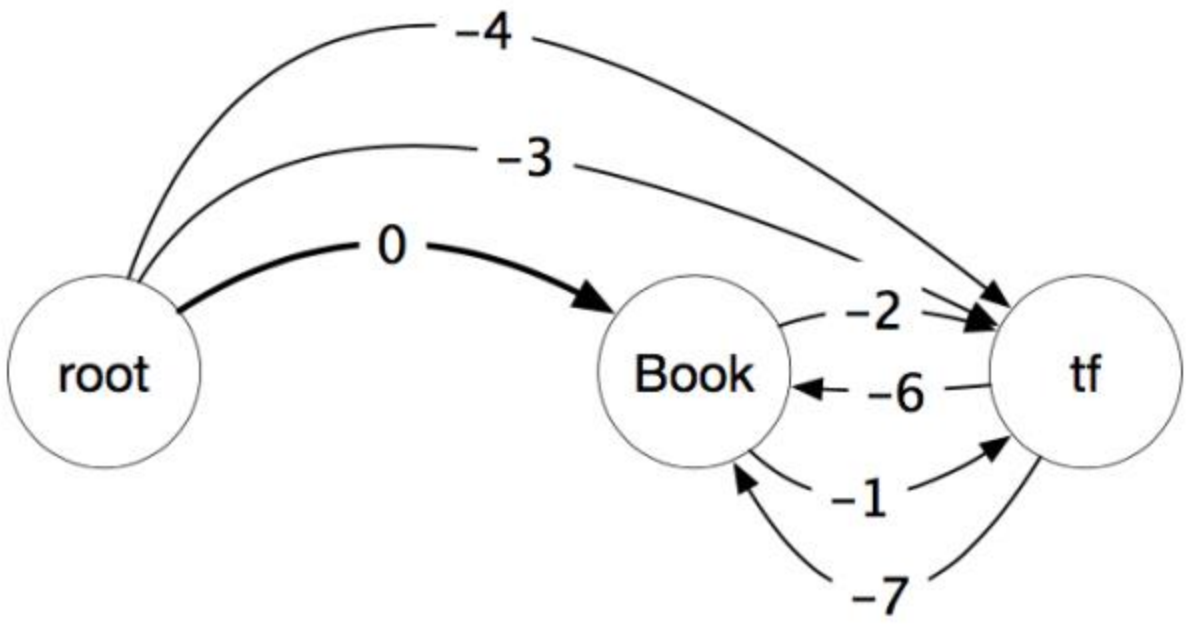
Chu-Liu-Edmond's Algorithm

❖ **Subtract the Max for Each**



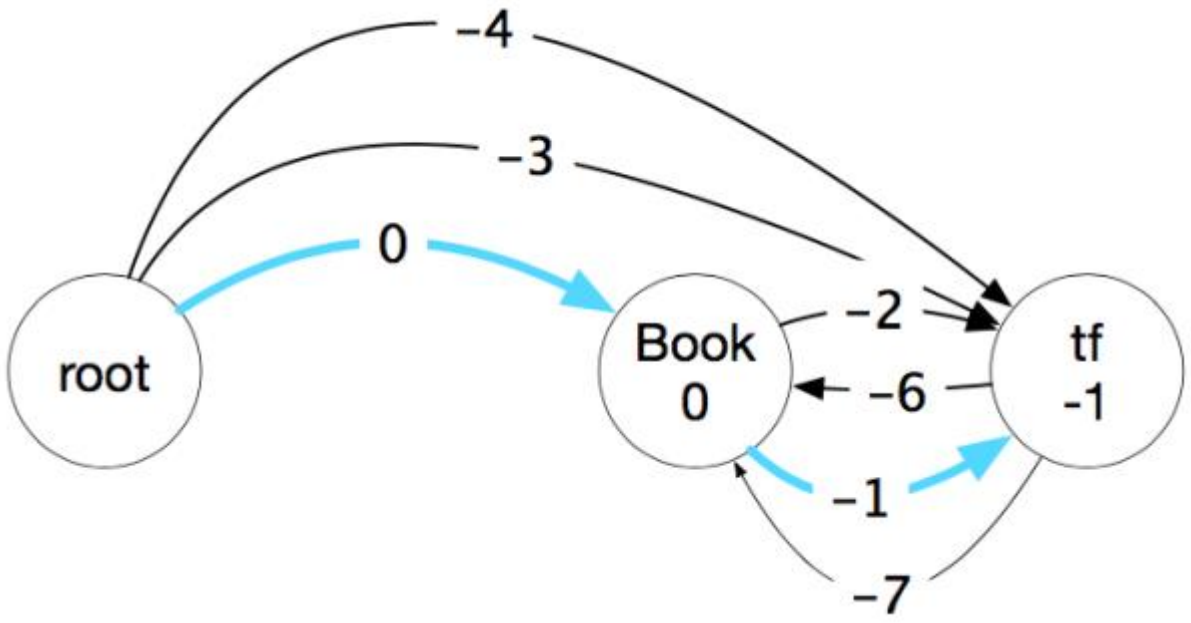
Chu-Liu-Edmond's Algorithm

❖ **Contract a Node**



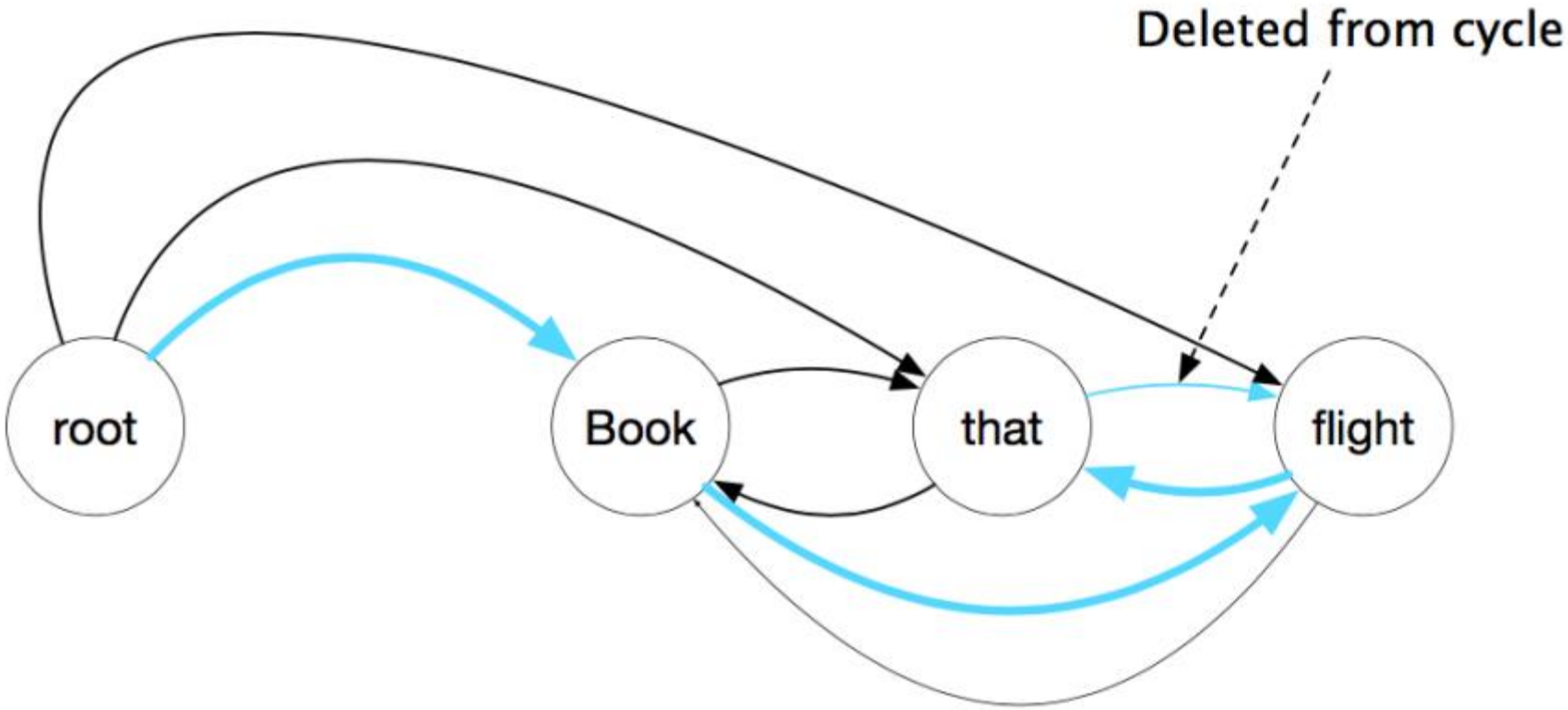
Chu-Liu-Edmond's Algorithm

❖ **Recursively Call Algorithm**



Chu-Liu-Edmond's Algorithm

❖ **Expand Nodes and Delete Edge**



Neural Models for Graph-based Dependency Parsing

Phrase Embeddings

BiLSTM Feature Extractors

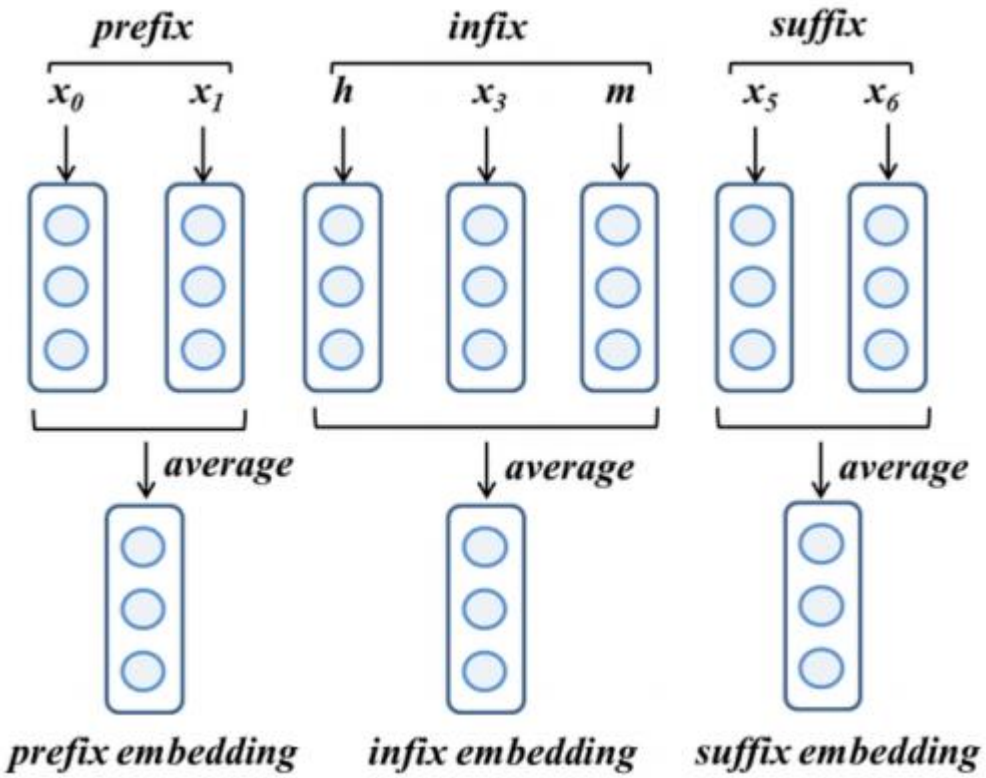
BiAffine Classifier

IV Phrase Embeddings

(Pei et al. 2015)

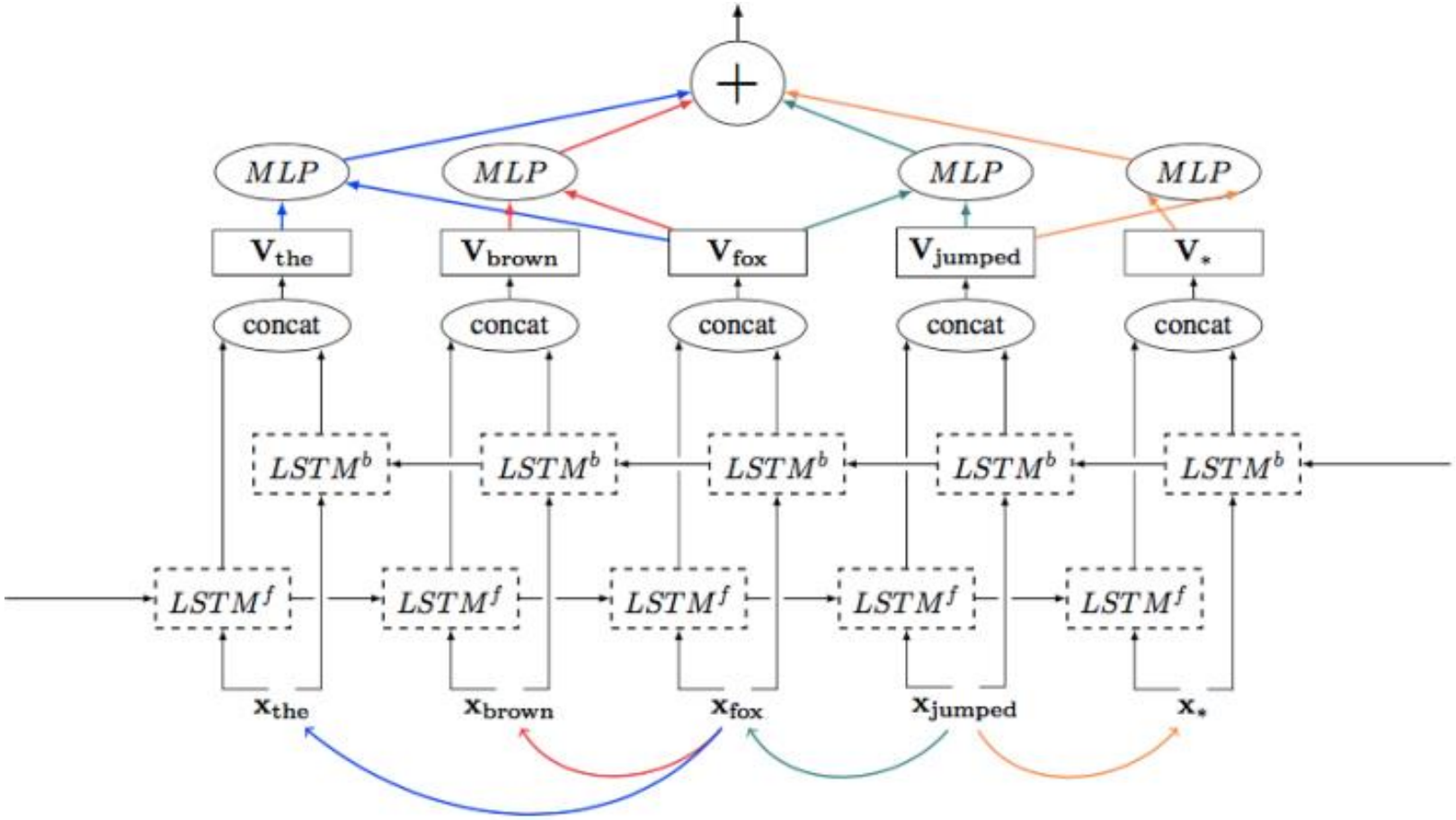
❖ Motivations

- words surrounding or between head and dependent are important clues



IV BiLSTM Feature Extractors

(Kipperwasser and Goldberg, 2016)



❖ Simpler and better accuracy than manual extraction

IV Pointer Network

❖ **attention**

- q가 vector x의 요소인 x(i) 각각에 어느 정도 attention을 둬야하는지 수치로 계산하는 방법.
- 따라서 관련도를 계산하는 score f(x(i), q)를 어떻게 정의하는 지가 중요
- Dependency parsing과 관련하여 attention mechanism은 LSTM기반 graph-based approach 에서 score(h, m)를 계산하는 데 그대로 활용됨

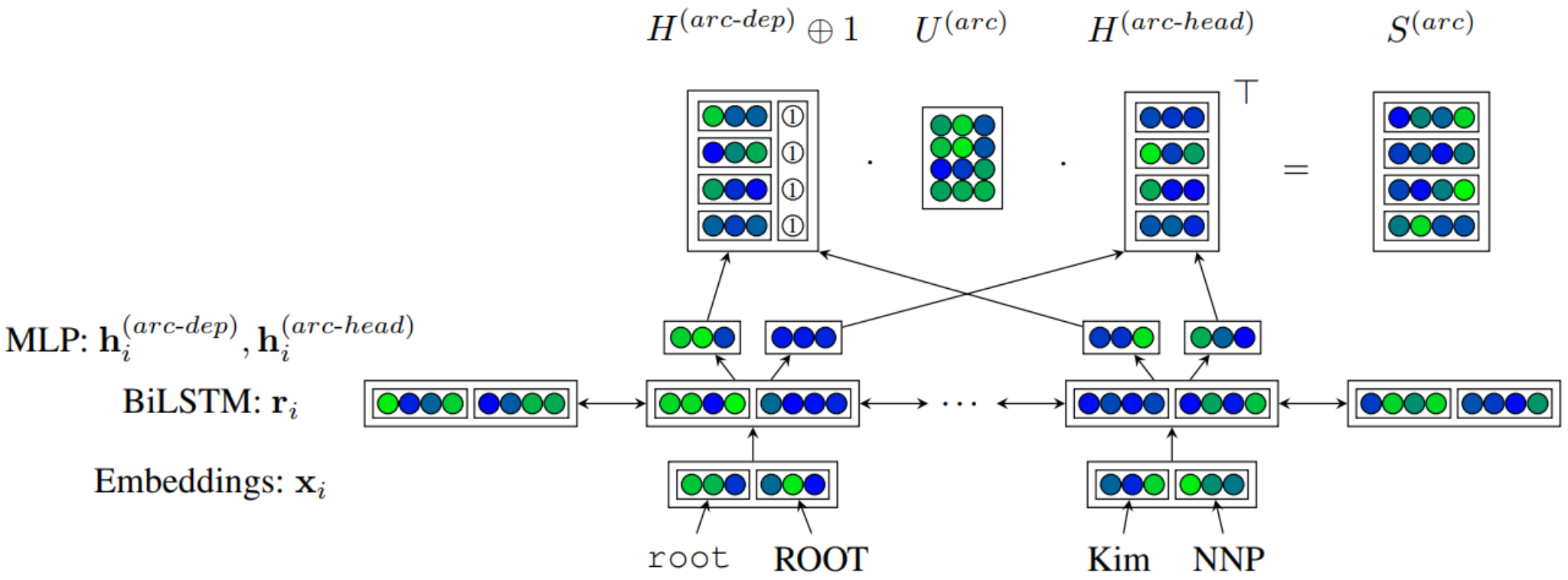
attention weight

$$u_j^i = v^T \tanh(W_1 e_j + W_2 d_i) \quad j \in (1, \ldots, n)$$
$$p(C_i | C_1, \ldots, C_{i-1}, \mathcal{P}) = \text{softmax}(u^i)$$

V BiAffine Classifier

(Dozat and Manning, 2017)

❖ vector를 MLP를 통해 한번 더 추상화 한 뒤 bilinear classifier로 attention scoring을 적용한 모델.



V BiAffine Classifier

(Dozat and Manning, 2017)

❖ deep bilinear attention mechanism: uses the recurrent states directly

- Learn specific representations for head/dependent for each word

$$\mathbf{h}_i^{(arc\text{-}dep)} = \mathrm{MLP}^{(arc\text{-}dep)}(\mathbf{r}_i)$$

$$\mathbf{h}_j^{(arc\text{-}head)} = \mathrm{MLP}^{(arc\text{-}head)}(\mathbf{r}_j)$$

$$\mathbf{s}_i^{(arc)} = H^{(arc\text{-}head)} U^{(1)} \mathbf{h}_i^{(arc\text{-}dep)} + H^{(arc\text{-}head)} \mathbf{u}^{(2)}$$

감 사 합 니 다.

남궁영
한국해양대학교 컴퓨터공학과
young_ng@kmou.ac.kr