2017年度III期

自然言語処理
（Natural Language Processing）

2017/11/02

Hiroyuki Shindo

目次

Advanced topics in natural language processing

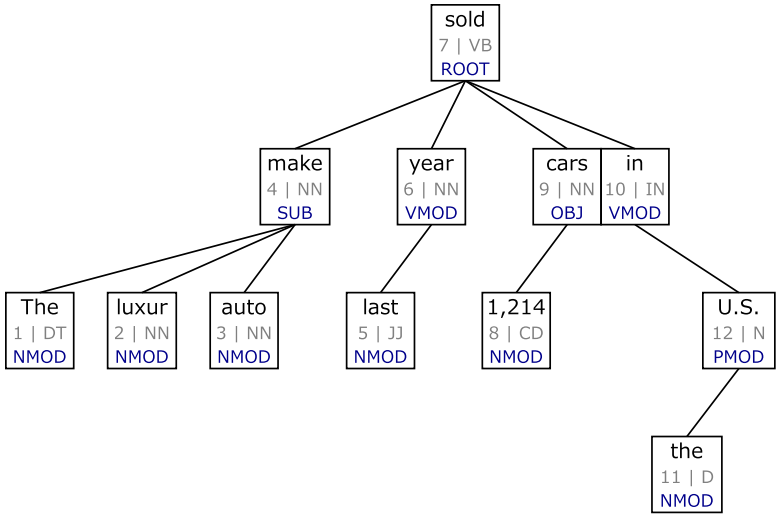
- 自然言語処理の概要
- 構文解析
  - 特に依存構造解析のアルゴリズム
- 意味解析
  - 述語項構造解析

Quick Review of What We've Learned （先週の復習）

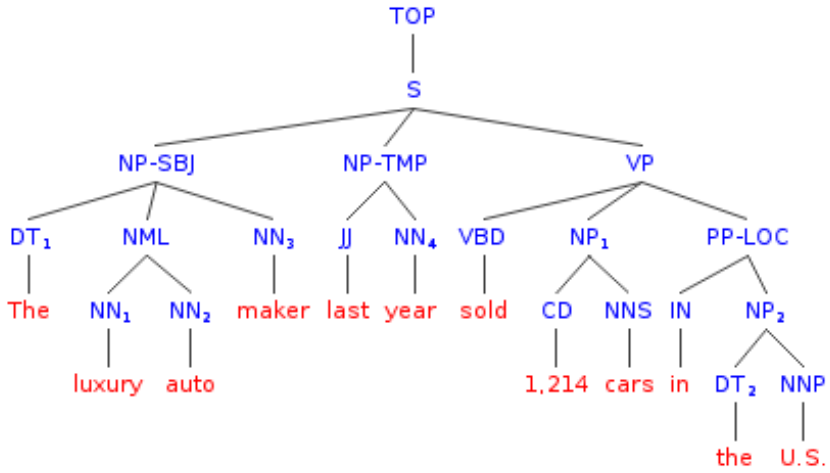
Syntactic Structures

2種類の統語構造

dependency structure（依存構造）

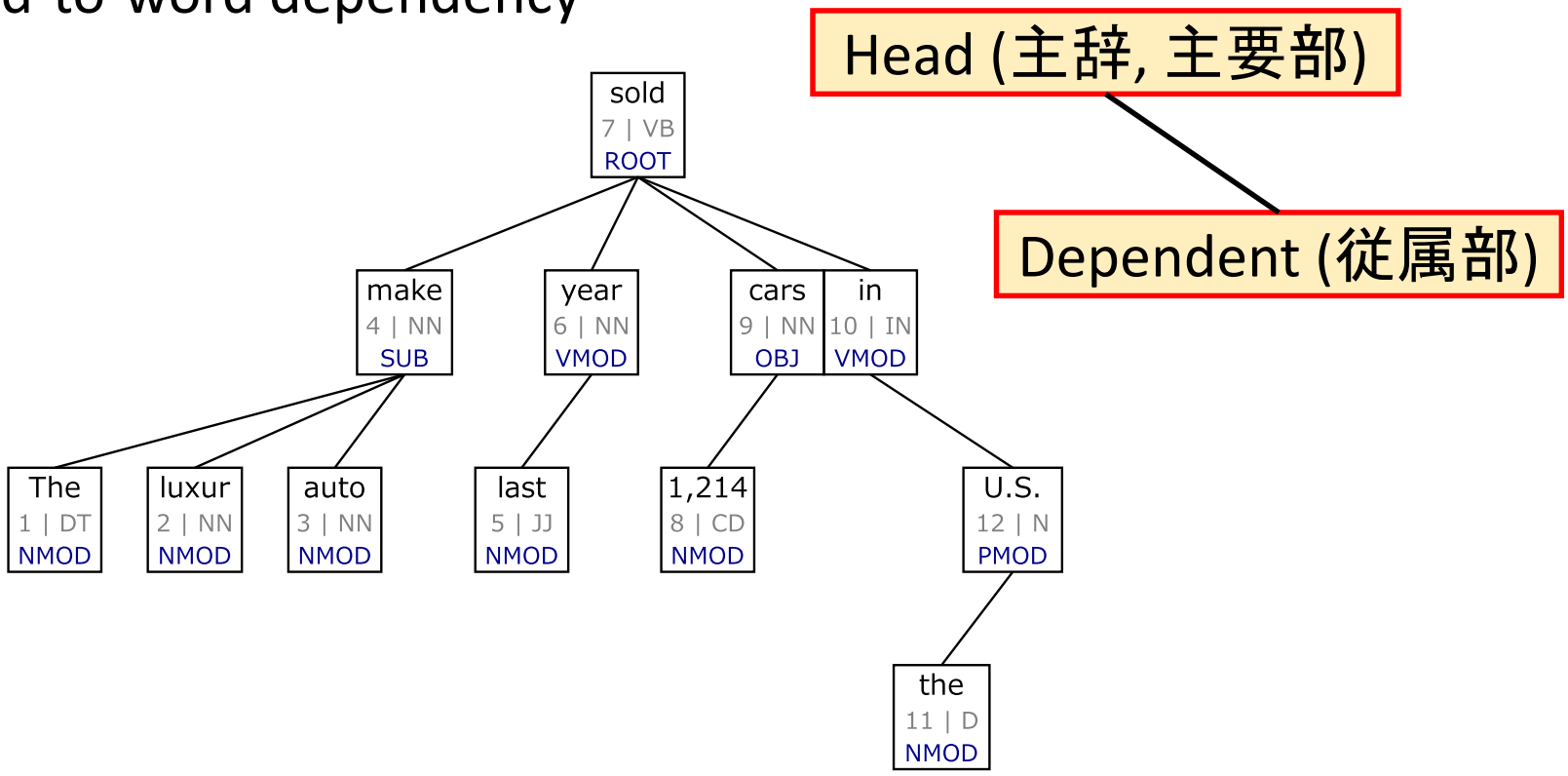


phrase structure（句構造）



Dependency Structure（依存構造）

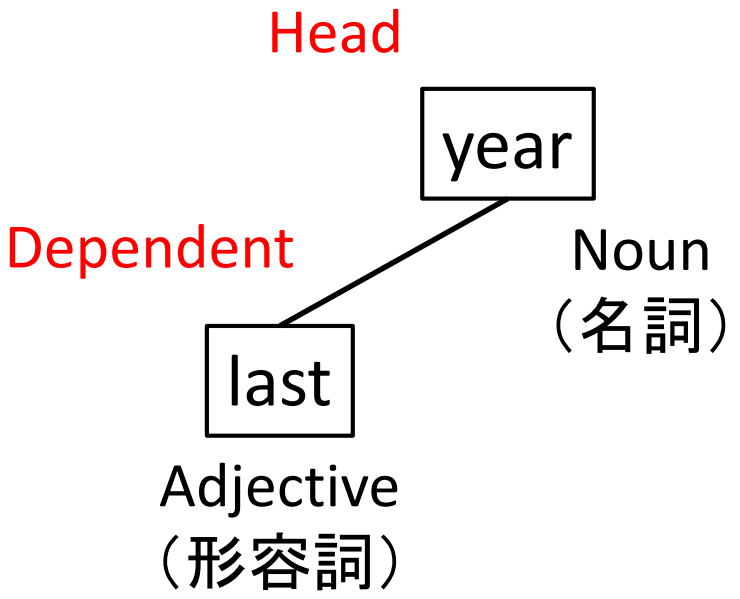
Word-to-word dependency



The luxury auto maker last year sold 1,214 cars in the U.S.

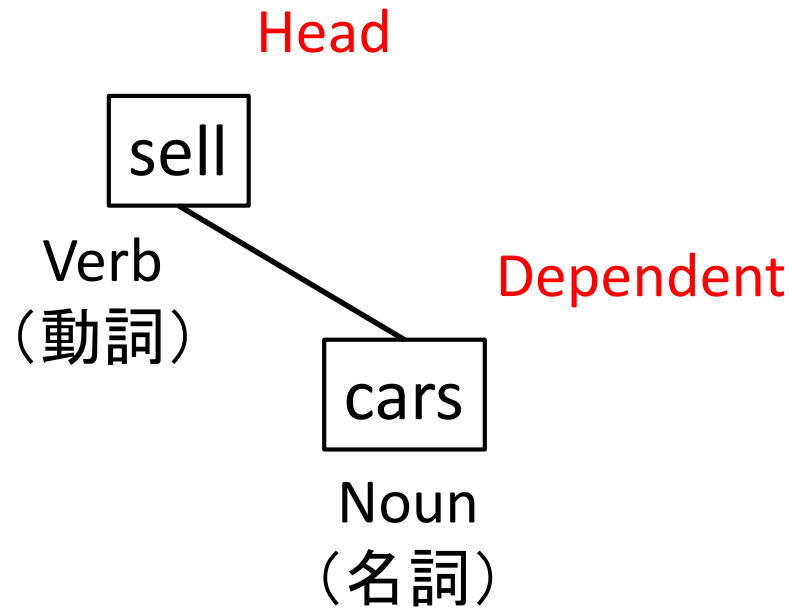
Dependency Structure（依存構造）

Examples:

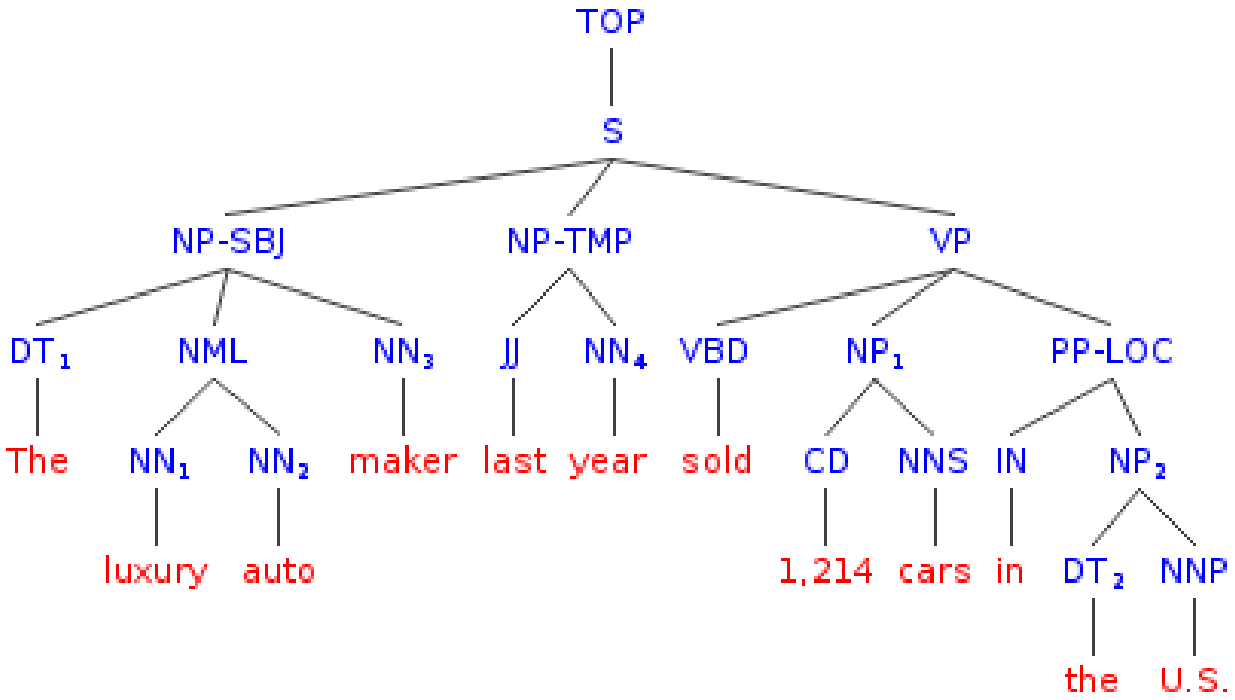


Dependency Structure（依存構造）

Examples:



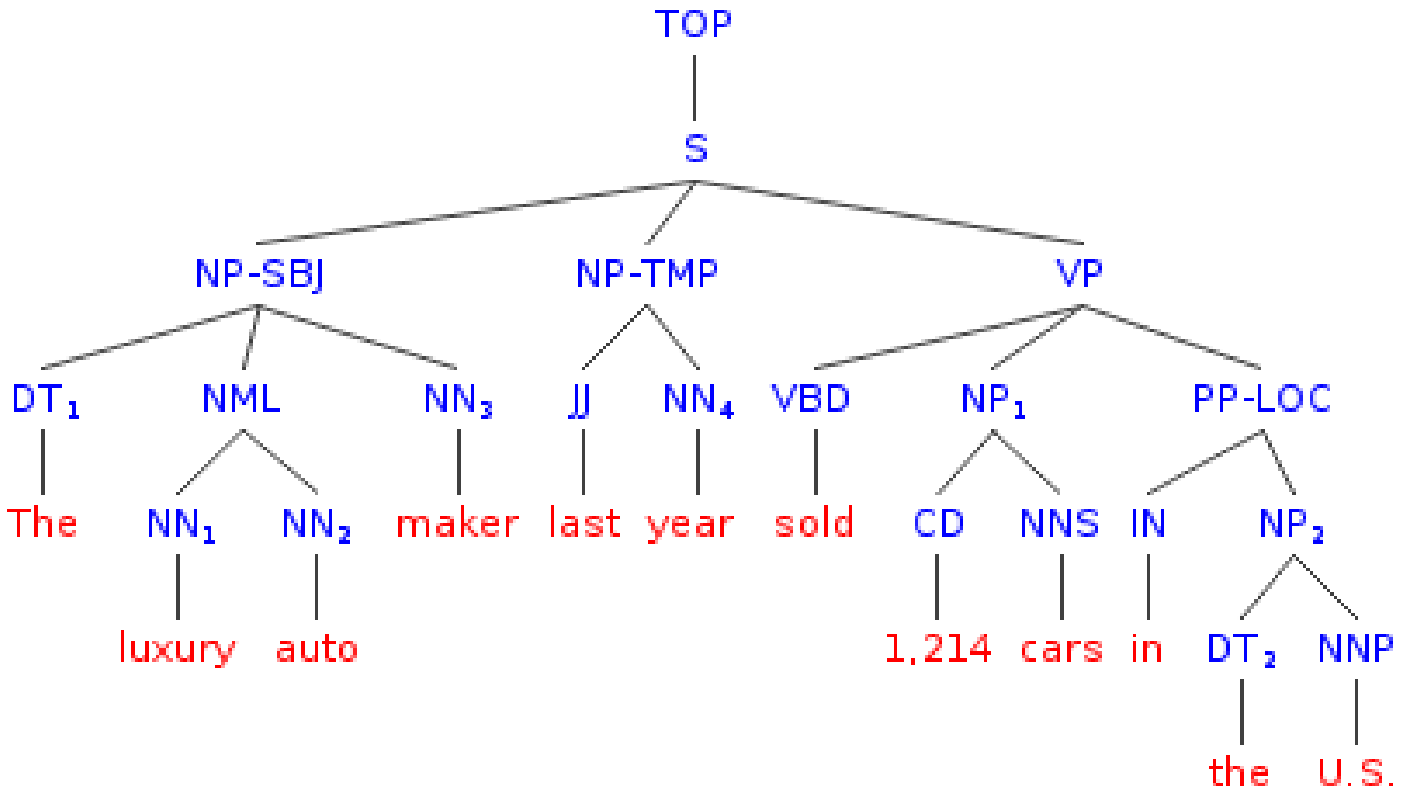
Phrase Structure（句構造）



The luxury auto maker last year sold 1,214 cars in the U.S.

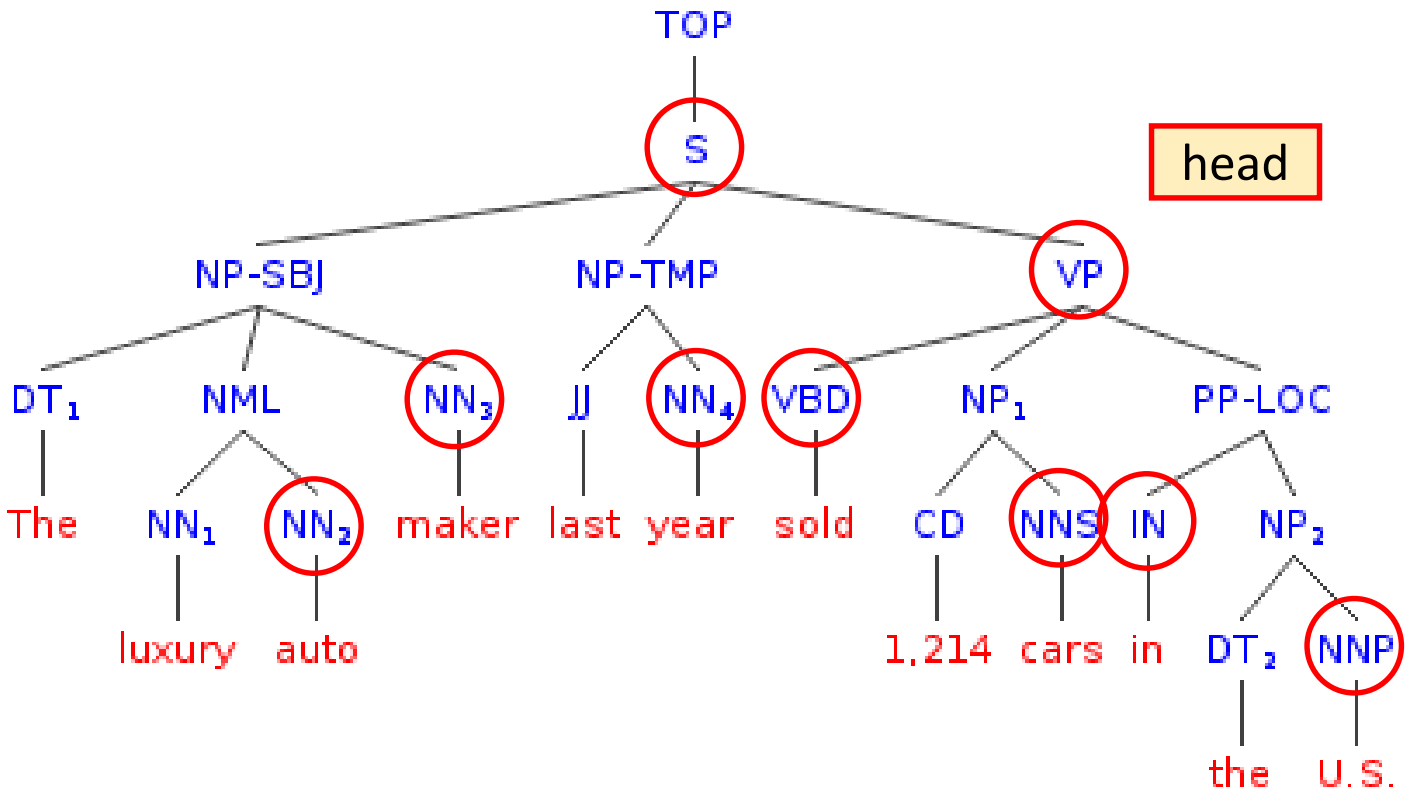
Phrase to Dependency Conversion

句構造を依存構造へ変換できる



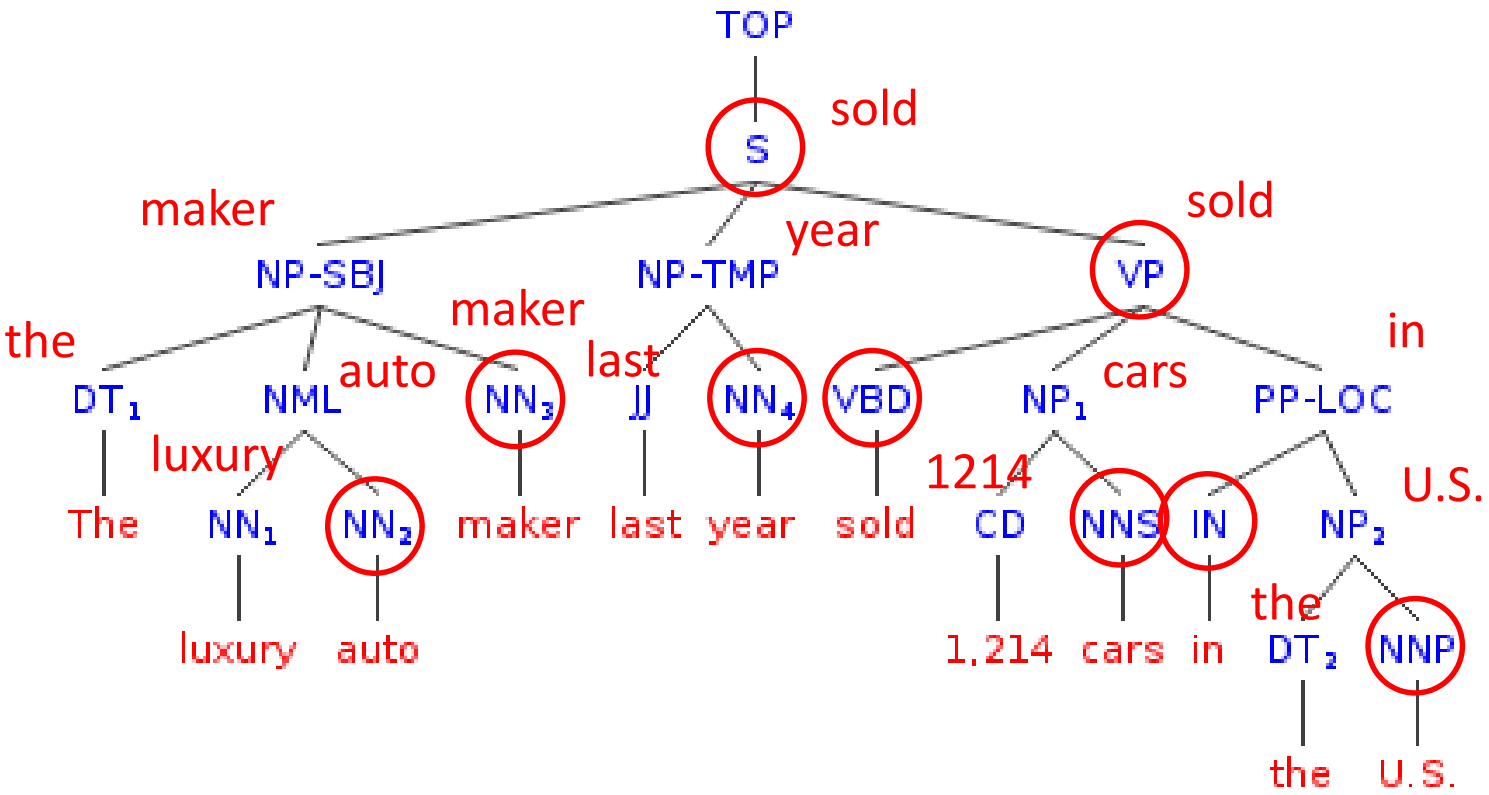
Phrase to Dependency Conversion

句構造を依存構造へ変換できる



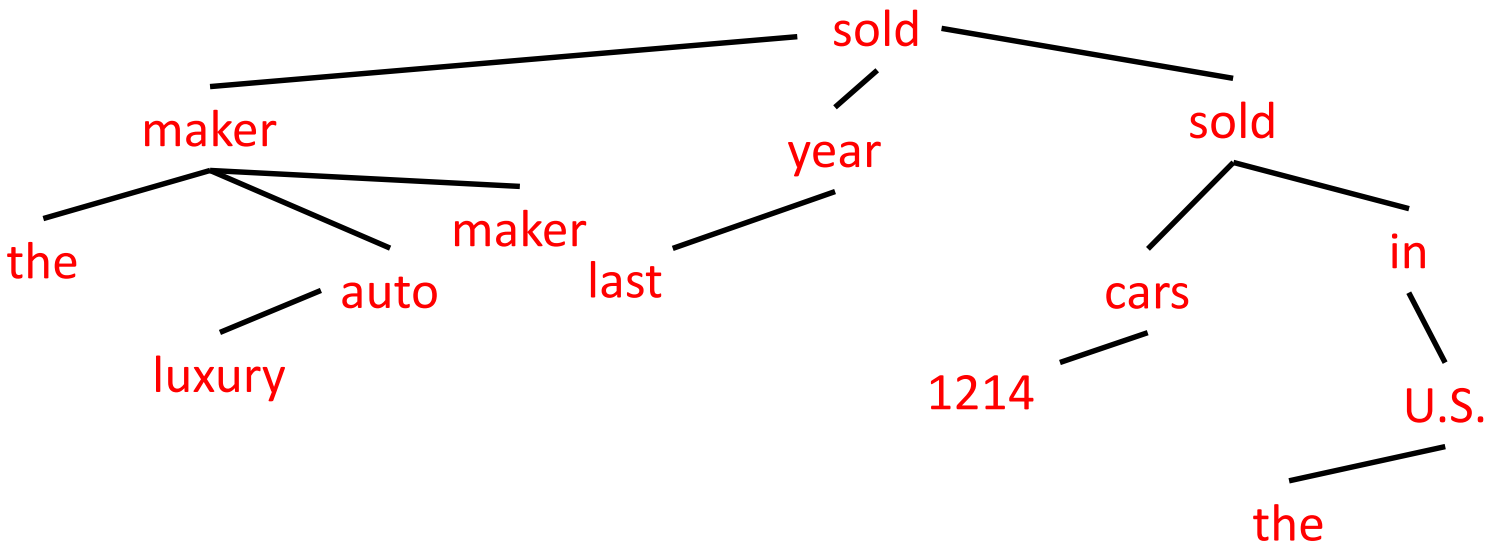
Phrase to Dependency Conversion

句構造を依存構造へ変換できる



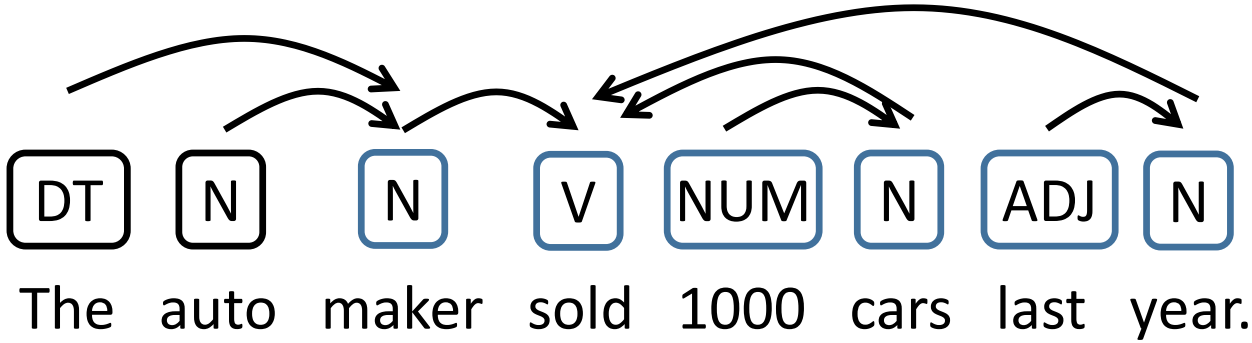
Phrase to Dependency Conversion

句構造を依存構造へ変換できる



Projective v.s. Non-projective

Projectiveな依存構造木

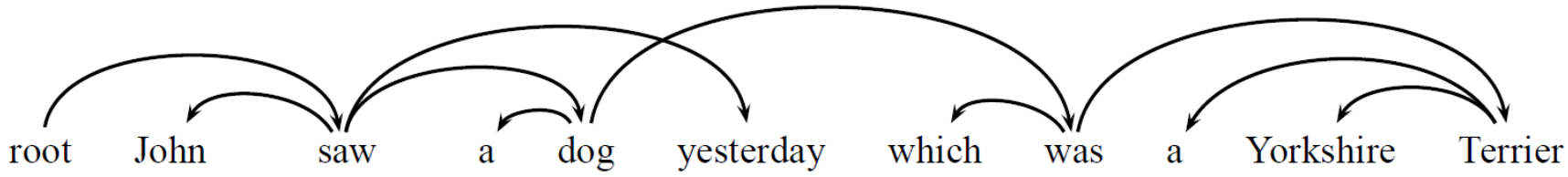


- 交差がない

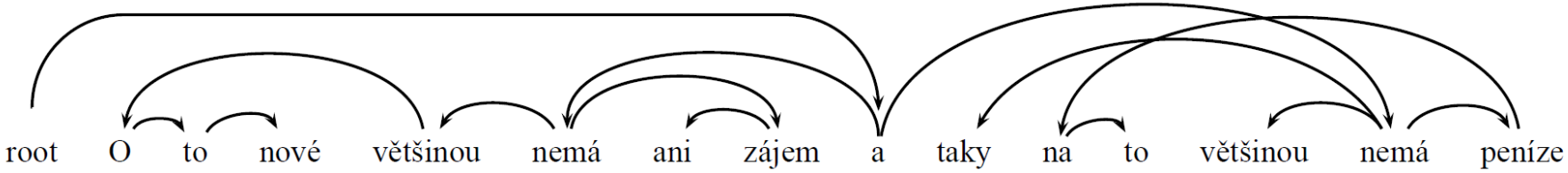
Projective v.s. Non-projective

Non-Projectiveな依存構造

English



Czech



He is mostly not even interested in the new things and in most cases, he has no money for it either.

Taken from “Non-projective Dependency Parsing using Spanning Tree Algorithms”, 2005

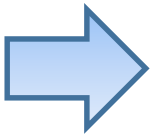
構文解析(依存構造解析)

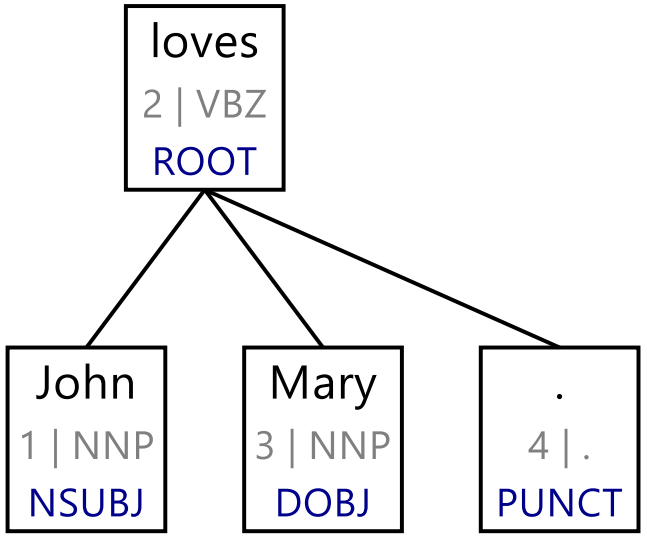
Input

Output

John loves Mary .

Parse





依存構造解析のアルゴリズム

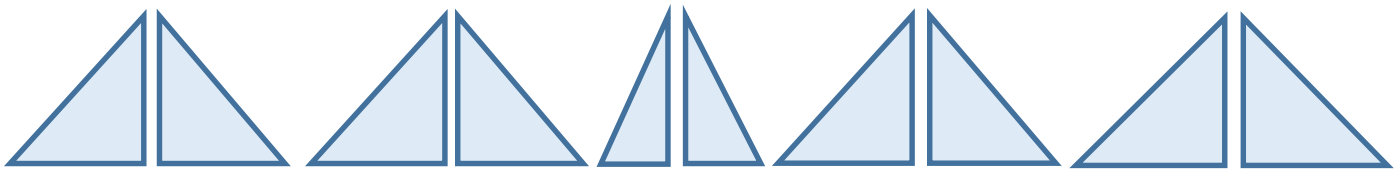
- Transition-based Parsing（遷移型構文解析）　projective
  - Shift-reduce algorithm [Yamada+ '03, Nivre '03]　projective
  - Easy-first (non-directional) algorithm [Goldberg+ '10]

- Graph-based Parsing（グラフ型構文解析）　projective
  - Eisner algorithm (CYK algorithm) [Eisner '96]
  - Chu-Liu-Edmonds algorithm [Chu+ '65, McDonald+ '05]　non-projective

- その他
  - Sampling-based algorithm [Zhang+ '14]　non-projective
  - Hill-climbing algorithm [Zhang+ '14]　non-projective

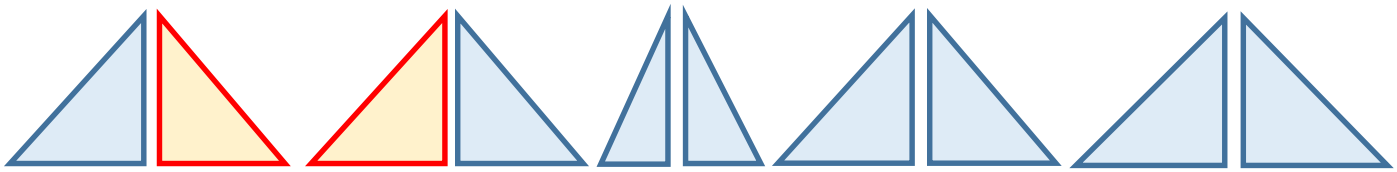
Eisner's Algorithm

Initialization



She	read	a	short	novel.
0	1	2	3	4

Eisner's Algorithm

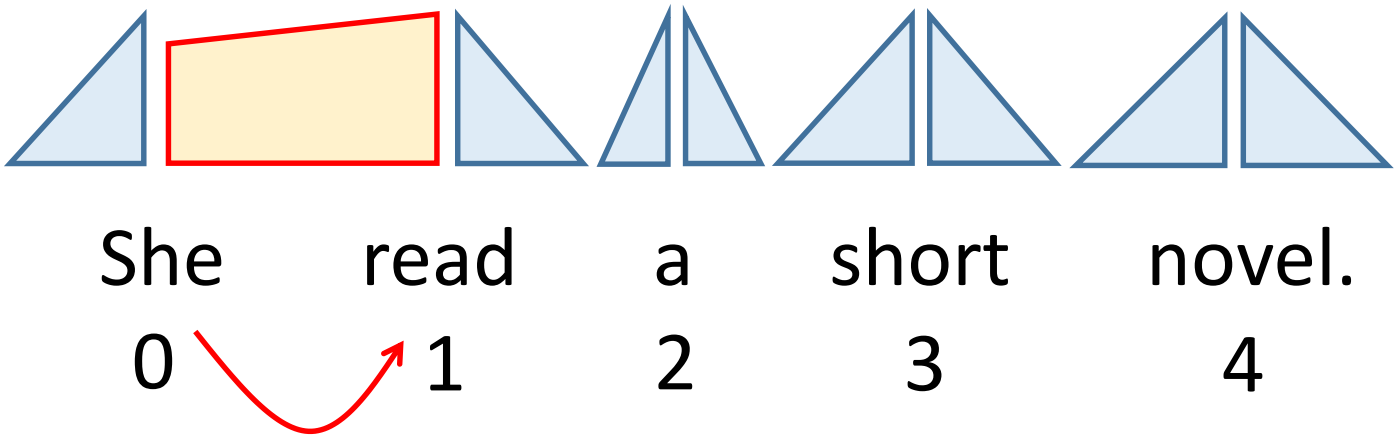


She	read	a	short	novel.
0	1	2	3	4

- Apply rule2:

[0, 1, comp] + [1, 2, comp] → [0, 2, incomp]

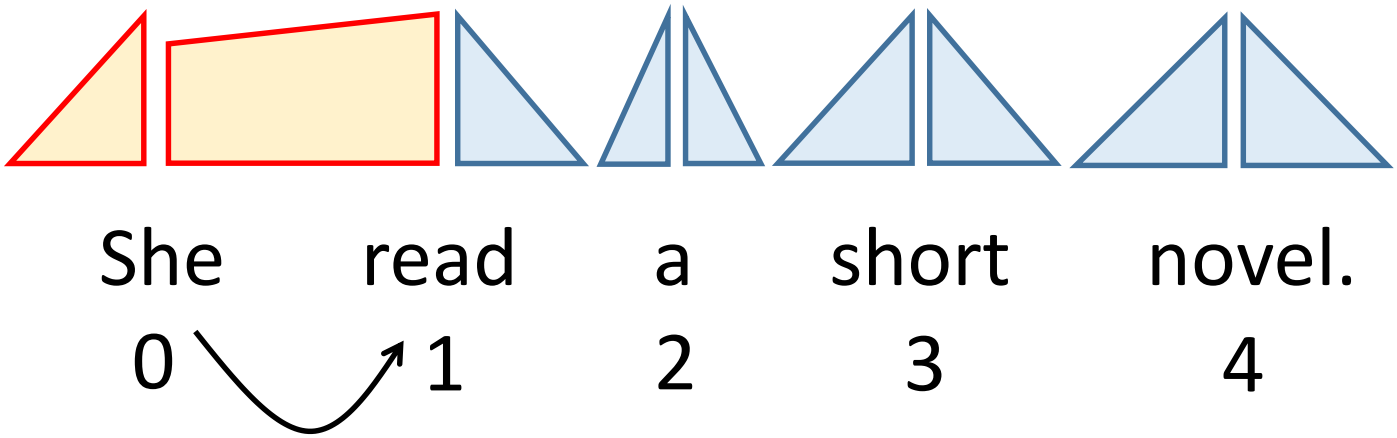
Eisner's Algorithm



- Apply rule2:

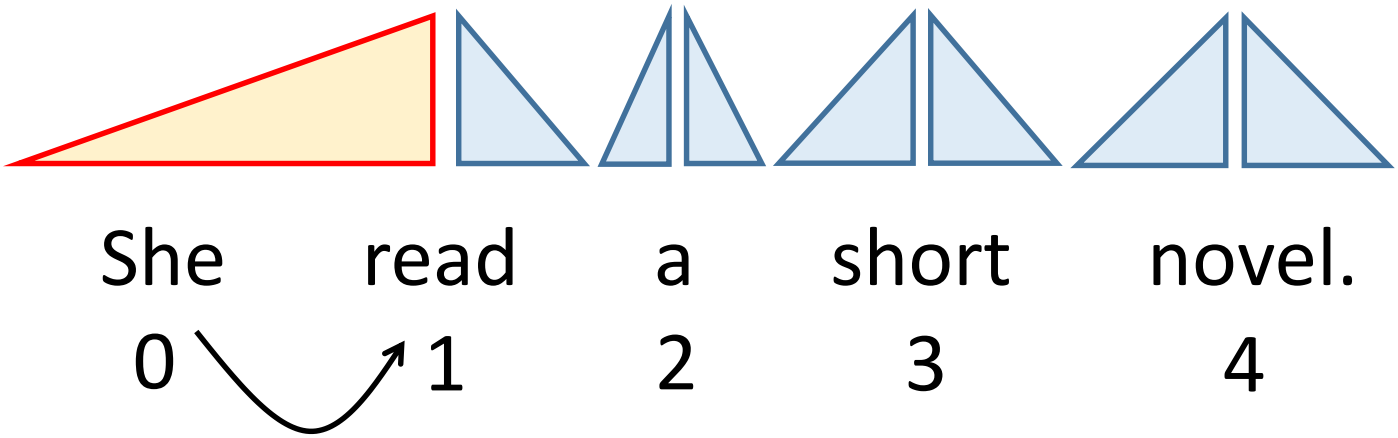
[0, 1, comp] + [1, 2, comp] → [0, 2, incomp]

Eisner's Algorithm



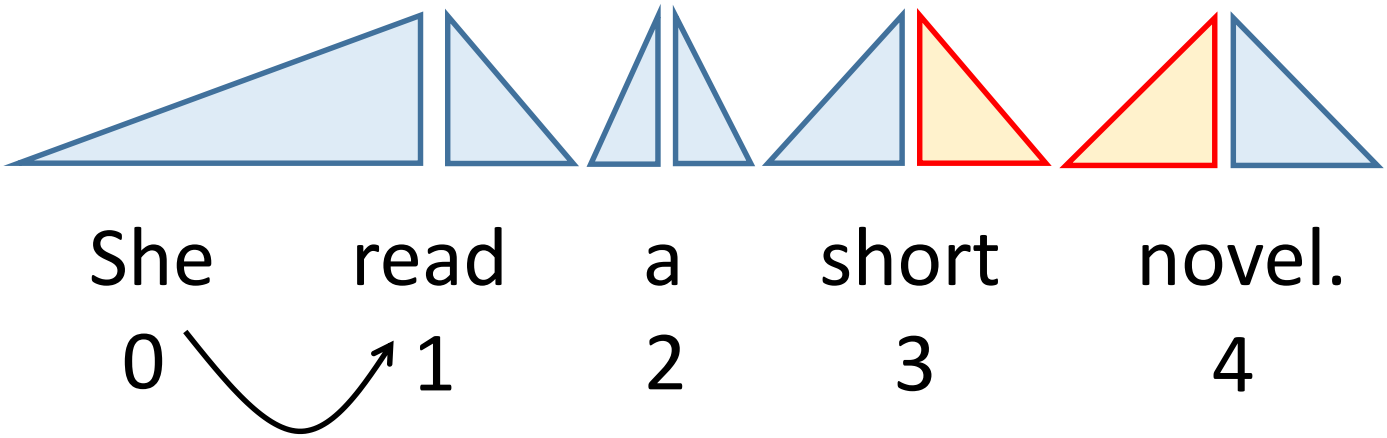
- Apply rule2:
  [0, 1, comp] + [1, 2, comp] → [0, 2, incomp]
- Apply rule4:
  [0, 1, comp] + [1, 2, incomp] → [0, 2, comp]

Eisner's Algorithm

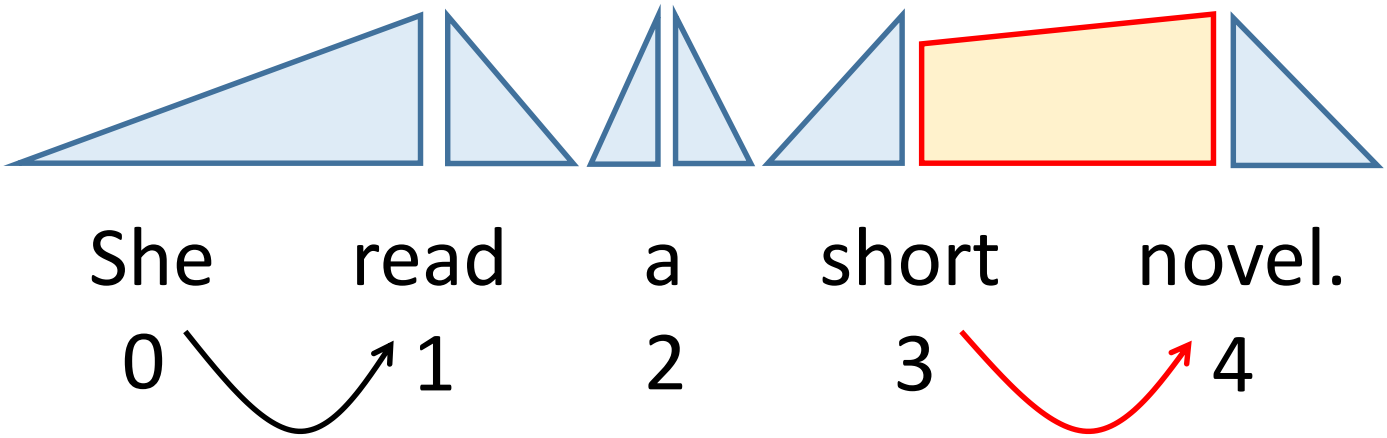


- Apply rule2:
  [0, 1, comp] + [1, 2, comp] → [0, 2, incomp]
- Apply rule4:
  [0, 1, comp] + [1, 2, incomp] → [0, 2, comp]

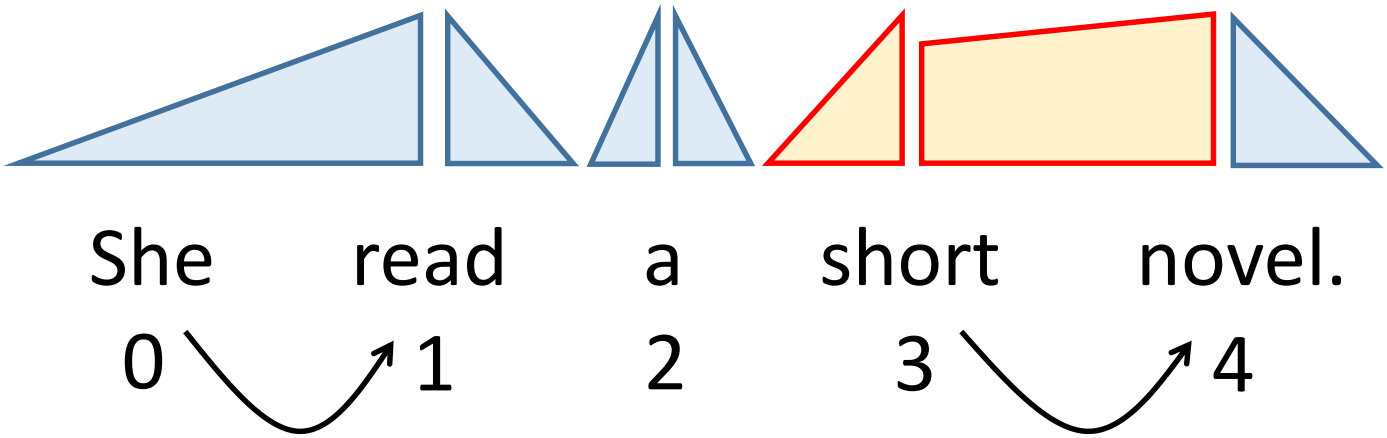
Eisner's Algorithm



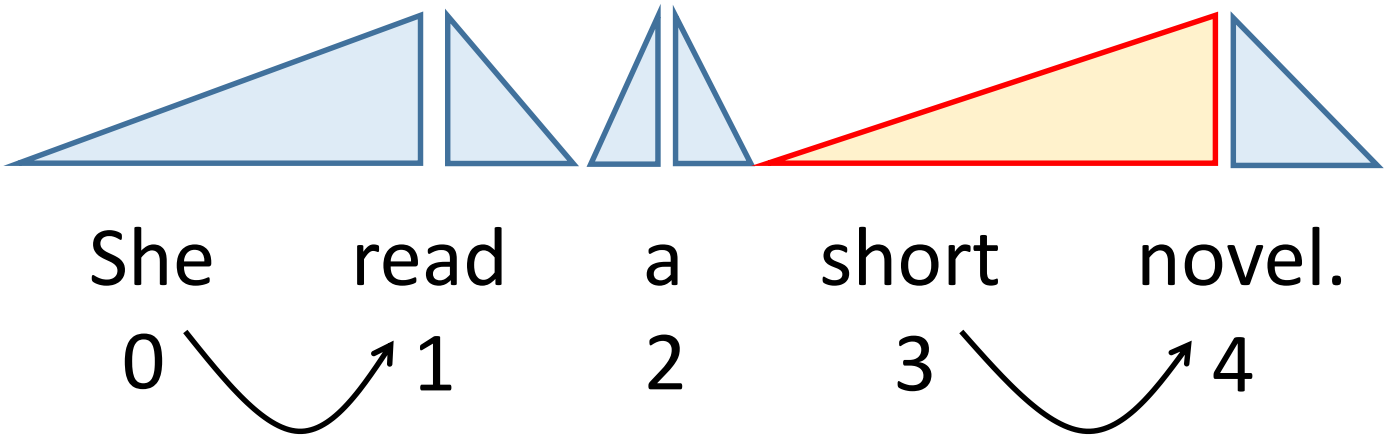
Eisner's Algorithm



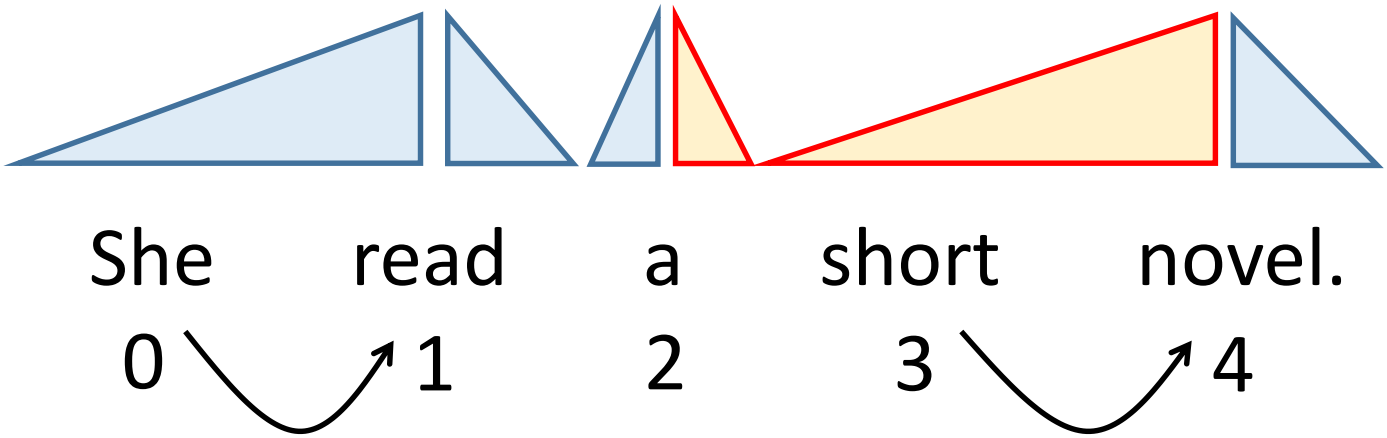
Eisner's Algorithm



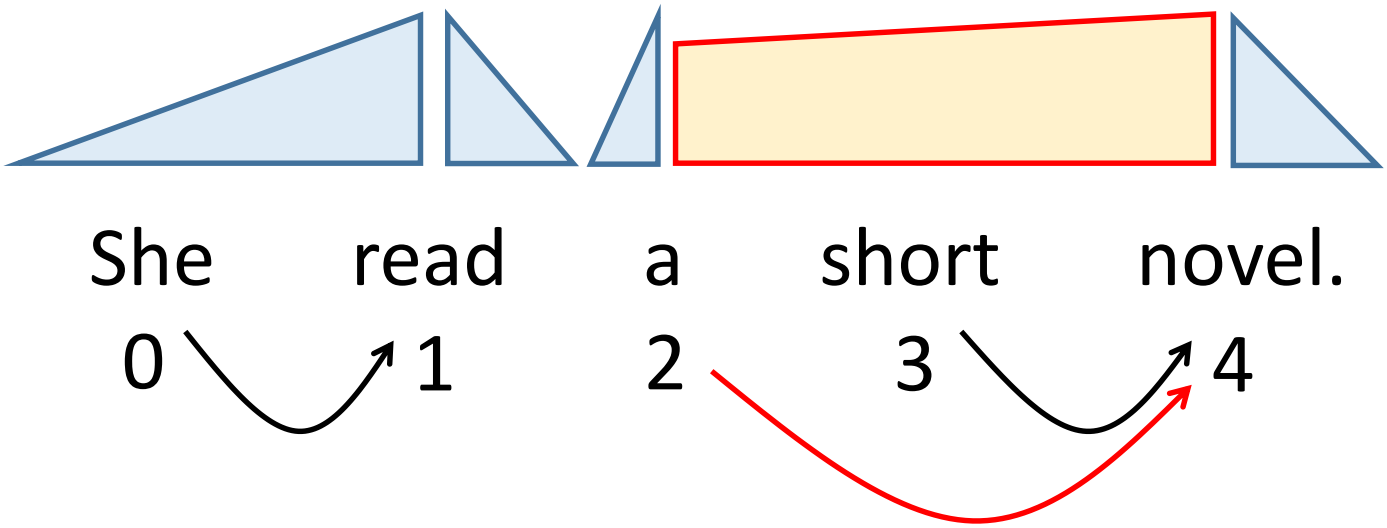
Eisner's Algorithm



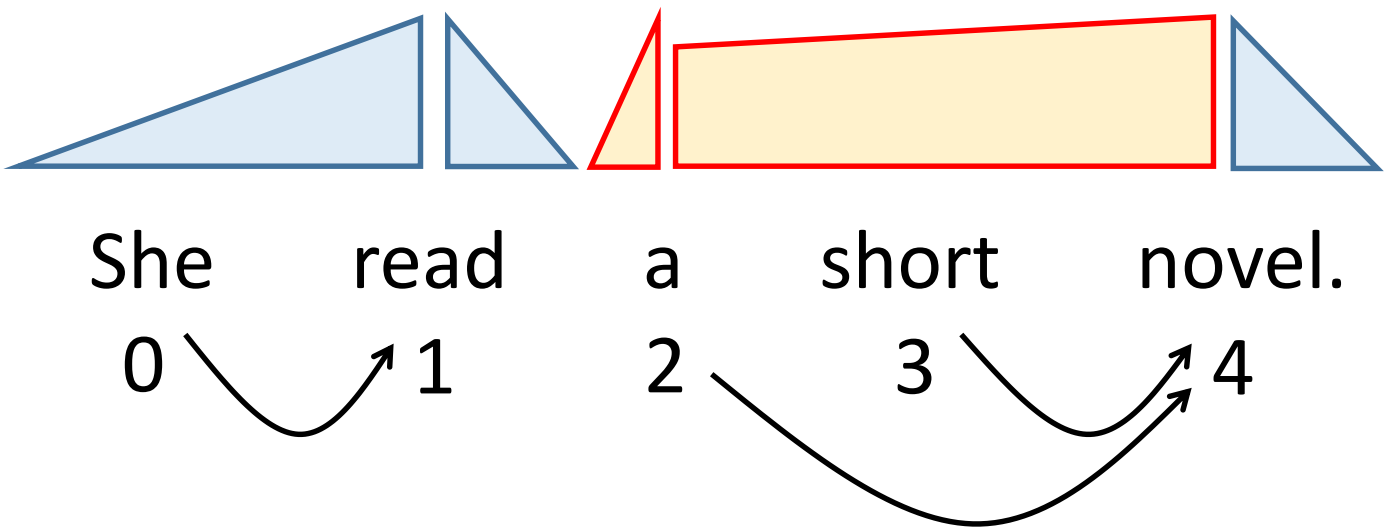
Eisner's Algorithm



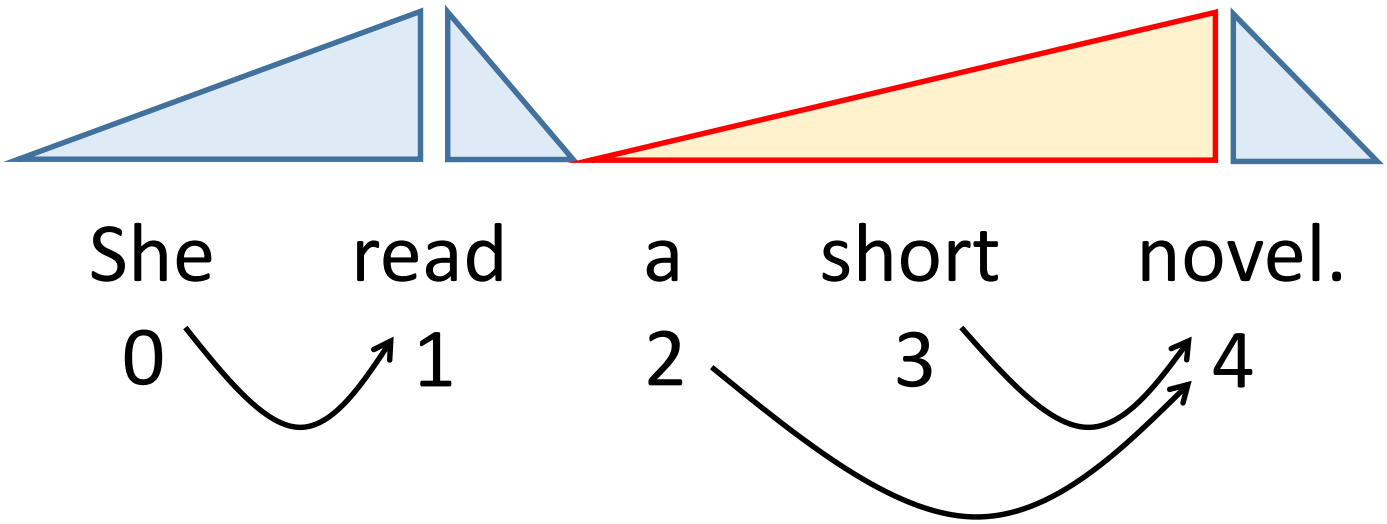
Eisner's Algorithm



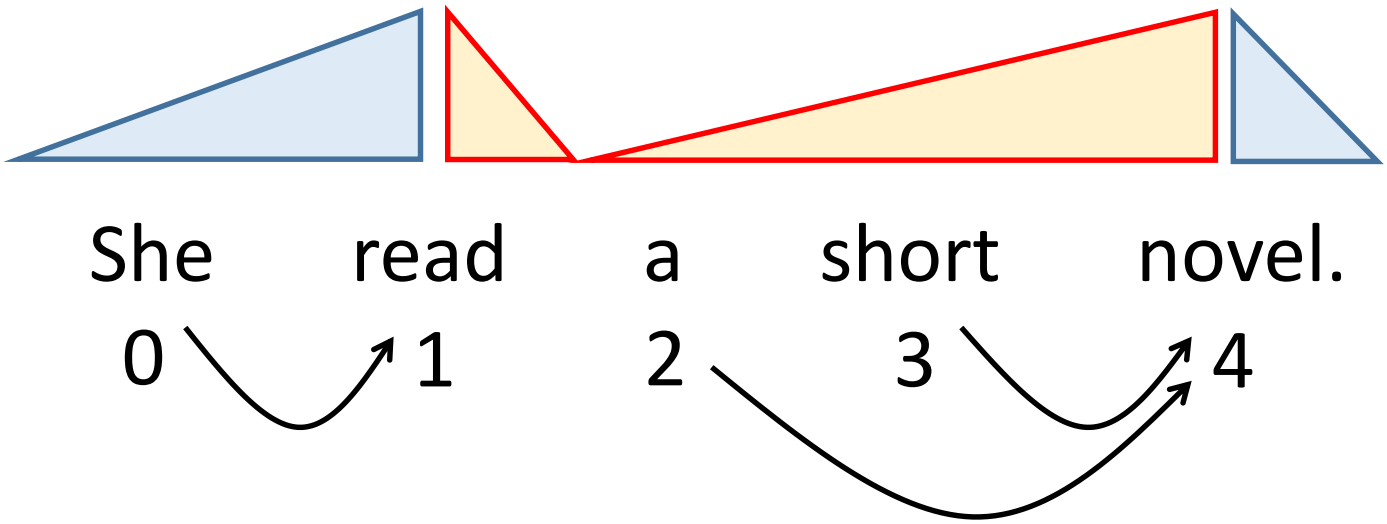
Eisner's Algorithm



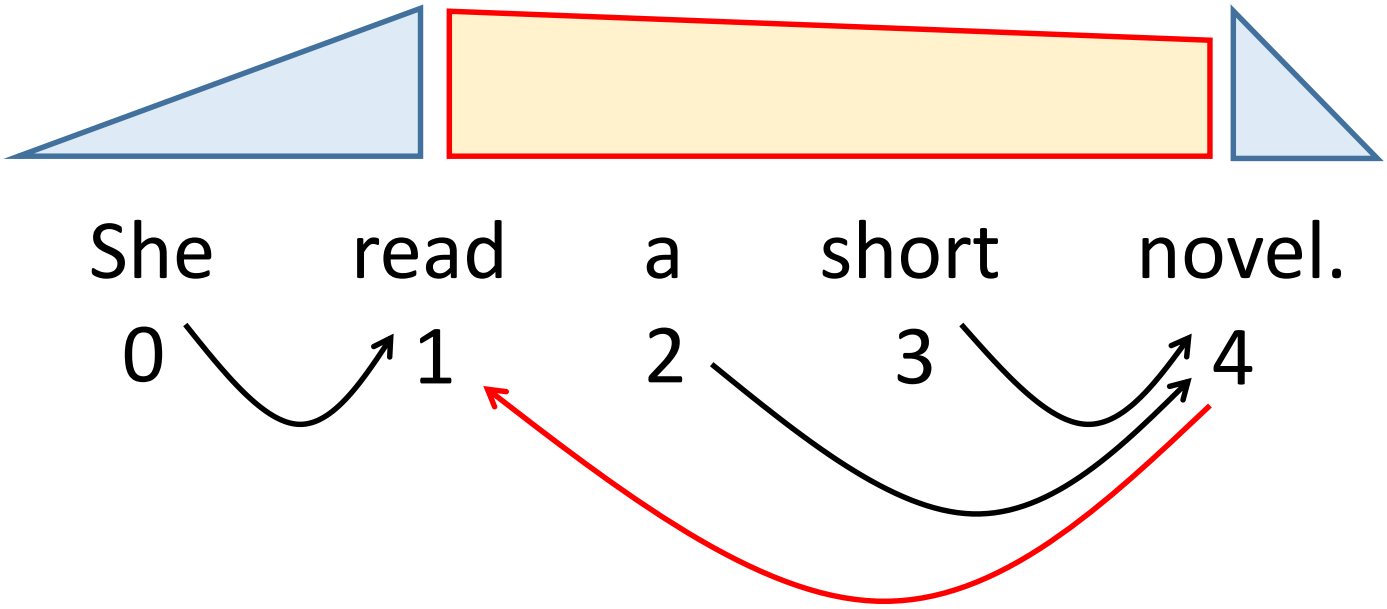
Eisner's Algorithm



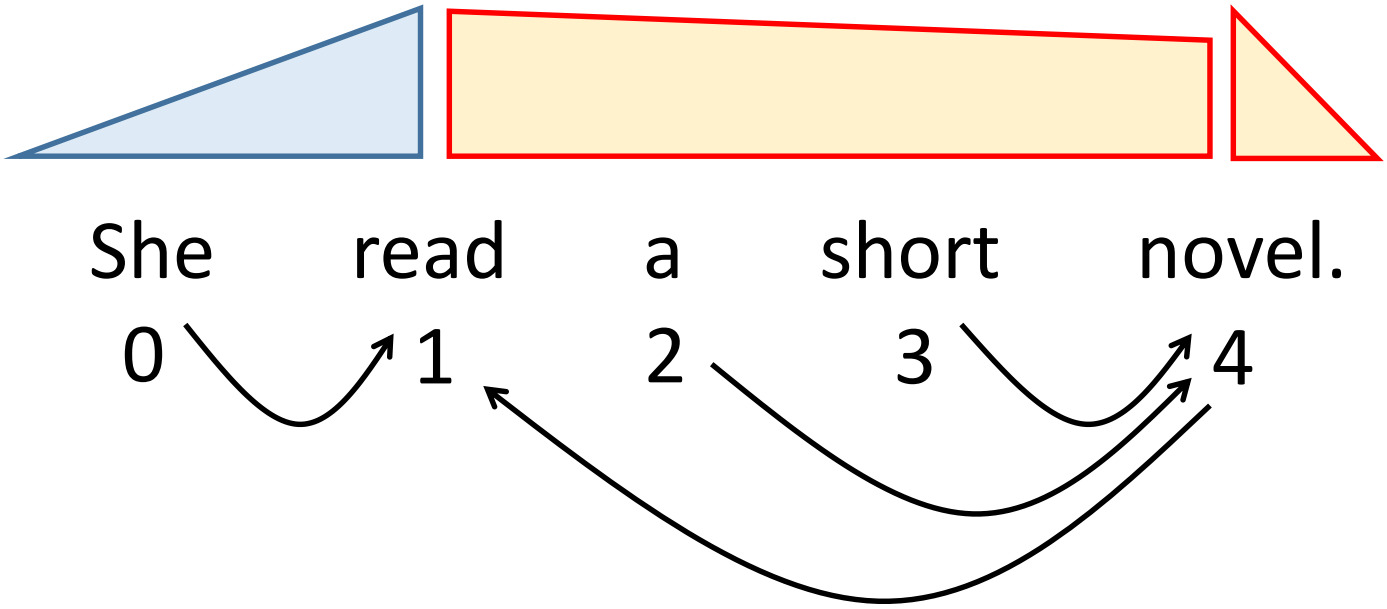
Eisner's Algorithm



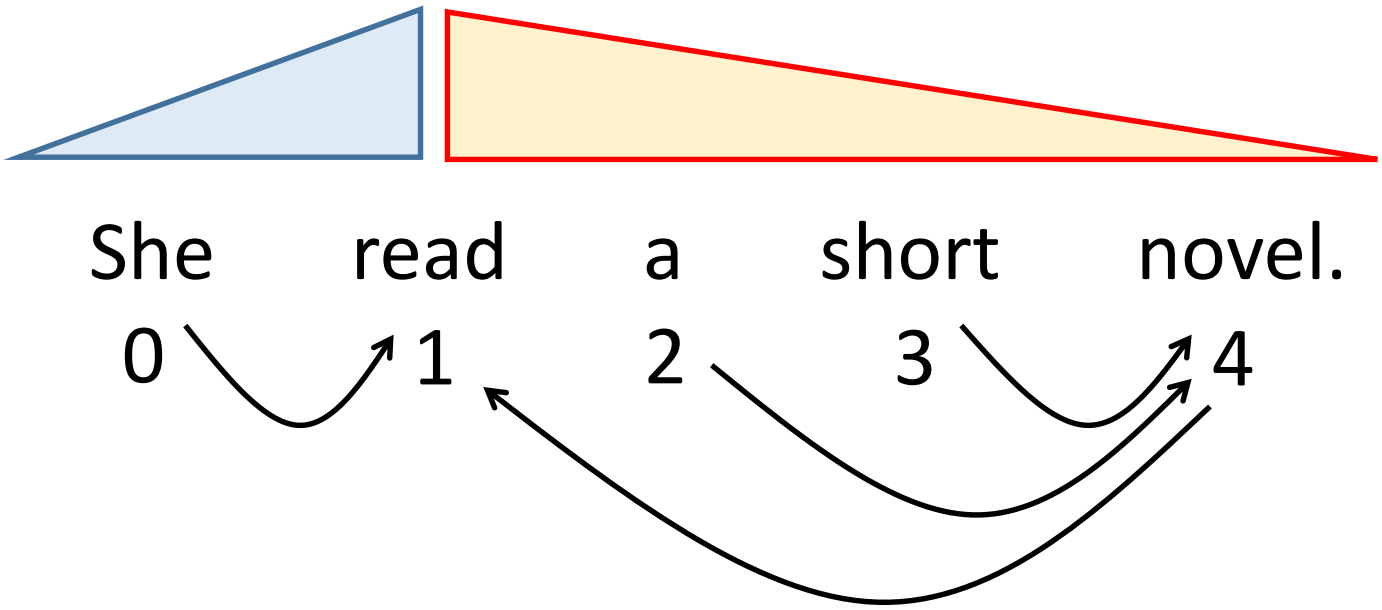
Eisner's Algorithm



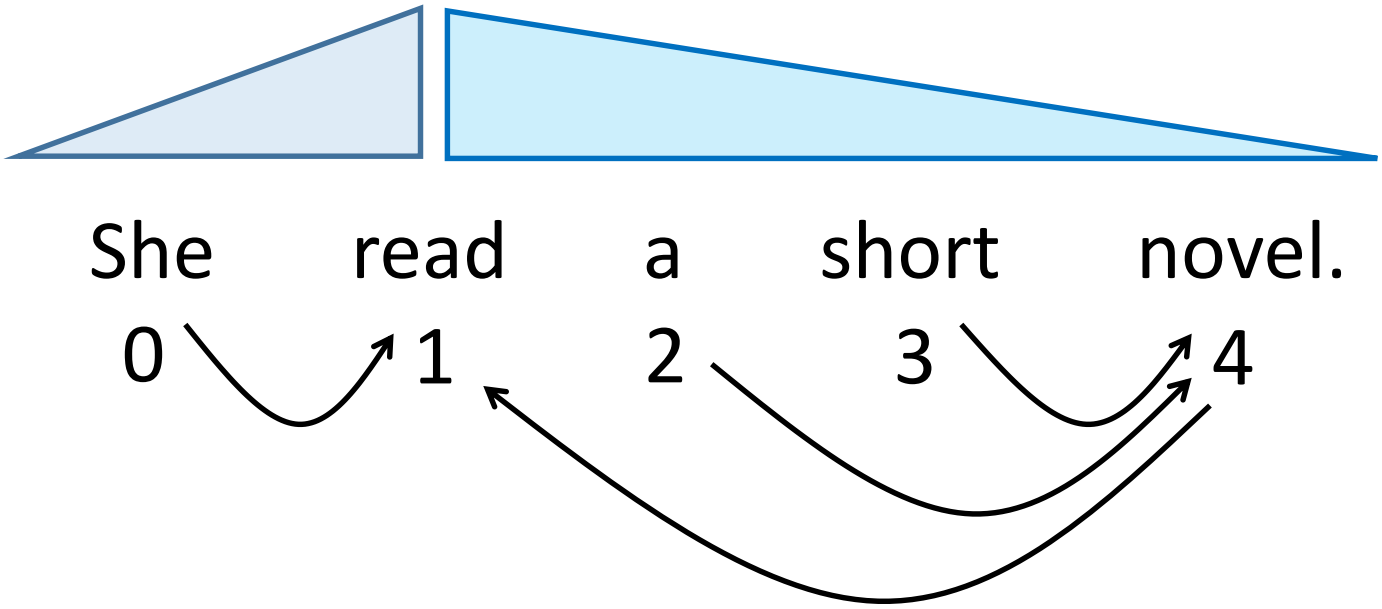
Eisner's Algorithm



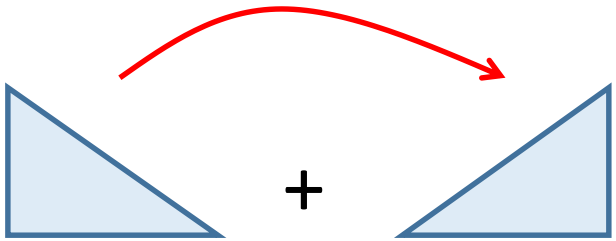
Eisner's Algorithm



Eisner's Algorithm



線形モデルによるスコア計算



w_d w_h

重みベクトル

$w \in R^d$

特徴ベクトル

$v \in R^d$

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ \dots \\ 0 \\ 1 \end{pmatrix} \bullet \begin{pmatrix} 1.1 \\ -0.5 \\ 0.0 \\ -0.1 \\ \dots \\ 3.7 \\ -2.1 \end{pmatrix} = -1.1$$

スコア

依存構造解析のアルゴリズム

- Transition-based Parsing（遷移型構文解析） projective
  - Shift-reduce algorithm [Yamada+ '03, Nivre '03] projective
  - Easy-first (non-directional) algorithm [Goldberg+ '10]

- Graph-based Parsing（グラフ型構文解析） projective
  - Eisner algorithm (CYK algorithm) [Eisner '96]
  - Chu-Liu-Edmonds algorithm [Chu+ '65, McDonald+ '05] non-projective

- その他
  - Sampling-based algorithm [Zhang+ '14] non-projective
  - Hill-climbing algorithm [Zhang+ '14] non-projective

Transition-based Dependency Parsing
（遷移型依存構造解析）

- Shift-reduce 解析
- Easy-first 解析

遷移型依存構造解析

- State（状態）とAction（行動）の２種類の概念を考える.
- 初期状態（＝入力文）から始めて，Actionを選択することを繰り返すことによって，少しずつ依存構造を構築していく.
- 依存構造解析の目的を達成するために，何を「状態」と定義するか，何を「行動」と定義するかによって様々なバリエーションがある.

Shift-Reduce Parsing

State（状態）

Stack
（解析済の単語列）

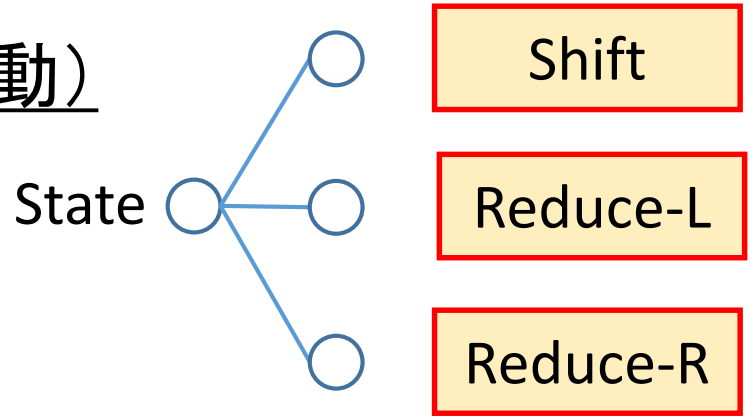
Buffer
（未解析の単語列）

[ROOT The luxury] [auto make last year sold ...]

未完成の依存構造

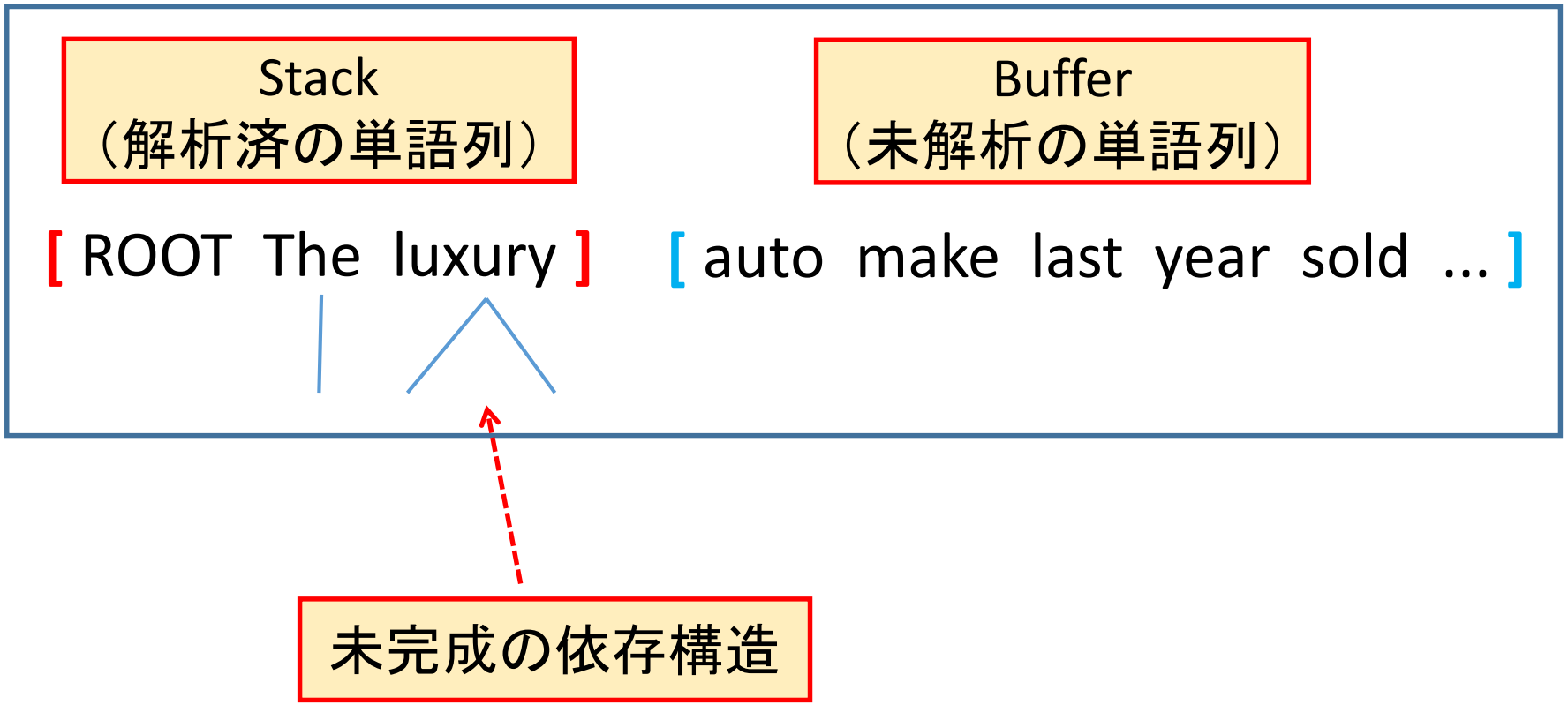
Shift-Reduce Parsing

Action（行動）



Shift-Reduce Parsing

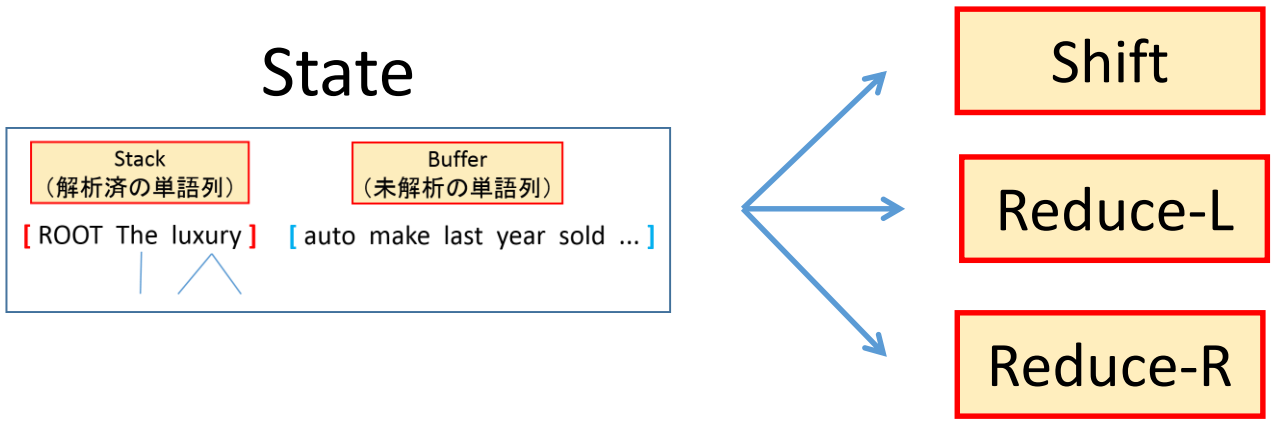
State（状態）



Shift-Reduce Parsing

Action（行動）

ある状態を次の状態へ遷移させる



Shift-Reduce Parsing (Arc-Standard)

初期状態

Shift

[ROOT] [The luxury auto maker last year sold ...]

[ROOT The] [luxury auto maker last year sold ...]

Shift-Reduce Parsing (Arc-Standard)

Shift

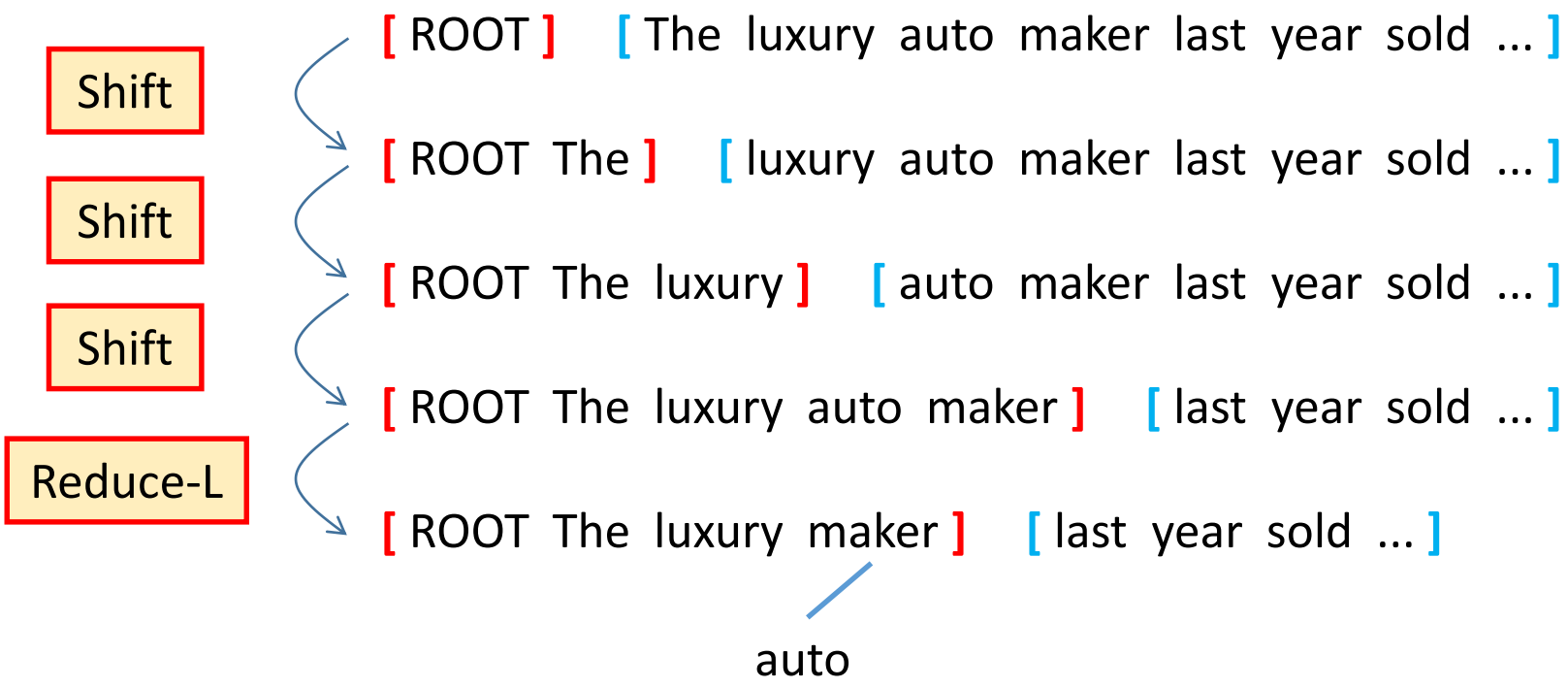
Shift

[ROOT] [The luxury auto maker last year sold ...]

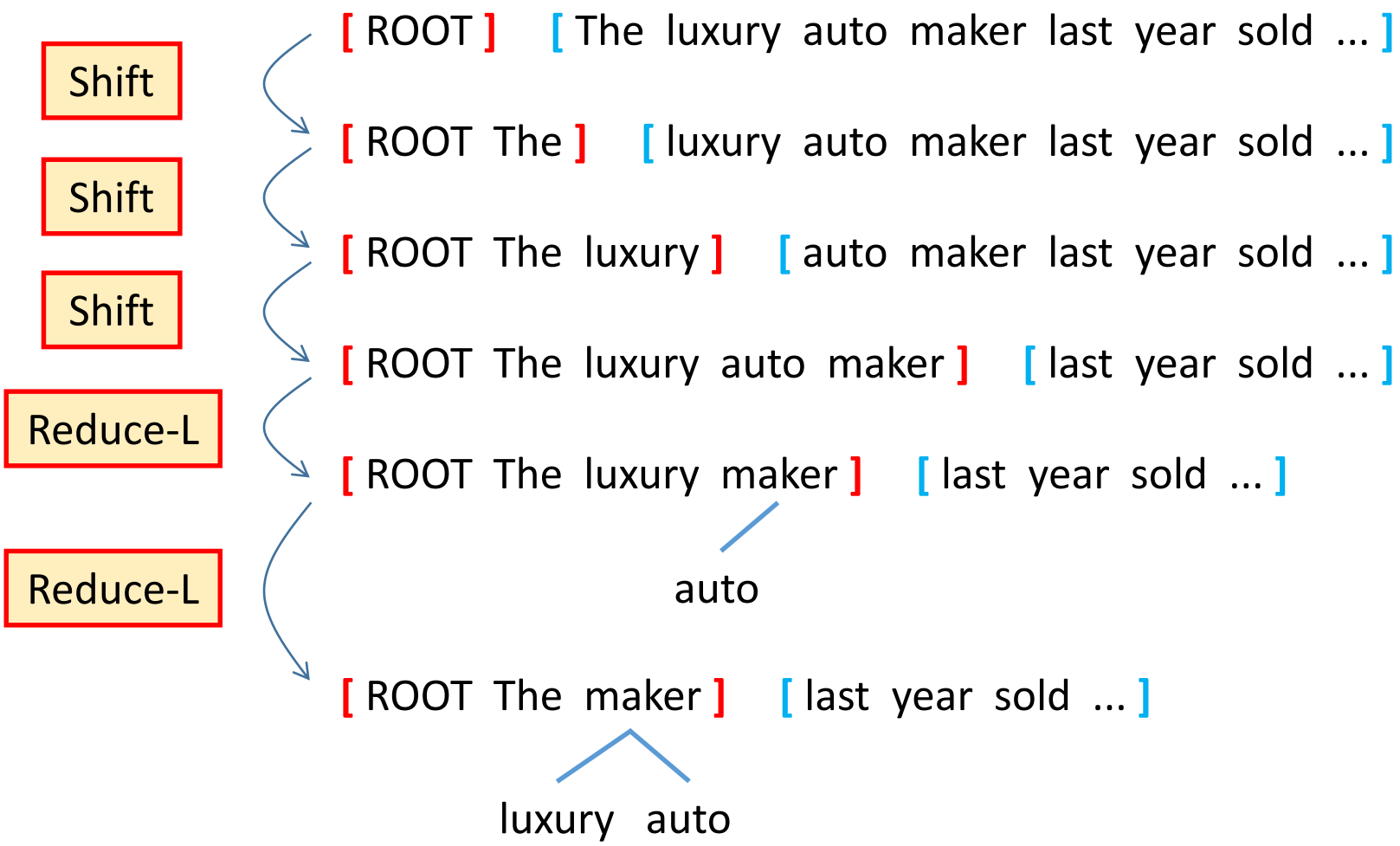
[ROOT The] [luxury auto maker last year sold ...]

[ROOT The luxury] [auto maker last year sold ...]

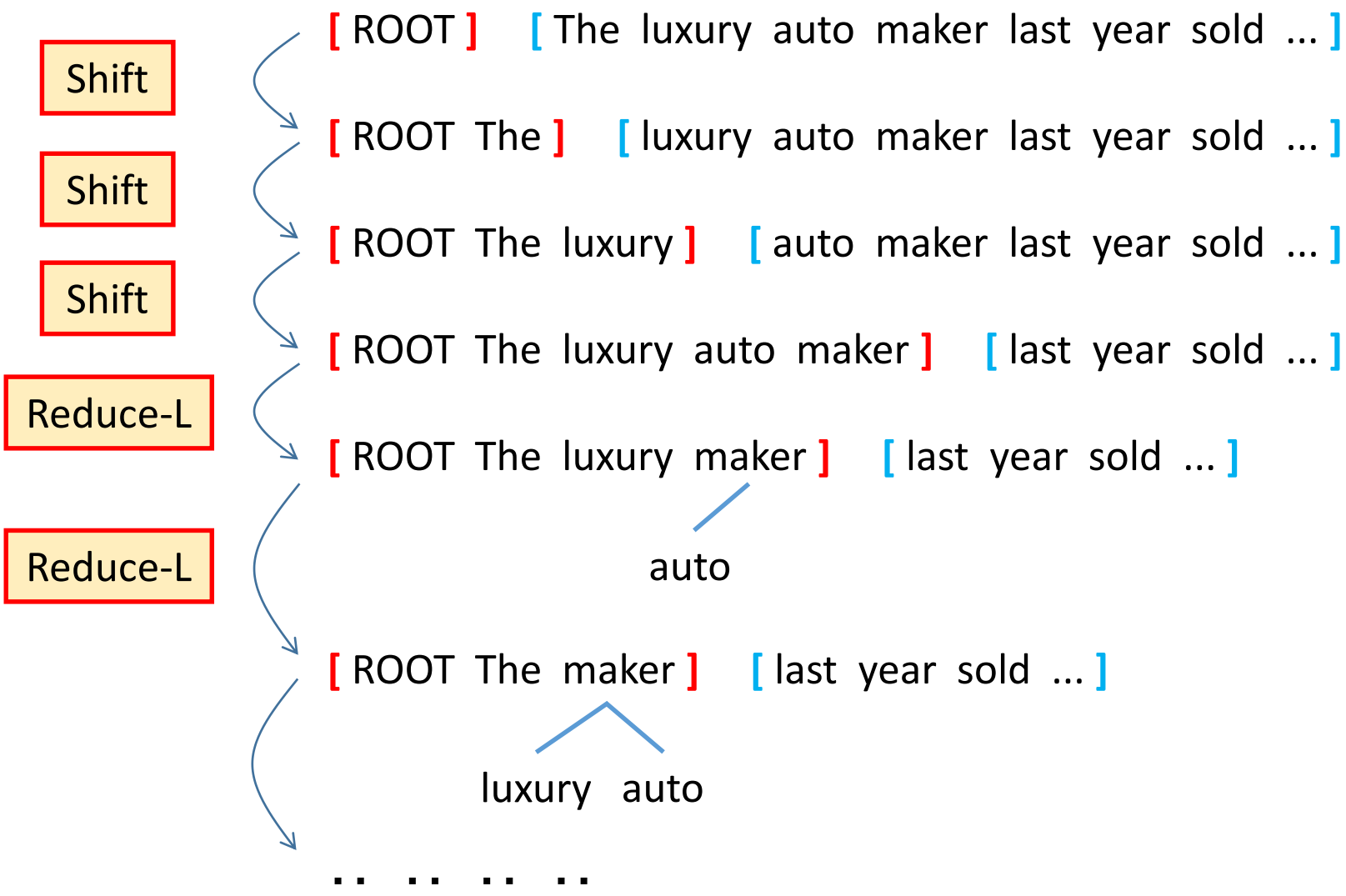
Shift-Reduce Parsing (Arc-Standard)



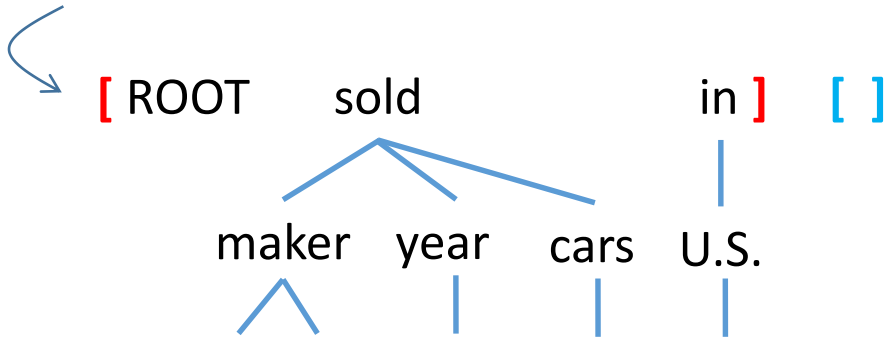
Shift-Reduce Parsing (Arc-Standard)



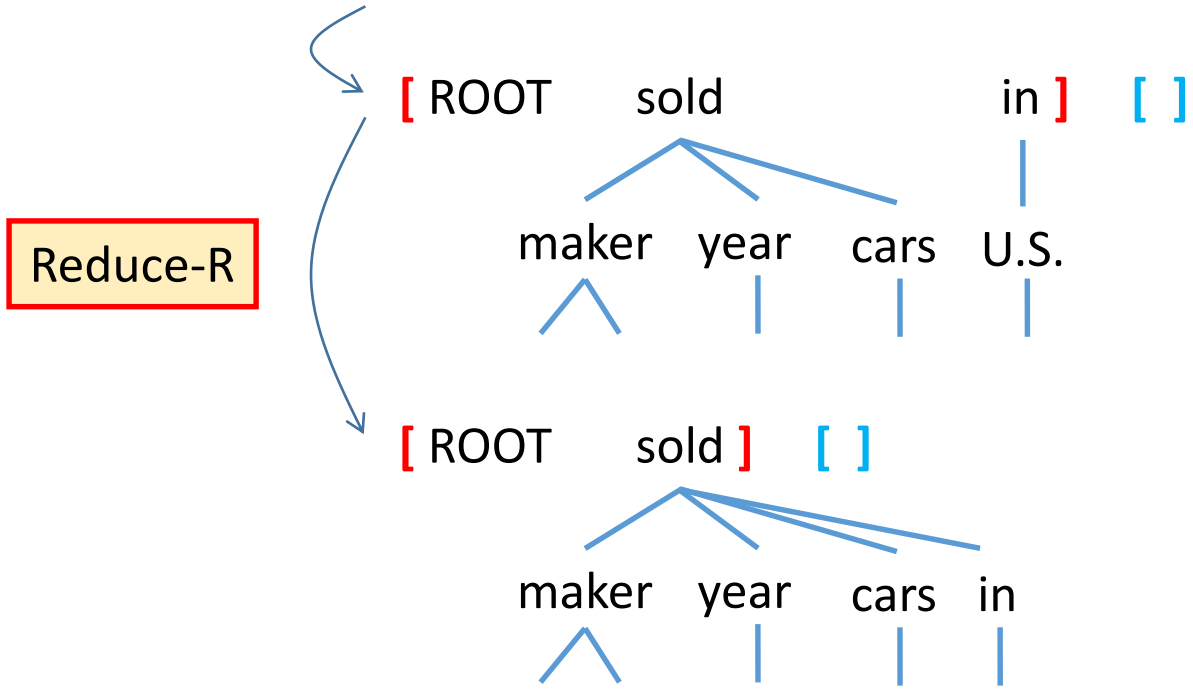
Shift-Reduce Parsing (Arc-Standard)



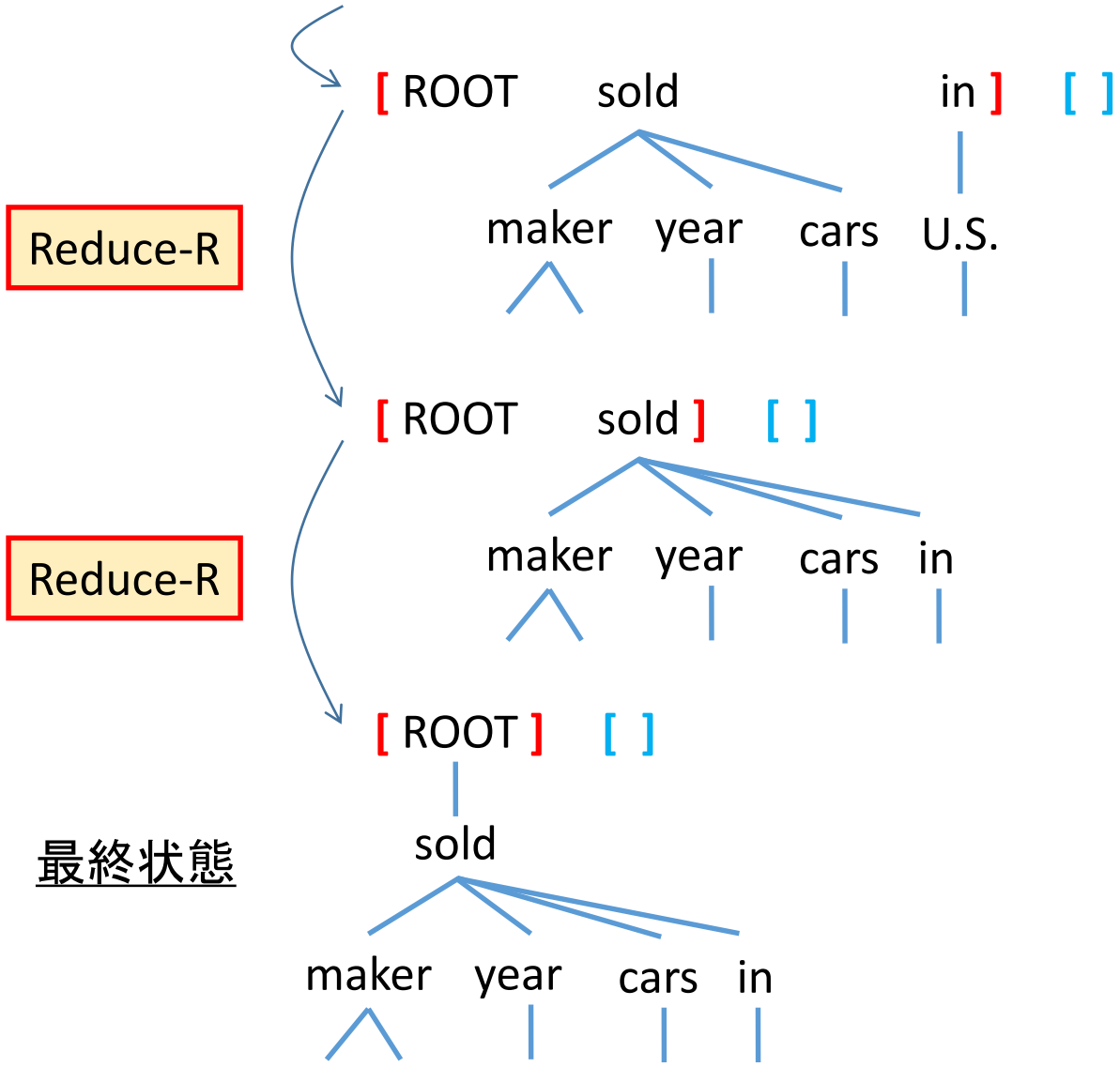
Shift-Reduce Parsing (Arc-Standard)



Shift-Reduce Parsing (Arc-Standard)



Shift-Reduce Parsing (Arc-Standard)



Shift-Reduce Parsing (Arc-Standard)

Reduce-L

[ROOT The luxury auto maker] [last year sold ...]

[ROOT The luxury maker] [last year sold ...]

auto

Shift-Reduce Parsing (Arc-Standard)

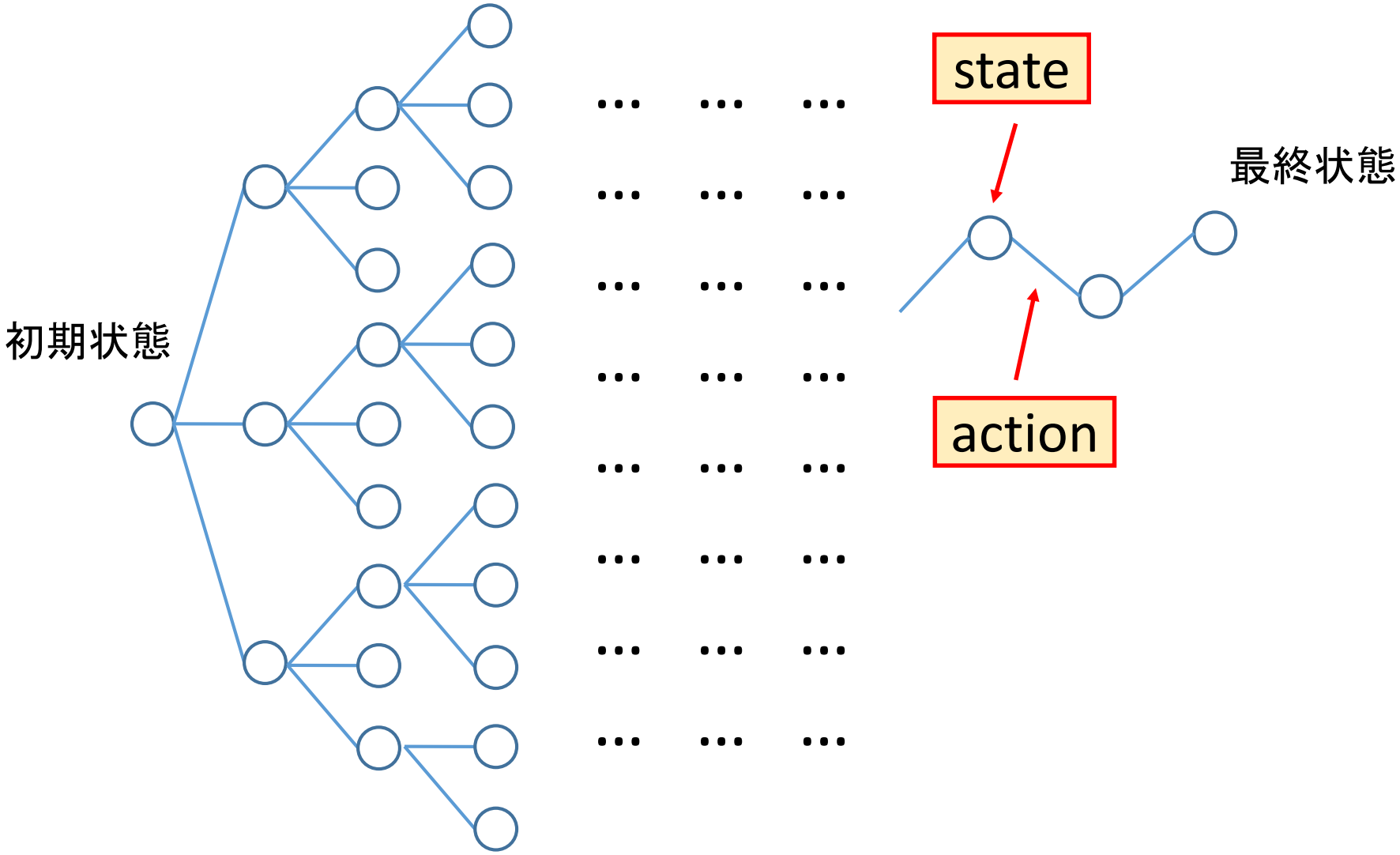
Reduce-R

[ROOT The luxury auto maker] [last year sold ...]

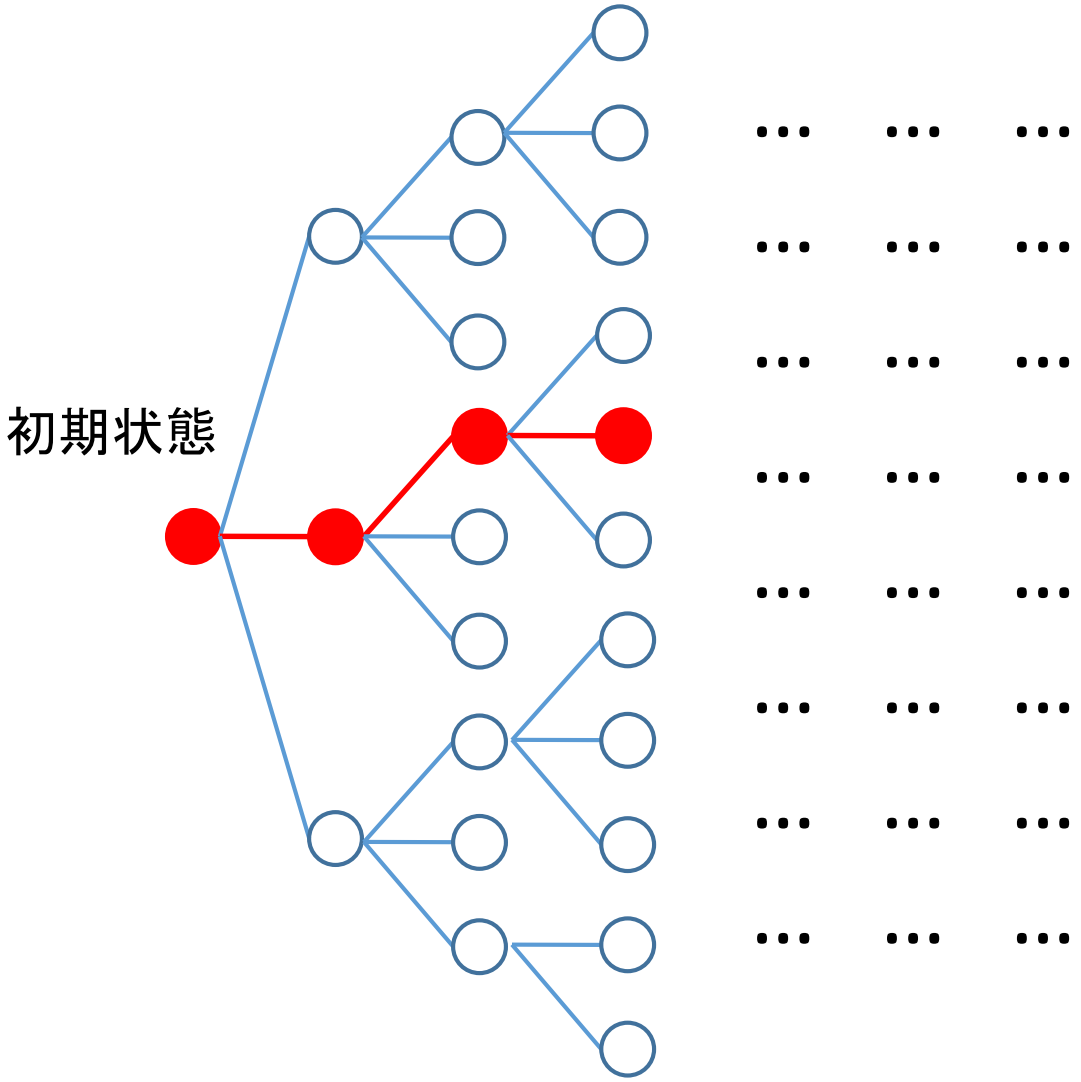
[ROOT The luxury auto] [last year sold ...]

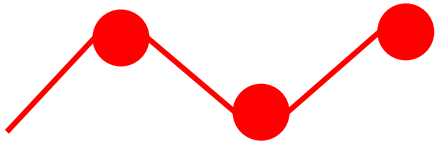
maker

状態遷移システム



状態遷移システム

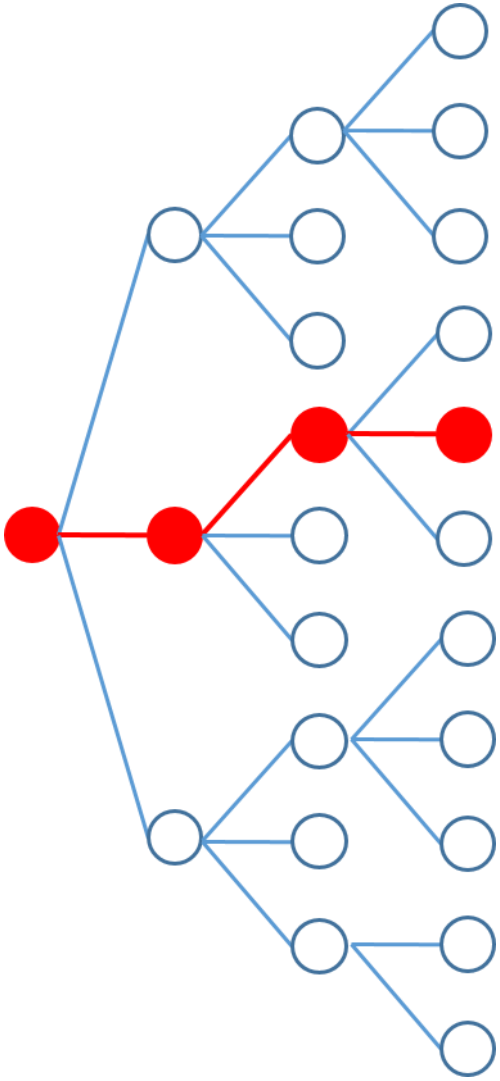




Shift-reduce解析：
状態遷移システムにおいて，
最適な経路を探索する

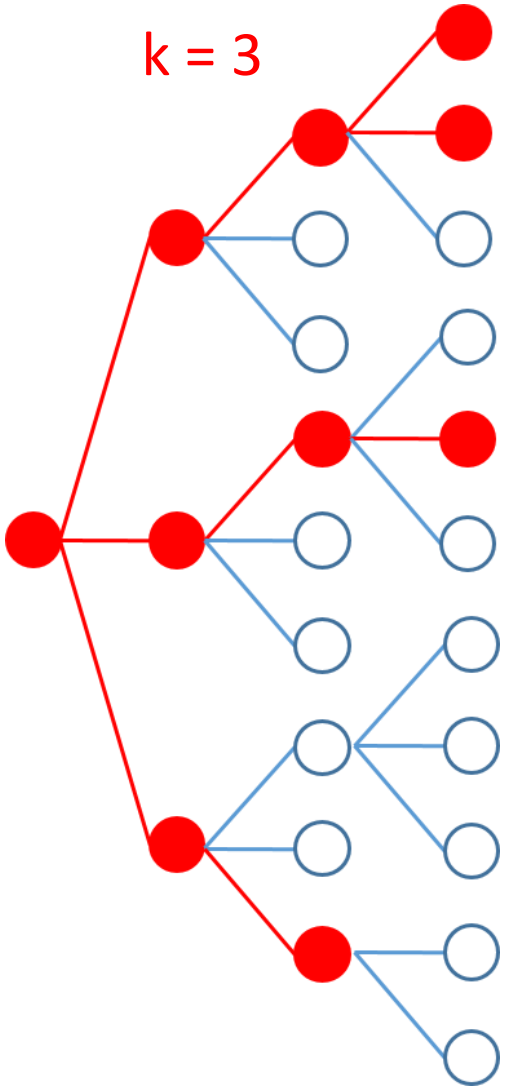
Greedy Search

スコアが最大の状態へ常に遷移する.



Beam Search

スコアの高い上位k個を保持しながら
状態遷移する.



Transition-based Dependency Parsing（遷移型依存構造解析）

- Shift-reduce 解析
- Easy-first 解析

Easy-First Parsing

Shift-reduce parsingの問題点

- Shift-reduce法は，左から右へ解析していく.
  これは多くの場合で上手くいくが，この順番が常に最適とは限らない.

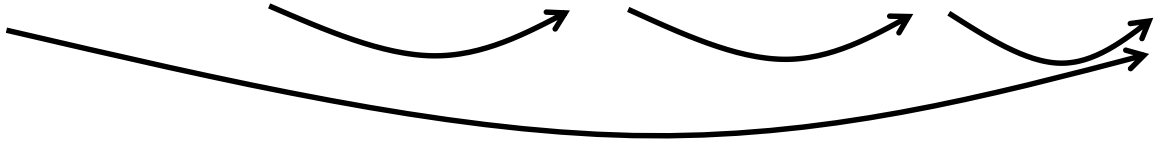
Easy-First Parsing

Garden-path sentence

男の子が三輪車に乗っていた女の子を見つめた

男の子が 三輪車に 乗っていた

男の子が 三輪車に 乗っていた 女の子を 見つめた



Easy-First Parsing

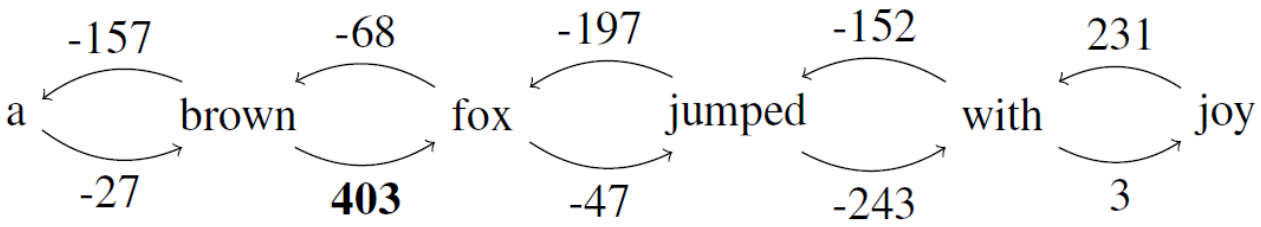
Shift-reduce parsingの問題点

- Shift-reduce法は，左から右へ解析していく.
  これは多くの場合で上手くいくが，この順番が常に最適とは限らない.

- Easy-first法では，簡単な順番（スコアの高い順番）に依存構造を構築していく.

Easy-First Parsing

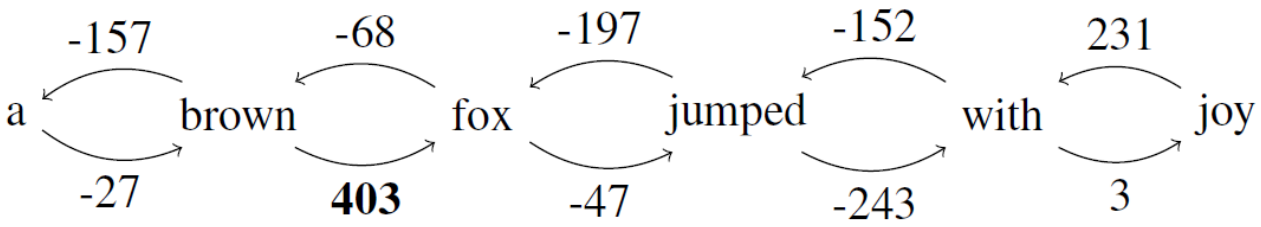
(1)



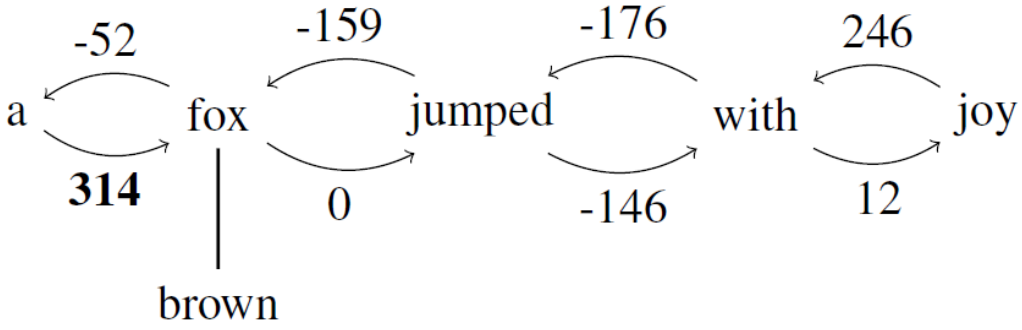
* Goldberg and Elhadad, "An Efficient Algorithm for Easy-First Non-Directional Dependency Parsing", NAACL 2010

Easy-First Parsing

(1)



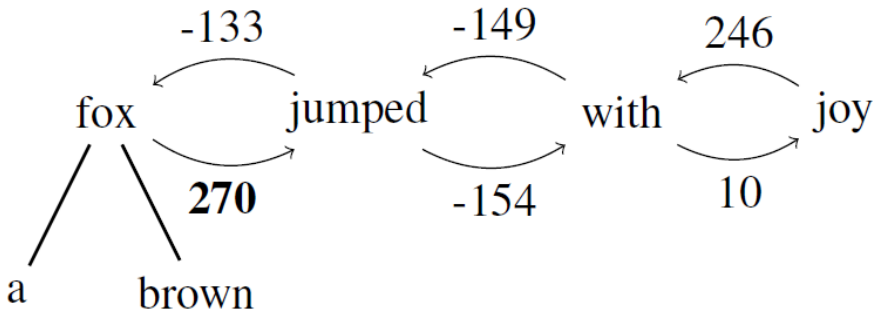
(2)



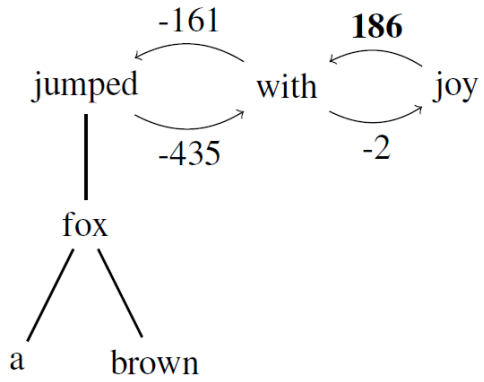
* Goldberg and Elhadad, "An Efficient Algorithm for Easy-First Non-Directional Dependency Parsing", NAACL 2010

Easy-First Parsing

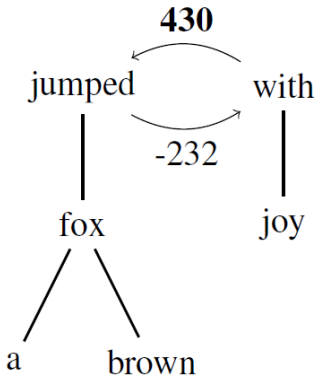
(3)



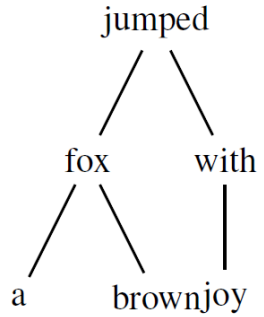
(4)



(5)



(6)



* Goldberg and Elhadad, "An Efficient Algorithm for Easy-First Non-Directional Dependency Parsing", NAACL 2010

アルゴリズムによる計算量の比較

Parser	Runtime
shift-reduce (greedy)	o (n)
shift-reduce (beam width: k)	o (k * n)
easy-first	o (n * log n)
Eisner (graph-based)	o (n^3)

※　Easy-first解析の“log n”項は，スコアの高い順番を決定するために優先度付きキューを使用することに起因する.

依存構造解析の性能比較 [Goldberg '10]

Parser	Accuracy	Complete
shift-reduce (greedy)	88.36	34.14
easy-first	89.70	37.50
Eisner (graph-based)	90.05	34.64

Accuracy: percentage of tokens which got assigned their correct parent.

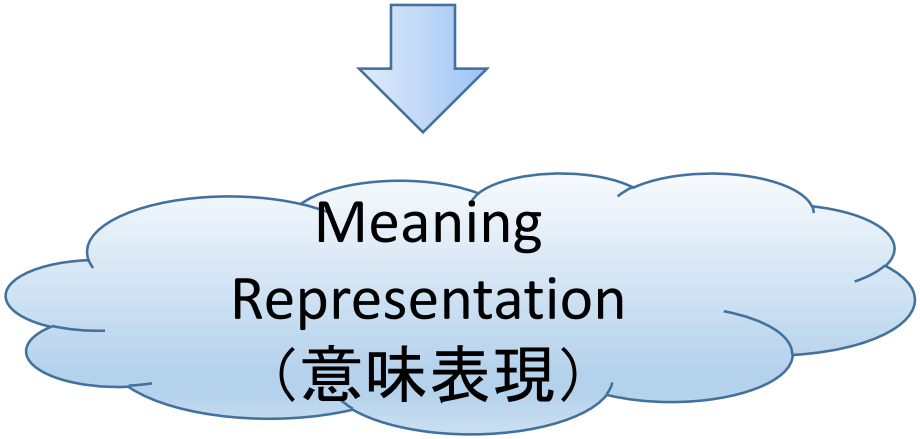
Complete: the percentage of sentences in which all tokens were assigned their correct parent.

Semantic Parsing
（意味解析）

Semantic Parsing（意味解析）

自然言語で書かれた文を意味表現へ変換すること

The luxury auto maker last year sold 1,214 cars in the U.S.

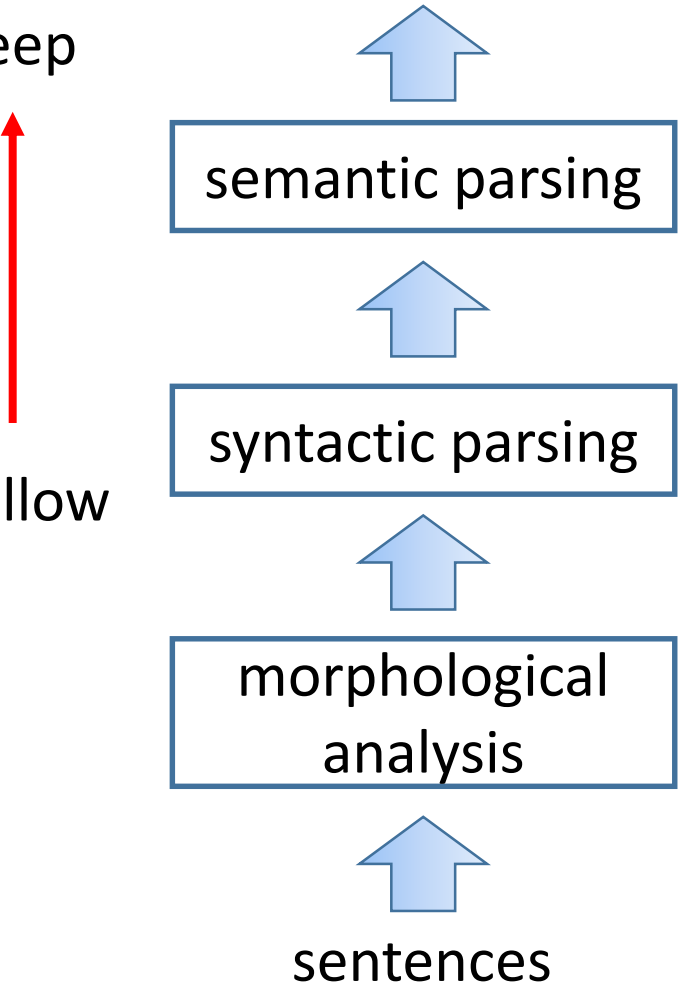


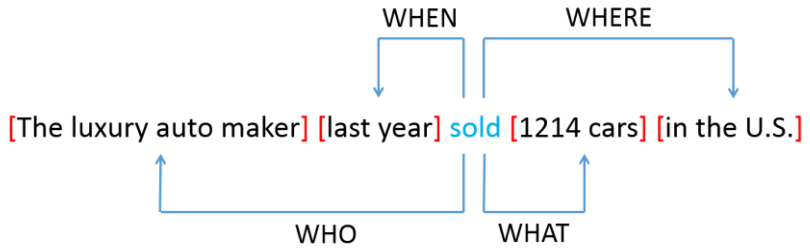
自然言語解析のパイプライン

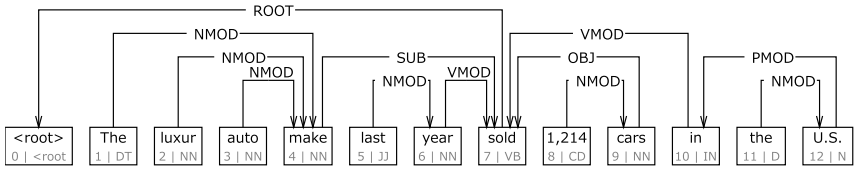
pragmatics, and more

deep

shallow



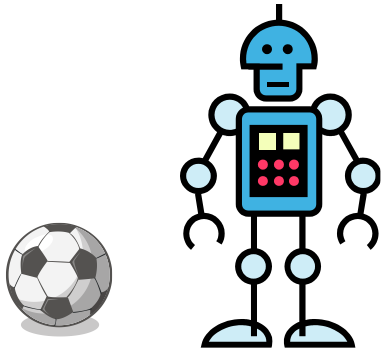




DT NN NN NN JJ NN VB CD NN IN DT NN
The luxury auto maker last year sold 1214 cars in the U.S.

RoboCup Coarch Language (CLang)

ロボットに言葉で指示を与えてゲームに勝つ.
[Wong & Mooney '06]



If our player 1 has the ball, position our player 1 in the penalty-area.

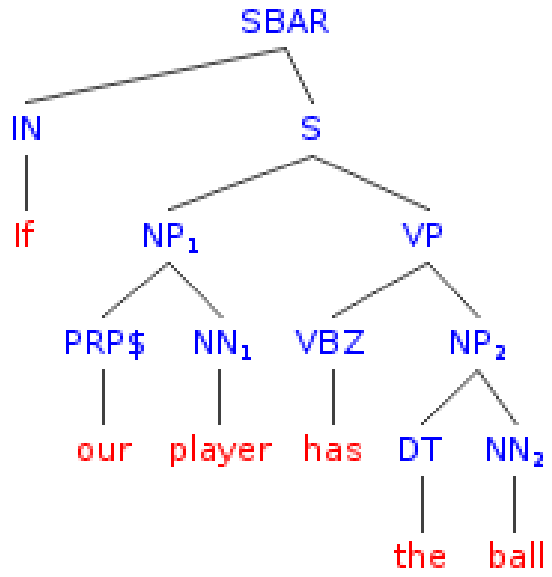
semantic parsing

((bowner (player our 1)) (do (player our 1) (pos (penalty-area))))

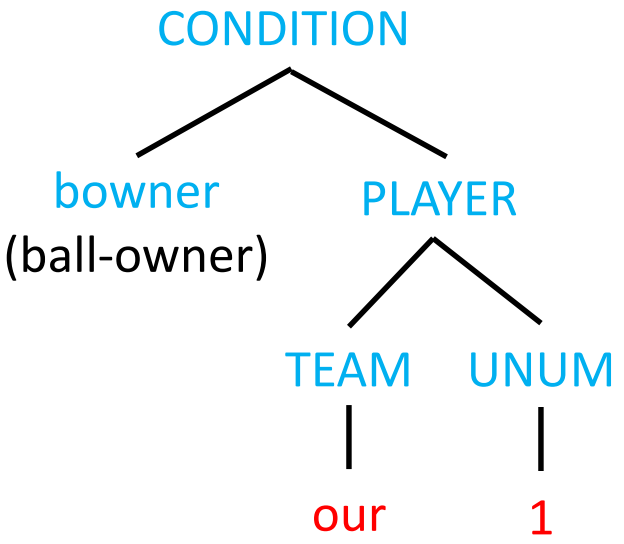
RoboCup Coarch Language (CLang)

If our player 1 has the ball, ...

統語構造



意味構造



Geoquery（地理情報の質問応答）

ラムダ計算

$$\lambda x.state\,(x)$$

argument（項）　predicate（述語）

Ex.

- What is the **largest** state?

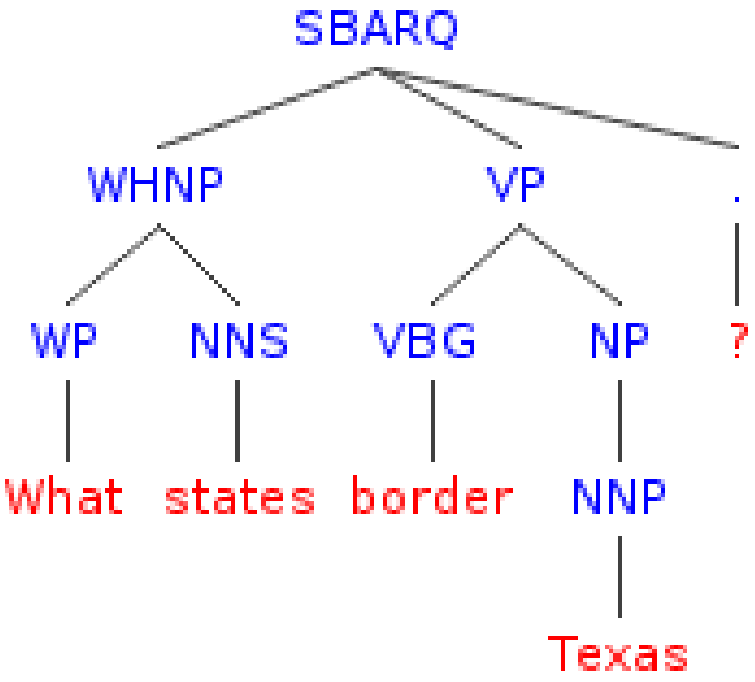
$$argmax\,(\lambda x.state\,(x)\,,\lambda x.size\,(x))$$

- What states border the largest state?

$$\lambda x.state\,(x) \wedge borders\,(x, argmax\,(\lambda y.state\,(y)\,,\lambda y.size\,(y)))$$

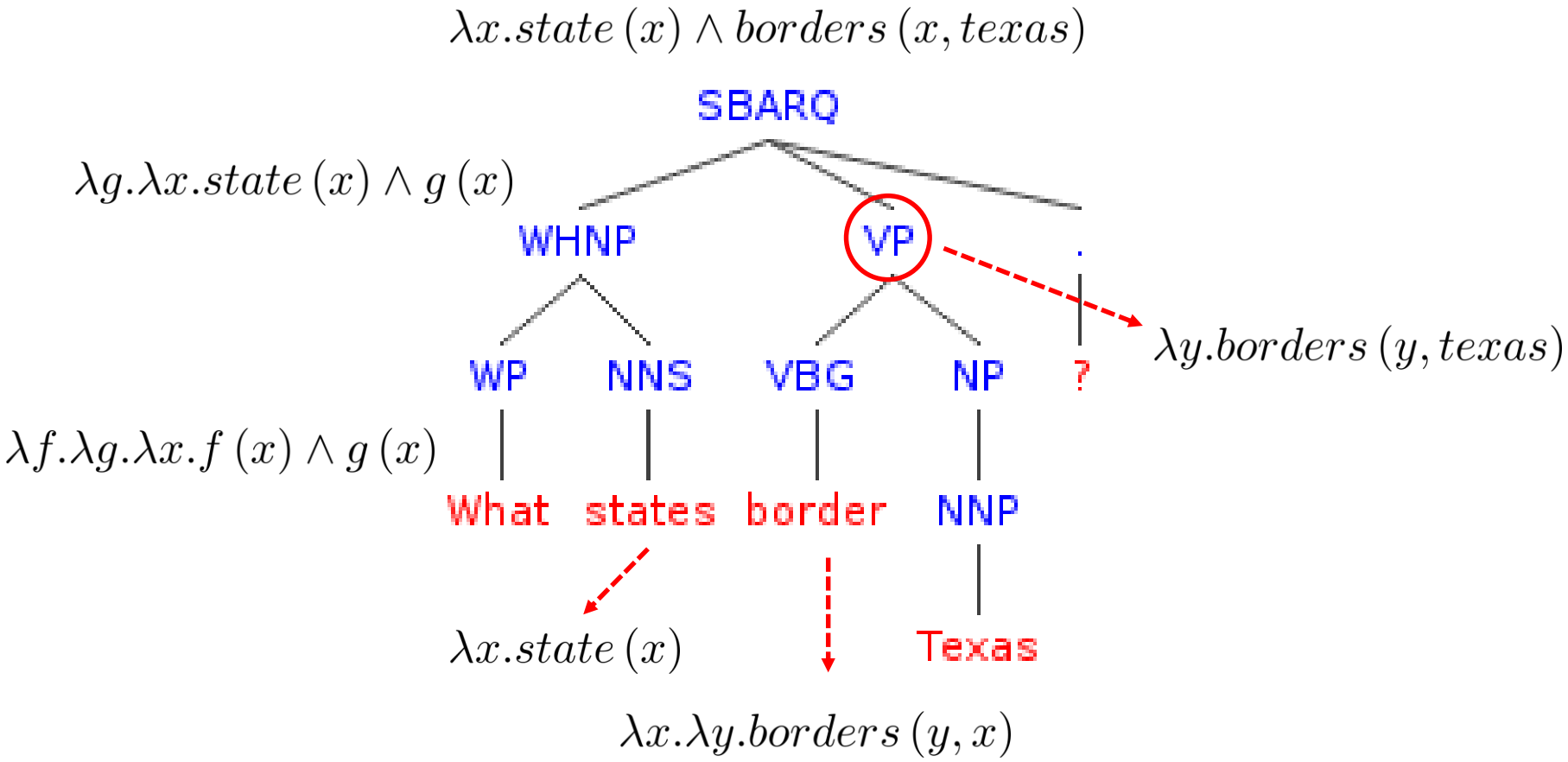
Geoquery（地理情報の質問応答）

統語構造



Geoquery（地理情報の質問応答）

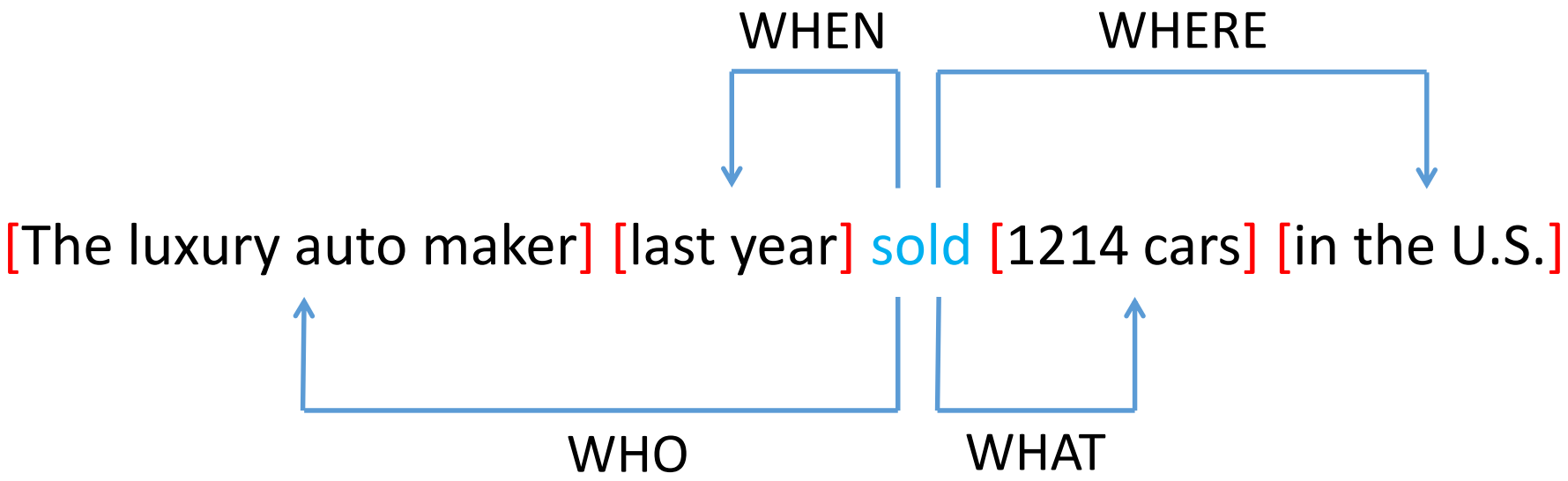
意味構造



Semantic Role Labeling
(意味役割付与，述語項構造解析)

Semantic Role Labeling (SRL)

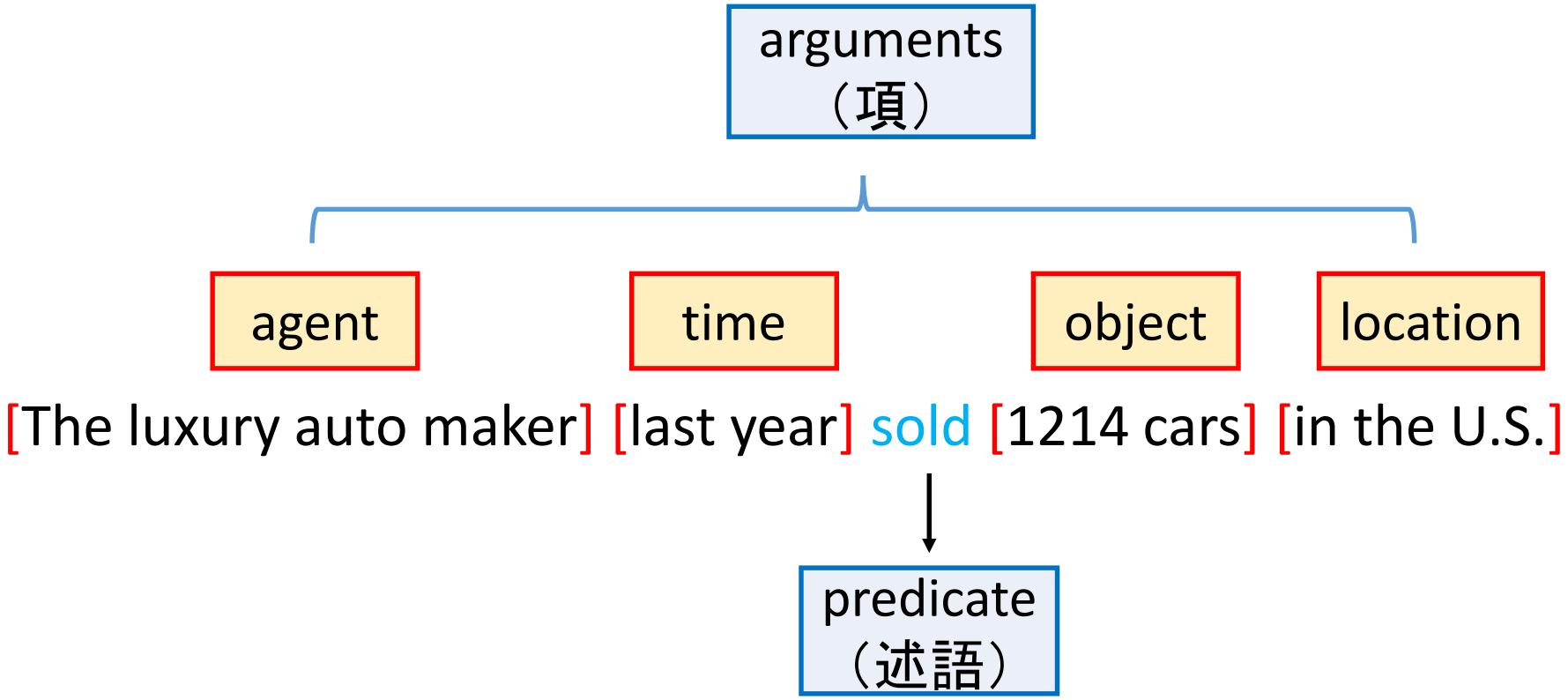
Who did what to whom, when, where and how?



Semantic Role Labeling (SRL)

Predicate-Argument Structure（述語項構造）

→ 文の意味表現の基本



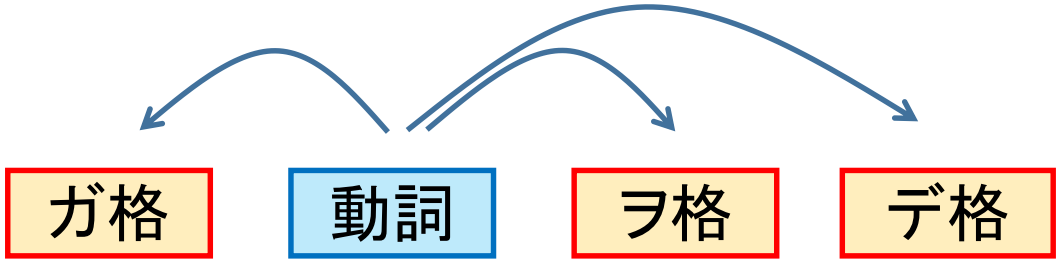
Semantic Role Labeling (SRL)

述語項構造

→ 文の意味表現の基本

The Case for Case [Fillmore '68]

格:
- ガ格
- ヲ格
- ニ格
- デ格



文の意味を，動詞を中心とした格構造によって理解する

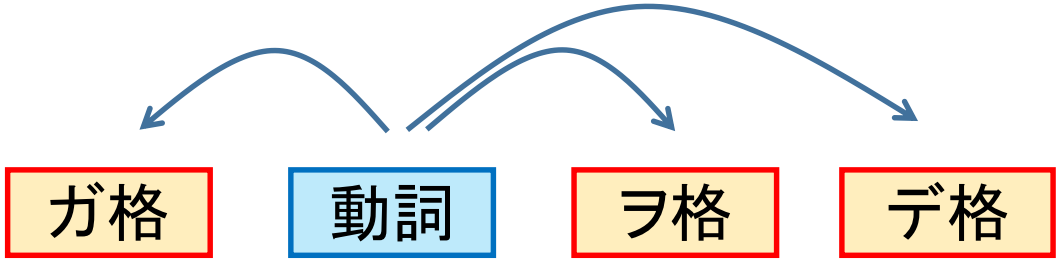
格文法

Case Grammar（格文法）

日本語の場合：

Cases:
- ガ格
- ヲ格
- ニ格
- デ格

Case frame
（格フレーム）



文の意味を，動詞を中心とした格構造によって整理する

Semantic Role Labeling (SRL)

誰が　いつ　どこで　何を　した？

Syntactic variations

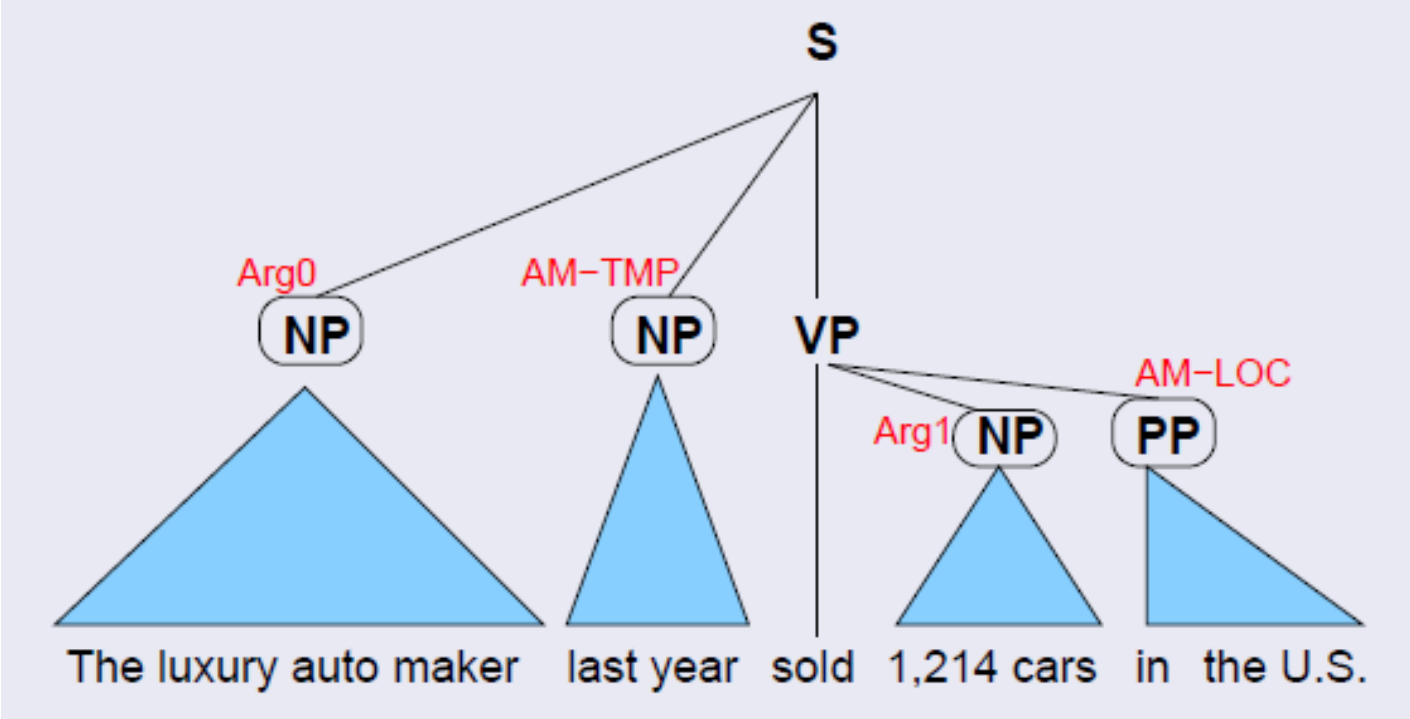
TEMP: Yesterday, HITTER: Kristina hit THING HIT: Scott INSTRUMENT: with a baseball

- Scott was hit by Kristina yesterday with a baseball
- Yesterday, Scott was hit with a baseball by Kristina
- With a baseball, Kristina hit Scott yesterday
- Yesterday Scott was hit by Kristina with a baseball
- Kristina hit Scott with a baseball yesterday

Example from (Yih & Toutanova, 2006)

PropBank (Proposition Bank)

Add a semantic layer to syntactic structure



[The luxury auto maker] [last year] sold [1214 cars] [in the U.S.]

Arg0	AM-TMP	sell.01	Arg1	AM-LOC

PropBank

Word senses for predicate verb
（述語動詞の語義）

sell.01:

- Arg0: seller
- Arg1: thing sold
- Arg2: buyer
- Arg3: price paid
- Arg4: benefactive

sell.02:

- Arg0: seller
- Arg1: person

PropBank

Arguments（項）

Arg0: agent, experiencer

Arg1: patient, theme

Arg2: benefactive, instrument, attribute

Arg3: benefactive, instrument, attribute

Arg4: end point

* Taken From Palmer et al. 2013, NAACL SRL Tutorial

PropBank

Adjuncts（付属語）

TMP: tomorrow, 3pm, etc.

LOC: in the living room, on the newspaper, etc.

DIR: to North, from America, etc.

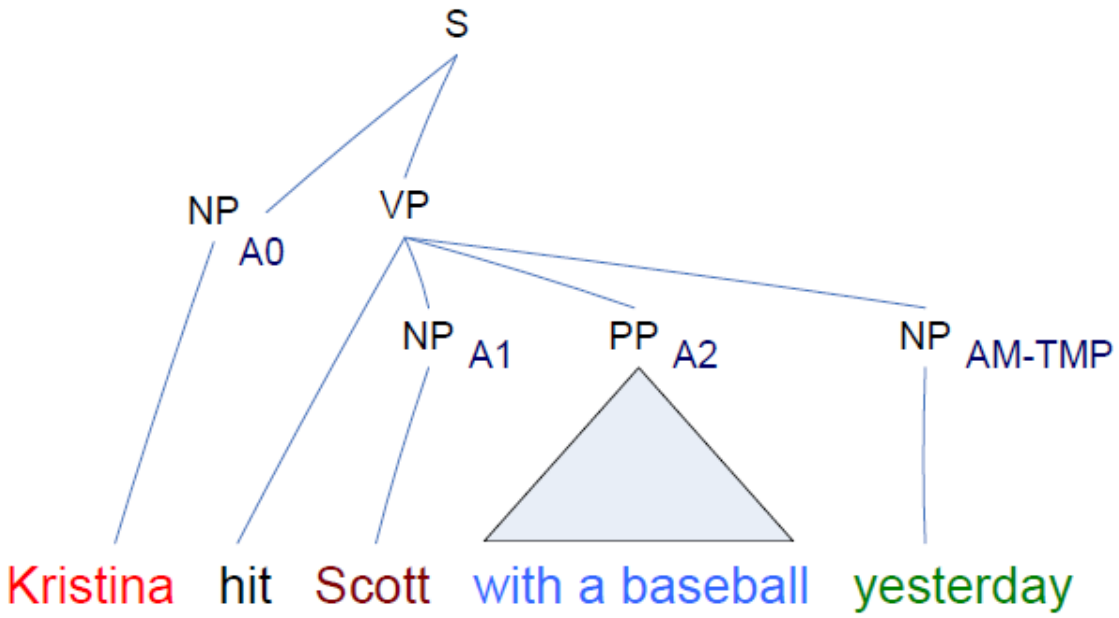
MNR: quickly, with much enthusiasm, etc.

PRP: because, so that, etc.

etc.

PropBank

Other examples

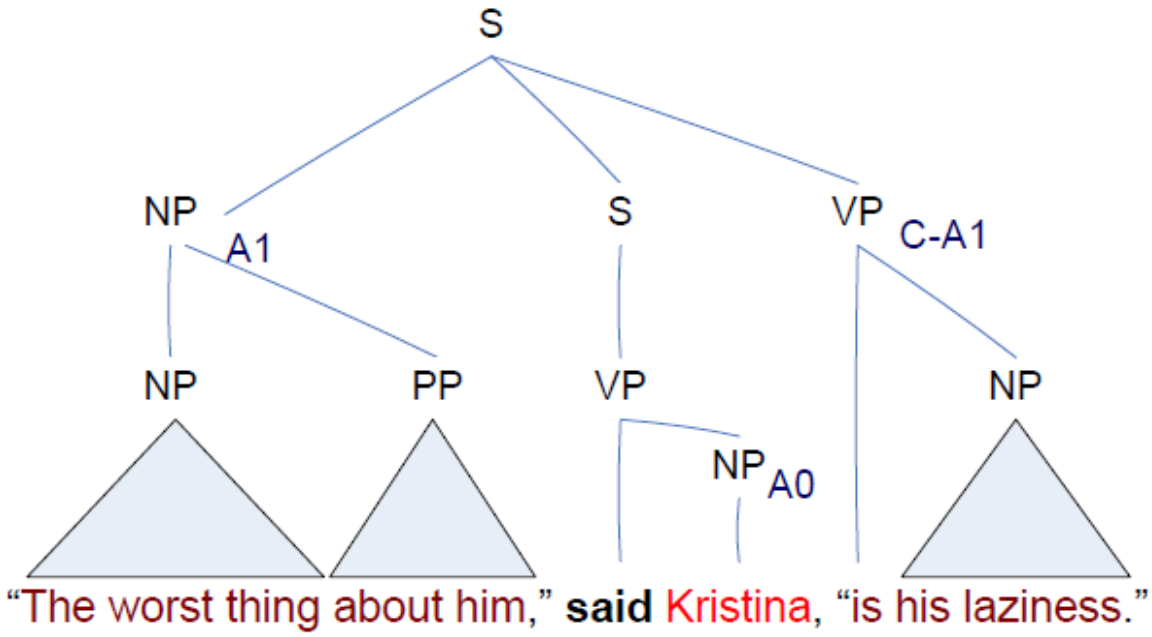


[$_{A0}$ *Kristina*] ***hit*** [$_{A1}$ *Scott*] [$_{A2}$ *with a baseball*] [$_{AM-TMP}$ *yesterday*].

* Taken From Yih et al. 2006, NAACL SRL Tutorial

PropBank

Other examples

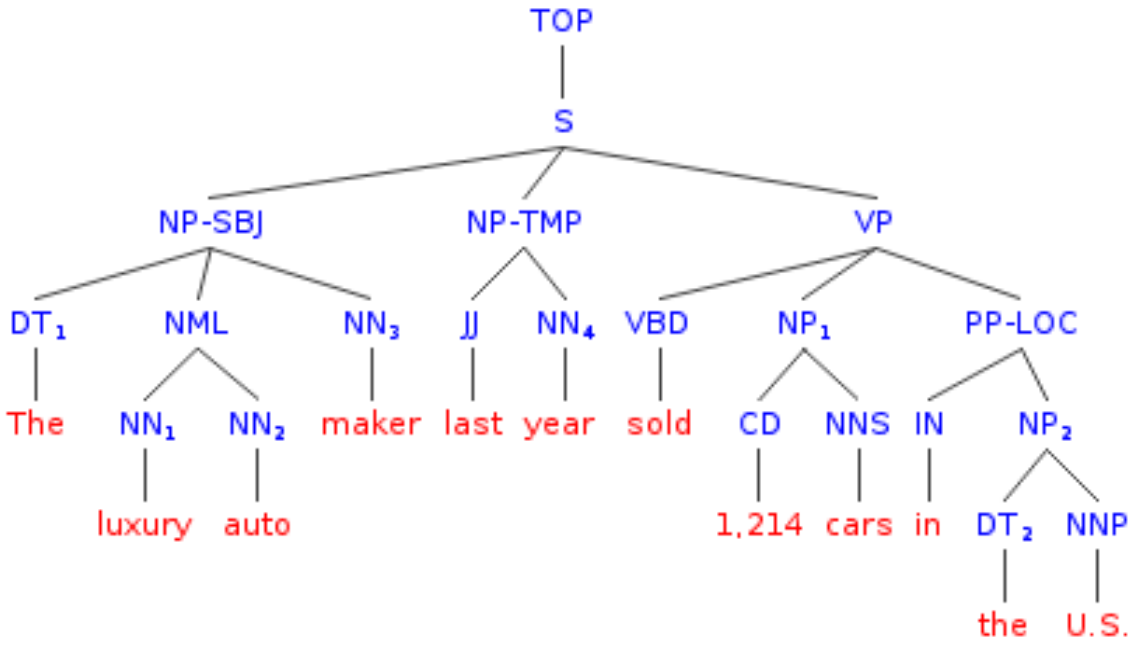


[$_{A1}$ *The worst thing about him*] ***said*** [$_{A0}$ *Kristina*] [$_{C\text{-}A1}$ *is his laziness*].

* Taken From Yih et al. 2006, NAACL SRL Tutorial

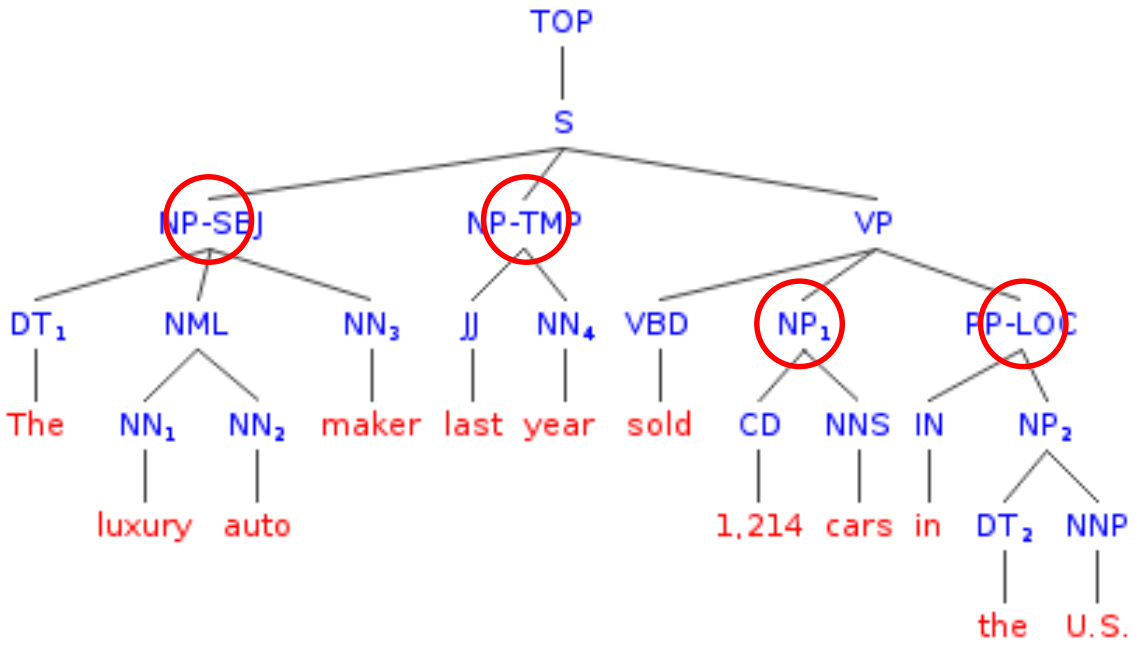
Algorithm for Semantic Role Labeling

1. 構文解析を行う(この場合，句構造)



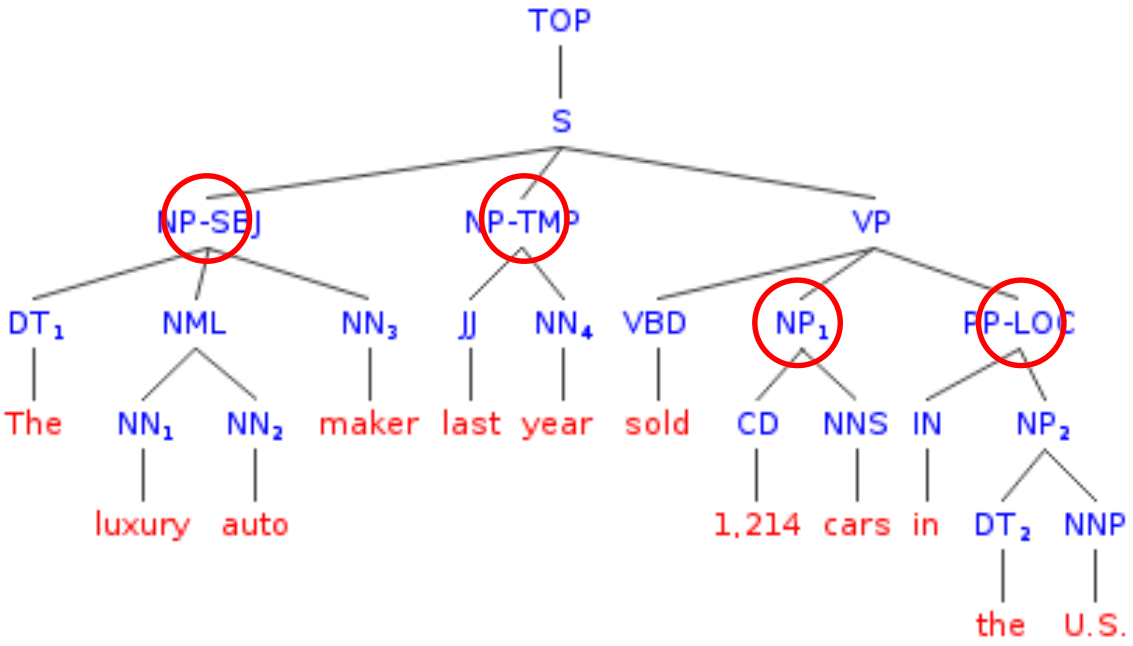
Algorithm for Semantic Role Labeling

2. 項の候補となる句を選択する



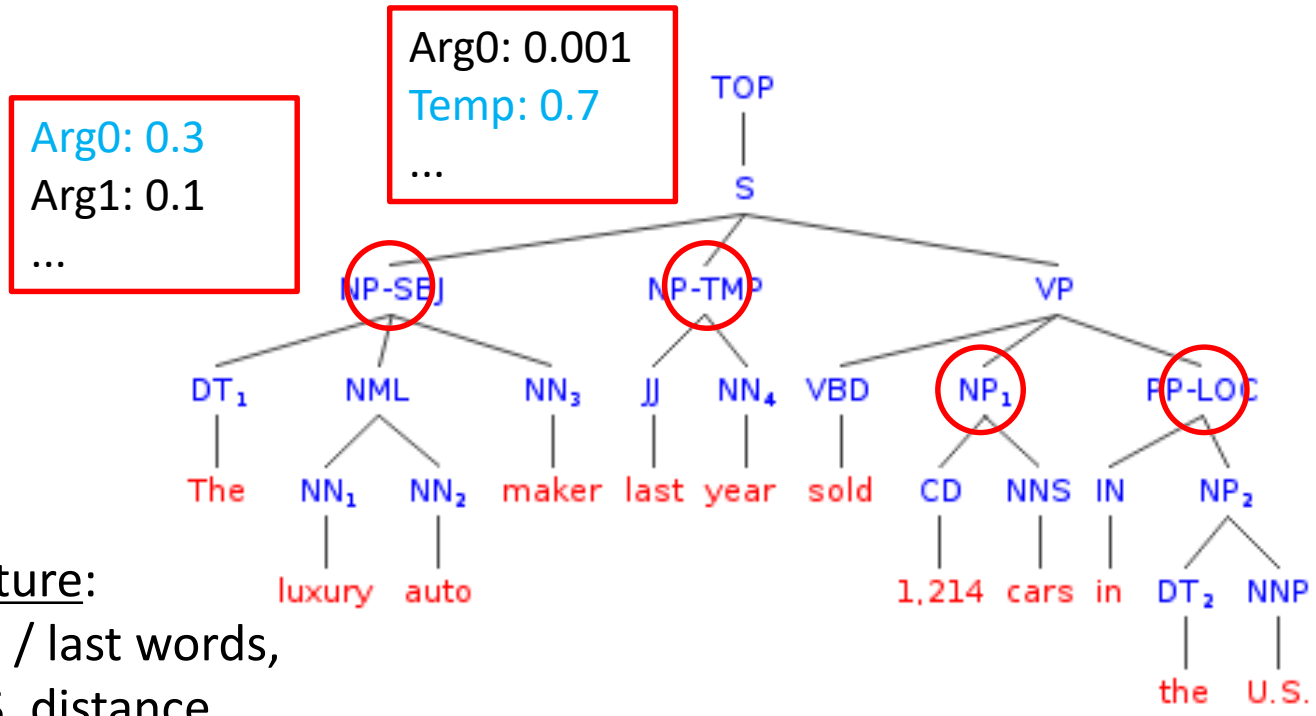
Algorithm for Semantic Role Labeling

3. 項の候補に対してスコアを計算して，項を決定する



Algorithm for Semantic Role Labeling

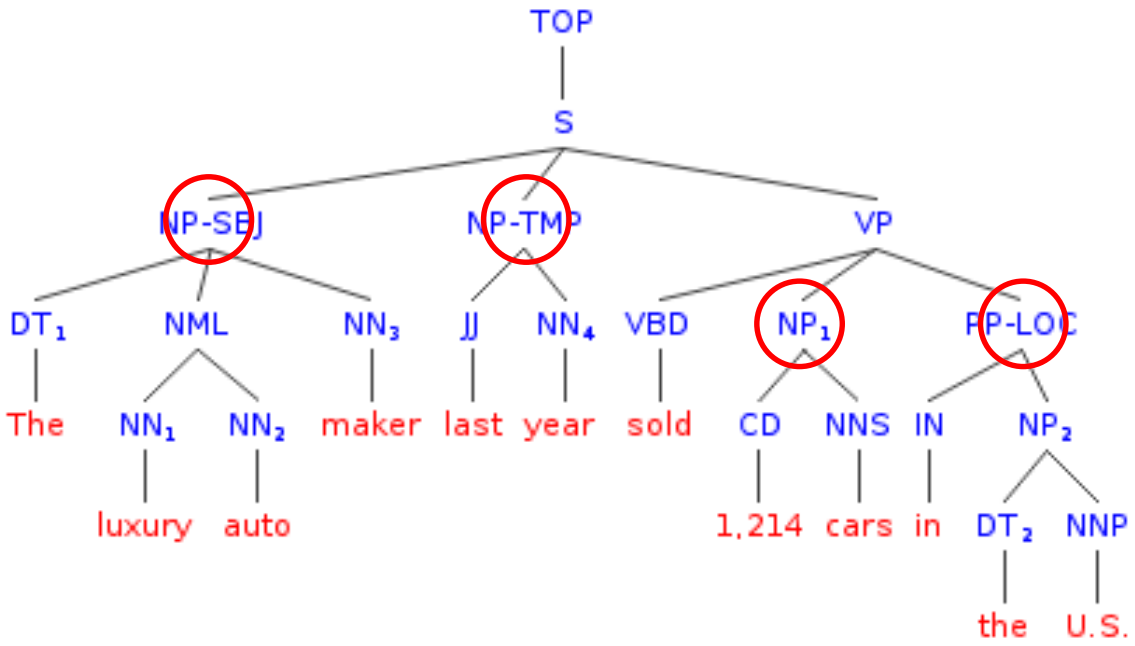
3. 項の候補に対してスコアを計算して，項を決定する



Feature:
first / last words,
POS, distance
from predicate

Algorithm for Semantic Role Labeling

3. 項の候補に対してスコアを計算して，項を決定する



[The luxury auto maker] [last year] sold [1214 cars] [in the U.S.]

Arg0 | AM-TMP | sell.01 | Arg1 | AM-LOC

Semantic Role Labeling (SRL)

誰が　いつ　どこで　何を　した？

IBM appointed John.

John was appointed by IBM.

IBM's appointment of John.

The appointment of John by IBM.

John is the current IBM appointee.

Semantic Role Labeling (SRL)

Argument Structures for Nouns
（名詞が述語になる場合がある）

predicate

[Her] gift of [a book] [to John]

Arg0 Arg1 Arg2

predicate

[His] promise [to make the trains run on time]

Arg0 Arg2-PRD