WestlakeNLP

Natural Language Processing

Yue Zhang
Westlake University

西湖大學
WESTLAKE UNIVERSITY

WestlakeNLP

Chapter 11

Transition-based Methods for Structured Prediction

Contents

Contents

WestlakeNLP

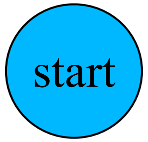
A Transition System

WestlakeNLP

- A different view from graph-based models for structured prediction
  - Encodes rich unbounded features without complexity constraints
  - A general framework that is easy to adapt to different tasks
- Maps output building process into a state transducer
  - State —— S_i
    - Corresponds to partial results during decoding
  - Action —— a_i
    - The operations that can be applied for state transition
    - Construct output incrementally

A Transition System

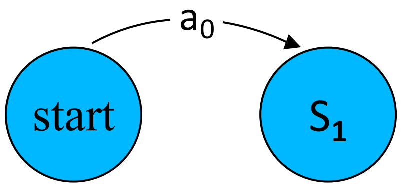
- **Automata:**
  - State —— S_i
  - Action —— a_i



A Transition System

WestlakeNLP

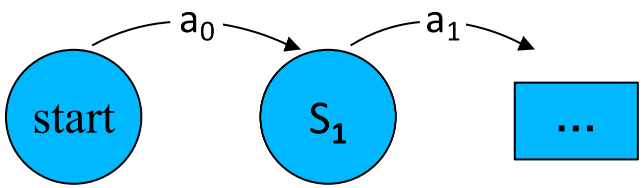
- **Automata:**
  - State —— S_i
  - Action —— a_i



A Transition System

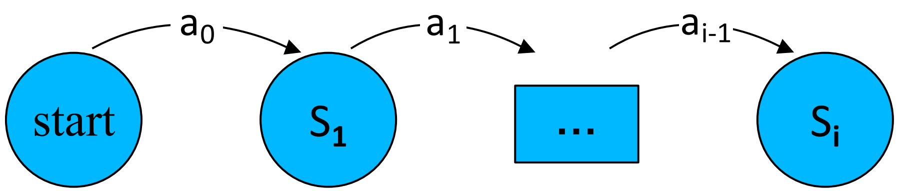
WestlakeNLP

- **Automata:**
  - State —— S_i
  - Action —— a_i



A Transition System

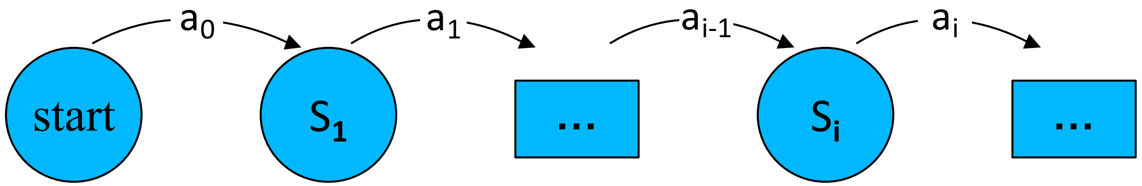
- **Automata:**
  - State —— S_i
  - Action —— a_i



A Transition System

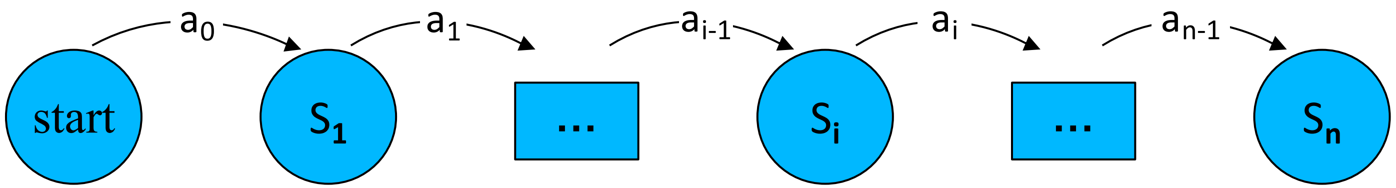
WestlakeNLP

- **Automata:**
  - State —— S_i
  - Action —— a_i



A Transition System

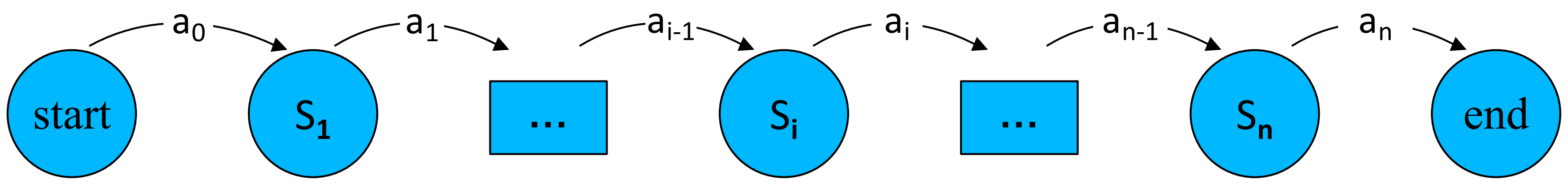
- **Automata:**
  - State —— S_i
  - Action —— a_i



A Transition System

WestlakeNLP

- **Automata:**
  - State —— S_i
  - Action —— a_i



Transition-based structured prediction

WestlakeNLP

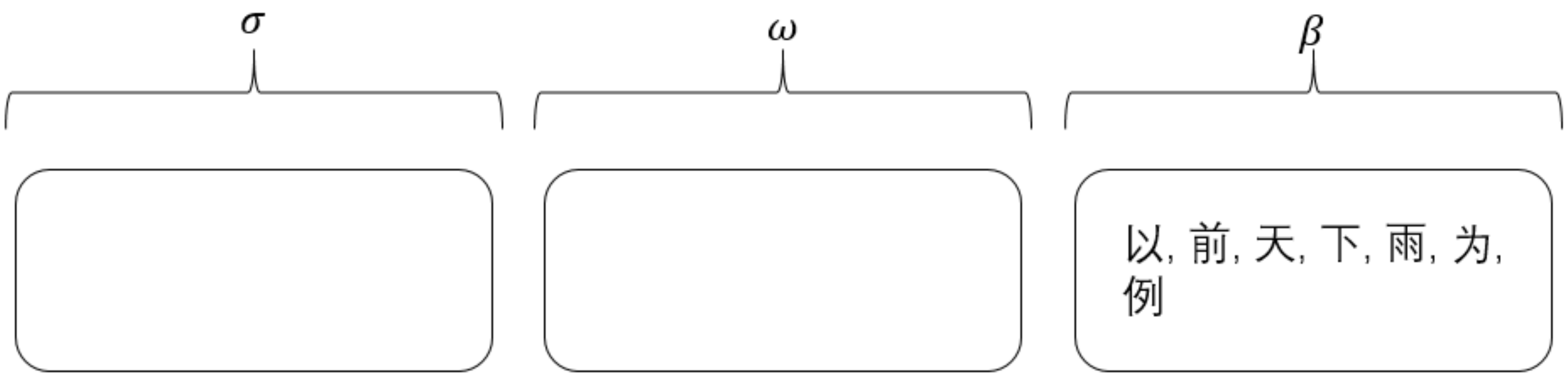
Transition-based structured prediction is a state transition process.

Automata:

- State
  - Start state —— an empty structure
  - End state —— the output structure
  - Intermediate state —— partially constructed structures
- Transition actions
  - Incremental steps that build output structures, change one state to another

Example (Word Segmentation)

- Input："以(take) 前(before) 天(day) 下(fall) 雨(rain) 为(be) 例(example)"



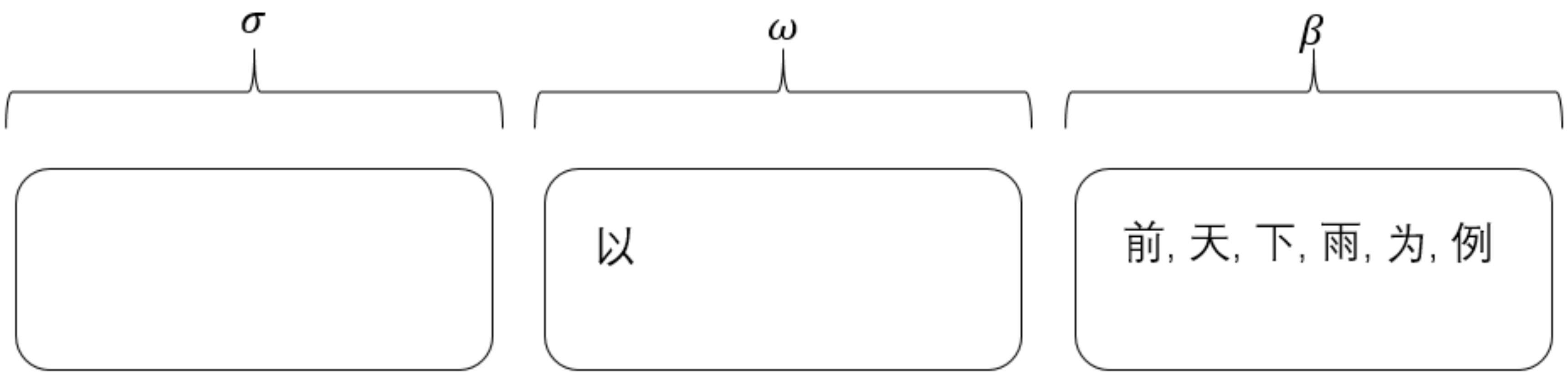
Next Action: ***SEP***

σ: partial output w: current partial word β: list of next incoming characters

Example (Word Segmentation)

WestlakeNLP

- Input："以(take) 前(before) 天(day) 下(fall) 雨(rain) 为(be) 例(example)"

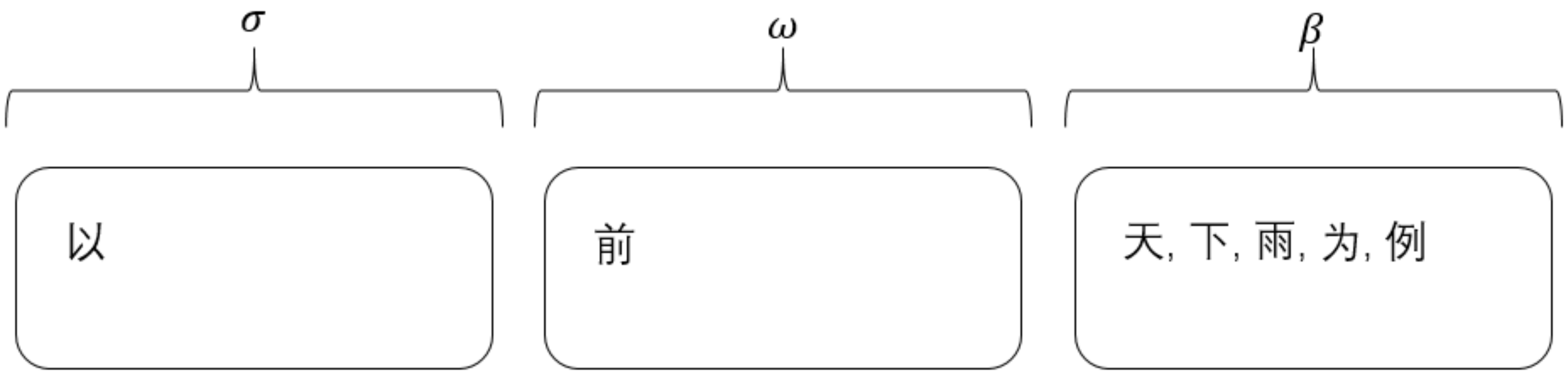


Next Action: *SEP*

σ: partial output w: current partial word β: list of next incoming characters

Example (Word Segmentation)

- Input："以(take) 前(before) 天(day) 下(fall) 雨(rain) 为(be) 例(example)"



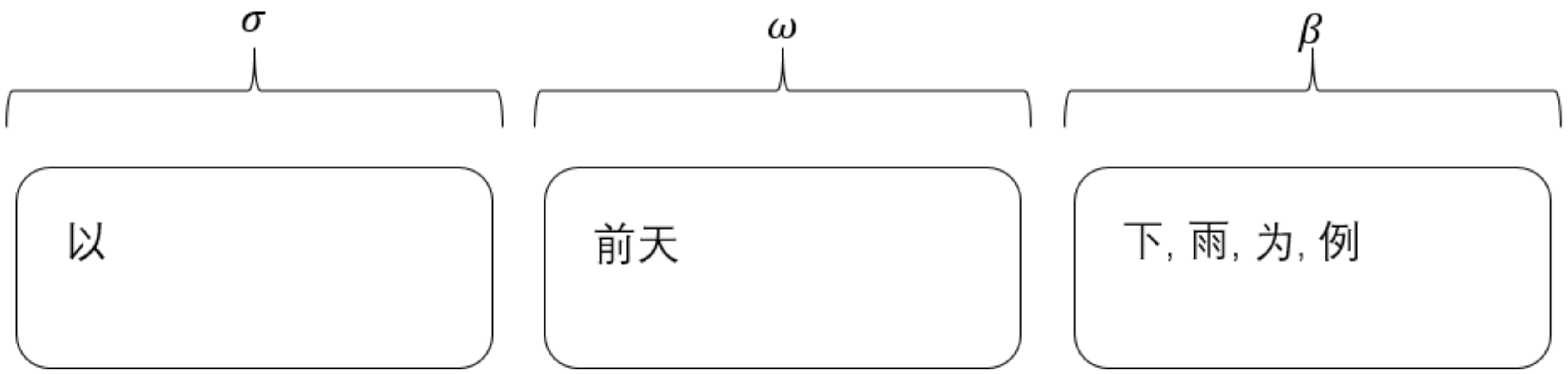
Next Action: ***APP***

σ: partial output w: current partial word β: list of next incoming characters

Example (Word Segmentation)

WestlakeNLP

- Input："以(take) 前(before) 天(day) 下(fall) 雨(rain) 为(be) 例(example)"

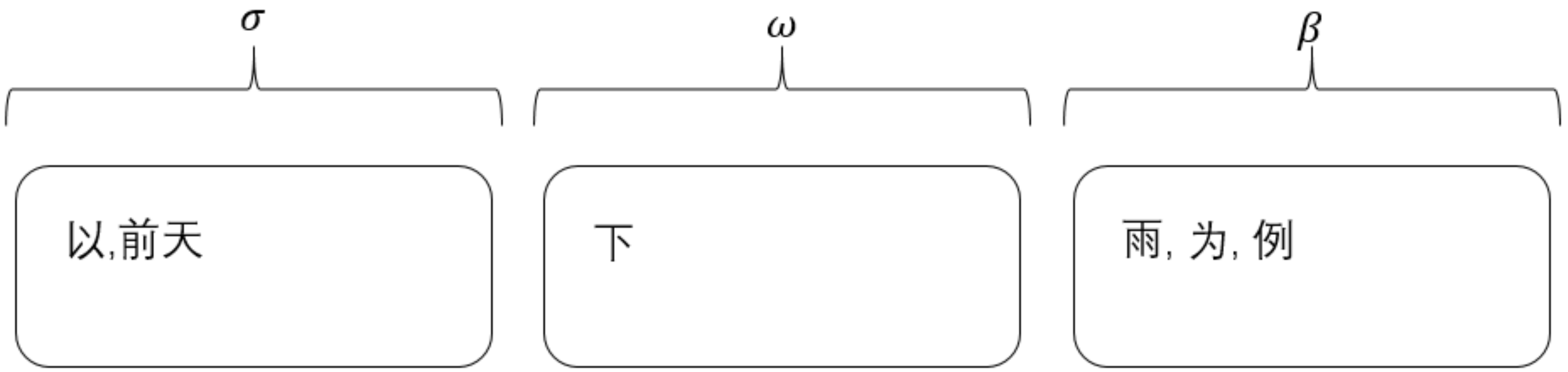


Next Action: ***SEP***

σ: partial output w: current partial word β: list of next incoming characters

Example (Word Segmentation)

- Input："以(take) 前(before) 天(day) 下(fall) 雨(rain) 为(be) 例(example)"

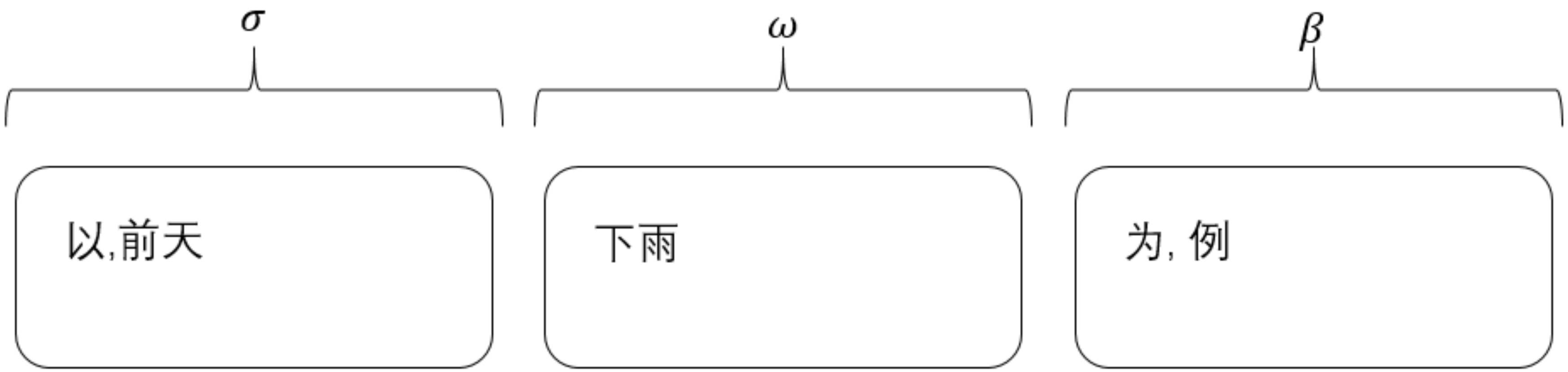


Next Action: ***APP***

σ: partial output w: current partial word β: list of next incoming characters

Example (Word Segmentation)

- Input："以(take) 前(before) 天(day) 下(fall) 雨(rain) 为(be) 例(example)"

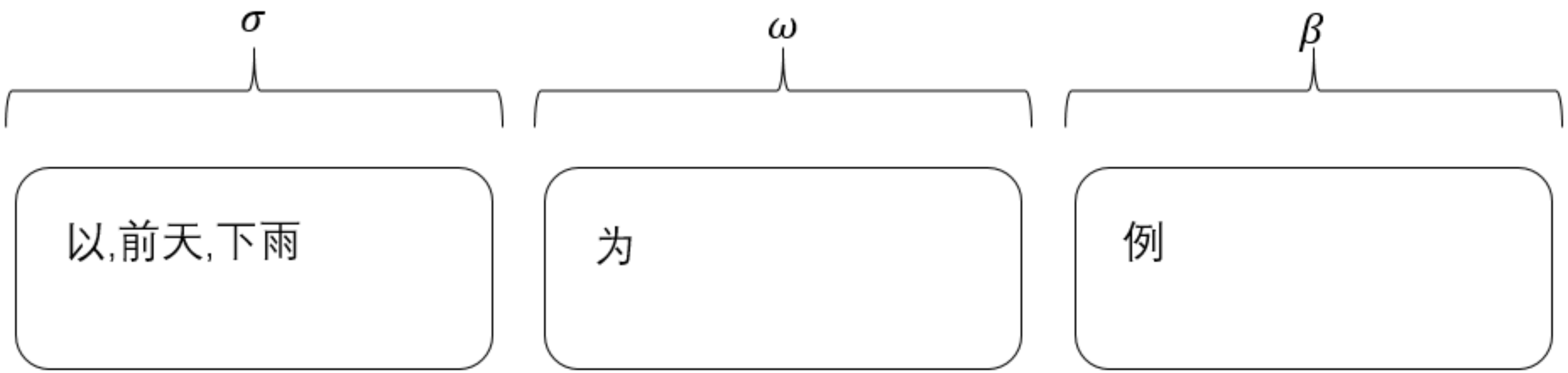


Next Action: *SEP*

σ: partial output w: current partial word β: list of next incoming characters

Example (Word Segmentation)

- Input："以(take) 前(before) 天(day) 下(fall) 雨(rain) 为(be) 例(example)"

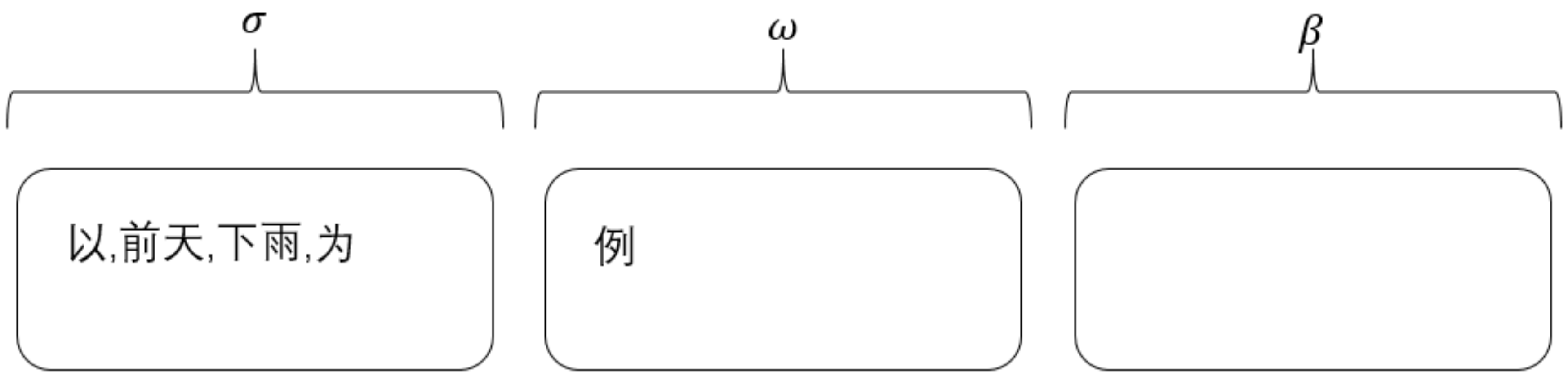


Next Action: ***SEP***

σ: partial output w: current partial word β: list of next incoming characters

Example (Word Segmentation)

- Input："以(take) 前(before) 天(day) 下(fall) 雨(rain) 为(be) 例(example)"

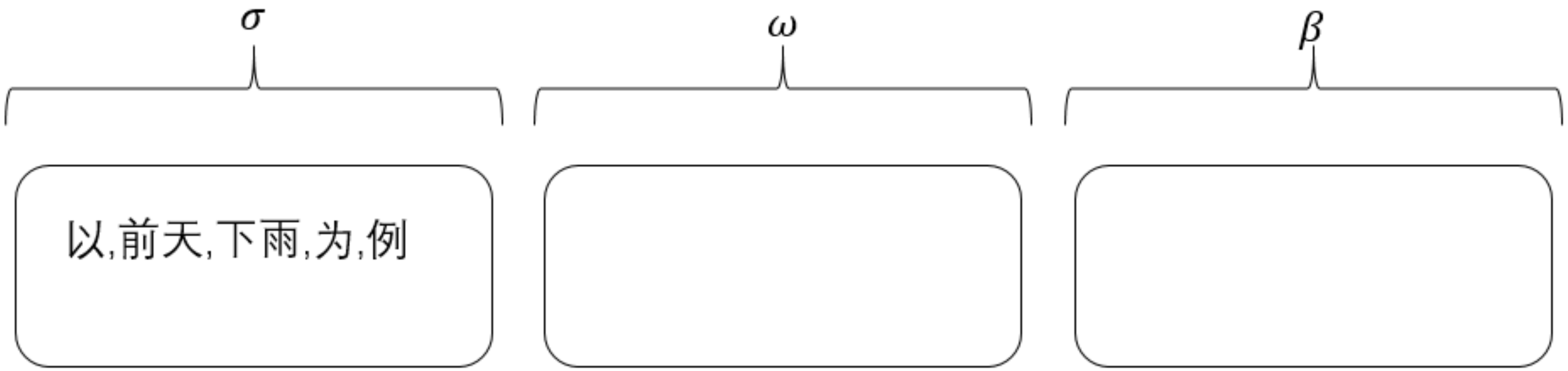


Next Action: ***FIN***

σ: partial output w: current partial word β: list of next incoming characters

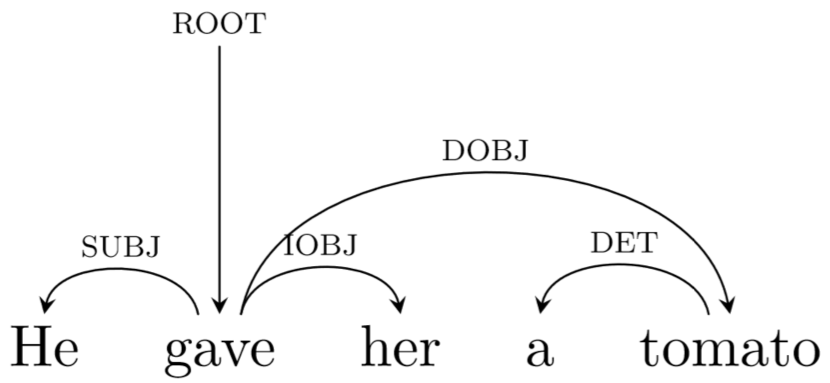
Example (Word Segmentation)

- Input："以(take) 前(before) 天(day) 下(fall) 雨(rain) 为(be) 例(example)"



σ: partial output w: current partial word β: list of next incoming characters

Example (Dependency Parsing)



Example (Dependency Parsing)

WestlakeNLP

$$\overbrace{\boxed{\phantom{\text{He does it}}}}^{\sigma} \quad \overbrace{\boxed{\text{He does it here}}}^{\beta}$$

σ: stack $\qquad$ β: buffer

Example (Dependency Parsing)

- NEXT ACTION: $Shift$

$$\overbrace{\boxed{\phantom{\text{He does it here}}}}^{\sigma} \quad \overbrace{\boxed{\text{He does it here}}}^{\beta}$$

σ: stack β: buffer

Example (Dependency Parsing)

WestlakeNLP

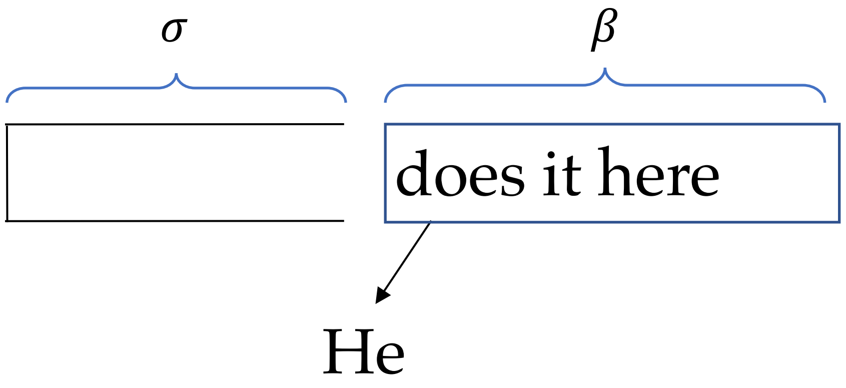
- NEXT ACTION: $Left - Arc$

$$\overbrace{\boxed{\text{He}}}^{\sigma} \quad \overbrace{\boxed{\text{does it here}}}^{\beta}$$

σ: stack β: buffer

Example (Dependency Parsing)

- NEXT ACTION: $Shift$



σ: stack β: buffer

Example (Dependency Parsing)

WestlakeNLP

- NEXT ACTION: $Right - Arc$

σ β

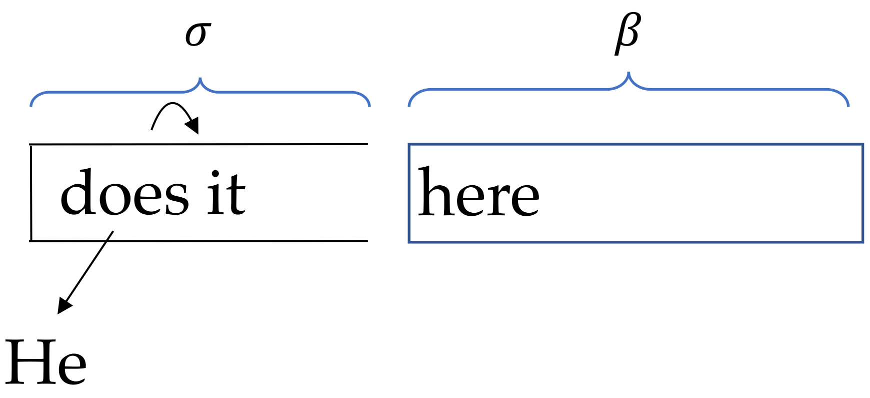
does | it here

He

σ: stack β: buffer

Example (Dependency Parsing)

- NEXT ACTION: *Reduce*

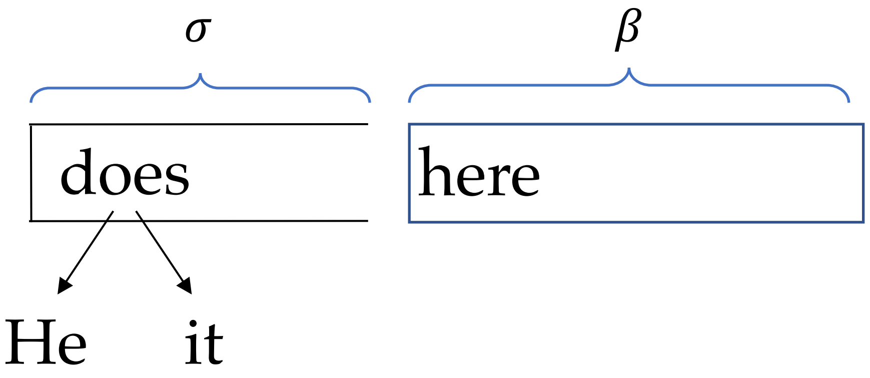


σ: stack β: buffer

Example (Dependency Parsing)

WestlakeNLP

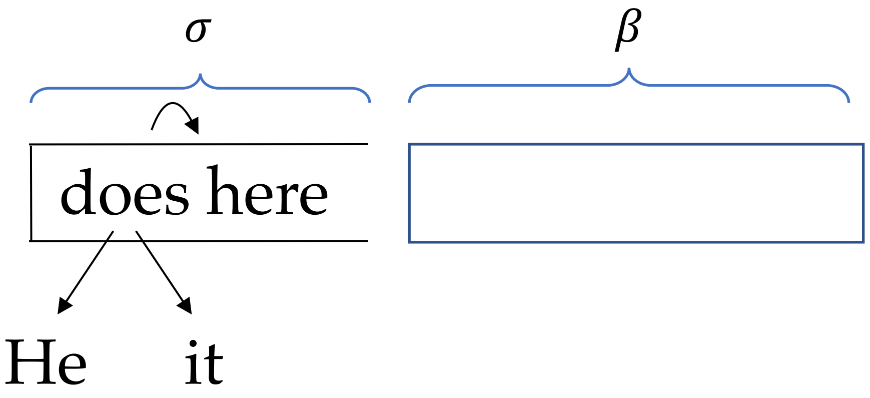
- NEXT ACTION: $Right - Arc$



σ: stack β: buffer

Example (Dependency Parsing)

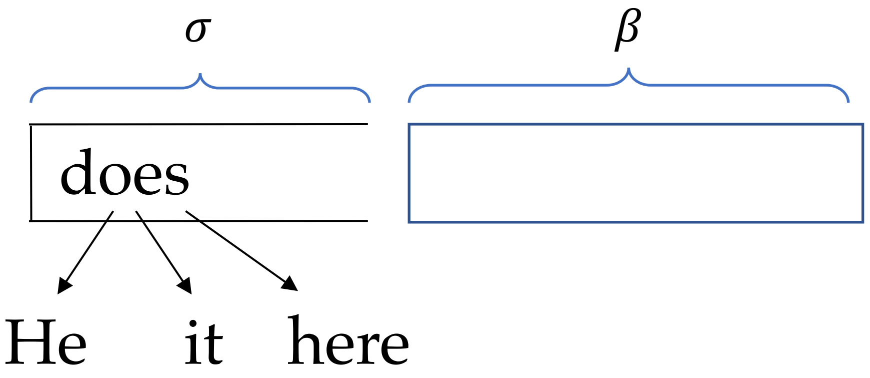
- NEXT ACTION: *Reduce*



σ: stack β: buffer

Example (Dependency Parsing)

- NEXT ACTION: *Finish*



σ: stack β: buffer

Contents

WestlakeNLP

Transition-based Modeling

WestlakeNLP

- Given a state s_{i-1}, our goal is to disambiguate all possible actions $a_i \in$POSSIBLEACTIONS(s_{i-1}) by using a discriminative model to score transition actions:

$$score(a_i|s_{i-1}) = \vec{\theta} \cdot \vec{\phi}(s_{i-1}, a_i)$$

- $\vec{\theta}$: model parameter vector
- $\vec{\phi}(s_{i-1}, a_i)$: feature vector on the input-output pair (s_{i-1}, a_i)

Greedy Local Method

- Training
  - Training dataset $D = \{(X_i, Y_i)\}|_{i=1}^{N}$

    break down

  - A sequence of gold transitions: $\left(s_{j-1}^{(i)}, a_j^{(i)}\right)$

    merge all the state-action pairs

  - A training set for the discriminative model
- Testing
  - Start from initial state $s_0(X)$
  - Repeatedly find $\hat{a}_i = argmax_{\alpha}\vec{\theta} \cdot \vec{\phi}(s_{i-1}, \alpha)$

Greedy Local Method

WestlakeNLP

- NEXT ACTION: $Shift$

σ β

He does it here

σ: stack β: buffer

Greedy Local Method

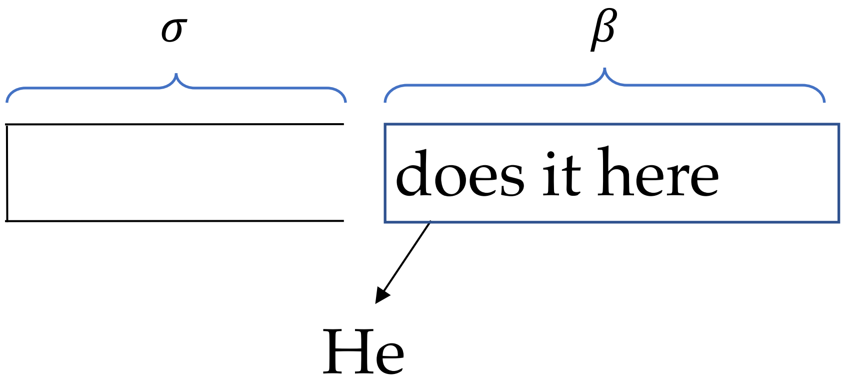
- NEXT ACTION: $Left - Arc$

$$\overbrace{\boxed{\text{He}}}^{\sigma} \quad \overbrace{\boxed{\text{does it here}}}^{\beta}$$

σ: stack β: buffer

Greedy Local Method

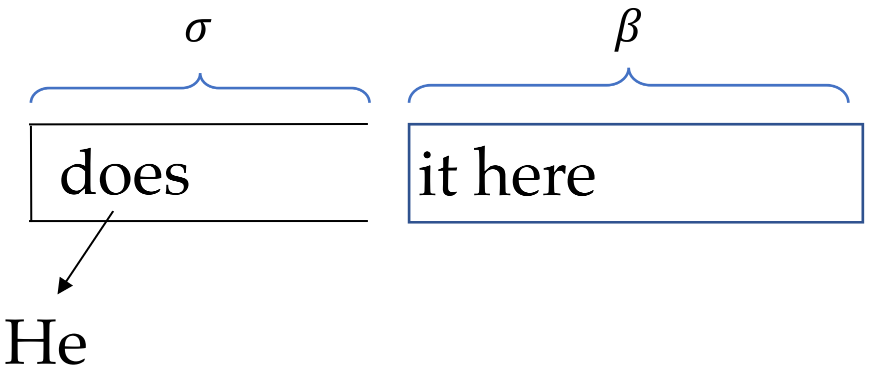
- NEXT ACTION: $Shift$



σ: stack β: buffer

Greedy Local Method

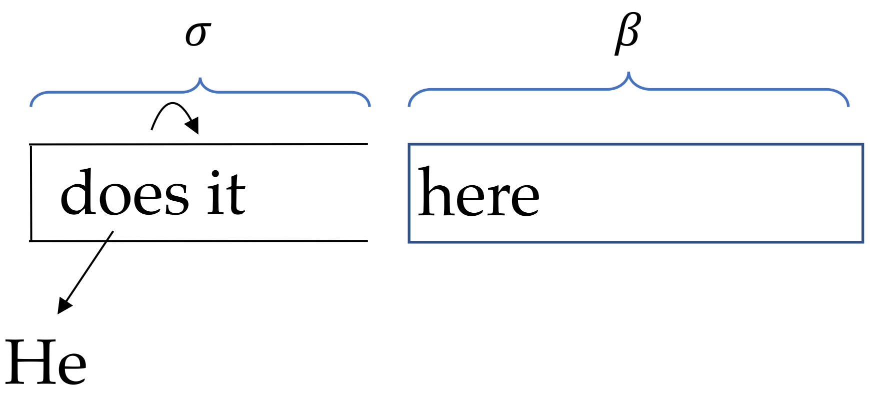
- NEXT ACTION: $Right - Arc$



σ: stack β: buffer

Greedy Local Method

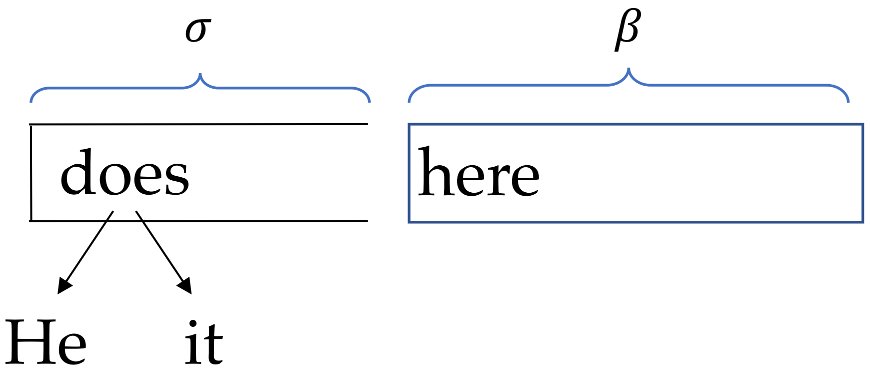
- NEXT ACTION: *Reduce*



σ: stack β: buffer

Greedy Local Method

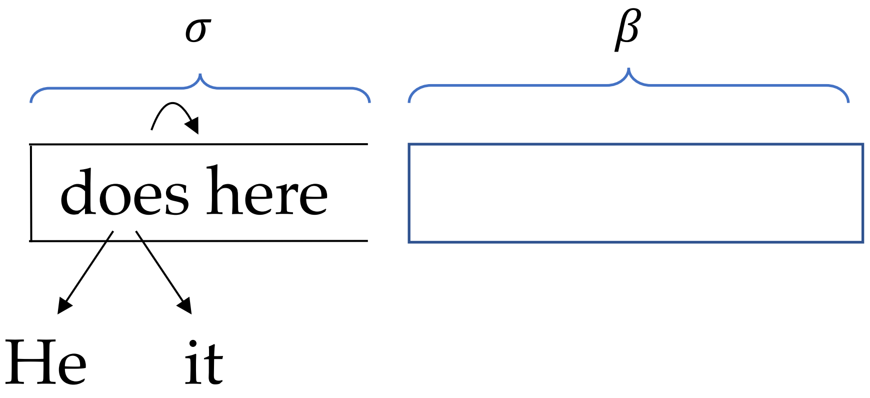
- NEXT ACTION: $Right - Arc$



σ: stack β: buffer

Greedy Local Method

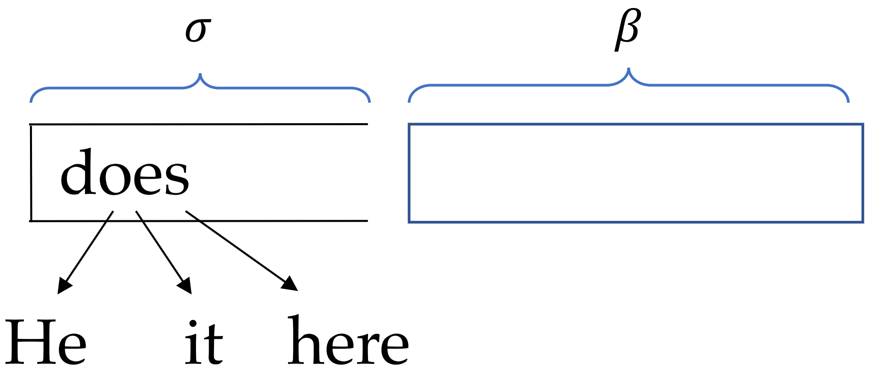
- NEXT ACTION: *Reduce*



σ: stack β: buffer

Greedy Local Method

- NEXT ACTION: *Finish*



σ: stack β: buffer

Greedy Local Method

- An Example

He does it here

Greedy Local Method

WestlakeNLP

- An Example

He does it here $\xrightarrow{S}$ He does it here

Greedy Local Method

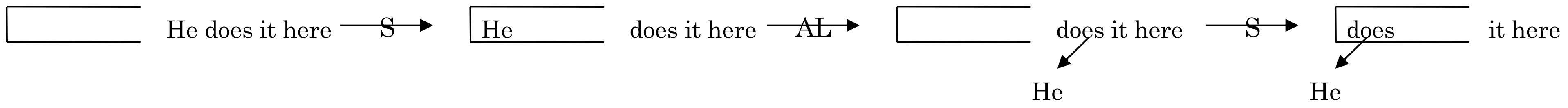
- An Example

[ ] He does it here $\xrightarrow{S}$ [He] does it here $\xrightarrow{AL}$ [ ] does it here

He

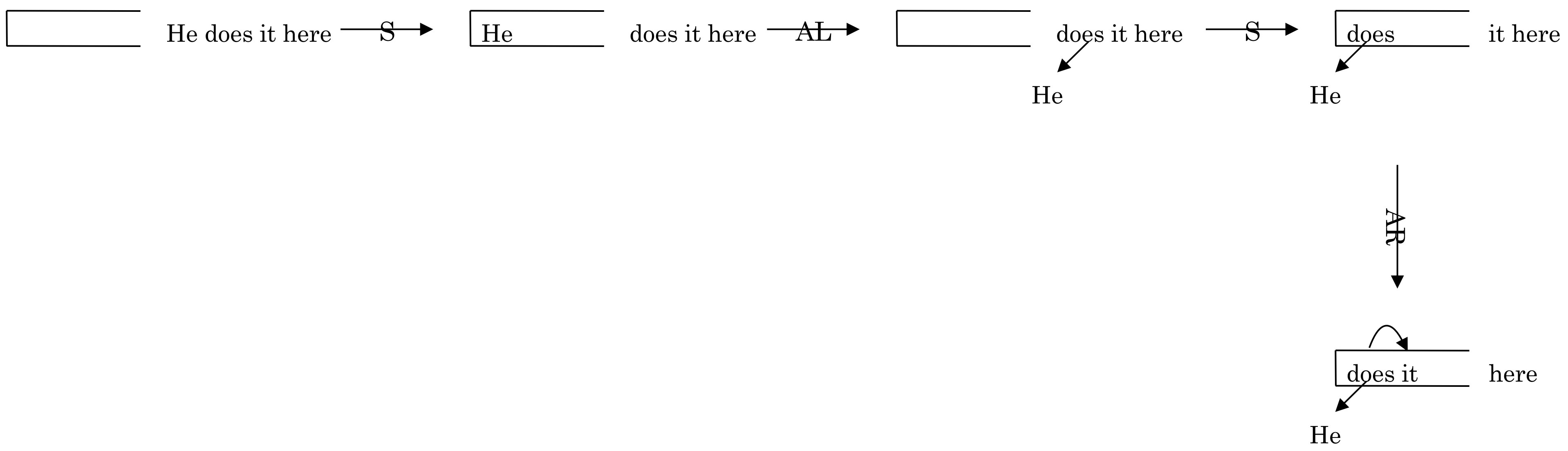
Greedy Local Method

- An Example



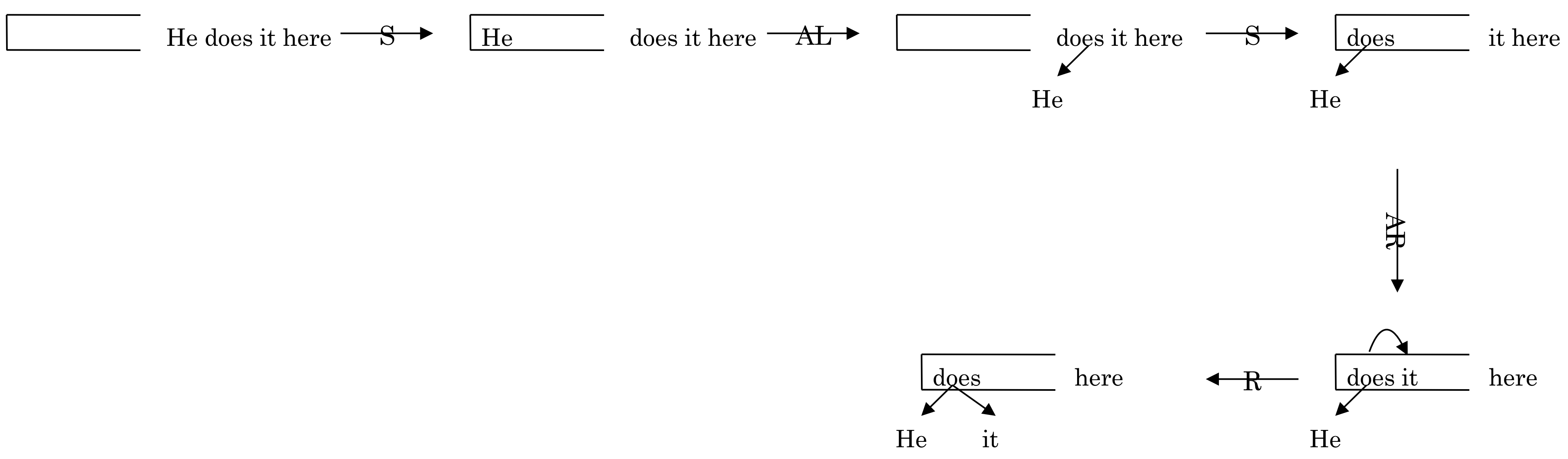
Greedy Local Method

- An Example



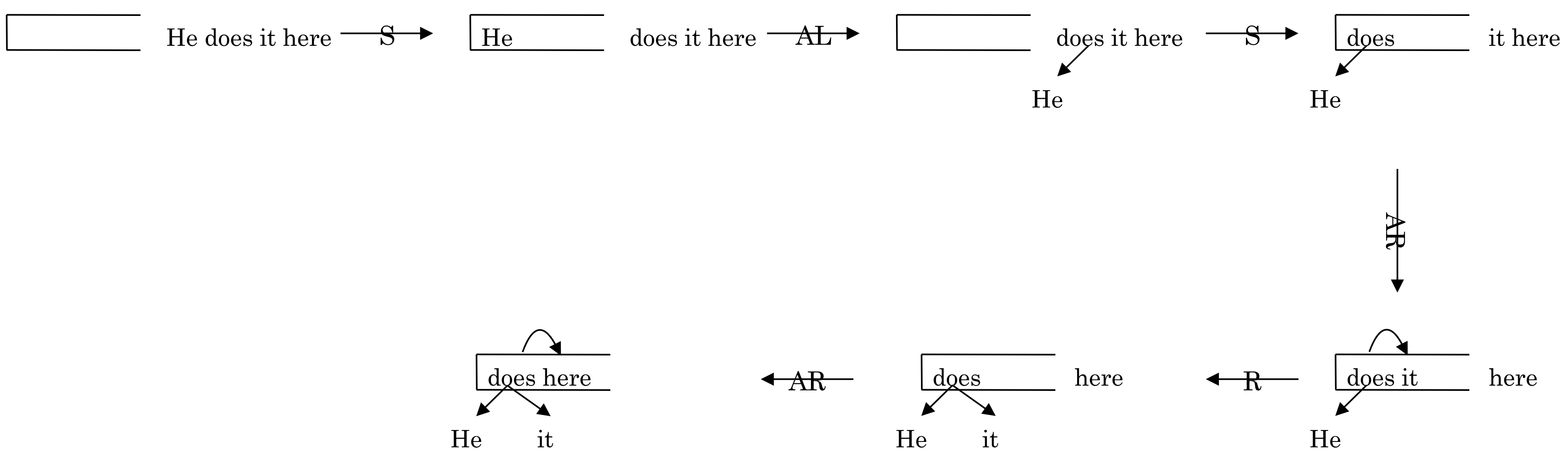
Greedy Local Method

- An Example



Greedy Local Method

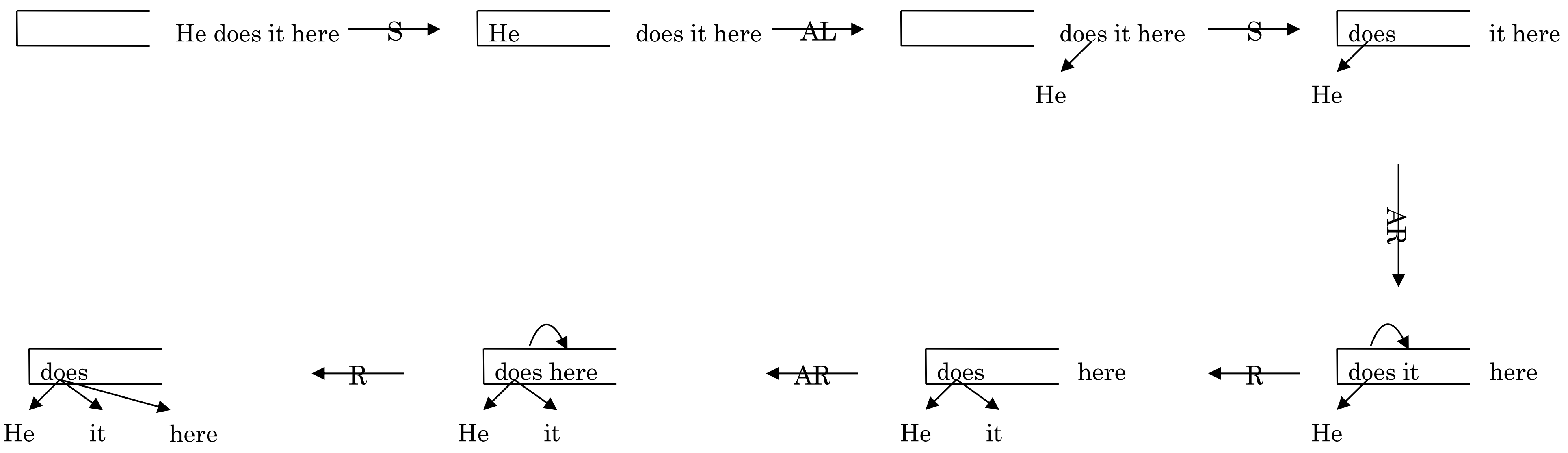
- An Example



Greedy Local Method

WestlakeNLP

- An Example



Problem of Greedy Local Modeling

WestlakeNLP

- In a globally optimal action sequence, each action may not necessarily be the optimal choice locally.
- Result in error propagation.
- Model does not see incorrect states during training.

Contents

WestlakeNLP

Problem of Greedy Local Modeling

WestlakeNLP

- In a globally optimal action sequence, each action may not necessarily be the optimal choice locally.
- Result in error propagation.

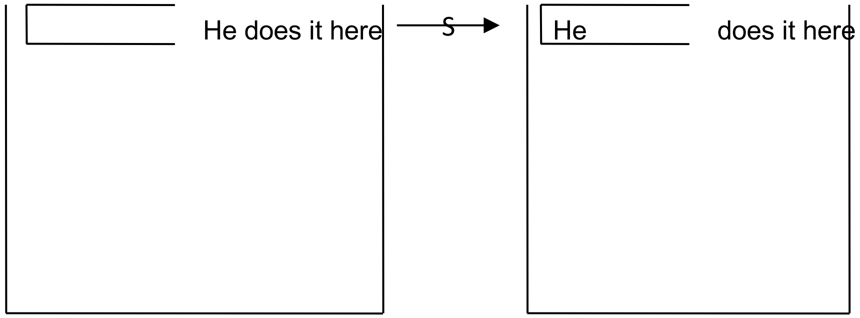
Solution:

- Beam search use a globally trained model:
  - Given an input X, a global transition-based model calculates $score(A|X)$ directly, where $A_{1:|A|} = a_1 a_2 \ldots a_{|A|}$ is a sequence of transition actions a_i for building an output structure for X.

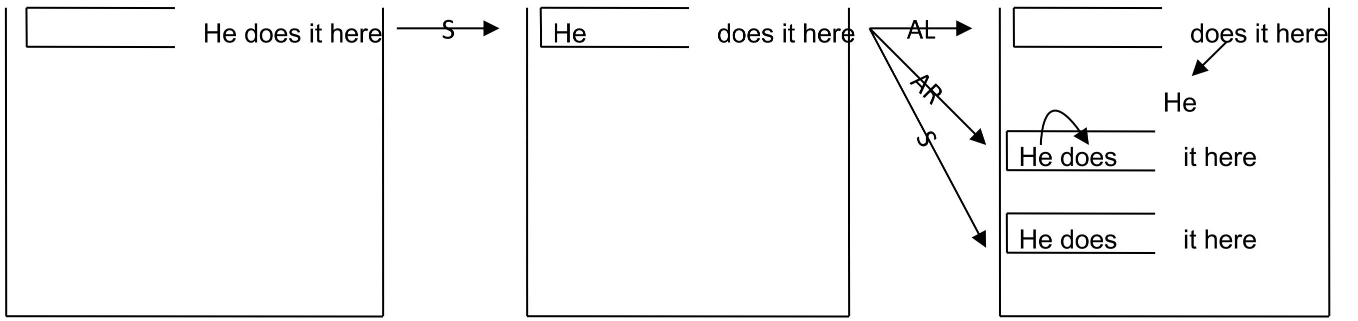
Beam search decoding example

He does it here

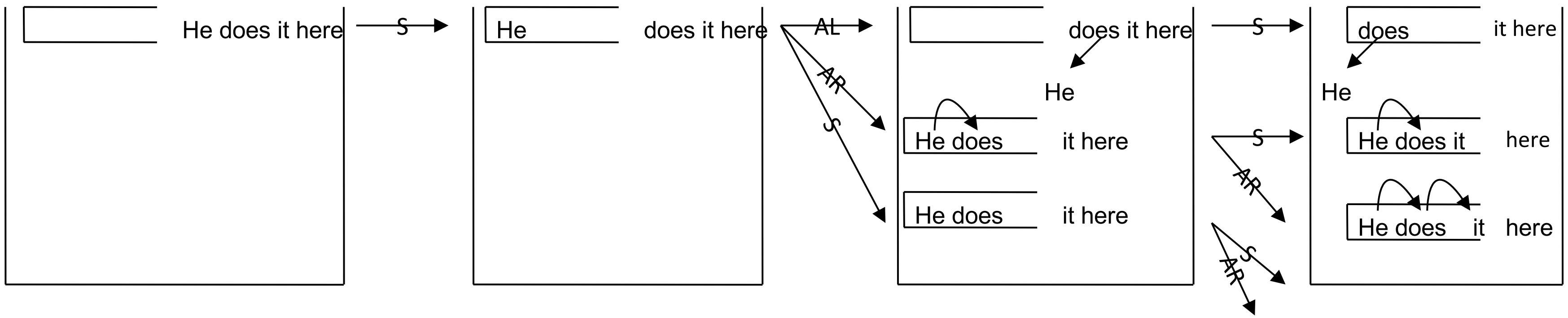
Beam search decoding example



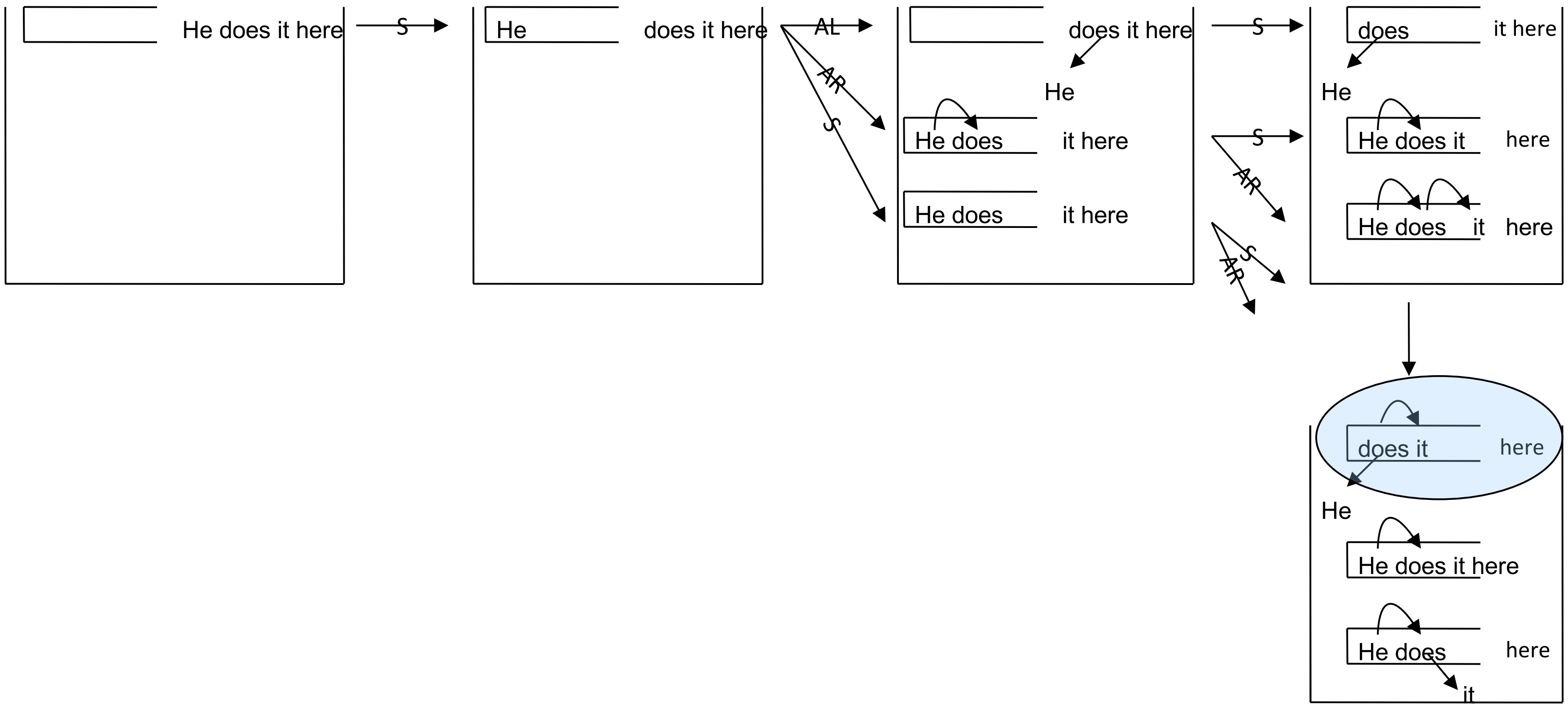
Beam search decoding example



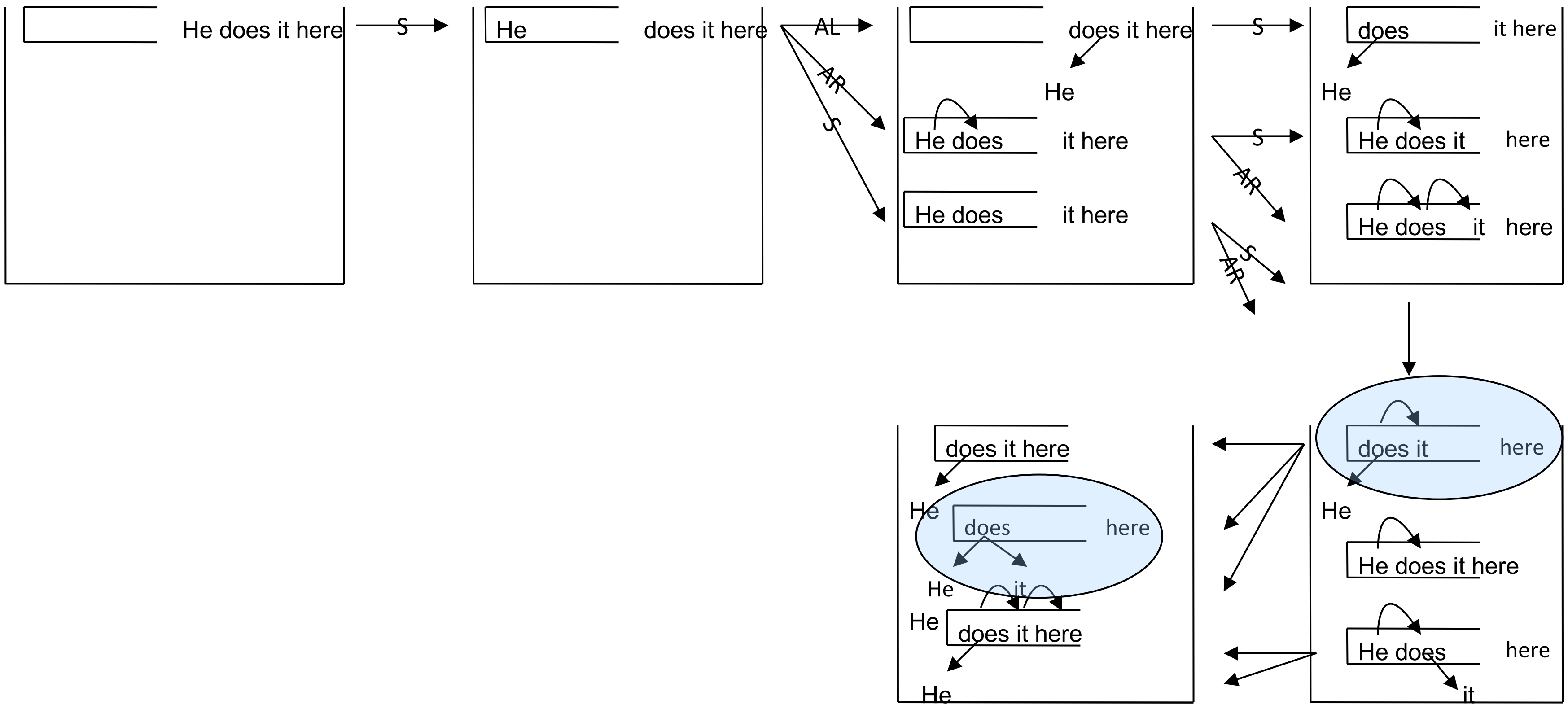
Beam search decoding example



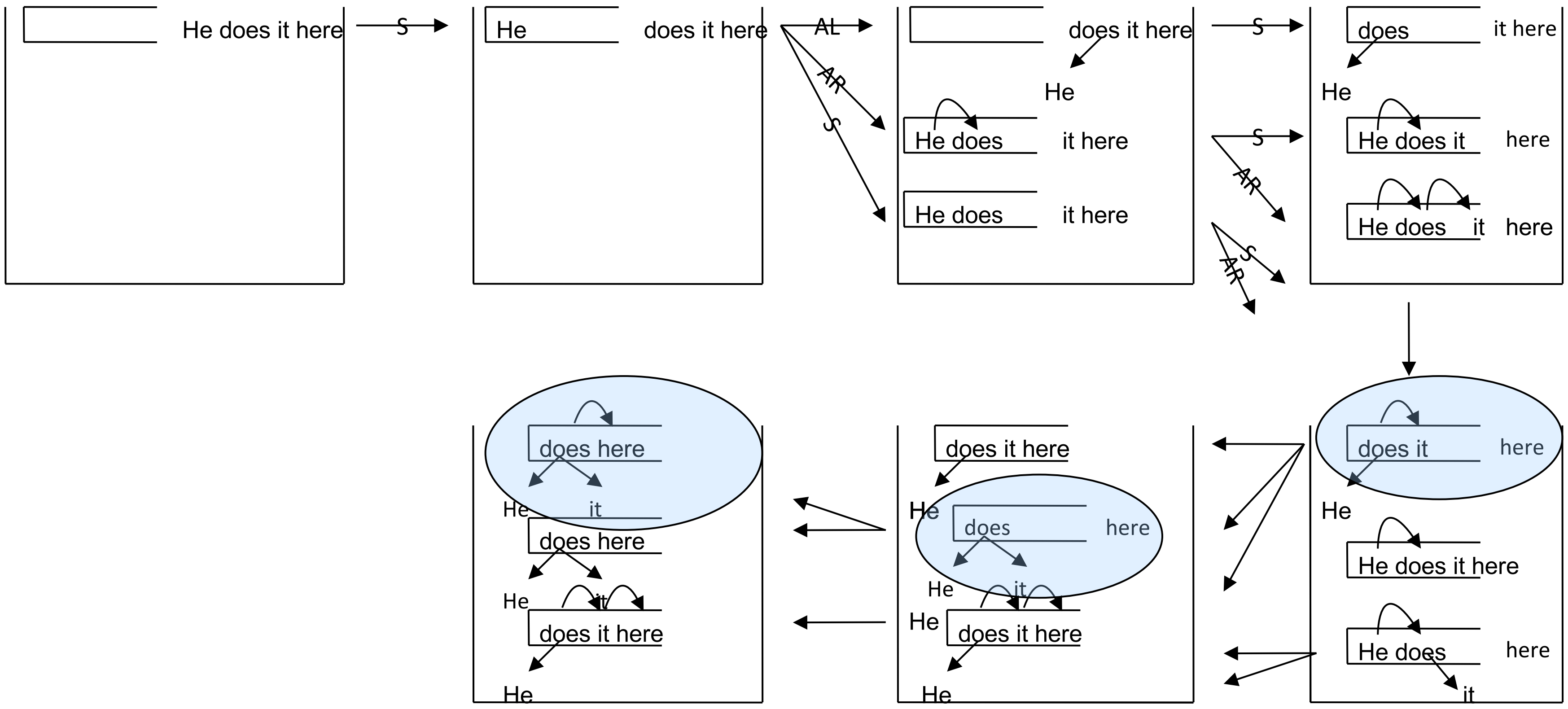
Beam search decoding example



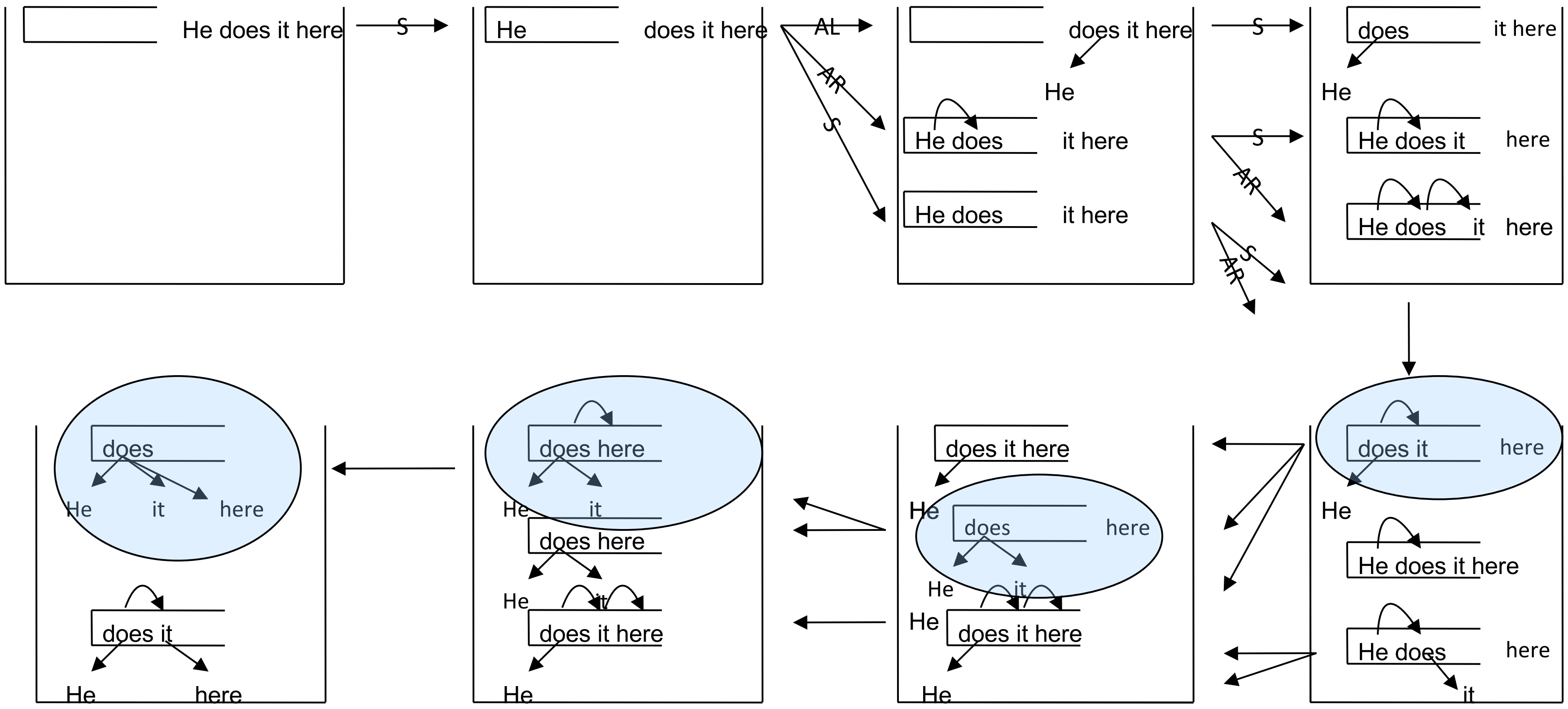
Beam search decoding example



Beam search decoding example



Beam search decoding example



Solution

WestlakeNLP

- Use global linear model to calculate $score(A|X)$

$$score(A|X) = \vec{\theta} \cdot \vec{\phi}(A, X)$$

decompose $\vec{\phi}(A|X)$ for incremental decoding

$\vec{\phi}(A|X) = \sum_{i=1}^{|A|} \vec{\phi}(a_i, s_{i-1})$

$$score(A|X) = \vec{\theta} \cdot \vec{\phi}(A, X) = \vec{\theta} \cdot \left(\sum_{i=1}^{|A|} \vec{\phi}(a_i, s_{i-1})\right)$$

$$= \sum_{i=1}^{|A|} \left(\vec{\theta}\right\} \cdot \left\{\vec{\phi}(a_i, s_{i-1})\right)$$

Beam search decoding algorithm

WestlakeNLP

Inputs: $\vec{\theta}$ —discriminative linear model parameters;
X — task input;
K — beam size;
Initialization: $agenda \leftarrow [(\textsc{StartState}(X), 0)]$;
Algorithm:
while **not** $\textsc{AllTerminal}(agenda)$ **do**
  $to\_expand \leftarrow agenda$;
  $agenda \leftarrow []$;
  for $(state, score) \in to\_expand$ **do**
    for $a \in \textsc{PossibleActions}(state)$ **do**
      $new\_state \leftarrow \textsc{Expand}(state, a)$;
      $new\_score \leftarrow score + \vec{\theta} \cdot \vec{\phi}(state, a)$;
      $\textsc{Append}(agenda, (new\_state, new\_score))$;
  $agenda \leftarrow \textsc{Top-k}(agenda, K)$;
Output: $\textsc{Top-k}(agenda, 1)[0]$;

Beam search decoding example

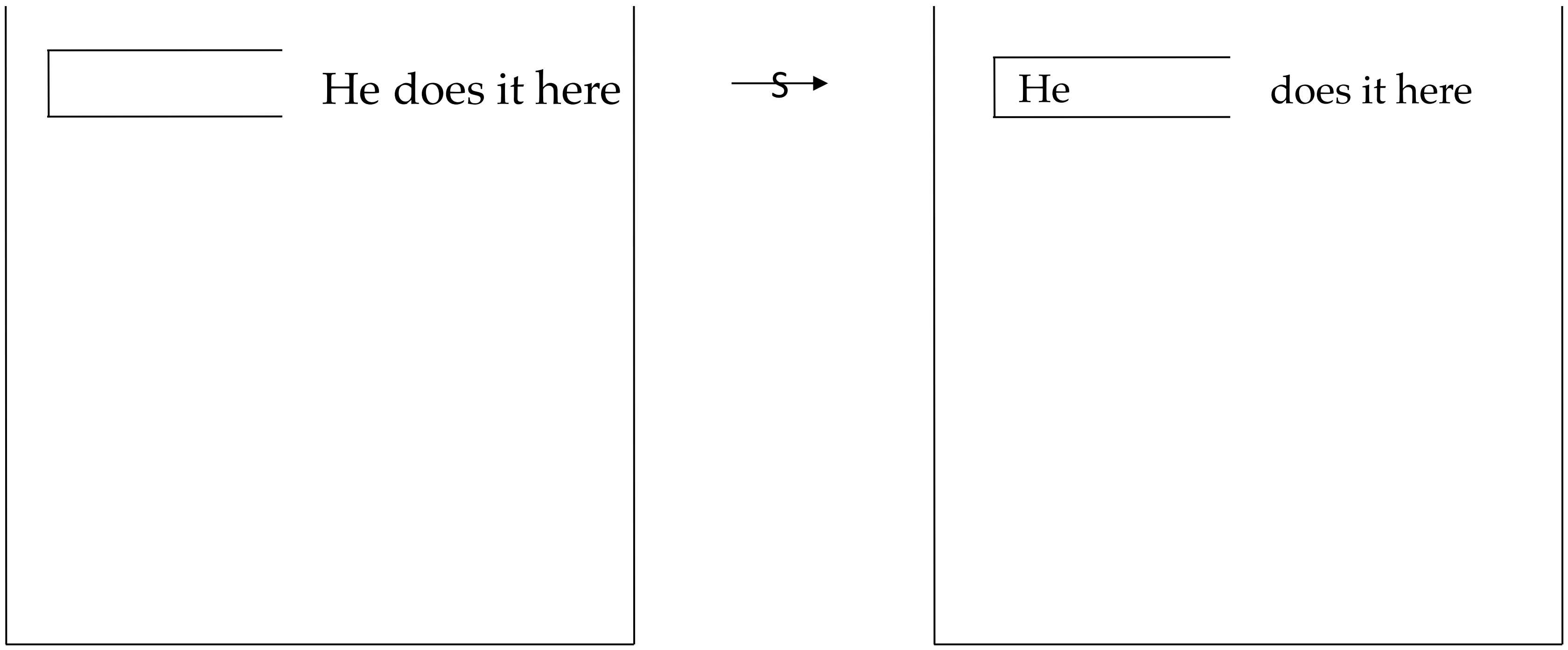
WestlakeNLP

- Dependency parsing

He does it here

Beam search decoding example

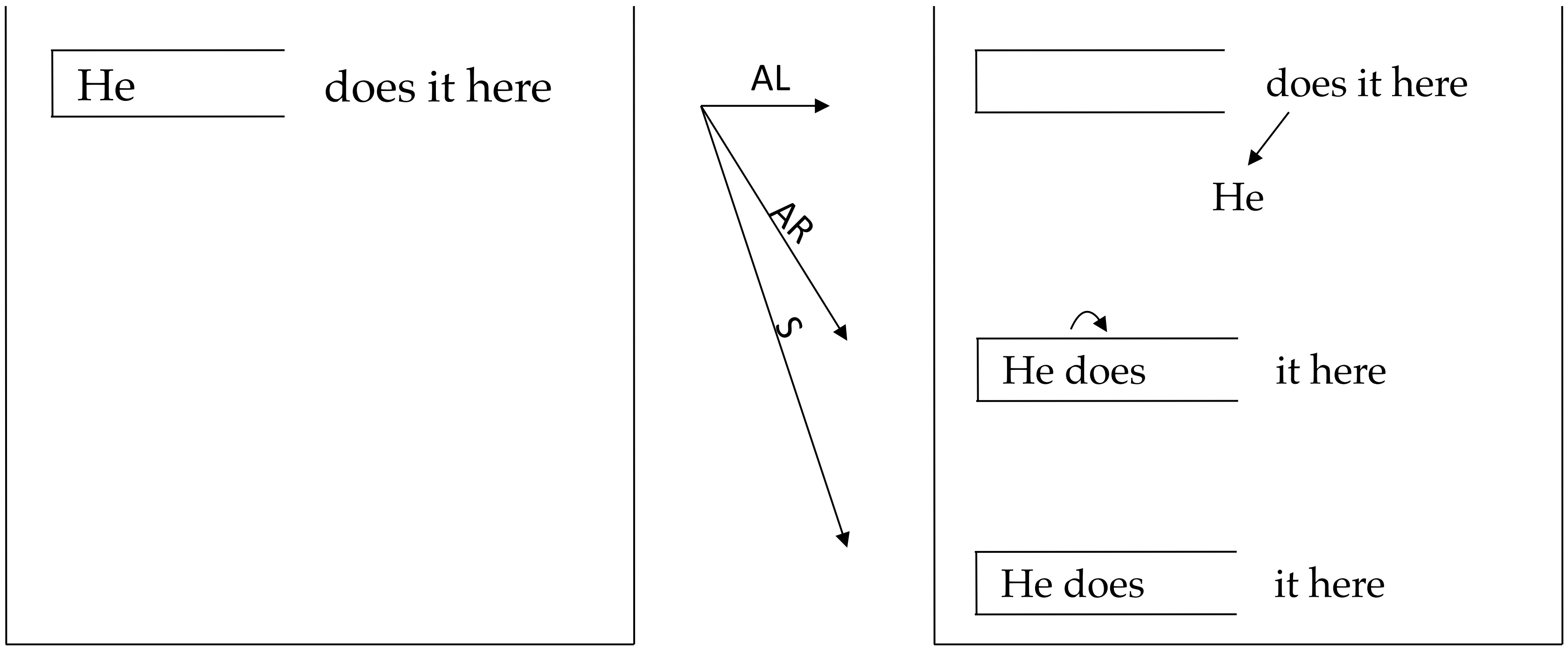
- Dependency parsing



S-SHIFT

Beam search decoding example

- Dependency parsing

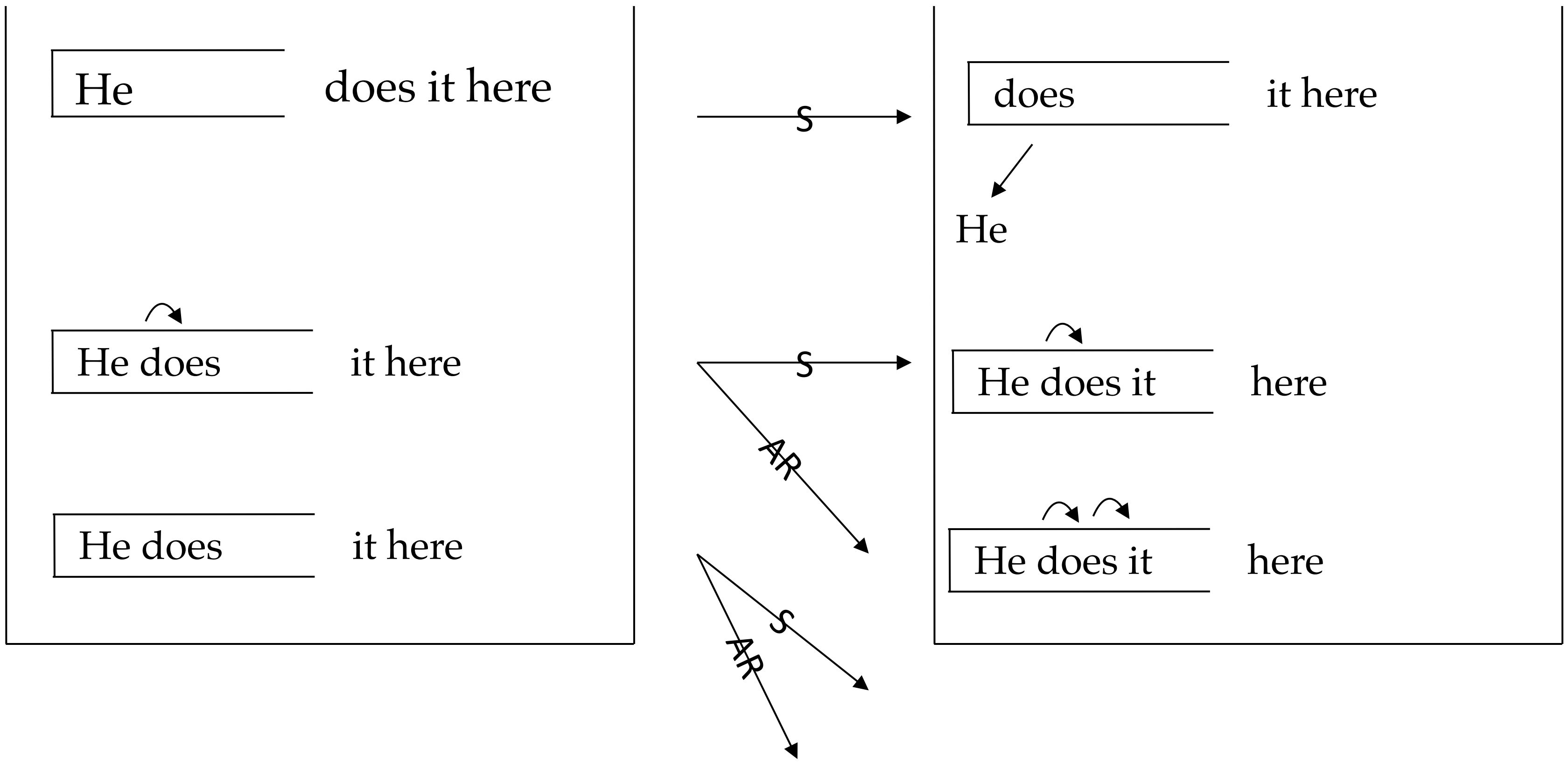


AL—LEFT-ARC

AR—RIGHT-ARC

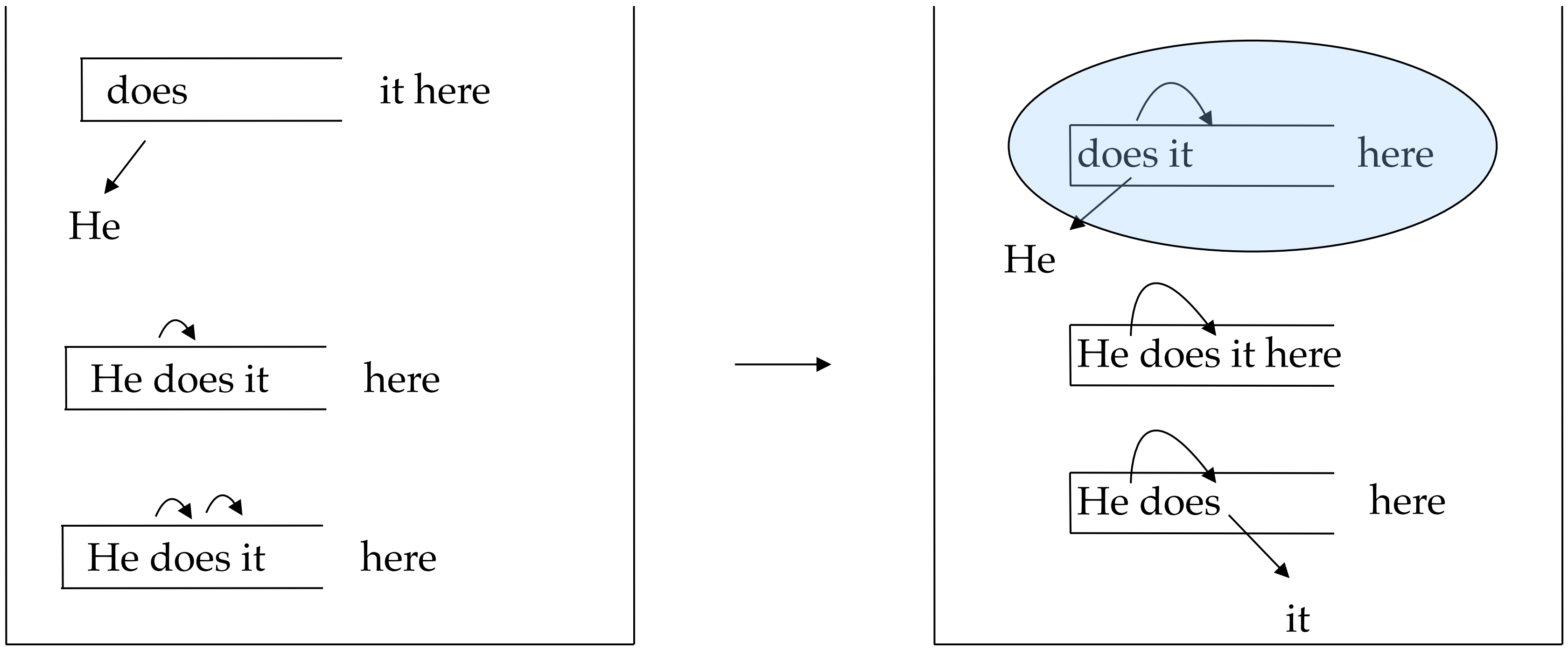
Beam search decoding example

- Dependency parsing



Beam search decoding example

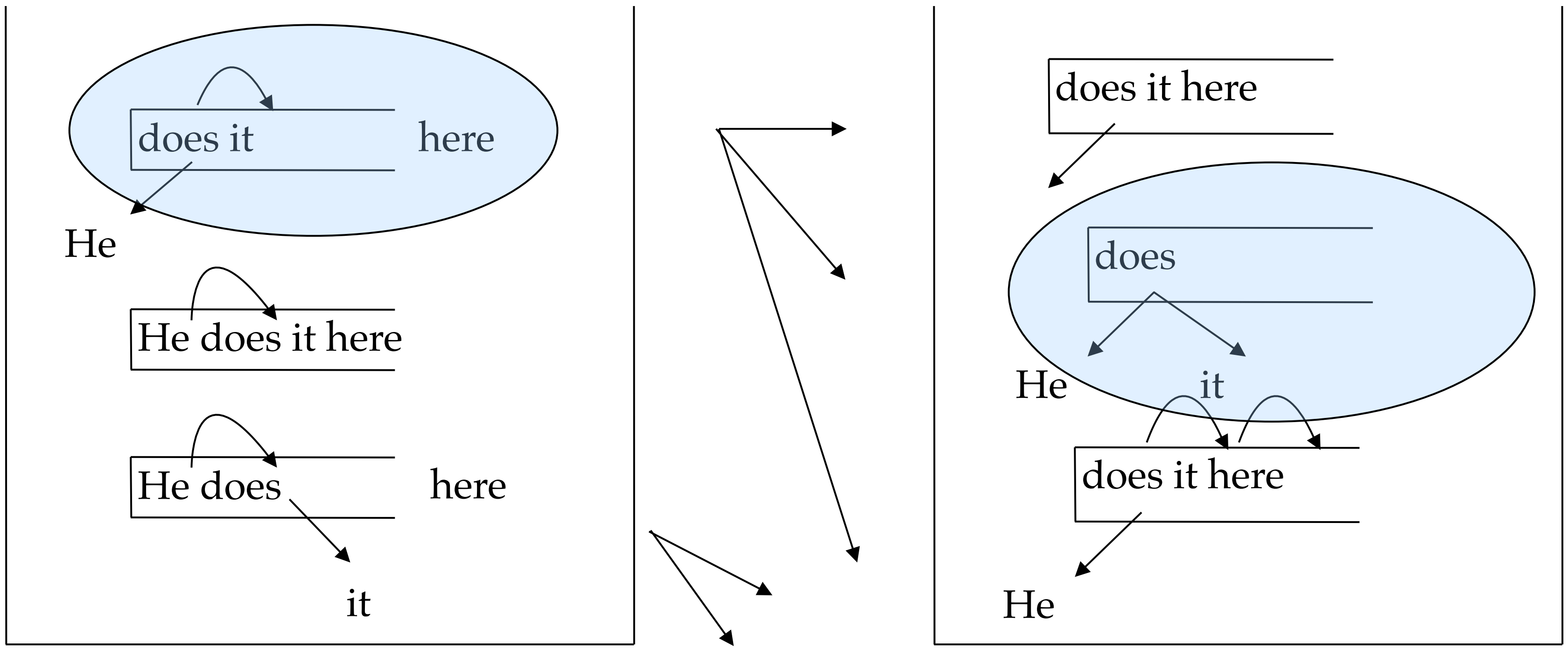
- Dependency parsing



Beam search decoding example

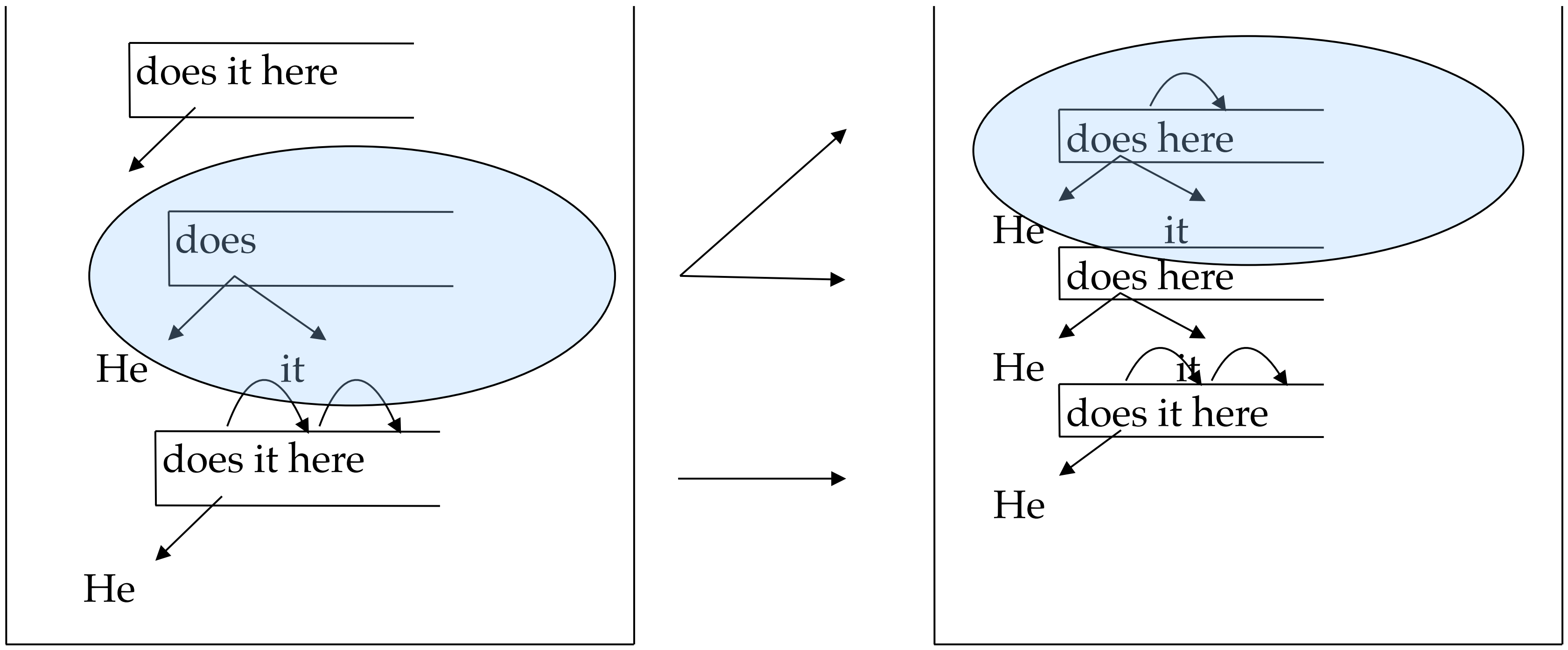
WestlakeNLP

- Dependency parsing



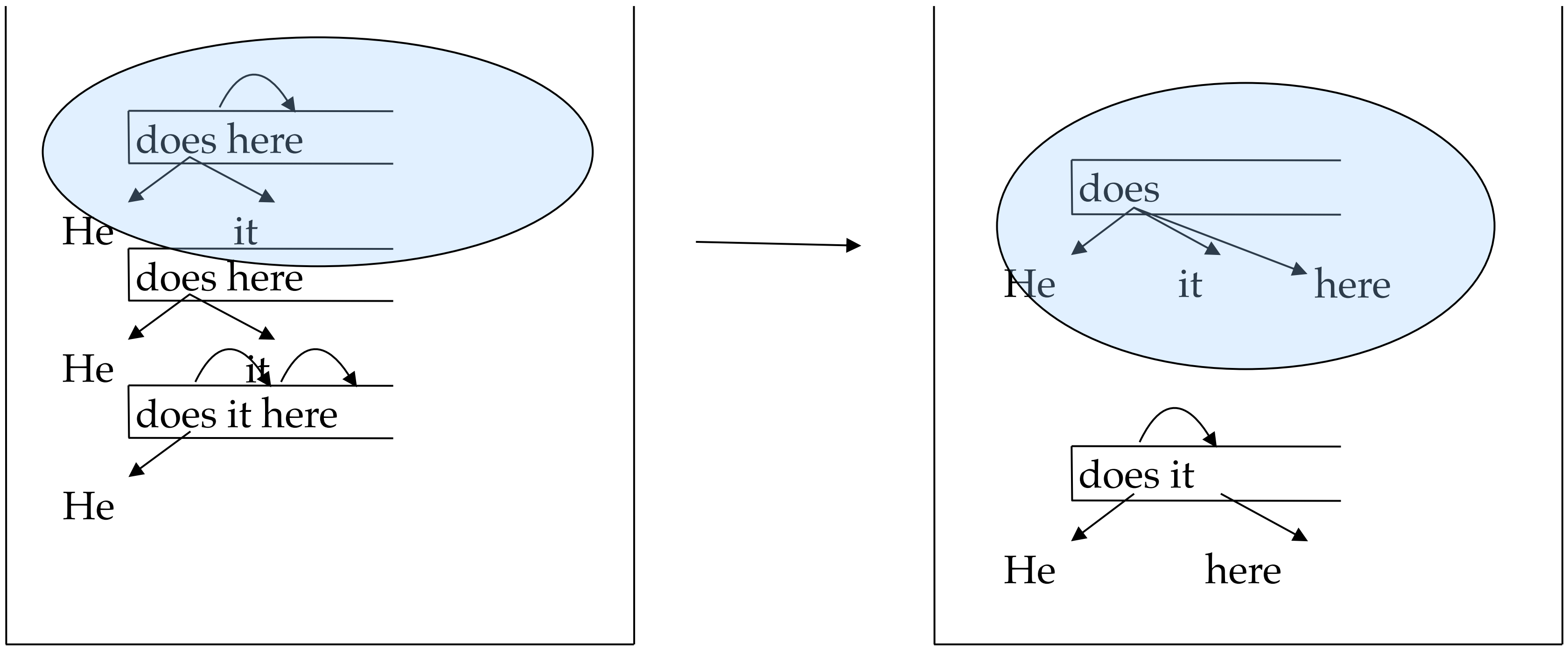
Beam search decoding example

- Dependency parsing

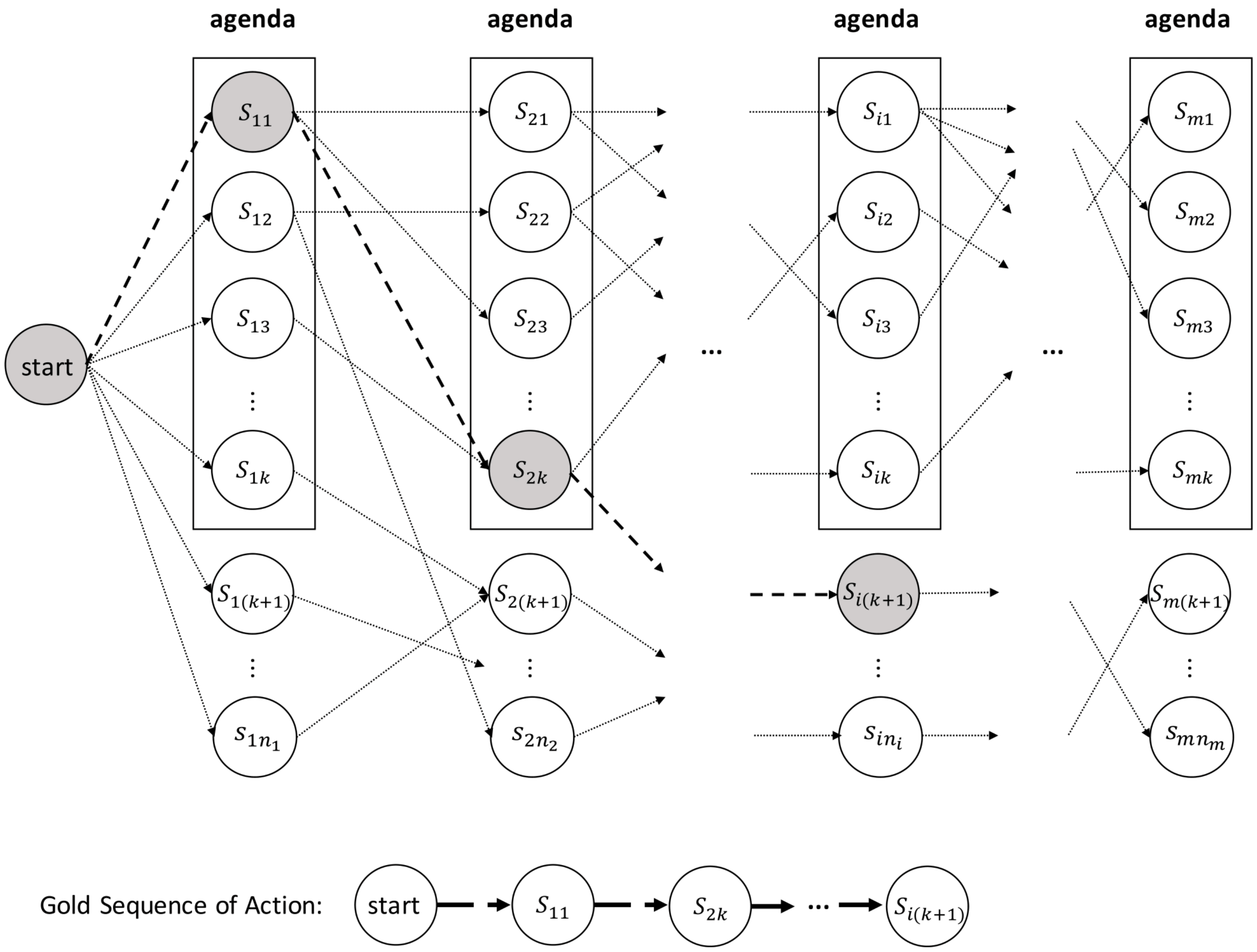


Beam search decoding example

- Dependency parsing



Beam search decoding example



Beam search training algorithm

WestlakeNLP

Inputs: D — gold standard training set;
K — beam size;
T — number of training iterations;
Initialisation: $\vec{\theta} \leftarrow 0$;
Algorithm:
for $t \in [1, \ldots, T]$ **do**
  for $(X, Y) \in D$ **do**
    $G \leftarrow$ GoldActionSeq(X, Y);
    $agenda \leftarrow$ [(StartState(X), 0)];
    $gold\_state \leftarrow$ StartState(X);
    $i \leftarrow 0$;
    while not AllTerminal$(agenda)$ **do**
      $i \leftarrow i + 1$;
      $to\_expand \leftarrow agenda$;
      $agenda \leftarrow []$;
      for $(state, score) \in to\_expand$ **do**
        for $a \in$ PossibleActions$(state)$ **do**
          $new\_state \leftarrow$ Expand$(state, a)$;
          $new\_score \leftarrow score + \vec{\theta} \cdot \vec{\phi}(state, a)$;
          Append$(agenda, (new_state, new_score))$;
      $agenda \leftarrow$ Top-k$(agenda, K)$;
      $gold\_state \leftarrow$ Expand$(gold_state, G[i])$;
      if not Contain$(agenda, gold_state)$ **then**
        $pos \leftarrow gold\_state$;
        $neg \leftarrow$ Top-k$(agenda, 1)[0]$;
        $\vec{\theta} \leftarrow \vec{\theta} + \vec{\phi}(pos) - \vec{\phi}(neg)$;
        continue((W, G) $\in D$)
    if $gold\_state \neq$ Top-k$(agenda, 1)[0]$ **then**
      $\vec{\theta} \leftarrow \vec{\theta} + \vec{\phi}(gold_state) - \vec{\phi}($Top-k$(agenda, 1)[0])$;
Output: $\vec{\theta}$;

Contents

WestlakeNLP

Transition-based constituent parsing

WestlakeNLP

- Three steps in transition-based modeling
  - Find the state transition process
  - Define global feature vector
  - Apply the standard learning and search framework

Transition-based constituent parsing

WestlakeNLP

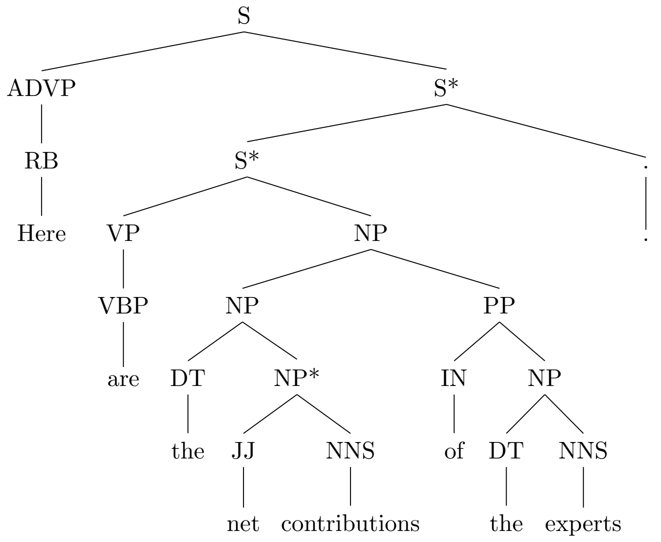
- **State**

  σ: stack of partially constituent outputs.

  β: buffer of the next incoming words.

- **actions**

  SHIFT, REDUCE-L/R-X, UNARY-X, IDLE



Transition-based constituent parsing

WestlakeNLP

Shift-reduce constituent parsing

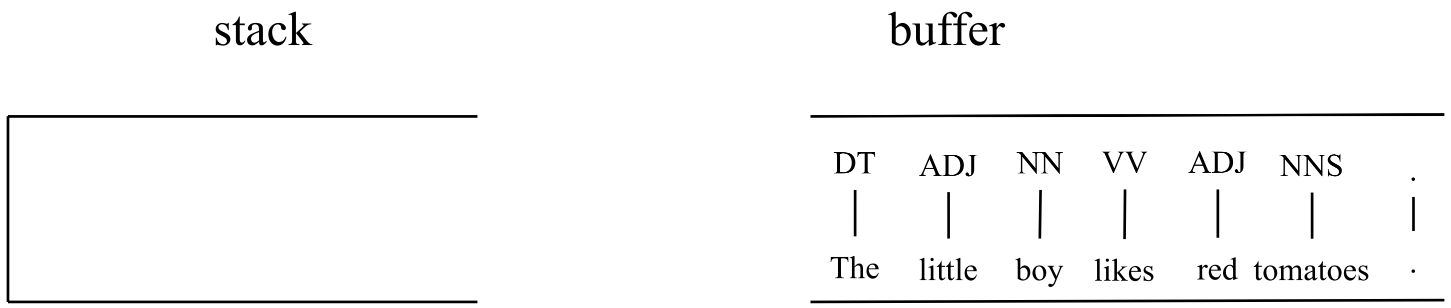
Axiom: Goal:	$([\,],\ W_{1:n})$ $(\sigma,\ [\,])$	SHIFT:	$\dfrac{(\sigma,\ w_0\|\beta)}{(\sigma\|w_0,\ \beta)}$
REDUCE-L-X:	$\dfrac{(\sigma\|s_1\|s_0,\ \beta)}{\left(\sigma\middle\| \begin{array}{c} \text{X} \\ \swarrow \quad \diagdown \\ s_1 \qquad s_0 \end{array},\ \beta\right)}$	REDUCE-R-X:	$\dfrac{(\sigma\|s_1\|s_0,\ \beta)}{\left(\sigma\middle\| \begin{array}{c} \text{X} \\ \diagup \quad \searrow \\ s_1 \qquad s_0 \end{array},\ \beta\right)}$
UNARY-X:	$\dfrac{(\sigma\|s_0,\ \beta)}{\left(\sigma\middle\| \begin{array}{c} \text{X} \\ \downarrow \\ s_0 \end{array},\ \beta\right)}$	IDLE:	$\dfrac{(\sigma,\ [\,])}{(\sigma,\ [\,])}$

Transition-based constituent parsing

WestlakeNLP

Shift-reduce constituent parsing

- Example
  - Shift



Transition-based constituent parsing

WestlakeNLP

Shift-reduce constituent parsing

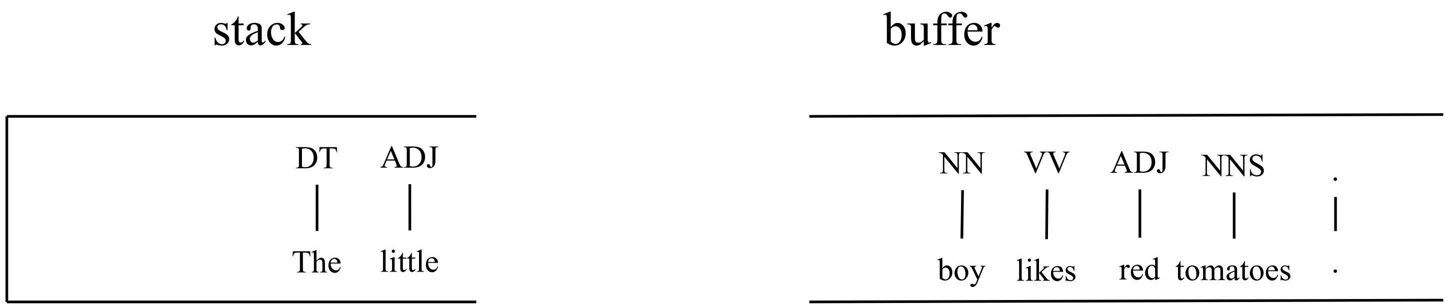
- Example
  - Shift

Transition-based constituent parsing

WestlakeNLP

Shift-reduce constituent parsing

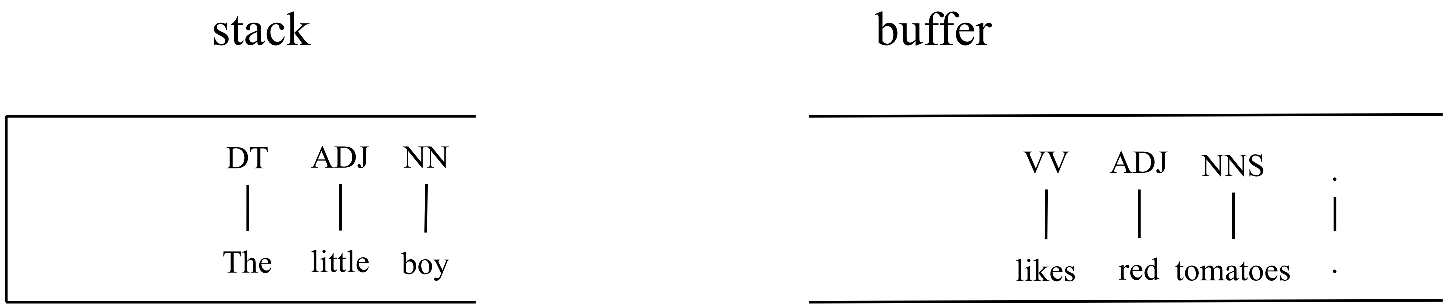
- Example
  - Shift



Transition-based constituent parsing

Shift-reduce constituent parsing

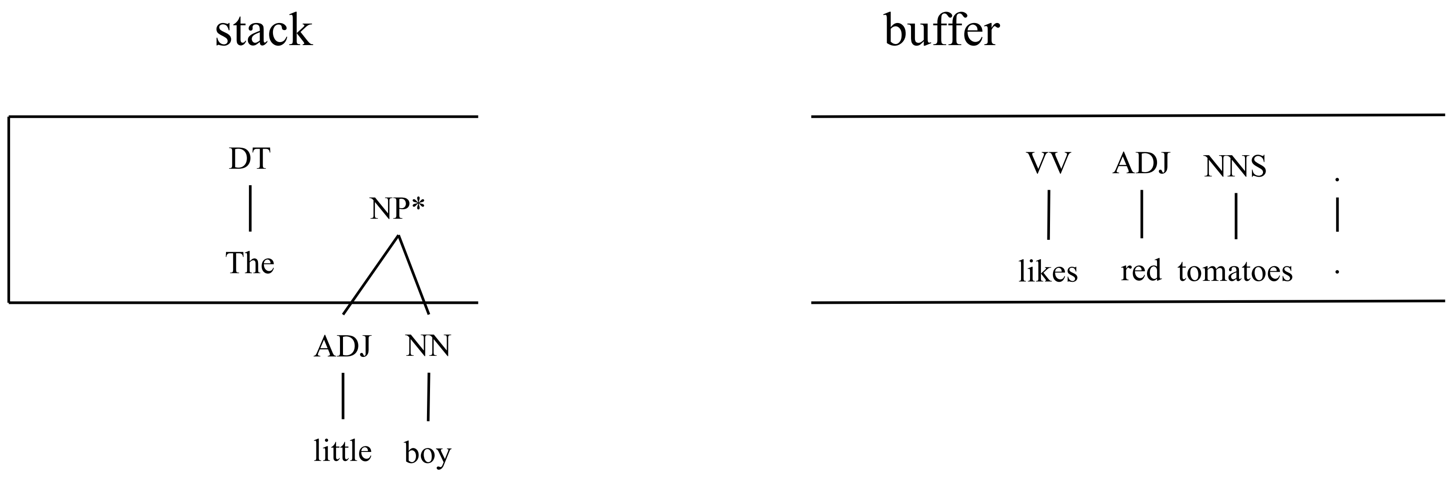
- Example
  - Reduce-R-NP



Transition-based constituent parsing

Shift-reduce constituent parsing

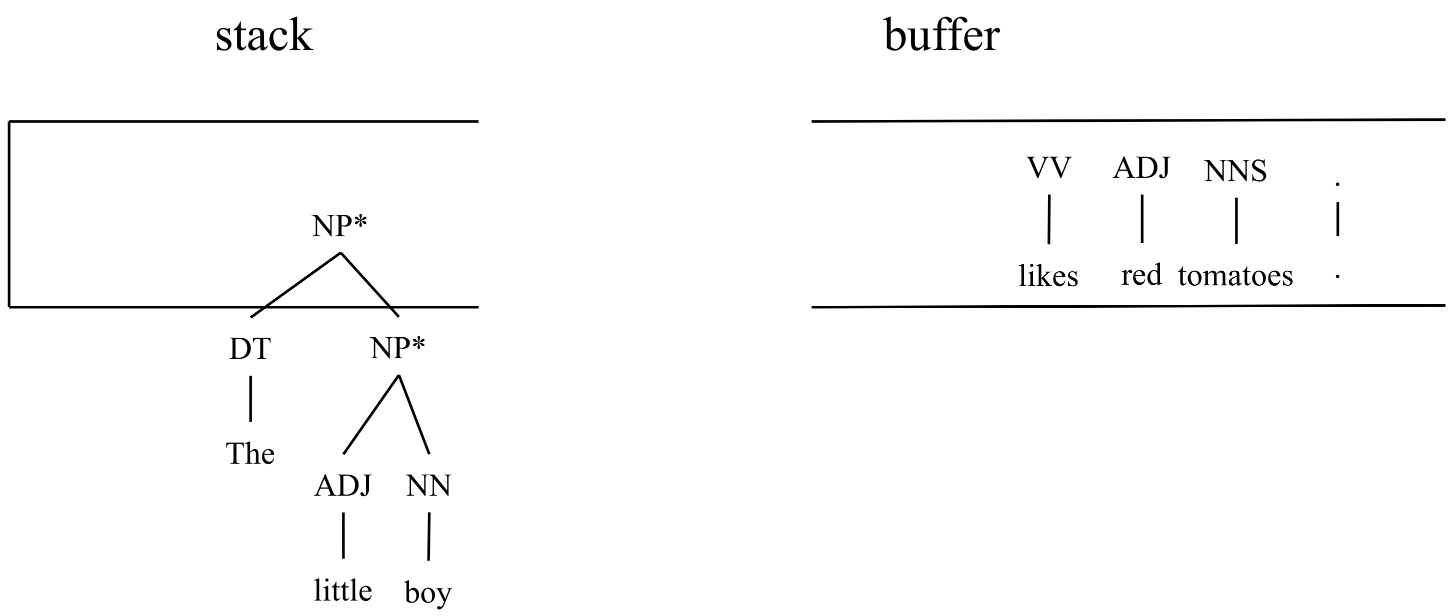
- Example
  - Reduce-R-NP



Transition-based constituent parsing

Shift-reduce constituent parsing

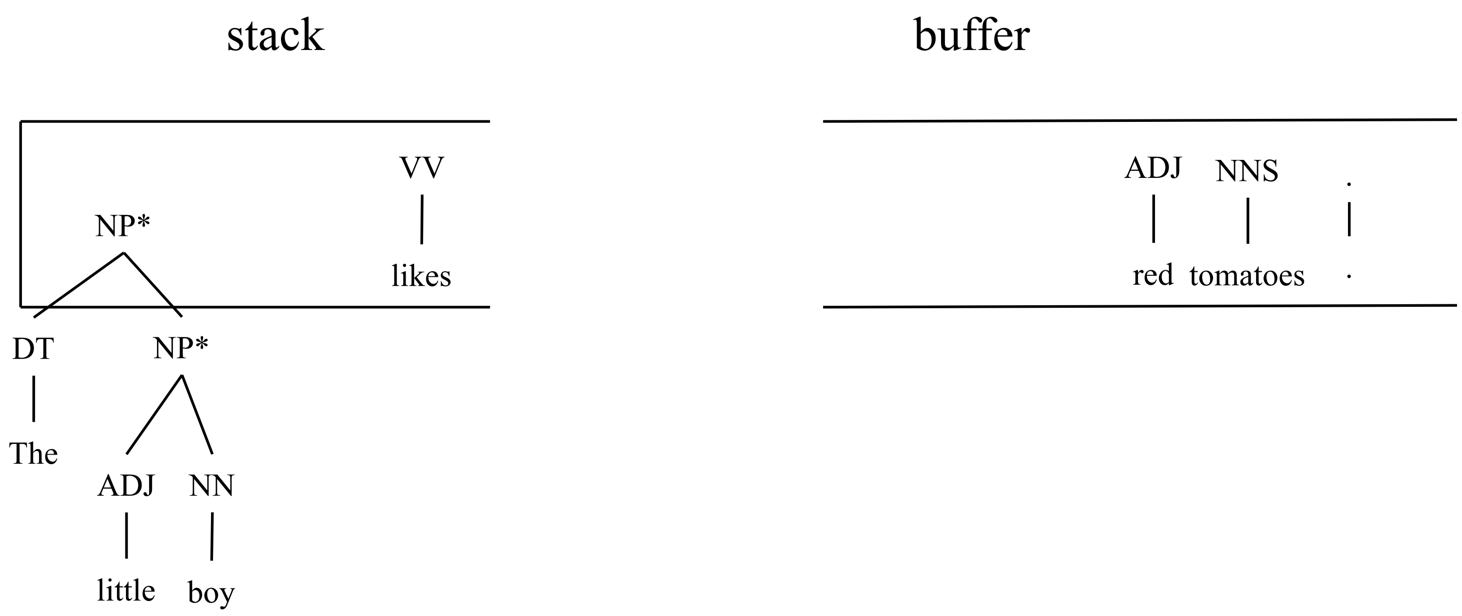
- Example
  - Shift



Transition-based constituent parsing

Shift-reduce constituent parsing

- Example
  - Shift

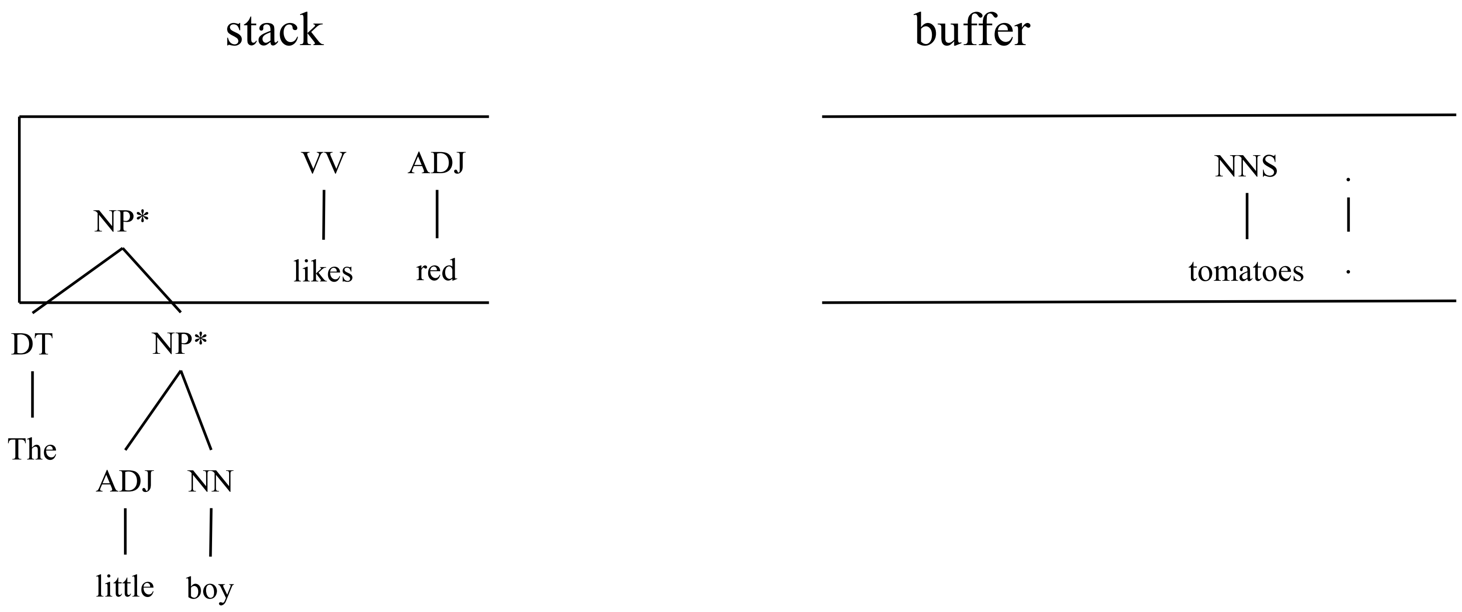


Transition-based constituent parsing

WestlakeNLP

Shift-reduce constituent parsing

- Example
  - Shift

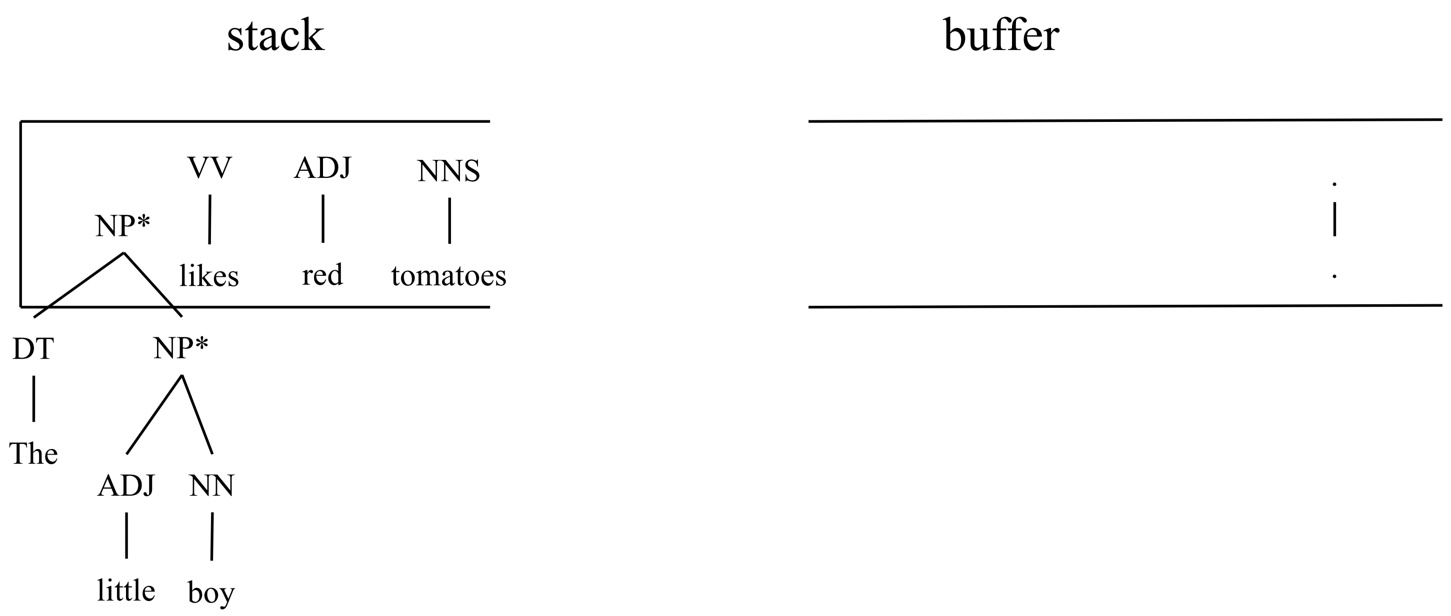


Transition-based constituent parsing

WestlakeNLP

Shift-reduce constituent parsing

- Example
  - Reduce-R-NP

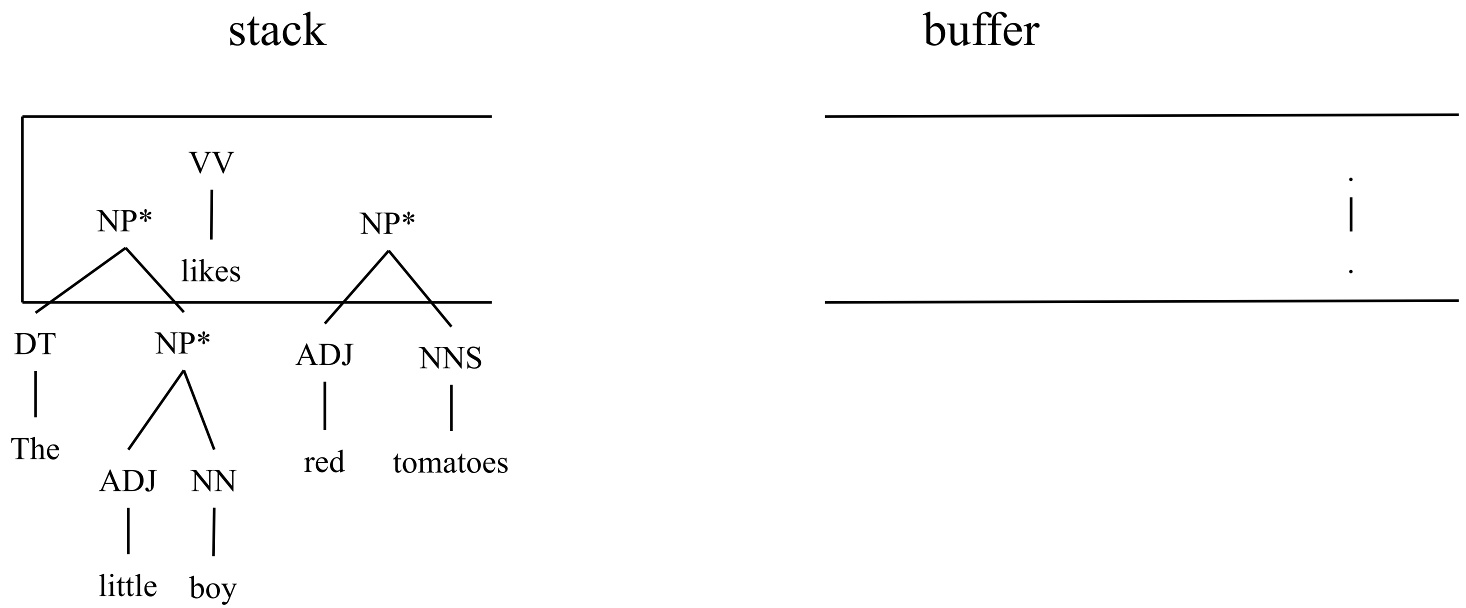


Transition-based constituent parsing

WestlakeNLP

Shift-reduce constituent parsing

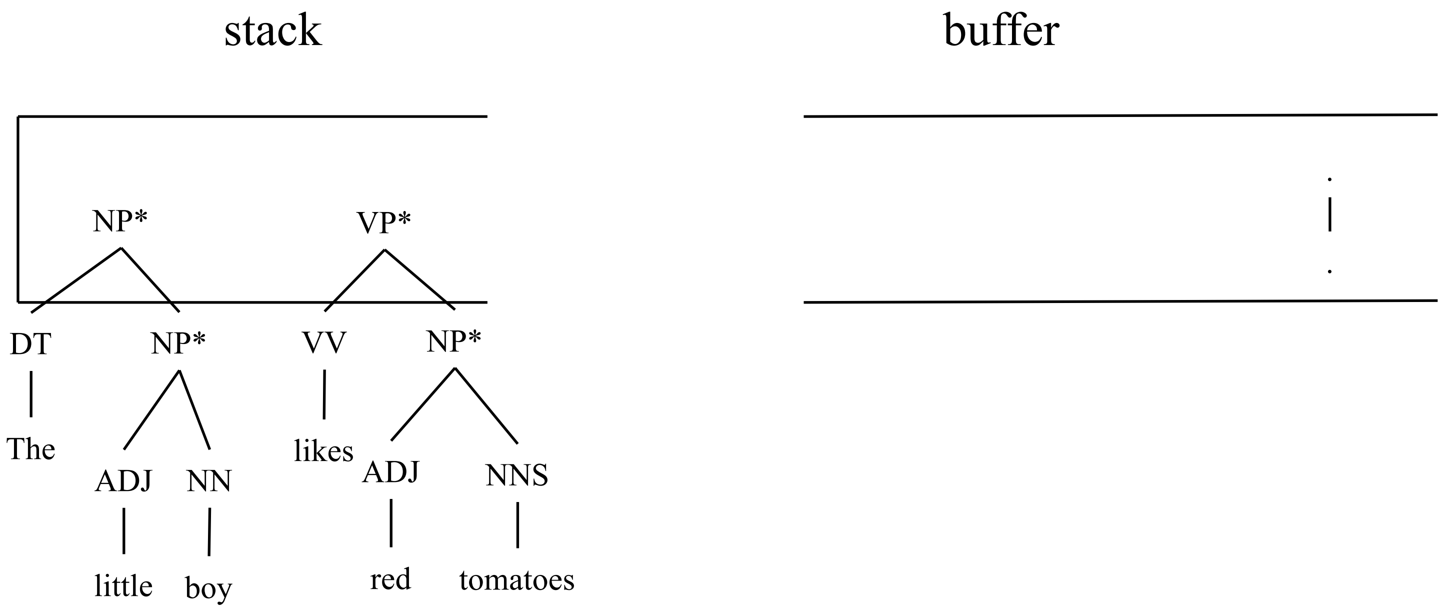
- Example
  - Reduce-L-VP



Transition-based constituent parsing

Shift-reduce constituent parsing

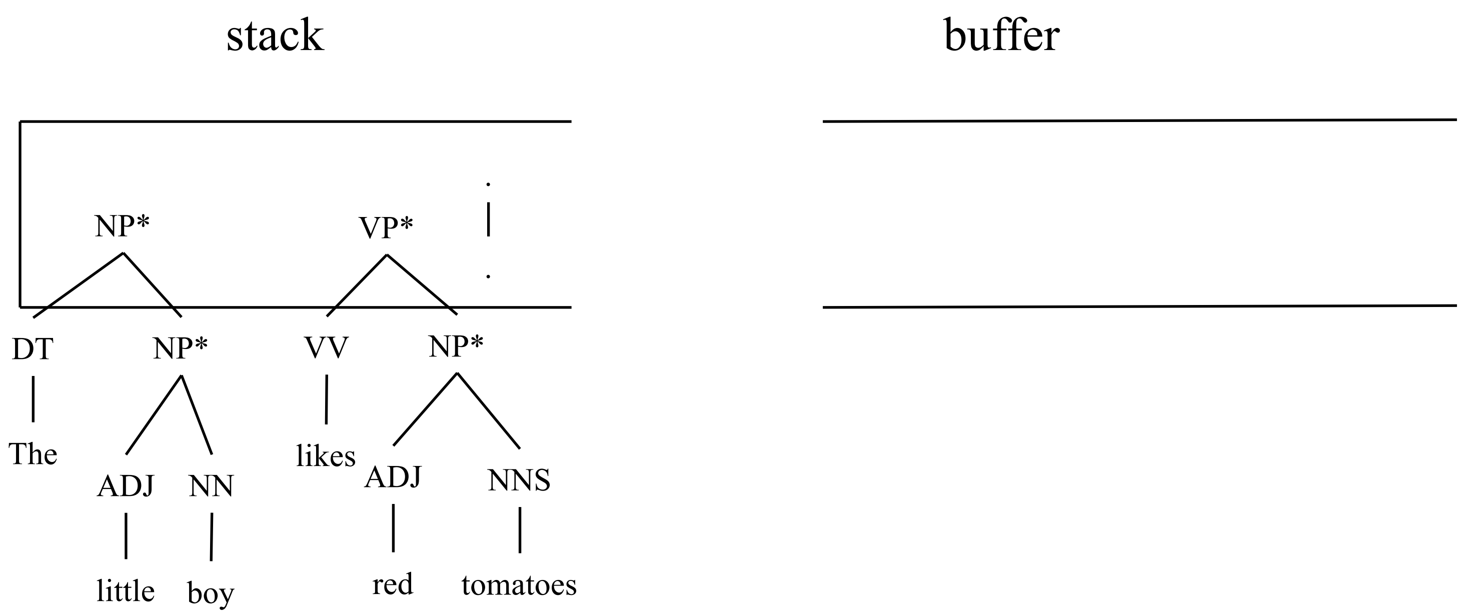
- Example
  - Shift



Transition-based constituent parsing

Shift-reduce constituent parsing

- Example
  - Reduce-L-S

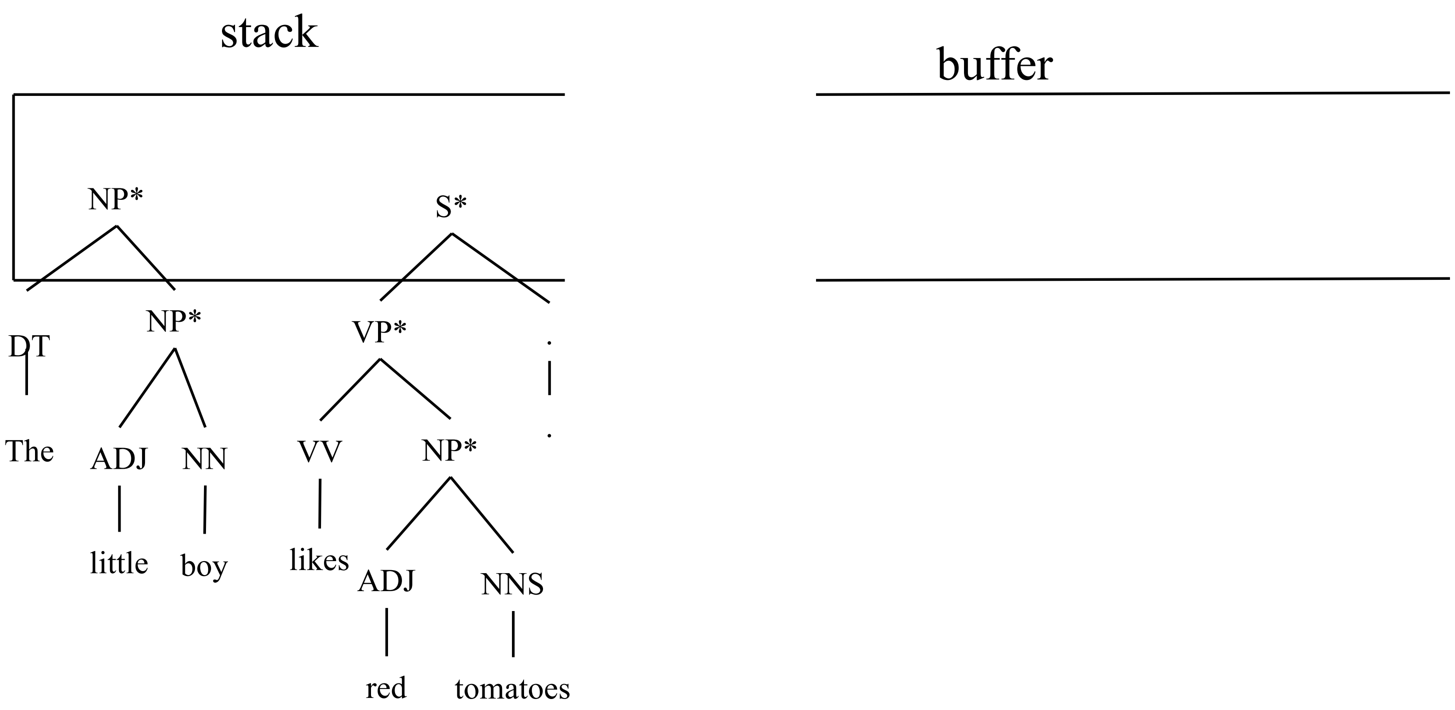


Transition-based constituent parsing

WestlakeNLP

Shift-reduce constituent parsing

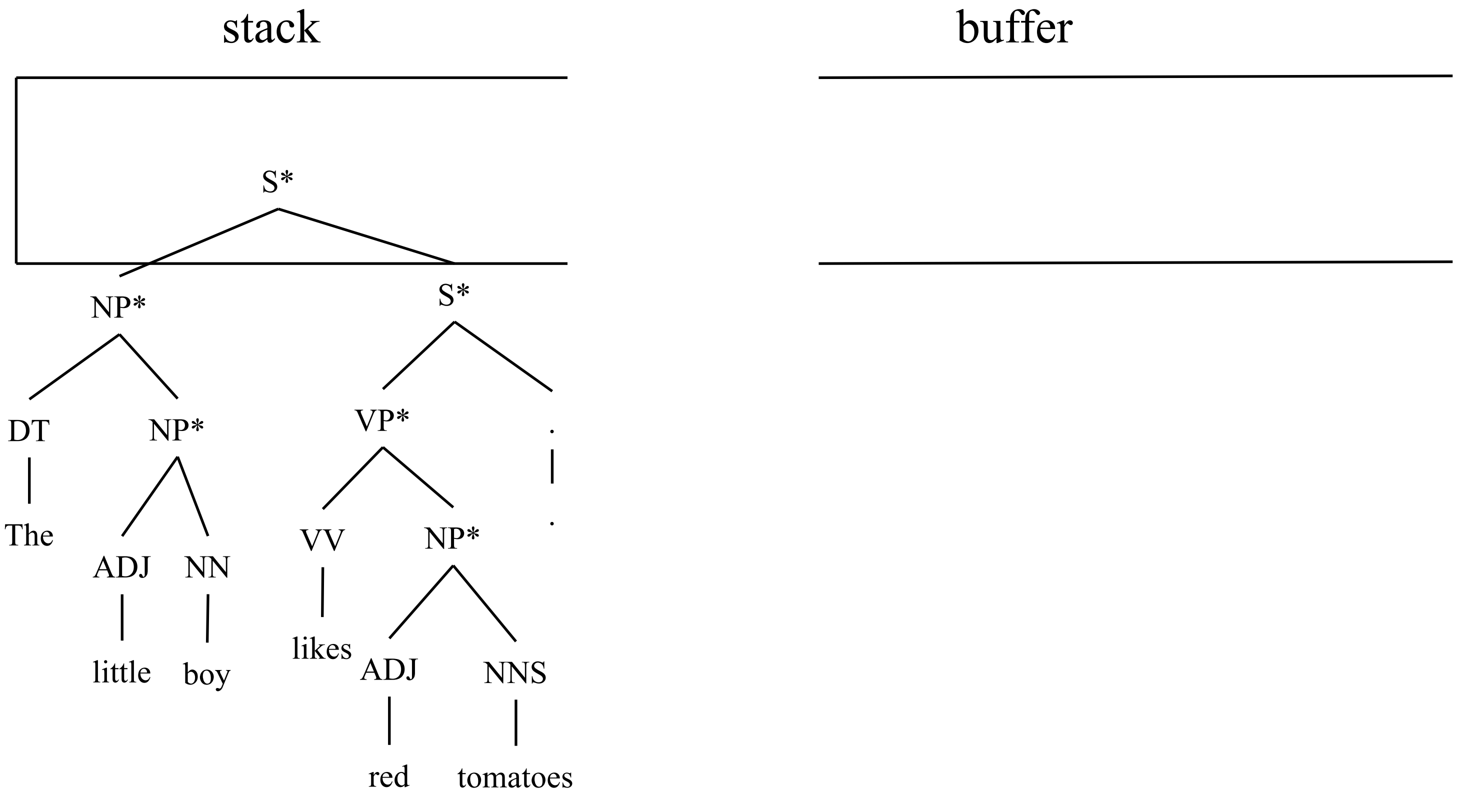
- Example
  - Reduce-R-S



Transition-based constituent parsing

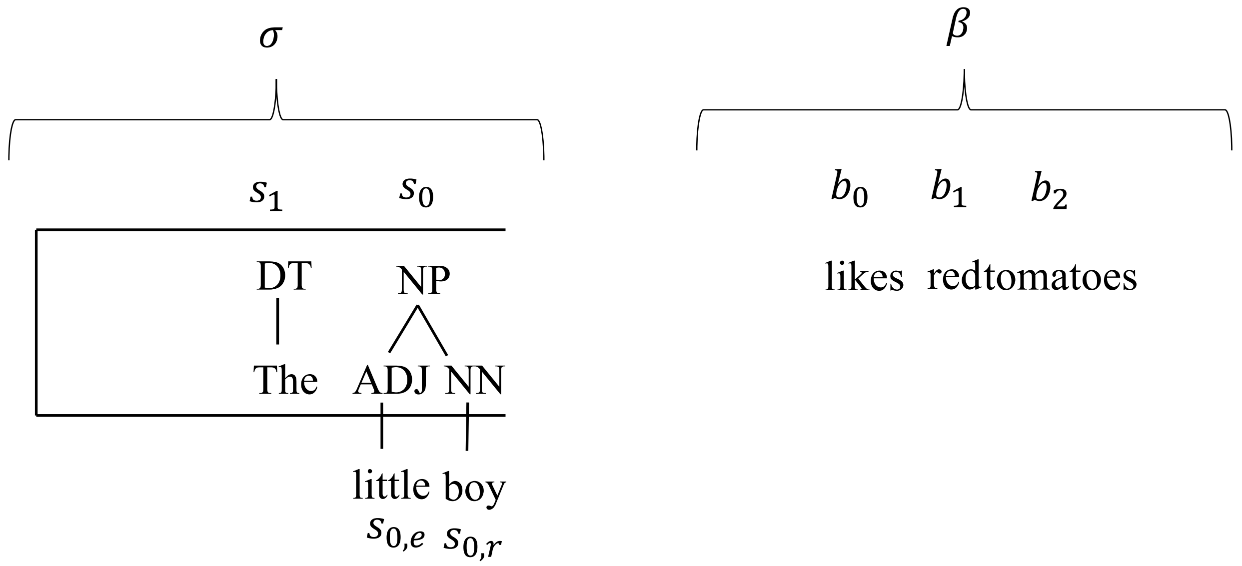
Shift-reduce constituent parsing

- Example
  - Terminate



Transition-based constituent parsing

- **Feature Templates for Shift-reduce Constituent Parser**
- Example state



Transition-based constituent parsing

WestlakeNLP

- **Feature Templates for Shift-reduce Constituent Parser**

Feature type	Feature Template
unigrams	s_0ps_0c, s_0ws_0c, s_1ps_1c, s_1ws_1c, s_2ps_2c, s_2ws_2c, s_3ps_3c, s_3ws_3c, b_0wb_0p, b_1wb_1p, b_2wb_2p, b_3wb_3p, $s_{0.l}ws_{0.l}c$, $s_{0.r}ws_{0.r}c$, $s_{0.u}ws_{0.u}c$, $s_{1.l}ws_{1.l}c$, $s_{1.r}ws_{1.r}c$, $s_{1.u}ws_{1.u}c$
bigrams	s_0ws_1w, s_0ws_1c, s_0cs_1w, s_0cs_1c, s_0wb_0w, s_0wb_0p, s_0cb_0w, s_0cb_0p, b_0wb_1w, b_0wb_1p, b_0pb_1w, b_0pb_1p, s_1wb_0w, s_1wb_0p, s_1cb_0w, s_1cb_0p
trigrams	$s_0cs_1cs_2c$, $s_0ws_1cs_2c$, $s_0cs_1wb_0p$, $s_0cs_1cs_2w$, $s_0cs_1cb_0p$, $s_0ws_1cb_0p$, $s_0cs_1wb_0p$, $s_0cs_1cb_0w$

s_i: top node of the stack; b_i: front word on the buffer; xw: the word form of x;

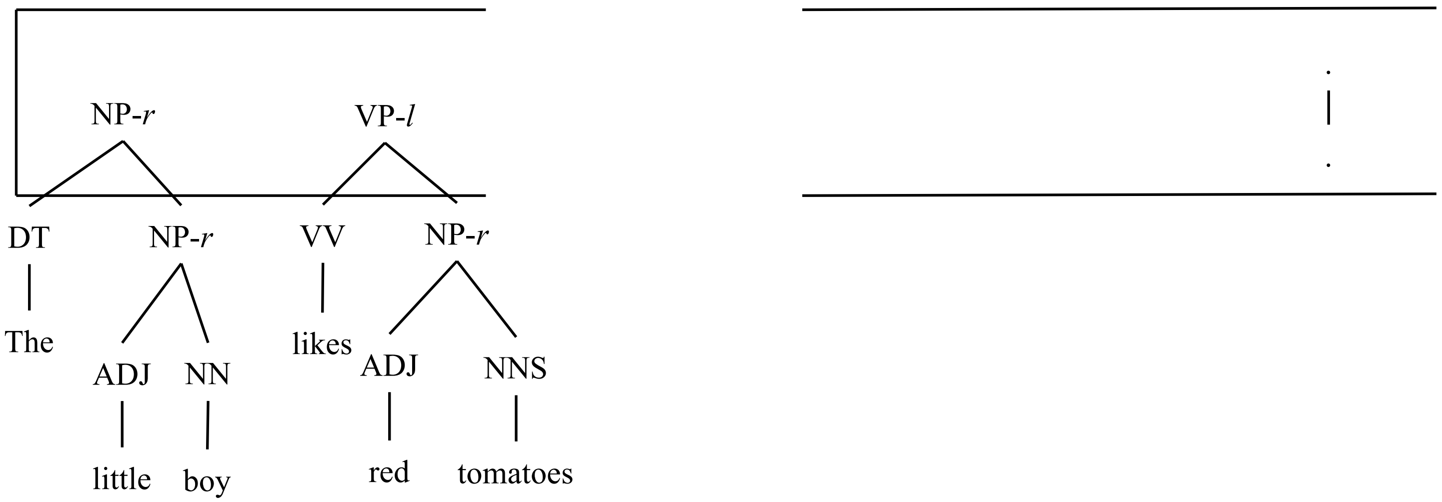
xp: the POS tag; xc: the constituent label of a non-terminal node x;

xl, xr, xu: the left child, the right child and the unary child of x, respectively.

Transition-based constituent parsing

- **Feature Templates for Shift-reduce Constituent Parser**
- Example

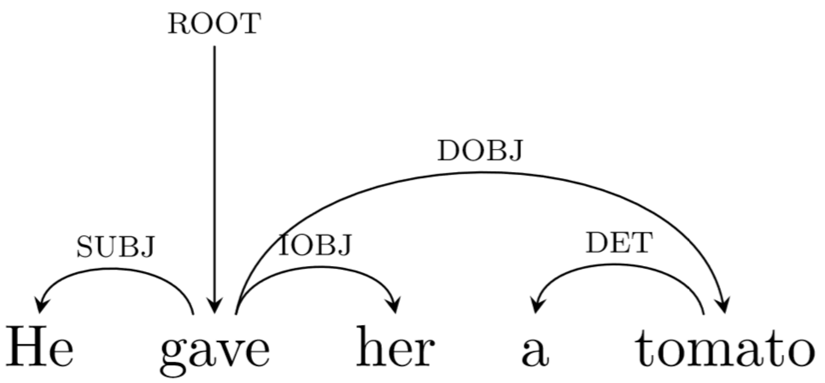
In the 10-th step of the example sentence "*The little boy likes red tomatoes*", the $s_0\ ps_0c$ feature is VBZ | VP and the s_1ps_1c feature is NN | NP.



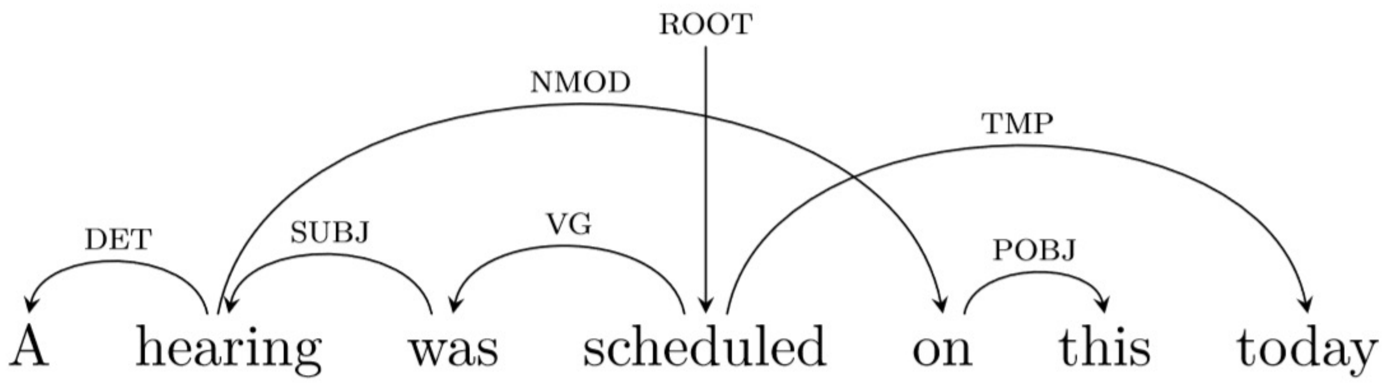
Contents

Shift-reduce dependency parsing

- Projective dependency tree



- Non-projective dependency tree



Contents

WestlakeNLP

Shift-reduce dependency parsing

- Arc-standard Parsing
  - **State**

    σ: stack of partially dependency outputs.

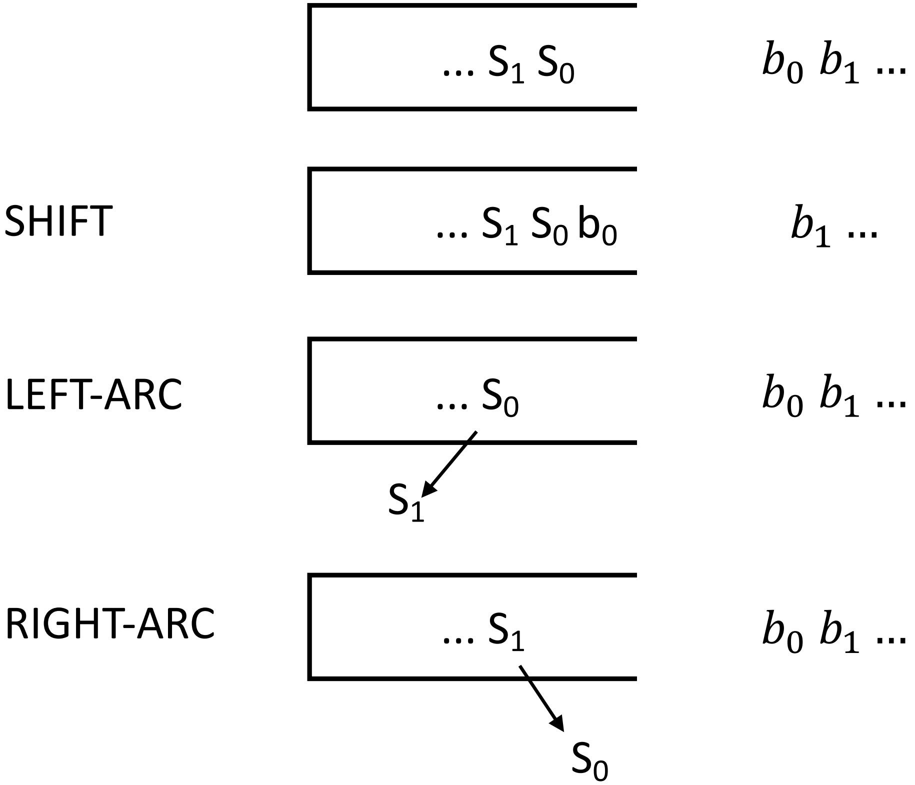
    β: buffer of the next incoming words.

  - **actions**

    SHIFT, LEFT-ARC-X, RIGHT-ARC-X

Shift-reduce dependency parsing

Actions



Shift-reduce dependency parsing

Arc-standard Dependency Parsing

Axiom:	$([\,],\ W_{1:n},\ \phi)$	LEFT-ARC-X:	$\dfrac{(\sigma\|s_1\|s_0,\ \beta,\ A)}{(\sigma\|s_0,\ \beta,\ A \cup \{s_1 \overset{X}{\curvearrowleft} s_0\})}$
Goal:	$([s_0],\ [\,],\ A)$		
SHIFT:	$\dfrac{(\sigma,\ b_0\|\beta,\ A)}{(\sigma\|b_0,\ \beta,\ A)}$	RIGHT-ARC-X:	$\dfrac{(\sigma\|s_1\|s_0,\ \beta,\ A)}{(\sigma\|s_1,\ \beta,\ A \cup \{s_1 \overset{X}{\curvearrowright} s_0\})}$

$State$: (σ, β, A);

σ: $stack$; β: $buffer$; A: $the\ set\ of\ dependency\ arcs\ that\ have\ been\ constructed$

Arc-standard dependency parsing

Example

Sentence “He gave her a tomato”

Next action: SHIFT

He gave her a tomato

Arc-standard dependency parsing

WestlakeNLP

Example

Sentence "He gave her a tomato"

Next action: SHIFT

He

gave her a tomato

Arc-standard dependency parsing

Example

Sentence “He gave her a tomato”

Next action: LEFT-ARC-SUBJ

He gave

her a tomato

Arc-standard dependency parsing

Example

Sentence “He gave her a tomato”

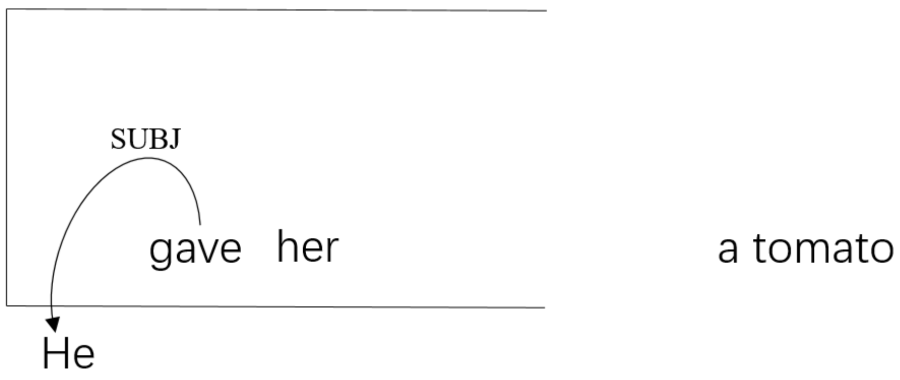
Next action: SHIFT

Arc-standard dependency parsing

Example

Sentence “He gave her a tomato”

Next action: RIGHT-ARC-IOBJ



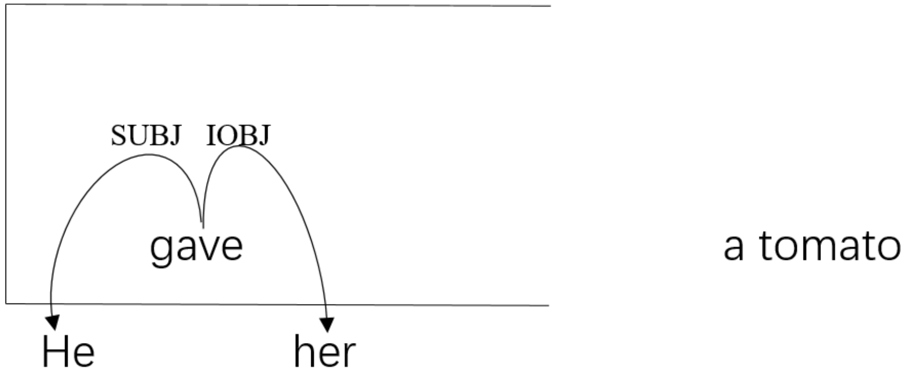
Arc-standard dependency parsing

WestlakeNLP

Example

Sentence “He gave her a tomato”

Next action: SHIFT

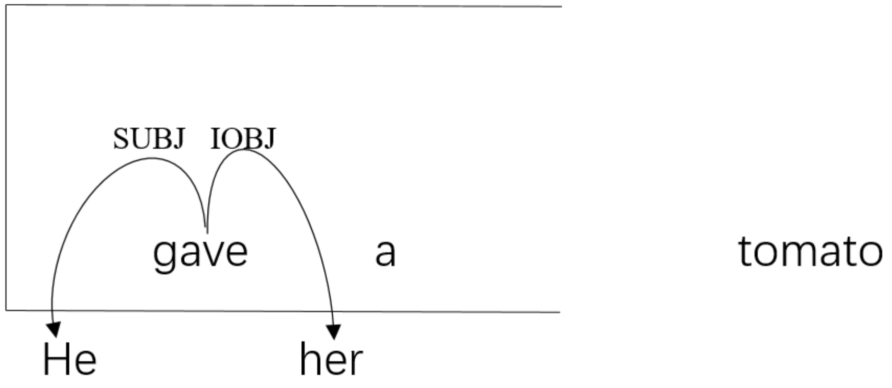


Arc-standard dependency parsing

Example

Sentence “He gave her a tomato”

Next action: SHIFT

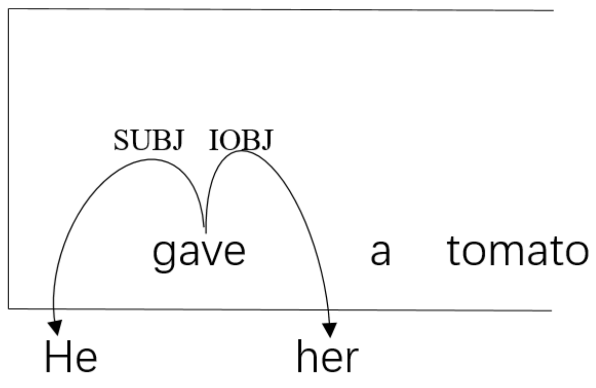


Arc-standard dependency parsing

Example

Sentence "He gave her a tomato"

Next action: LEFT-ARC-DET

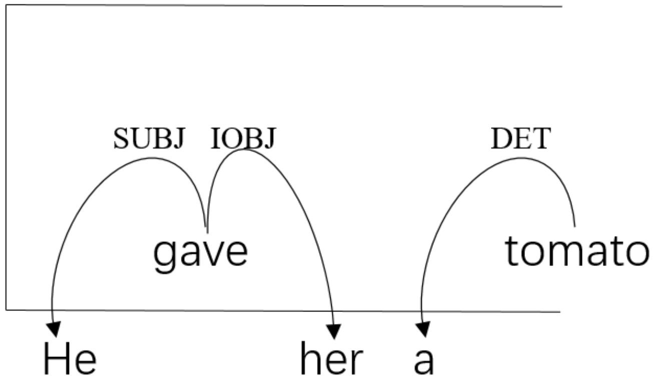


Arc-standard dependency parsing

Example

Sentence "He gave her a tomato"

Next action: RIGHT-ARC-DOBJ



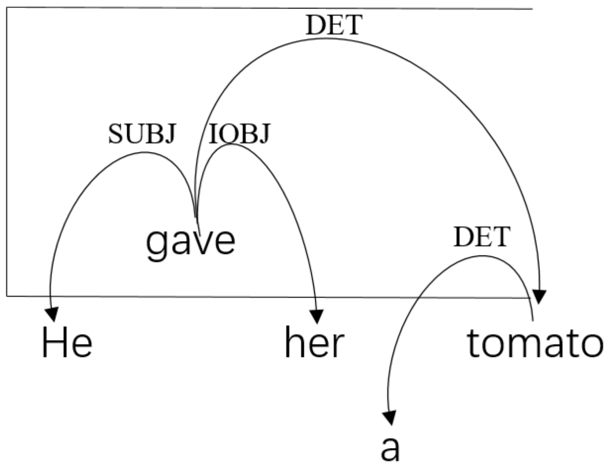
Arc-standard dependency parsing

WestlakeNLP

Example

Sentence "He gave her a tomato"

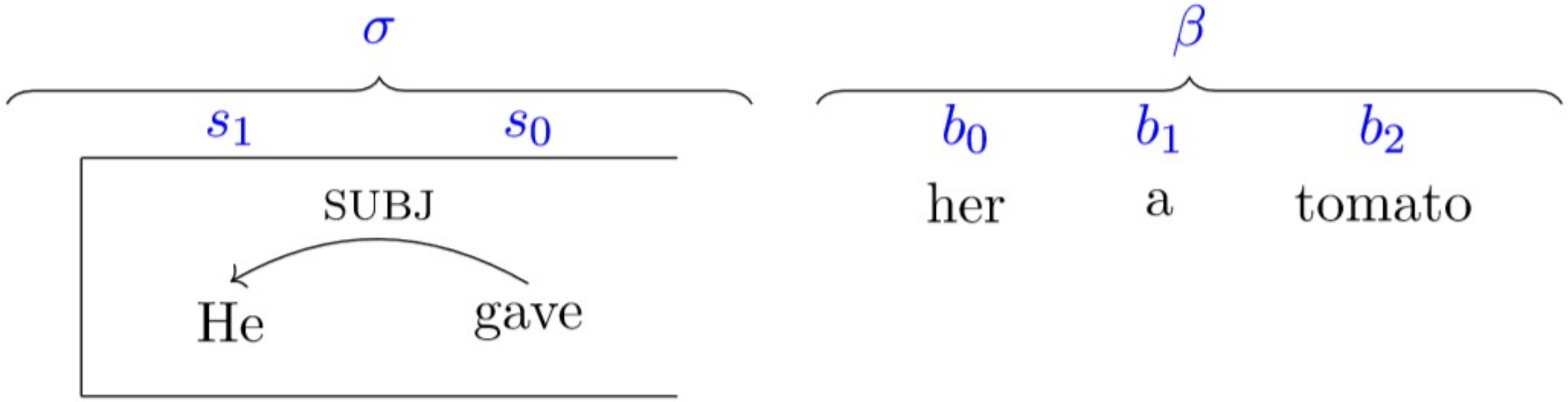
Next action: END



Shift-reduce dependency parsing

Arc-standard Dependency Parsing

- Example state



Arc-standard dependency parsing

WestlakeNLP

Feature Templates for arc-standard dependency parsing

Feature Type	Feature Template	Feature Type	Feature Template
from single words	s_0wp; s_0w; s_0p; b_0wp; b_0w; b_0p; b_1wp; b_1w; b_1p; b_2wp; b_2w; b_2p;	valency	s_0wv_r; s_0pv_r; s_0wv_l; s_0pv_l; b_0wv_l; b_0pv_l;
from word pairs	s_0wpb_0wp; s_0wpb_0w; s_0wpb_0p; s_0wb_0wp; s_0pb_0wp; s_0wb_0w; s_0pb_0p; b_0pb_1p;	unigrams	$s_{0.h}w$; $s_{0.h}p$; $s_{0.l}$; $s_{0.l}w$; $s_{0.l}p$; $s_{0.l}l$; $s_{0.r}w$; $s_{0.r}p$; $s_{0.r}l$; $b_{0.l}w$; $b_{0.l}p$; $b_{0.l}l$;
from three words	$b_0pb_1pb_2p$; $s_0pb_0pb_1p$; $s_{0.h}ps_0pb_0p$; $s_0ps_{0.l}pb_0p$; $s_0ps_{0.r}pb_0p$; $s_0pb_0pb_{0.l}p$;	third-order	$s_{0.h_2}w$; $s_{0.h_2}p$; $s_{0.h}l$; $s_{0.l_2}p$; $s_{0.l_2}l$; $s_{0.r_2}w$; $s_{0.r_2}p$; $s_{0.r_2}l$; $b_{0.l_2}w$; $b_{0.l_2}p$; $b_{0.l_2}l$; $s_0ps_{0.l}ps_{0.l_2}p$; $s_0ps_{0.r}ps_{0.r_2}p$; $s_0ps_{0.h}ps_{0.h_2}p$; $b_0pb_{0.l}pb_{0.l_2}p$;
distance	s_0wd; s_0pd; b_0wd; b_0pd; s_0wb_0wd; s_0pb_0pd;	label set	s_0ws_r; s_0ps_r; s_0ws_l; s_0ps_l; n_0ws_l; n_0ps_l

Contents

Shift-reduce dependency parsing

WestlakeNLP

- Arc-eager Parsing
  - **State**

    σ: stack of partially constituent outputs.

    β: buffer of the next incoming words.

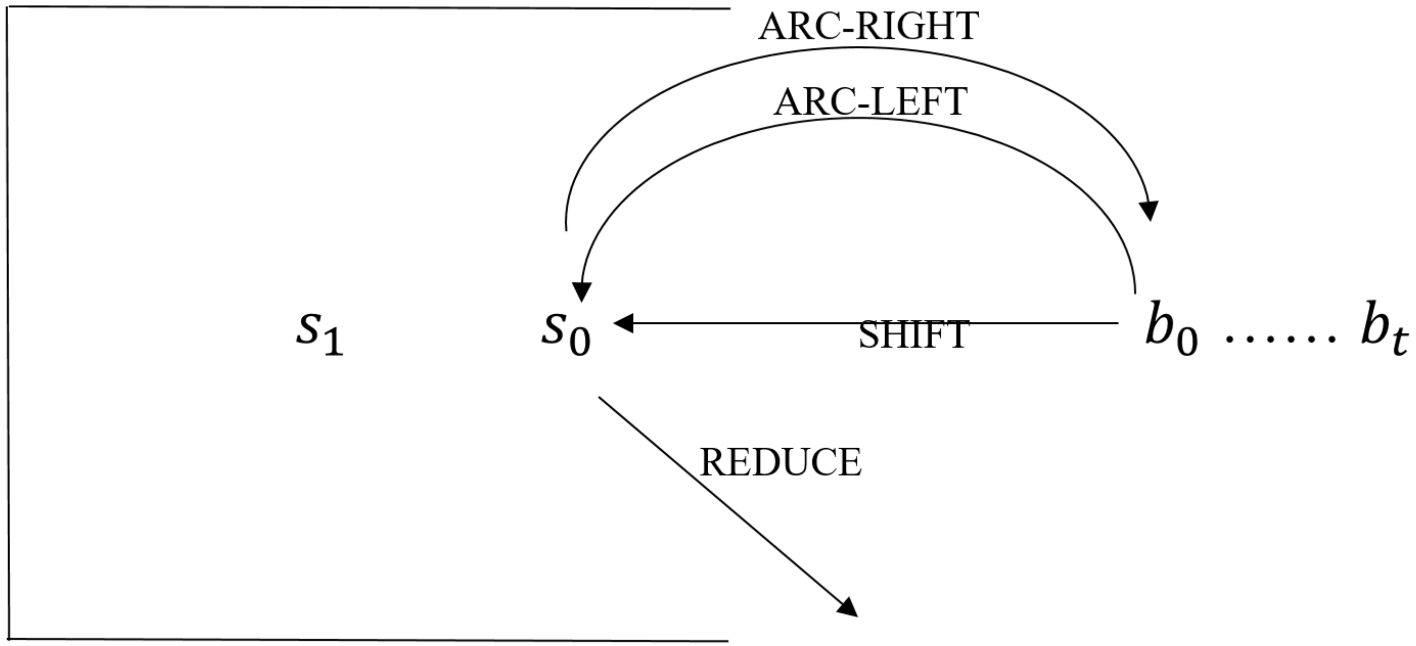
  - **actions**

    SHIFT, LEFT-ARC-X, RIGHT-ARC-X, REDUCE

Shift-reduce dependency parsing

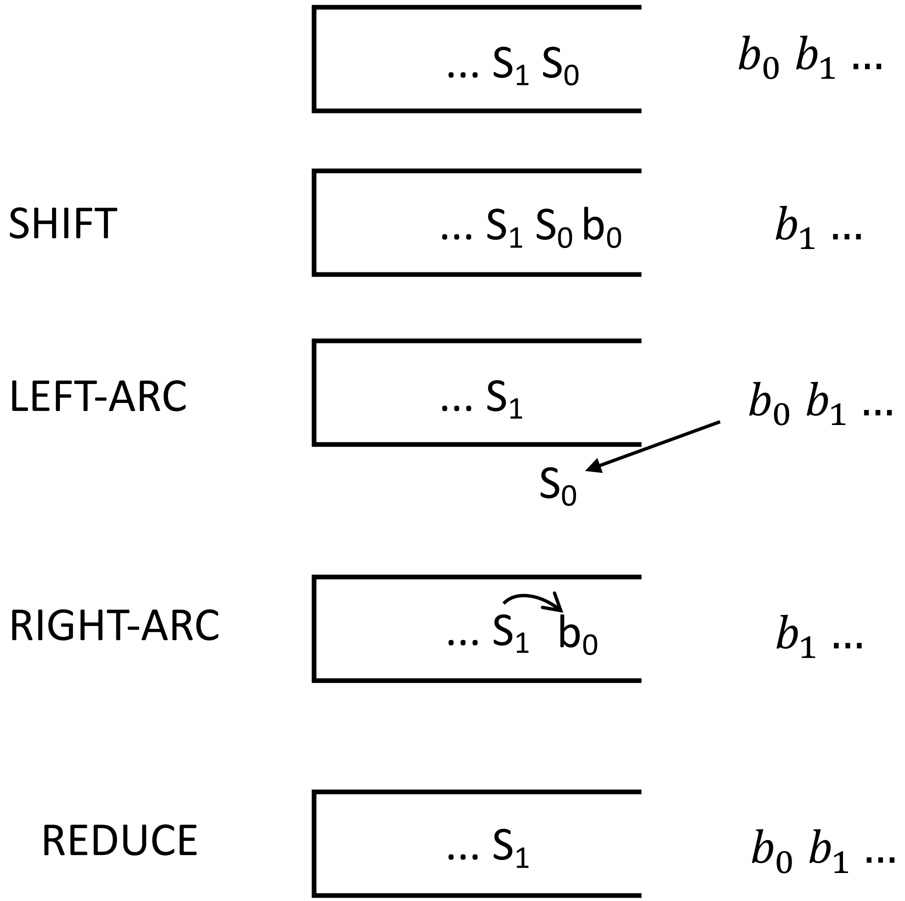
Arc-eager dependency parsing

- Example



Shift-reduce dependency parsing

Actions



Shift-reduce dependency parsing

Arc-eager dependency parsing

Axiom: $([\,],\ W_{1:n},\ \phi)$

Goal: $([s_0],\ [\,],\ A)$

$$\textsc{Shift}:\ \frac{(\sigma,\ b_0|\beta,\ \phi)}{(\sigma|b_0,\ \beta,\ \phi)}$$

$$\textsc{Left-Arc-X}:\ \frac{(\sigma|s_0,\ b_0|\beta,\ A),\ \text{such that}\ \neg\left(\exists(k,\textsc{l})\ w_k \overset{\textsc{l}}{\curvearrowright} s_0 \in A\right)}{(\sigma,\ b_0|\beta,\ A \cup \{s_0 \overset{\textsc{x}}{\curvearrowleft} b_0\})}$$

$$\textsc{Right-Arc-X}:\ \frac{(\sigma|s_0,\ b_0|\beta,\ A)}{(\sigma|s_0|b_0,\ \beta,\ A \cup \{s_0 \overset{\textsc{x}}{\curvearrowright} b_0\})}$$

$$\textsc{Reduce}:\ \frac{(\sigma|s_0,\ \beta,\ A),\ \text{such that}\ \left(\exists(k,\textsc{l})\ w_k \overset{\textsc{l}}{\curvearrowright} s_0 \in A\right)}{(\sigma,\ \beta,\ A)}$$

Main differences comparing with arc-standard:

- $Left-Arc-X/Right-Arc-X$: construct a dependency arc from the front word on the buffer to the top word on the stack, but not as that (from the top two words on the stack) in arc-standard.
- *Reduce*: pop the top word off the stack.

Example (Dependency Parsing)

- NEXT ACTION: $Shift$

$\overbrace{\boxed{\qquad\qquad\qquad}}^{\sigma} \quad \overbrace{\boxed{\text{He does it here}}}^{\beta}$

σ: stack $\qquad$ β: buffer

Example (Dependency Parsing)

- NEXT ACTION: $Left - Arc - SUBJ$

$$\underbrace{\boxed{\text{He}}}_{\sigma} \quad \underbrace{\boxed{\text{does it here}}}_{\beta}$$

σ: stack β: buffer

Example (Dependency Parsing)

- NEXT ACTION: $Shift$

σ β

does it here

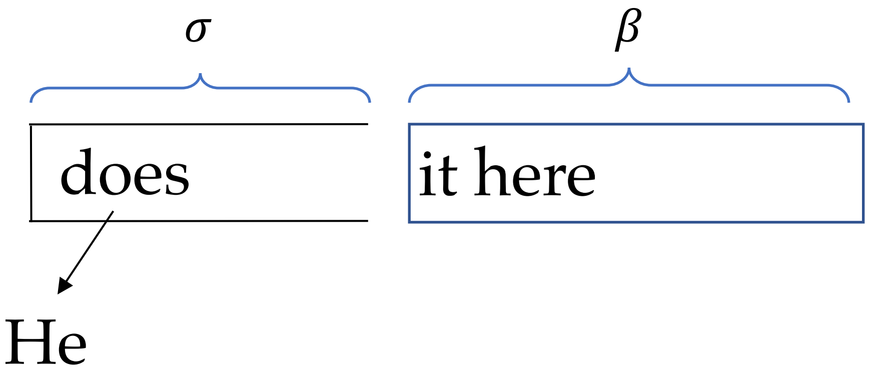
He

σ: stack β: buffer

Example (Dependency Parsing)

WestlakeNLP

- NEXT ACTION: $Right - Arc - OBJ$

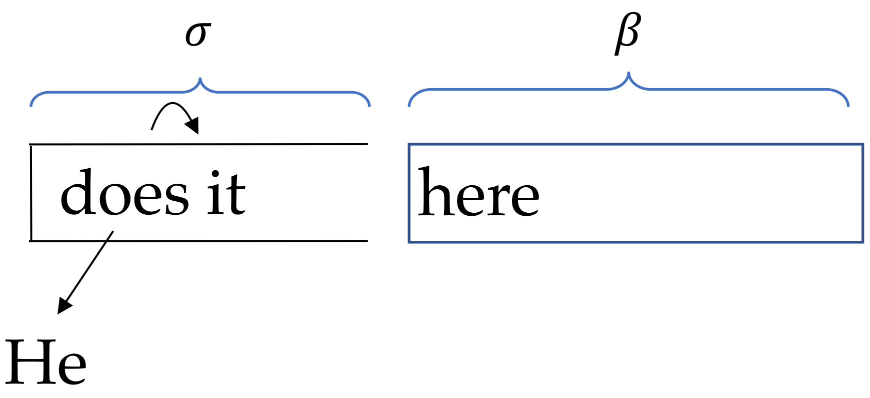


σ: stack β: buffer

Example (Dependency Parsing)

WestlakeNLP

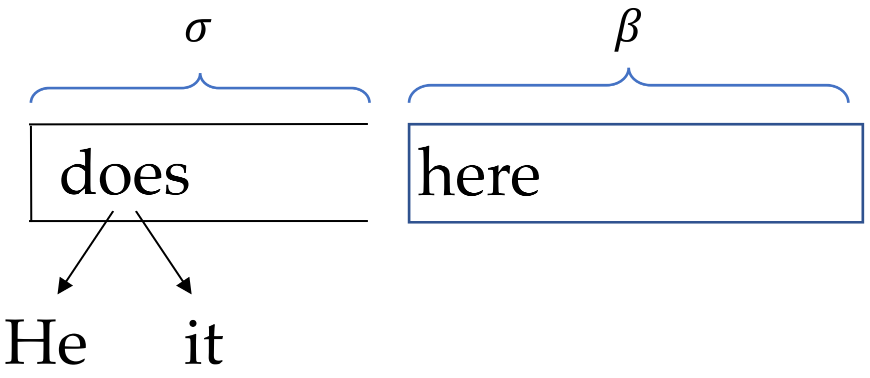
- NEXT ACTION: *Reduce*



σ: stack β: buffer

Example (Dependency Parsing)

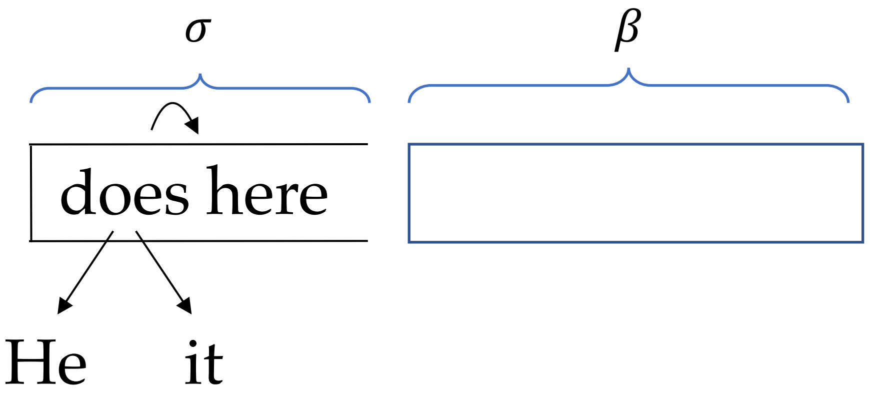
- NEXT ACTION: $Right - Arc - MOD$



σ: stack β: buffer

Example (Dependency Parsing)

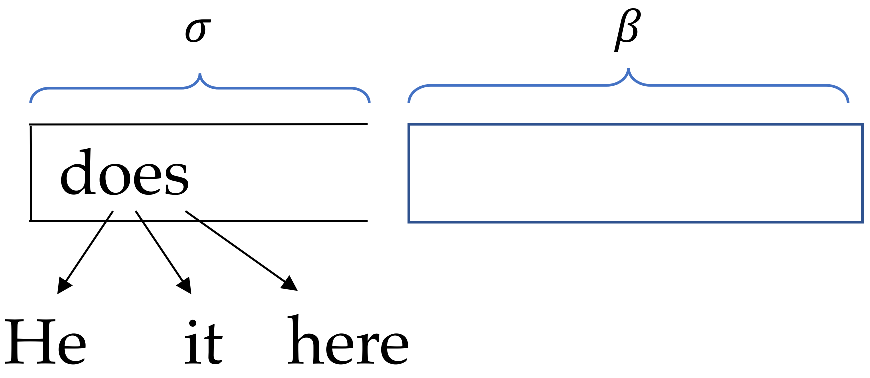
- NEXT ACTION: *Reduce*



σ: stack β: buffer

Example (Dependency Parsing)

- NEXT ACTION: *Finish*



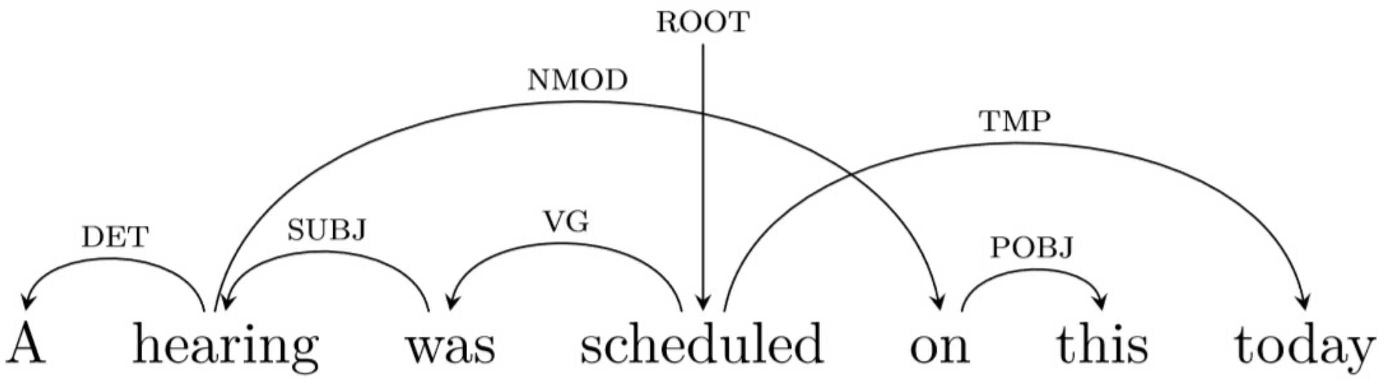
σ: stack $\qquad$ β: buffer

Contents

WestlakeNLP

Shift-reduce dependency parsing

- **The Swap Action and Non-projective Trees**
- To allow constructing non-projective trees, the arc-standard system can be extended by adding a new action:
  - *Swap*: remove the second top word from the stack, pushing it onto the buffer front.



Shift-reduce dependency parsing

Arc-standard Dependency Parsing with Swap Action

Axiom:	$([\,],\ W_{1:n},\ \phi)$	LEFT-ARC-X:	$\dfrac{([\sigma\|s_1\|s_0],\ \beta,\ A)}{([\sigma\|s_0],\ \beta,\ A \cup \{s_1 \overset{X}{\curvearrowleft} s_0\})}$
Goal:	$([s_0],\ [\,],\ A)$		
SHIFT:	$\dfrac{(\sigma,\ [b_0\|\beta],\ \phi)}{([\sigma\|b_0],\ \beta,\ \phi)}$	RIGHT-ARC-X:	$\dfrac{([\sigma\|s_1\|s_0],\ \beta,\ A)}{([\sigma\|s_1],\ \beta,\ A \cup \{s_1 \overset{X}{\curvearrowright} s_0\})}$
		SWAP:	$\dfrac{([\sigma\|s_1\|s_0],\ \beta,\ A),\ \text{such that } \mathrm{IDX}(s_1) < \mathrm{IDX}(s_0)}{([\sigma\|s_0],\ [s_1\|\beta],\ A)}$

State: (σ, β, A); σ: stack; β: buffer; A: the set of dependency arcs that have been constructed; $IDX(w)$: return the index of w in the sentence $W_{1:n}$.

Shift-reduce dependency parsing

WestlakeNLP

Arc-standard Dependency Parsing with Swap Action

- Example

Next action: SHIFT

A hearing was scheduled on this today

Shift-reduce dependency parsing

WestlakeNLP

Arc-standard Dependency Parsing with Swap Action

- Example

Next action: SHIFT

A

hearing was scheduled on this today

Shift-reduce dependency parsing

Arc-standard Dependency Parsing with Swap Action

- Example

Next action: LEFT-ARC-DET

A hearing	was scheduled on this today

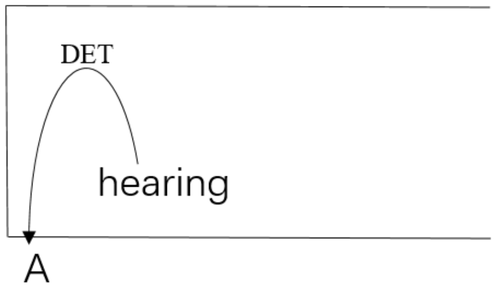
Shift-reduce dependency parsing

WestlakeNLP

Arc-standard Dependency Parsing with Swap Action

- Example

Next action: SHIFT



was scheduled on this today

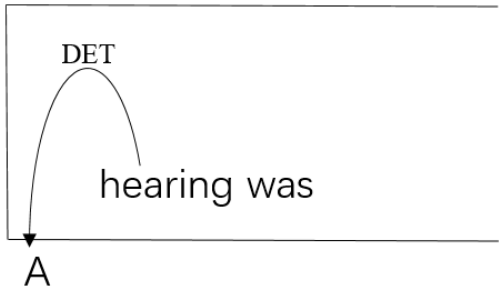
Shift-reduce dependency parsing

WestlakeNLP

Arc-standard Dependency Parsing with Swap Action

- Example

Next action: SHIFT



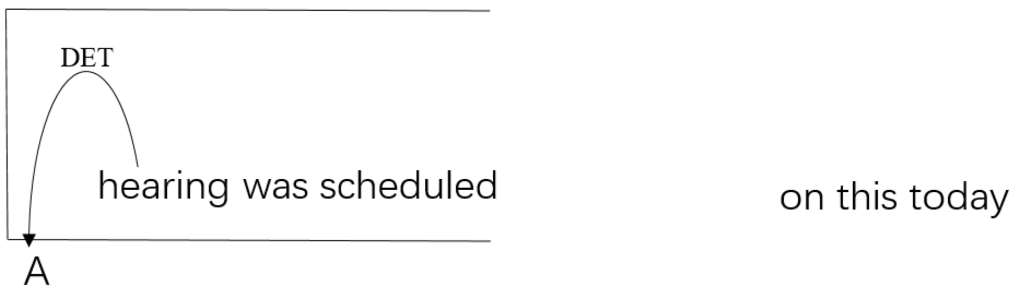
scheduled on this today

Shift-reduce dependency parsing

Arc-standard Dependency Parsing with Swap Action

- Example

Next action: SHIFT



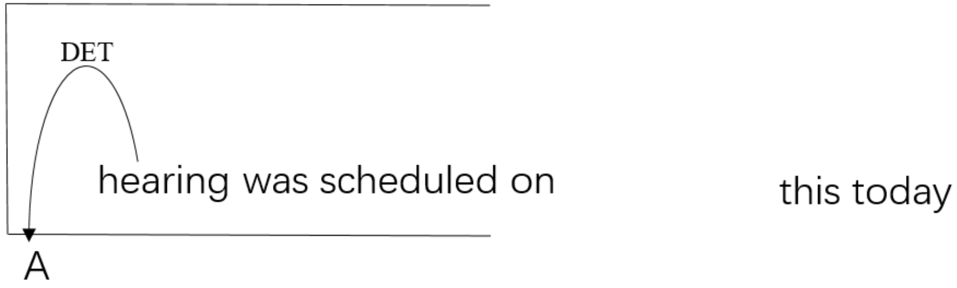
Shift-reduce dependency parsing

WestlakeNLP

Arc-standard Dependency Parsing with Swap Action

- Example

Next action: SWAP

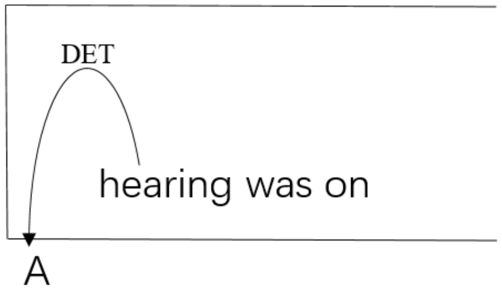


Shift-reduce dependency parsing

Arc-standard Dependency Parsing with Swap Action

- Example

Next action: SWAP



scheduled this today

Shift-reduce dependency parsing

Arc-standard Dependency Parsing with Swap Action

- Example

Next action: SHIFT

Shift-reduce dependency parsing

Arc-standard Dependency Parsing with Swap Action

- Example

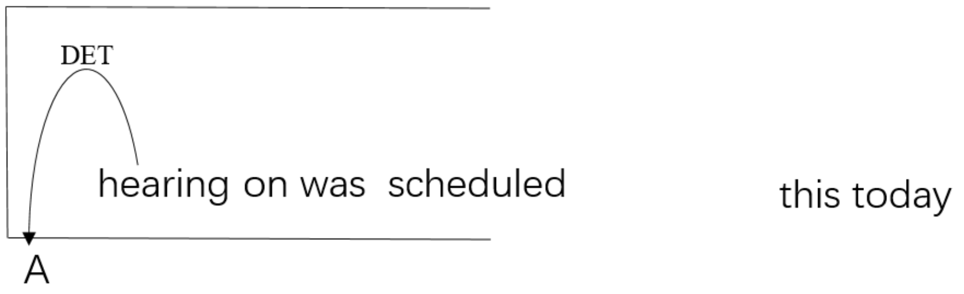
  Next action: SHIFT

Shift-reduce dependency parsing

Arc-standard Dependency Parsing with Swap Action

- Example

Next action: SHIFT

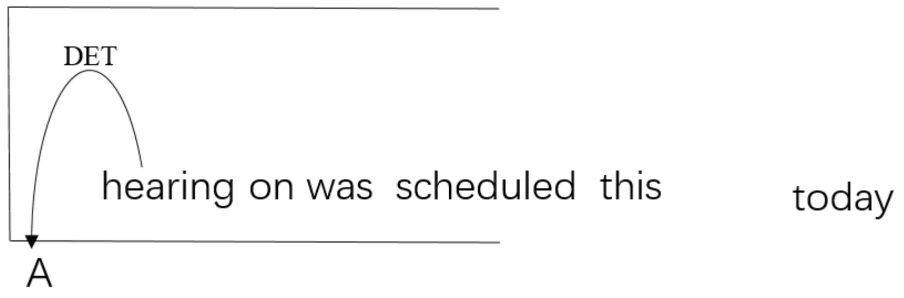


Shift-reduce dependency parsing

Arc-standard Dependency Parsing with Swap Action

- Example

Next action: SWAP



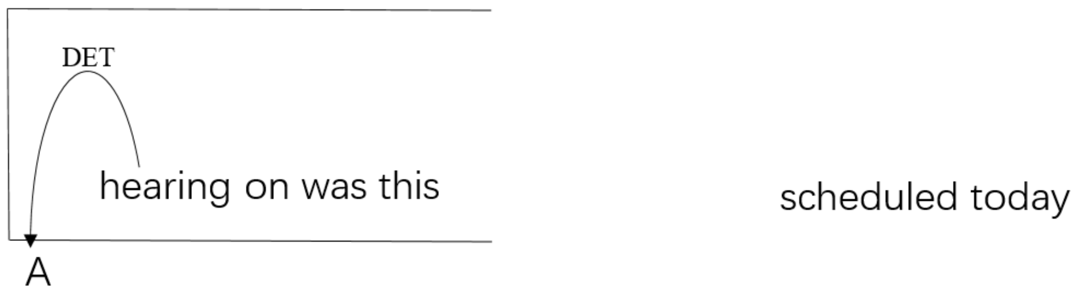
Shift-reduce dependency parsing

WestlakeNLP

Arc-standard Dependency Parsing with Swap Action

- Example

Next action: SWAP



Shift-reduce dependency parsing

WestlakeNLP

Arc-standard Dependency Parsing with Swap Action

- Example

Next action: RIGHT-ARC-POBJ

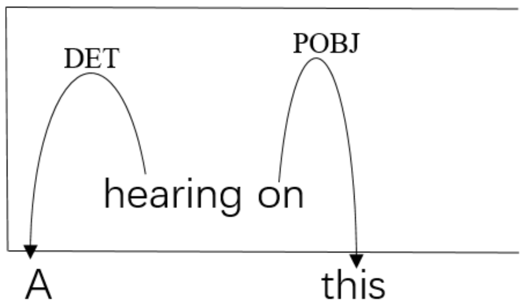
Shift-reduce dependency parsing

WestlakeNLP

Arc-standard Dependency Parsing with Swap Action

- Example

Next action: RIGHT-ARC-NMOD



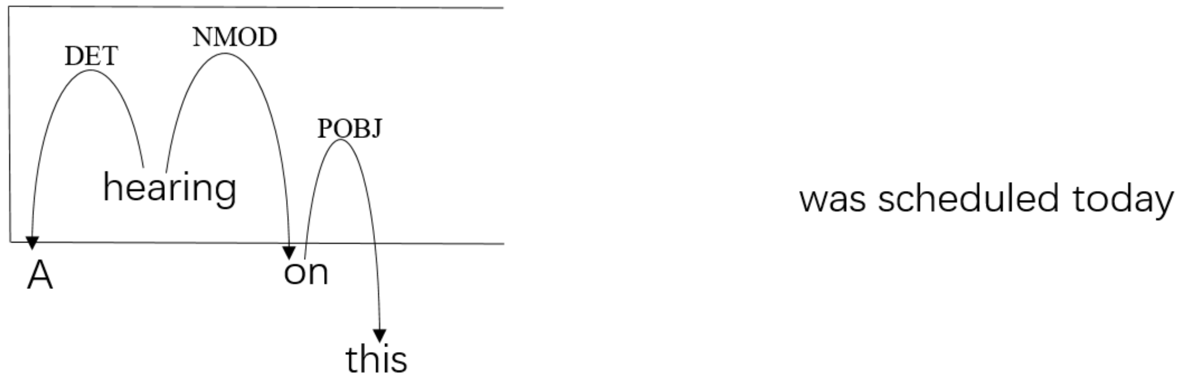
was scheduled today

Shift-reduce dependency parsing

Arc-standard Dependency Parsing with Swap Action

- Example

Next action: SHIFT



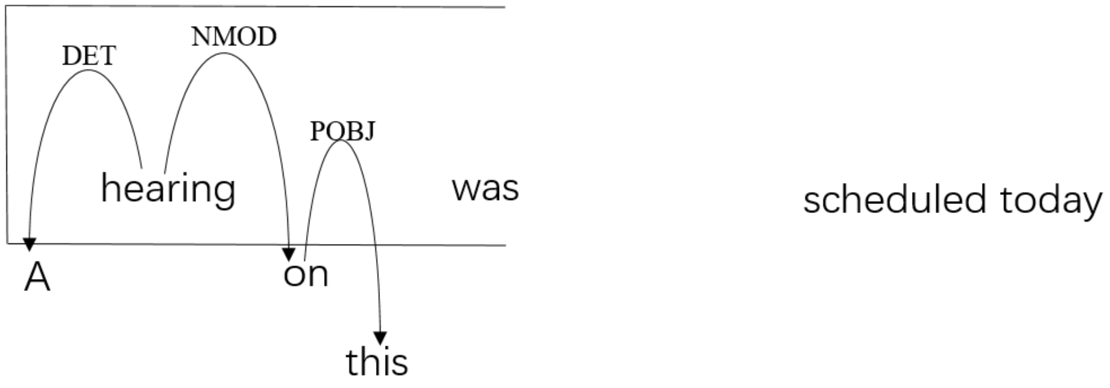
Shift-reduce dependency parsing

WestlakeNLP

Arc-standard Dependency Parsing with Swap Action

- Example

Next action: LEFT-ARC-SUBJ



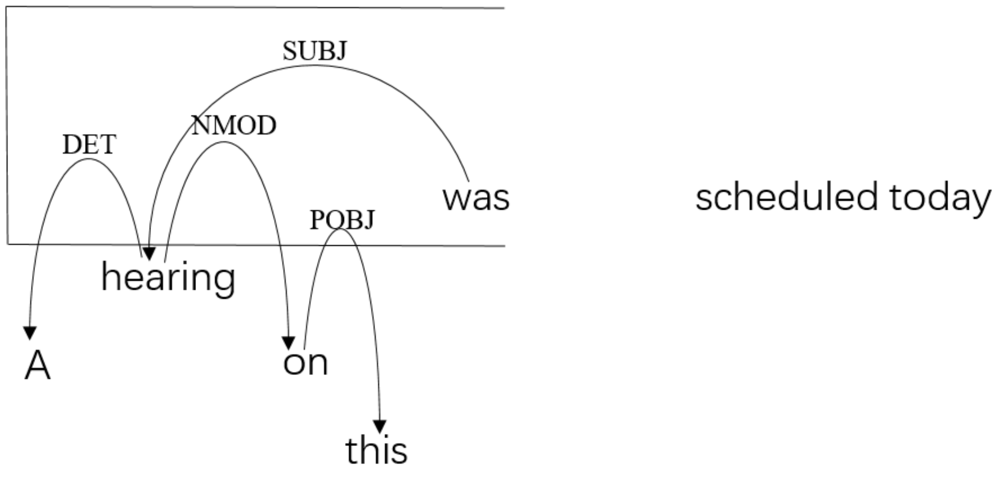
Shift-reduce dependency parsing

WestlakeNLP

Arc-standard Dependency Parsing with Swap Action

- Example

Next action: SHIFT



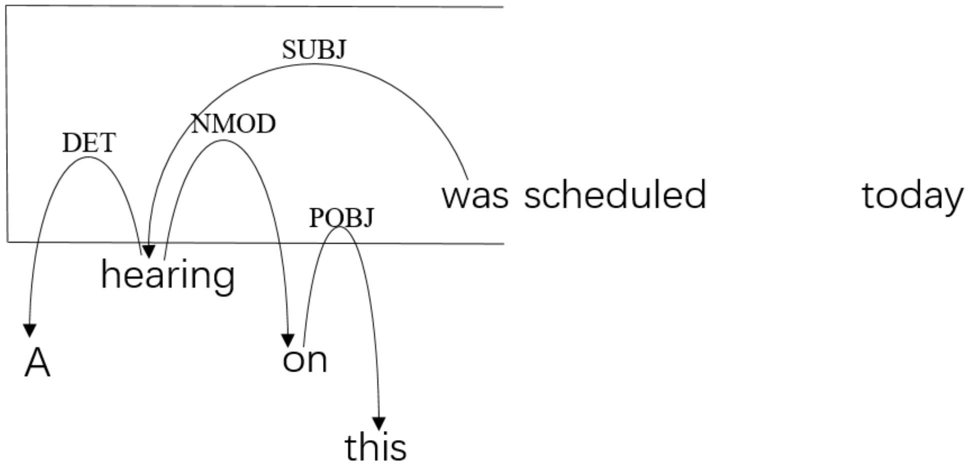
Shift-reduce dependency parsing

WestlakeNLP

Arc-standard Dependency Parsing with Swap Action

- Example

Next action: SHIFT

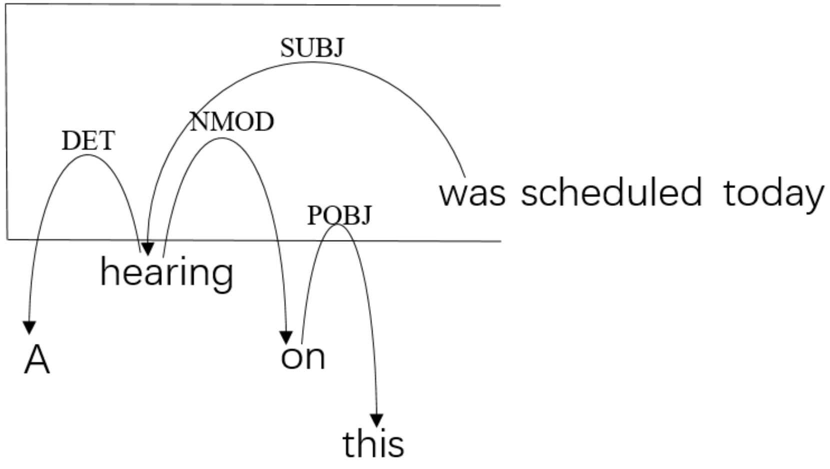


Shift-reduce dependency parsing

Arc-standard Dependency Parsing with Swap Action

- Example

Next action: RIGHT-ARC-TMP

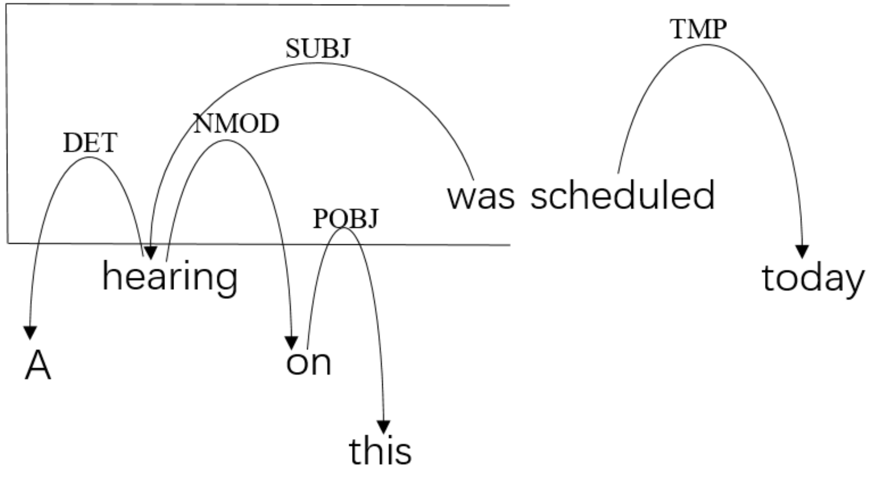


Shift-reduce dependency parsing

Arc-standard Dependency Parsing with Swap Action

- Example

Next action: LEFT-ARC-VG



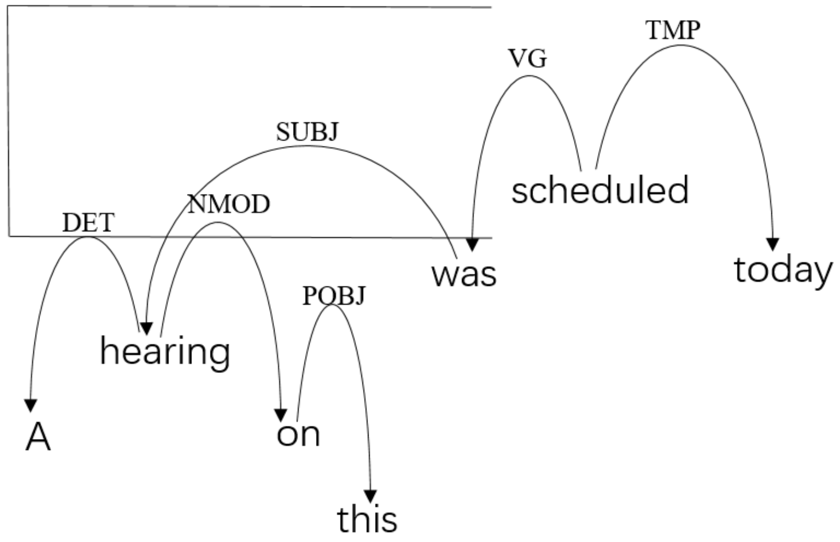
Shift-reduce dependency parsing

WestlakeNLP

Arc-standard Dependency Parsing with Swap Action

- Example

Next action: END



Contents

Joint Models

WestlakeNLP

- Motivations
  - Cross-task information sharing
  - Reduction of error propagation
- Example
  - For joint POS-tagging and syntactic parsing, the $SHIFT$ action of arc-standard algorithm can be replaced with $SHIFT - X$ action, where X refers to the POS label.

Joint Models

WestlakeNLP

Joint POS-tagging and Dependency Parsing

- Input: $W_{1:n} = w_1, \dots, w_n$
- Output: $(T_{1:n}, A)$, where t_i is POS for w_i.
- The arc-standard algorithm can be extended by replacing $SHIFT$ action with $SHIFT - X$ action
    - $SHIFT - X$, which removes the front word from the buffer, assigning the POS label X to the word, and pushing it onto the stack.

Joint Models

Joint POS-tagging and Dependency Parsing

- Example

Next action: SHIFT——PRP

Stack [S]

Buffer [B]

He_1 won_2 the_3 game_4

Joint Models

Joint POS-tagging and Dependency Parsing

- Example

Next action: SHIFT——VBD

Stack [S]	Buffer [B]
$\text{He}_{1/\text{PRP}}$	won_2 the_3 game_4

Joint Models

Joint POS-tagging and Dependency Parsing

- Example

Next action: LEFT——ARC——NSUBJ

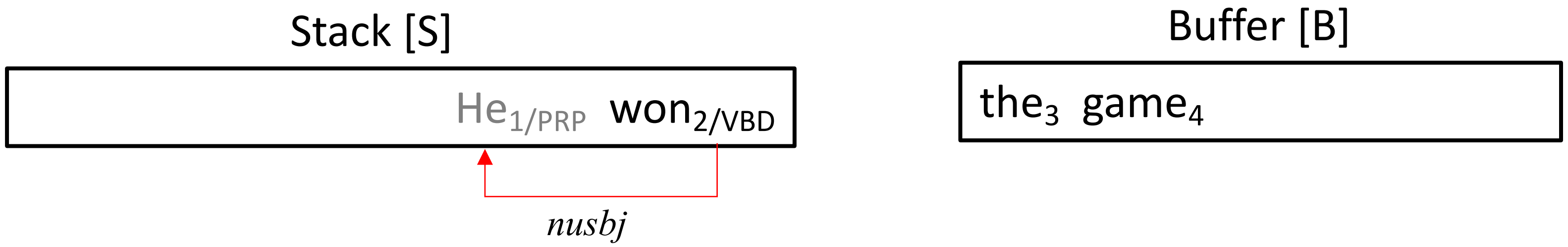
Stack [S]	Buffer [B]
$\text{He}_{1/\text{PRP}}$ won_2	the_3 game_4

Joint Models

Joint POS-tagging and Dependency Parsing

- Example

Next action: SHIFT——DT

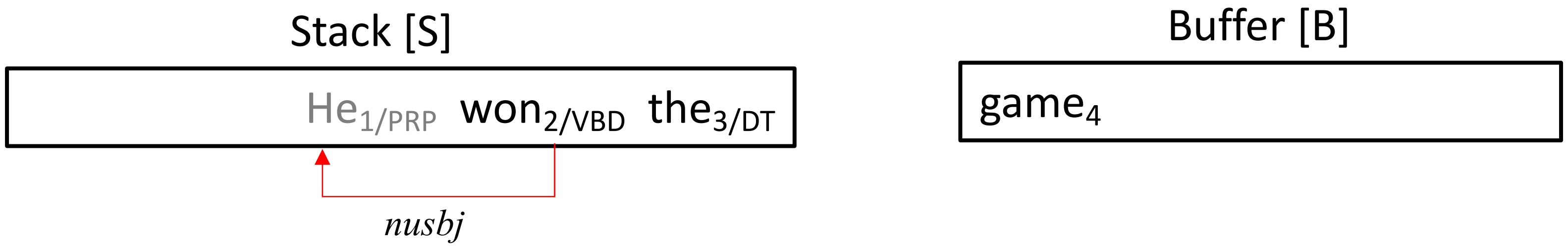


Joint Models

Joint POS-tagging and Dependency Parsing

- Example

Next action: SHIFT——NN



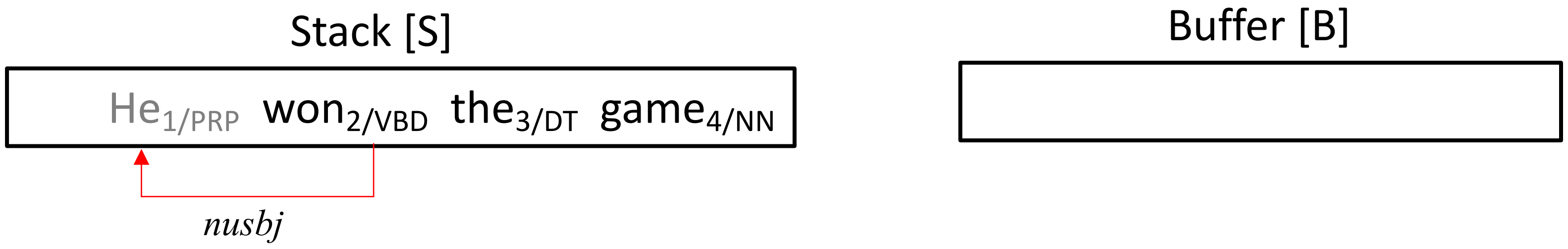
Joint Models

WestlakeNLP

Joint POS-tagging and Dependency Parsing

- Example

Next action: LEFT——ARC——DET

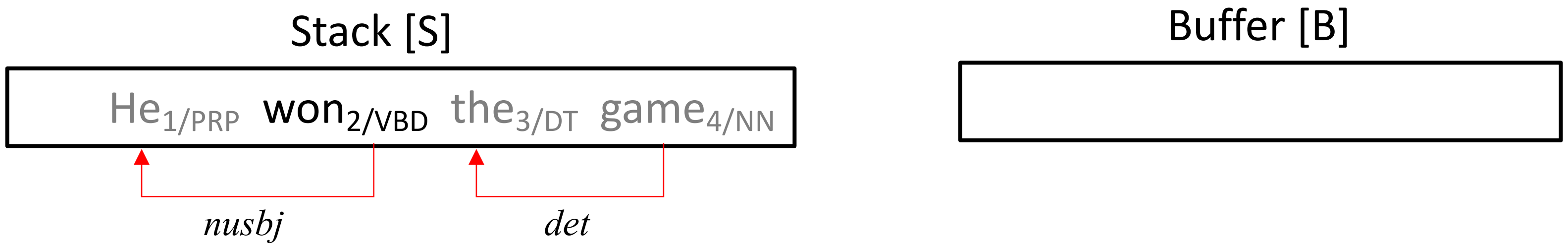


Joint Models

Joint POS-tagging and Dependency Parsing

- Example

Next action: RIGHT——ARC——DOBJ

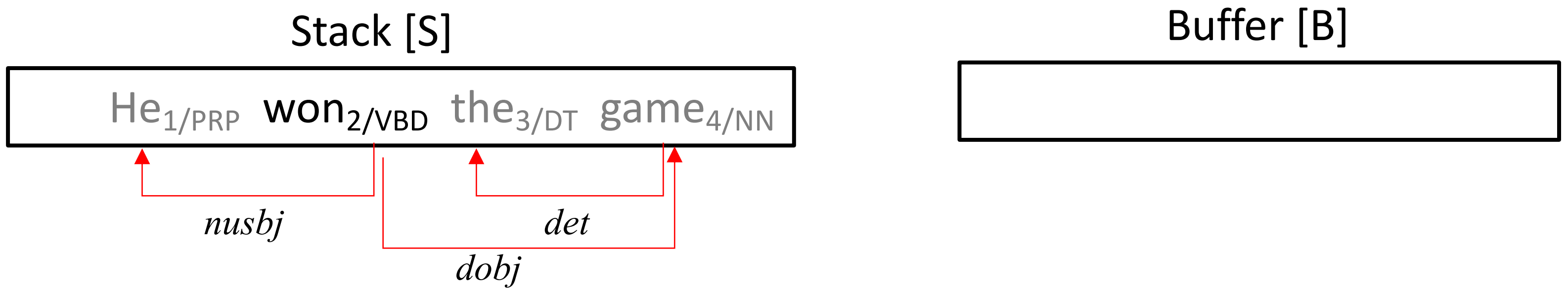


Joint Models

Joint POS-tagging and Dependency Parsing

- Example

Next action: END



Joint Models

WestlakeNLP

Joint Word Segmentation, POS-tagging and Dependency Parsing

- Input: $C_{1:n} = C_1 \dots C_n$
- Output: $(W_{1:m}, T_{1:m}, A)$ where $w_1 \dots w_m$ are words, $t_1 \dots t_n$ are POS, and A is the set of dependency arcs.
- State:

  σ -- partially built outputs

  δ -- words

  β -- incoming characters

  A_c -- character dependencies

  A_w -- word dependencies

Joint Models

Joint Word Segmentation, POS-tagging and Dependency Parsing

Axiom:		$([\,],\ [\,],\ C_{1:n},\ \phi,\ \phi)$
Goal:		$([S_0],\ [\,],\ [\,],\ A_c,\ A_w)$
LEFT-ARC-C:	$(\sigma,\ \delta\|d_0,\ b_0\|\beta,\ A_c,\ A_w)$ such that $\neg(\exists d \in \delta, d \curvearrowright d_0 \in A_c)$	$(\sigma,\ \delta,\ b_0\|\beta,\ A_c \cup \{d_0 \curvearrowleft b_0\},\ A_w)$
LEFT-ARC-X:	$(\sigma\|s_0,\ [d_0],\ \beta,\ A_c,\ A_w)$ such that $\neg(\exists s \in \sigma, s \overset{l}{\curvearrowright} s_0 \in A_w)$	$(\sigma,\ [d_0],\ \beta,\ A_c,\ A_w \cup \{s_0 \curvearrowleft d_0\})$
SHIFT:	$(\sigma,\ [d_0],\ \beta,\ A_c,\ A_w)$	$(\sigma\|d_0,\ [\,],\ \beta,\ A_c,\ A_w)$
SHIFT-C:	$(\sigma,\ \delta,\ b_0\|\beta,\ A_c,\ A_w)$	$(\sigma,\ \delta\|b_0,\ \beta,\ A_c,\ A_w)$
RIGHT-ARC-C:	$(\sigma,\ \delta\|d_0,\ b_0\|\beta,\ A_c,\ A_w)$	$(\sigma,\ \delta\|d_0\|b_0,\ \beta,\ A_c \cup \{d_0 \curvearrowright b_0\},\ A_w)$
RIGHT-ARC-X:	$(\sigma\|s_0,\ [d_0],\ \beta,\ A_c,\ A_w \cup \{s_0 \curvearrowright d_0\})$	$(\sigma\|s_0\|d_0,\ [\,],\ \beta,\ A_c,\ A_w)$
POP-X:	$(\sigma,\ [d_0],\ \beta,\ A_c,\ A_w)$	$(\sigma,\ [\textsc{Subtree}(d_0, A_c)/X],\ \beta,\ A_c,\ A_w)$
REDUCE-C:	$(\sigma,\ \delta\|d_0,\ \beta,\ A_c,\ A_w)$ such that $\exists d \in \delta,\ d \curvearrowright d_0 \in A_c$	$(\sigma,\ \delta,\ \beta,\ A_c,\ A_w)$
REDUCE:	$(\sigma\|s_0,\ \delta,\ \beta,\ A_c,\ A_w)$ such that $\exists s \in \sigma, s \curvearrowright s_0 \in A_w$	$(\sigma,\ \delta,\ \beta,\ A_c,\ A_w)$

State: $(\sigma, \delta, \beta, A\_c, A\_w)$ σ: stack; β:buffer; δ: partial-word buffer;

A_c: the set of character dependencies; A_w: the set of word dependencies

Joint Models

Joint Word Segmentation, POS-tagging and Dependency Parsing

- Example

Step	σ	δ	β	A_c	A_w	*Action*
0			我，来，到，会，客，室			*SHIFT-C*

我, 来, 到, 会, 客, 室

Joint Models

Joint Word Segmentation, POS-tagging and Dependency Parsing

- Example

Step	σ	δ	β	A_c	A_w	*Action*
1		我	来，到，会，客，室			*POP-PN*

我

来, 到, 会, 客, 室

Joint Models

Joint Word Segmentation, POS-tagging and Dependency Parsing

- Example

Step	σ	δ	β	A_c	A_w	*Action*
2		我/PN	来，到，会，客，室			*SHIFT-W*

我/PN

来, 到, 会, 客, 室

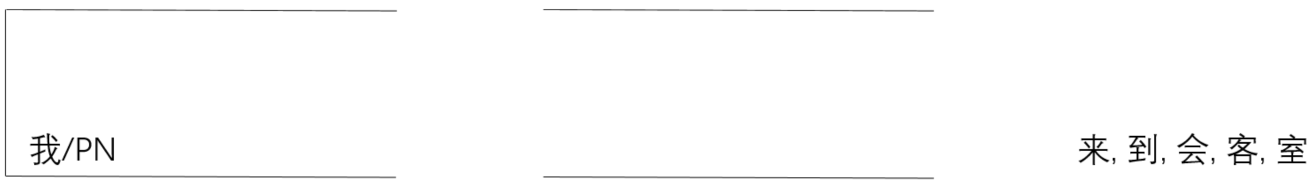
Joint Models

WestlakeNLP

Joint Word Segmentation, POS-tagging and Dependency Parsing

- Example

Step	σ	δ	β	A_c	A_w	*Action*
3	我/PN		来，到，会，客，室			*SHIFT-C*



Joint Models

Joint Word Segmentation, POS-tagging and Dependency Parsing

- Example

Step	σ	δ	β	A_c	A_w	*Action*
4	我/PN	来	到，会，客，室			*LEFTARC-C*

我/PN

来

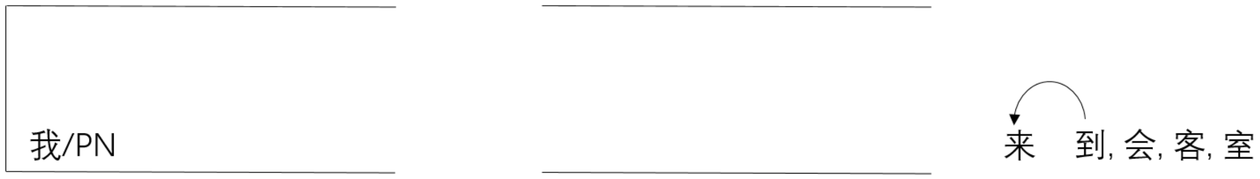
到, 会, 客, 室

Joint Models

Joint Word Segmentation, POS-tagging and Dependency Parsing

- Example

Step	σ	δ	β	A_c	A_w	*Action*
5	我/PN		到，会，客，室	来← 到		*SHIFT*

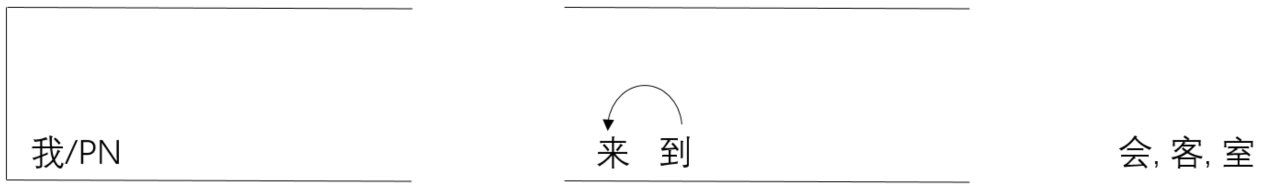


Joint Models

Joint Word Segmentation, POS-tagging and Dependency Parsing

- Example

Step	σ	δ	β	A_c	A_w	*Action*
6	我/PN	到	到，会，客，室	来← 到		*POP-VV*

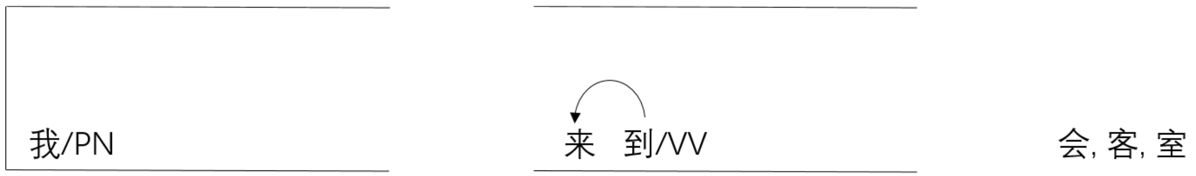


Joint Models

Joint Word Segmentation, POS-tagging and Dependency Parsing

- Example

Step	σ	δ	β	A_c	A_w	*Action*
7	我/PN	来到/VV	会，客，室	来← 到		*LEFTARC-SUBJ*

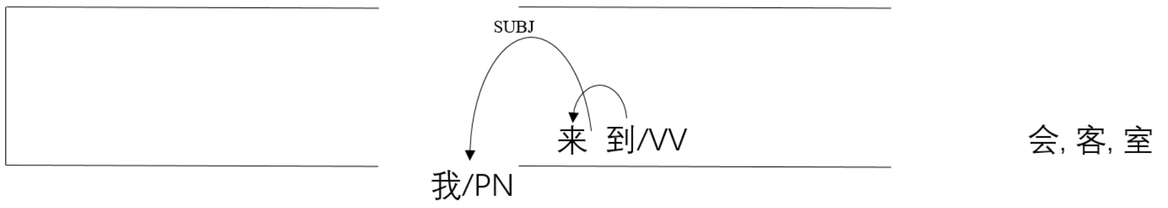


Joint Models

Joint Word Segmentation, POS-tagging and Dependency Parsing

- Example

Step	σ	δ	β	A_c	A_w	*Action*
8		来到/VV	会，客，室	来← 到	我/PN$\leftarrow_{SUBJ}$ 来到/VV	*SHIFT*

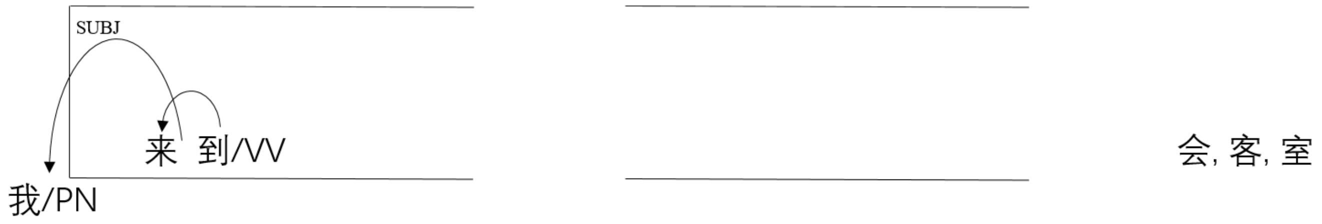


Joint Models

Joint Word Segmentation, POS-tagging and Dependency Parsing

- Example

Step	σ	δ	β	A_c	A_w	*Action*
9	来到/VV		会，客，室	来← 到	我/PN $\xleftarrow{SUBJ}$ 来到/VV	*SHIFT-C*



Joint Models

Joint Word Segmentation, POS-tagging and Dependency Parsing

- Example

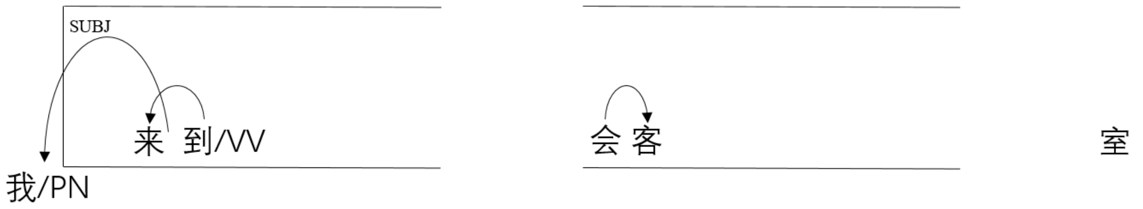
$Step$	σ	δ	β	A_c	A_w	$Action$
10	来到/VV	会	客，室	来← 到	我/PN$\leftarrow_{SUBJ}$ 来到/VV	$RIGHTARC\text{-}C$

Joint Models

Joint Word Segmentation, POS-tagging and Dependency Parsing

- Example

Step	σ	δ	β	A_c	A_w	*Action*
11	来到/VV	会，客	室	来← 到，会→客	我/PN$\leftarrow$ *SUBJ* 来到/VV	*REDUCE-C*

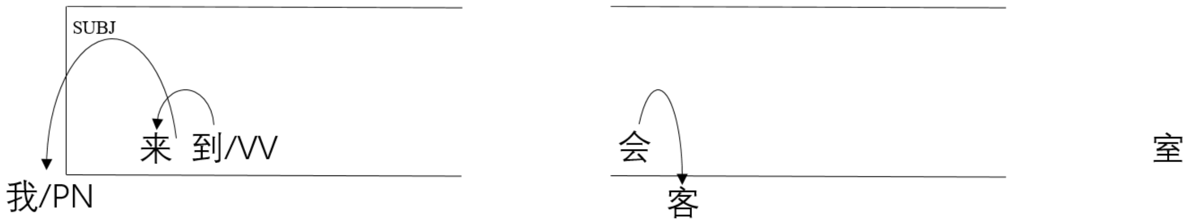


Joint Models

Joint Word Segmentation, POS-tagging and Dependency Parsing

- Example

Step	σ	δ	β	A_c	A_w	*Action*
12	来到/VV	会	室	来$\leftarrow$ 到，会$\rightarrow$客	我/PN$\xleftarrow{\textsc{subj}}$ 来到/VV	*LEFTARC-C*

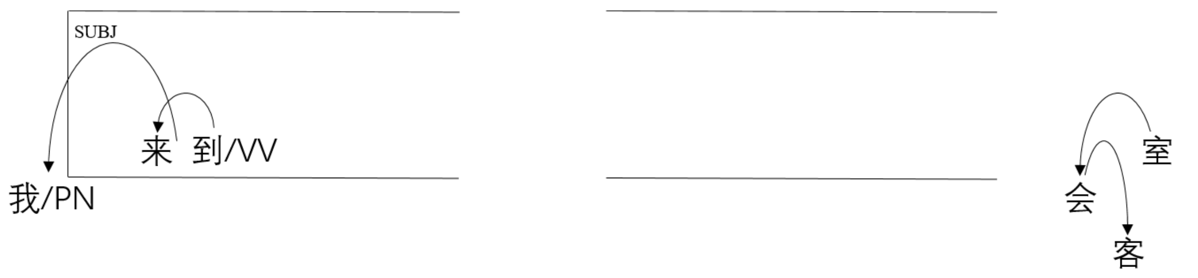


Joint Models

Joint Word Segmentation, POS-tagging and Dependency Parsing

- Example

$Step$	σ	δ	β	A_c	A_w	$Action$
13	来到/VV		室	来← 到，会→客，会←室	我/PN$\leftarrow_{SUBJ}$ 来到/VV	$SHIFT\text{-}C$

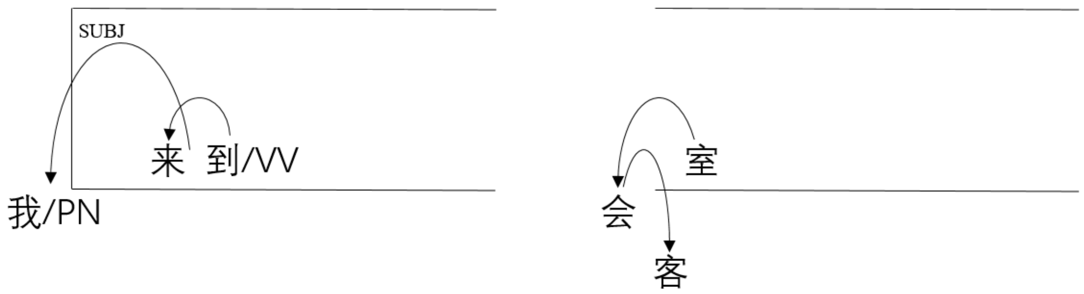


Joint Models

Joint Word Segmentation, POS-tagging and Dependency Parsing

- Example

$Step$	σ	δ	β	A_c	A_w	$Action$
14	来到/VV	室		来← 到，会→客，会←室	我/PN$\leftarrow_{SUBJ}$ 来到/VV	$POP\text{-}NN$



Joint Models

Joint Word Segmentation, POS-tagging and Dependency Parsing

- Example

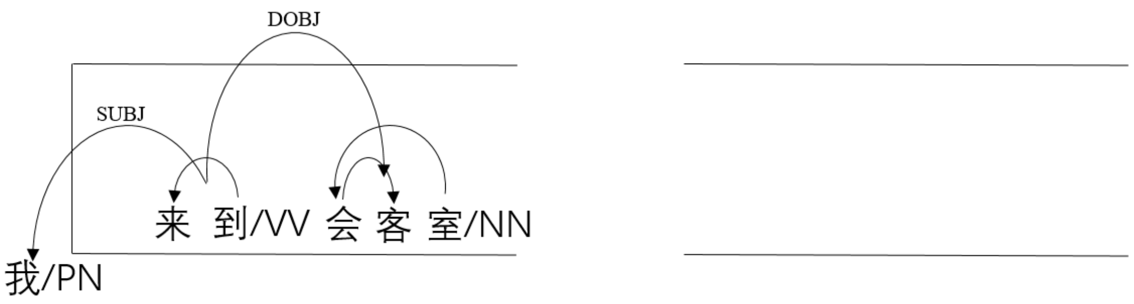
Step	σ	δ	β	A_c	A_w	*Action*
15	来 到/VV	会客 室/NN		来← 到，会→客，会 ←室	我/PN$\leftarrow_{SUBJ}$ 来 到/VV	*RIGHTARC-DOBJ*

Joint Models

Joint Word Segmentation, POS-tagging and Dependency Parsing

- Example

Step	σ	δ	β	A_c	A_w	*Action*
16	来到/VV，会客室/NN			来← 到，会→客，会←室	我/PN← *SUBJ* 来到/VV，来到/VV *DOBJ* →会客室/NN	*REDUCE*

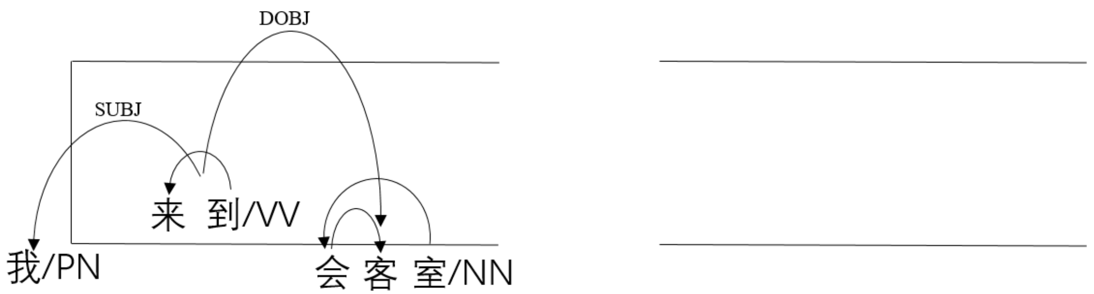


Joint Models

Joint Word Segmentation, POS-tagging and Dependency Parsing

- Example

Step	σ	δ	β	A_c	A_w	*Action*
17	来 到/VV			来← 到，会→客，会 ←室	我/PN← *SUBJ* 来到/VV，来到/VV *DOBJ* →会客室/NN	*END*



Summary

- What is transition-based method?
- Apply transition-based methods on different tasks.
- Joint modeling with transition-based methods.