Dependency Parsing
lecture 3

Jakub Waszczuk, Kilian Evang

Heinrich Heine Universität

Summer semester 2021

Grammar based dependency parsing [Kubler et al., 2009]

Features

- Grammar-based methods rely on an explicitly defined formal grammar
- More restrictive parsing model (in comparison with purely data-driven approaches)

Main approaches

- Constraint dependency parsing
- Context free dependency parsing

Grammar based dependency parsing [Kubler et al., 2009]

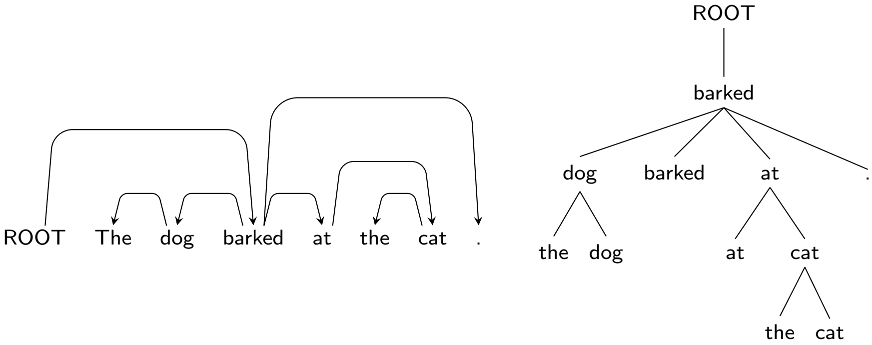
Features

- Grammar-based methods rely on an explicitly defined formal grammar
- More restrictive parsing model (in comparison with purely data-driven approaches)

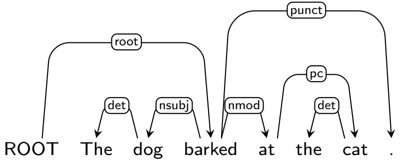
Main approaches

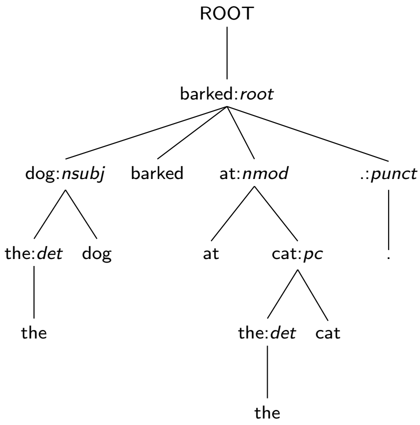
- Constraint dependency parsing
- Context free dependency parsing ← **today's focus**

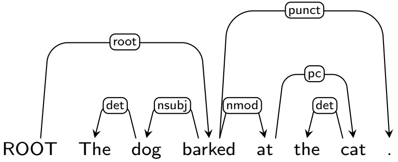
Dependency tree as a constituency tree

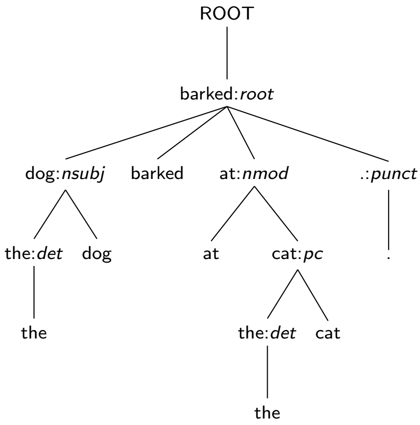


Dependency tree as a constituency tree









We focus on unlabeled trees for simplicity

Context free grammar

Definition: Context Free Grammar

A Context Free Grammar is a 4-tuple (N,Σ,Π,S) where:

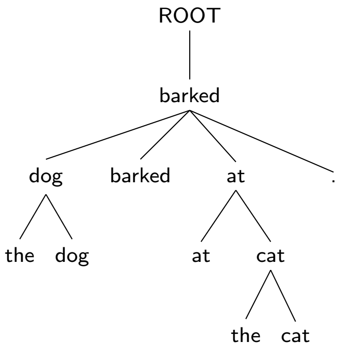
- N is a set of non-terminal symbols
- Σ is a set of terminal symbols
- Π is a set of production rules of the form $X \rightarrow \alpha$, where $X \in N$ and α is a string of terminal and non terminal symbols
- $S \in N$ is the start symbol

Example

- $\Sigma = \{\text{the}, \text{dog}, \text{barked}, \text{at}, \text{the}, \text{cat}\}$, $N = \Sigma \cup \{\text{ROOT}\}$, $S = \text{ROOT}$
- $\Pi = \{\text{ROOT} \rightarrow \text{barked}, \text{barked} \rightarrow \text{dog barked at .}, \text{dog} \rightarrow \text{the dog}, \text{at} \rightarrow \text{at cat}, \text{cat} \rightarrow \text{the cat}\}$

Context free grammar

- $\Sigma = \{\text{the}, \text{dog}, \text{barked}, \text{at}, \text{the}, \text{cat}, .\}$
- $N = \Sigma \cup \{\text{ROOT}\}$
- $S = \text{ROOT}$
- Π :
  - ROOT $\rightarrow$ barked
  - barked $\rightarrow$ dog barked at .
  - dog $\rightarrow$ the dog
  - at $\rightarrow$ at cat
  - cat $\rightarrow$ the cat



Context free dependency parsing

Advantages

- Possibility to reuse the well-studied CFG parsing algorithm: CYK [Younger, 1967], Earley's algorithm [Earley, 1970], etc.

Disadvantages

- Considerably higher number of nonterminals in comparison to traditional CFGs

Bilexical context free grammar

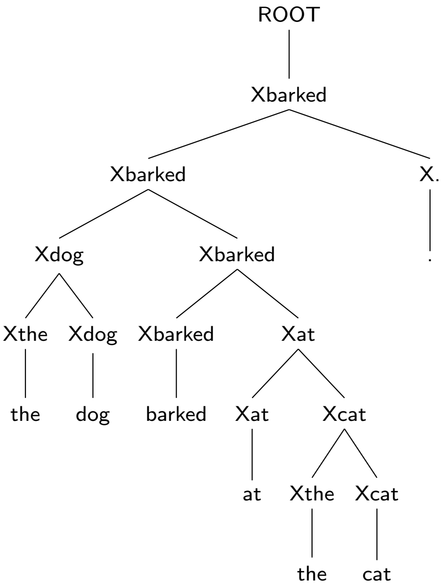
Definition

A Bilexical Context Free Grammar is a context free grammar in which the set of production rules Π consists of a set of

- root dependencies $ROOT \rightarrow H$
- left dependencies $H \rightarrow NH\ H$
- right dependencies $H \rightarrow H\ NH$
- and terminal dependencies $H \rightarrow h$

Bilexical grammar: Example

- $\Sigma =$ $\{\text{the}, \text{dog}, \text{barked}, \text{at}, \text{the}, \text{cat}, .\}$
- $N = \{\text{Xt} : \text{t} \in \Sigma\} \cup \{\text{ROOT}\}$
- $S = \text{ROOT}$
- Π :
  - ROOT → Xbarked
  - Xbarked → Xdog Xbarked
  - Xbarked → Xbarked Xat
  - Xbarked → Xbarked X.
  - Xdog → Xthe Xdog
  - Xat → Xat Xcat
  - Xcat → Xthe Xcat
  - Xthe → the
  - Xdog → dog
  - Xbarked → barked
  - Xat → at
  - Xcat → cat



CYK

Input

- Bilexical CFG $G = (N, \Sigma, \Pi, S)$
- Input sentence $w = w_1 w_2 \dots w_n \in \Sigma^*$ of length n

Item

- Each item has the form $[A, i, j]$ where $A \in N$ and $1 \leq i \leq j \leq n$
- Item $[A, i, j]$ states that $A \Rightarrow^* w_i \dots w_j$

Rules

- Axiom:
$$\frac{}{[A, i, i]} \quad A \rightarrow w_i \in \Pi$$
- Combine:
$$\frac{[B, i, j] \quad [C, j+1, k]}{[A, i, k]} \quad A \rightarrow BC \in \Pi$$
- Root:
$$\frac{[A, 1, n]}{[\mathrm{ROOT}, 1, n]} \quad \mathrm{ROOT} \rightarrow A \in \Pi$$

CYK

Input

- Bilexical CFG $G = (N, \Sigma, \Pi, S)$
- Input sentence $w = w_1 w_2 \dots w_n \in \Sigma^*$ of length n

Item

- Each item has the form $[A, i, j]$ where $A \in N$ and $1 \leq i \leq j \leq n$
- Item $[A, i, j]$ states that $A \Rightarrow^* w_i \dots w_j$

Rules

- Axiom:

$$\frac{}{[A, i, i]} \quad A \rightarrow w_i \in \Pi$$

- Combine:

$$\frac{[B, i, j] \quad [C, j+1, k]}{[A, i, k]} \quad A \rightarrow BC \in \Pi$$

- Root: **(non-standard)**

$$\frac{[A, 1, n]}{[\mathrm{ROOT}, 1, n]} \quad \mathrm{ROOT} \rightarrow A \in \Pi$$

CYK: Example

Rules

$$\frac{}{[A, i, i]}\ A \rightarrow w_i \in \Pi \quad ; \quad \frac{[B, i, j]\ [C, j+1, k]}{[A, i, k]}\ A \rightarrow BC \in \Pi$$

Grammar

$\Sigma = \{\text{the}, \text{dog}, \text{barked}\}$ and Π :

- ROOT $\rightarrow$ Xbarked
- Xbarked $\rightarrow$ Xdog Xbarked | barked
- Xdog $\rightarrow$ Xthe Xdog | dog
- Xthe $\rightarrow$ the

Chart

the dog barked

CYK: Example

Rules

$$\frac{}{[A, i, i]}\ A \to w_i \in \Pi \quad ; \quad \frac{[B, i, j]\ [C, j+1, k]}{[A, i, k]}\ A \to BC \in \Pi$$

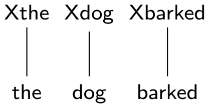
Grammar

$\Sigma = \{\text{the}, \text{dog}, \text{barked}\}$ and Π :

- ROOT $\to$ Xbarked
- Xbarked $\to$ Xdog Xbarked | barked
- Xdog $\to$ Xthe Xdog | dog
- Xthe $\to$ the

Chart

- $\to$ [Xthe, 1, 1], [Xdog, 2, 2], [Xbarked, 3, 3] (axiom × 3)



CYK: Example

Rules

$$\frac{}{[A, i, i]} A \rightarrow w_i \in \Pi \quad ; \quad \frac{[B, i, j]\ [C, j+1, k]}{[A, i, k]} A \rightarrow BC \in \Pi$$

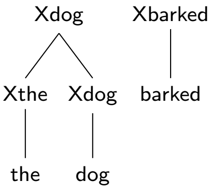
Grammar

$\Sigma = \{\text{the}, \text{dog}, \text{barked}\}$ and Π :

- ROOT $\rightarrow$ Xbarked
- Xbarked $\rightarrow$ Xdog Xbarked | barked
- Xdog $\rightarrow$ Xthe Xdog | dog
- Xthe $\rightarrow$ the

Chart

- $\rightarrow$ [Xthe, 1, 1], [Xdog, 2, 2], [Xbarked, 3, 3] (axiom × 3)
- [Xthe, 1, 1], [Xdog, 2, 2] $\rightarrow$ [Xdog, 1, 2] (combine)



CYK: Example

Rules

$$\frac{}{[A, i, i]} A \rightarrow w_i \in \Pi \quad ; \quad \frac{[B, i, j]\ [C, j+1, k]}{[A, i, k]} A \rightarrow BC \in \Pi$$

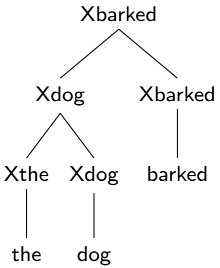
Grammar

$\Sigma = \{$the, dog, barked$\}$ and Π :

- ROOT $\rightarrow$ Xbarked
- Xbarked $\rightarrow$ Xdog Xbarked | barked
- Xdog $\rightarrow$ Xthe Xdog | dog
- Xthe $\rightarrow$ the

Chart

- $\rightarrow$ [Xthe, 1, 1], [Xdog, 2, 2], [Xbarked, 3, 3] (axiom × 3)
- [Xthe, 1, 1], [Xdog, 2, 2] $\rightarrow$ [Xdog, 1, 2] (combine)
- [Xdog, 1, 2], [Xbarked, 3, 3] $\rightarrow$ [Xbarked, 1, 3] (combine)



CYK: Example

Rules

$$\frac{}{[A, i, i]} \; A \to w_i \in \Pi \quad ; \quad \frac{[B, i, j] \; [C, j+1, k]}{[A, i, k]} \; A \to BC \in \Pi$$

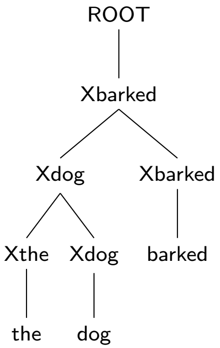
Grammar

$\Sigma = \{\text{the}, \text{dog}, \text{barked}\}$ and Π :

- ROOT $\to$ Xbarked
- Xbarked $\to$ Xdog Xbarked | barked
- Xdog $\to$ Xthe Xdog | dog
- Xthe $\to$ the

Chart

- $\to$ [Xthe, 1, 1], [Xdog, 2, 2], [Xbarked, 3, 3] (axiom × 3)
- [Xthe, 1, 1], [Xdog, 2, 2] $\to$ [Xdog, 1, 2] (combine)
- [Xdog, 1, 2], [Xbarked, 3, 3] $\to$ [Xbarked, 1, 3] (combine)
- [Xbarked, 1, 3] $\to$ [ROOT, 1, 3] (root)



Bilexical grammar: Issues

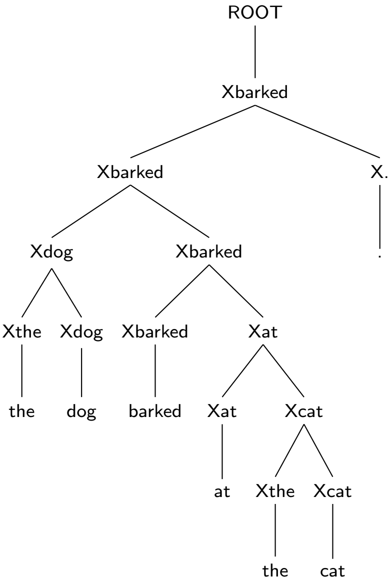
Parsing complexity

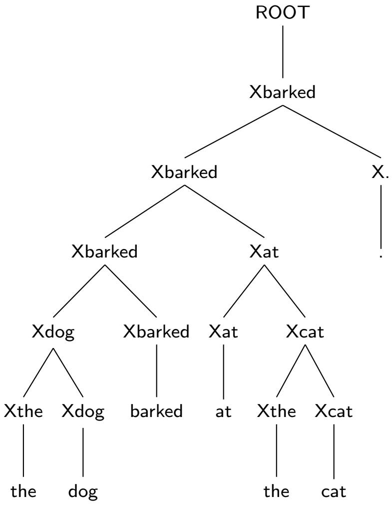
$$\mathcal{O}(n^5)$$

Spurious structures

Several constituency trees can correspond to one and the same dependency tree

Bilexical grammar: Spurious ambiguity

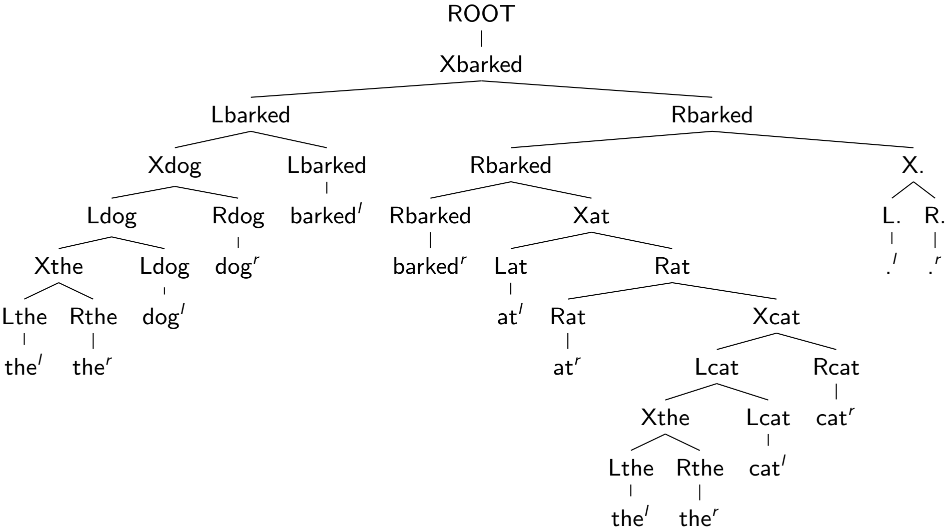




Split-head representation

- Word w_i can be a root:
  $ROOT \rightarrow X_i$
- We collect left and right dependents separately:
  $X_i \rightarrow L_i R_i$
- Left ($w_j \leftarrow w_i$) and right ($w_i \rightarrow w_j$) dependencies:
  $L_i \rightarrow X_j\ L_i$
  $R_i \rightarrow R_i\ X_j$
- Two lexical rules per word:
  $L_i \rightarrow w_i^l$
  $R_i \rightarrow w_i^r$

Split-head representation: Example



Split-head representation: Complexity

Parsing complexity

$$\mathcal{O}(n^4)$$

Spurious structures

None

Other approaches

Methods

- Unfold-fold transformation
- Eisner algorithm (arc-factored parsing), without detour via CFG

Parsing complexity

$$\mathcal{O}(n^3)$$

T H E

E N D

Earley, J. (1970).
An efficient context-free parsing algorithm.
Commun. ACM, 13(2):94–102.

Kubler, S., McDonald, R., Nivre, J., and Hirst, G. (2009).
Dependency Parsing.
Morgan and Claypool Publishers.

Younger, D. H. (1967).
Recognition and parsing of context-free languages in time n3.
Information and Control, 10(2):189–208.