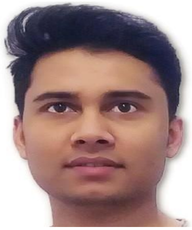
Deep Learning for NLP 2020

Lecture 5 - Dependency Parsing, Guiding Questions

Dr. Steffen Eger
Niraj D Pandey
Wei Zhao





Natural Language Learning Group (NLLG)
Technische Universität Darmstadt

Question 1

- What does NP → DET A* N PP* mean?
- Which phrases could we derive from this grammar rule?

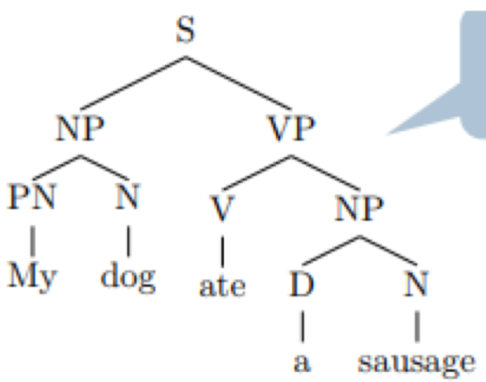
Answer 1

- So-called phrase structure (aka contex-free grammar) rule
- “NP” (noun phrase) expands to determiner, zero or more adjectives, noun, and zero or more prepositional phrases
- We could derive e.g. “a happy happy cow in a castle” as an NP

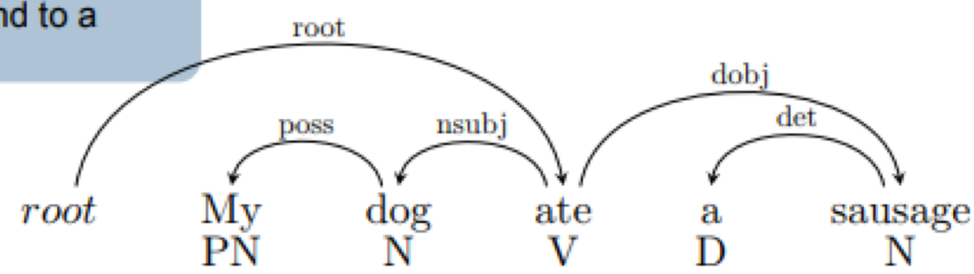
Question 2

- Which are two ways to specify the grammar of sentences?
- How do they differ? What do they have in common?

Answer 2



Each internal node correspond to a phrase



Constuent (a.k.a. phrase-structure) tree

Dependency tree

3

Answer 2

- Difference: CFG uses non-terminal symbols, describes how sentence is structured in components/**phrases**. Dependency Parsing gives head/dependent **relationships** between words
- Similarities: one can convert one into the other using heuristic techniques or machine learning; they are not equivalent

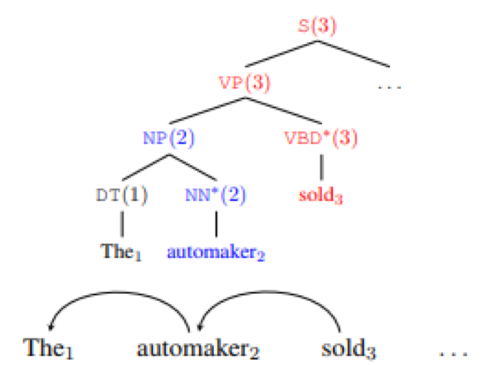


Figure 1: Illustration of c-parse to d-parse conversion with head rules $\{\texttt{VP} \rightarrow \texttt{NP VBD}^*, \texttt{NP} \rightarrow \texttt{DT NN}^*, \ldots\}$. The c-parse is an ordered tree with fringe $x_1, \ldots, x_n$. Each vertex is annotated with a terminal or nonterminal symbol and a derived head index. The blue and red vertices have the words $\texttt{automaker}_2$ and $\texttt{sold}_3$ as heads respectively. The vertex $\texttt{VP}(3)$ implies that $\texttt{automaker}_2$ is a left-dependent of $\texttt{sold}_3$, and that $2 \in \mathcal{L}(3)$ in the d-parse.

Question 3

- What's ambiguous about "He saw the girl with the telescope"? Give two dependency trees that reveal this ambiguity.

Answer 3

- It’s not clear who has the telescope

Question 4

- What's the relationship between dependency parsing and semantics?

Answer 4

- By considering the attachment decisions, several meaning relevant questions about the sentence are answered, e.g.:
  - Who has the telescope?
- There are also approaches that determine the meaning of a sentence using e.g. phrase structure (or similarly dependency structure) such as **Recursive Neural Nets**

Question 7

- Which property do dependency relations have?

TECHNISCHE UNIVERSITÄT DARMSTADT

Answer 7

- Binary
- Asymmetric
- Sometimes typed

Question 10

- Does the arrow go from head to dependent or vice versa?

Answer 10

- That's a matter of convention, you can draw it either way round, but be consistent (e.g. head→modifier OR head←modifier)

Question 11

- What are 4 sources for determining dependency relations?

Answer 11

TECHNISCHE UNIVERSITÄT DARMSTADT

- Bilexical affinities
- Dependency distance
- Intervening material
- Valency of heads

Question 12

- What is projectivity?

Answer 12

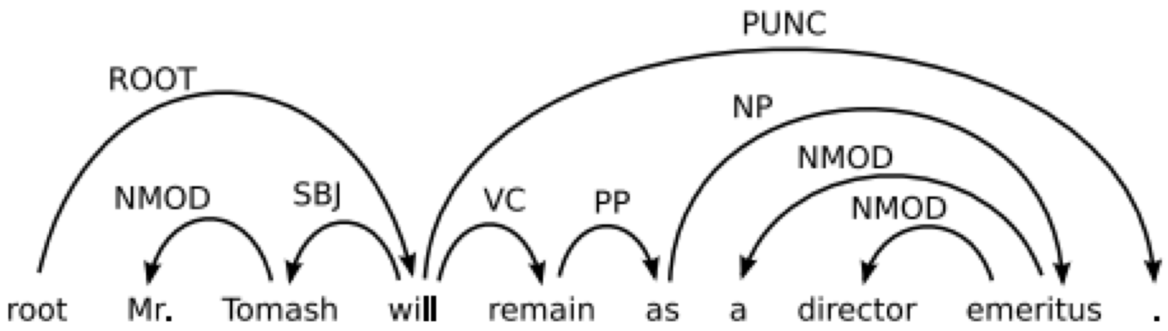


Figure 1: A projective dependency graph.

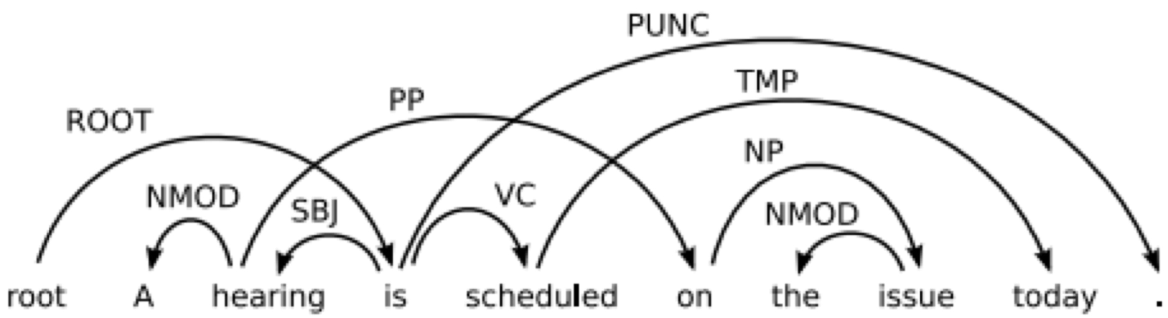


Figure 2: Non-projective dependency graph.

Answer 12

TECHNISCHE UNIVERSITÄT DARMSTADT

- nesting structure relative to linear order
- Non-projective: crossing edges in dependency tree
- More “complex” languages are non-projective (German, Czech, ...)
- Arc-standard can’t handle non-projective phenomena (need a swap operation)

Question 14

- Which components does a transition-based system have and which operations?

Answer 14

- **Buffer**: stores the words, puts them on the stack for processing
- **Stack**: relationships between words can only be constructed when words are on the stack
- **A set of arcs**
- **Transitions**: E.g. left-arc, right-arc, shift (can also be typed)

Question 15

- What does arc-standard dependency parsing mean?

Answer 15

- Dependency parsing with arc-standard operations
  - Left-arc, right-arc, shift
- Can also include other operations (e.g. swap), then it’s not called arc-standard anymore
- There are also other systems (e.g., arc-eager) with other operations

Parsing Example

TECHNISCHE UNIVERSITÄT DARMSTADT

Arc-standard transition-based parser

(there are other transition schemes …)

Analysis of "I ate fish"

Start

[root] | I ate fish

Shift

[root] I | ate fish

Shift

[root] I ate | fish

Start: σ = [ROOT], $\beta = w_1, \ldots, w_n$, A = ∅

1. Shift $\sigma, w_i|\beta, A \rightarrow \sigma|w_i, \beta, A$
2. Left-Arc$_r$ $\sigma|w_i|w_j, \beta, A \rightarrow \sigma|w_j, \beta, A \cup \{r(w_j, w_i)\}$
3. Right-Arc$_r$ $\sigma|w_i|w_j, \beta, A \rightarrow \sigma|w_i, \beta, A \cup \{r(w_i, w_j)\}$

Finish: β = ∅

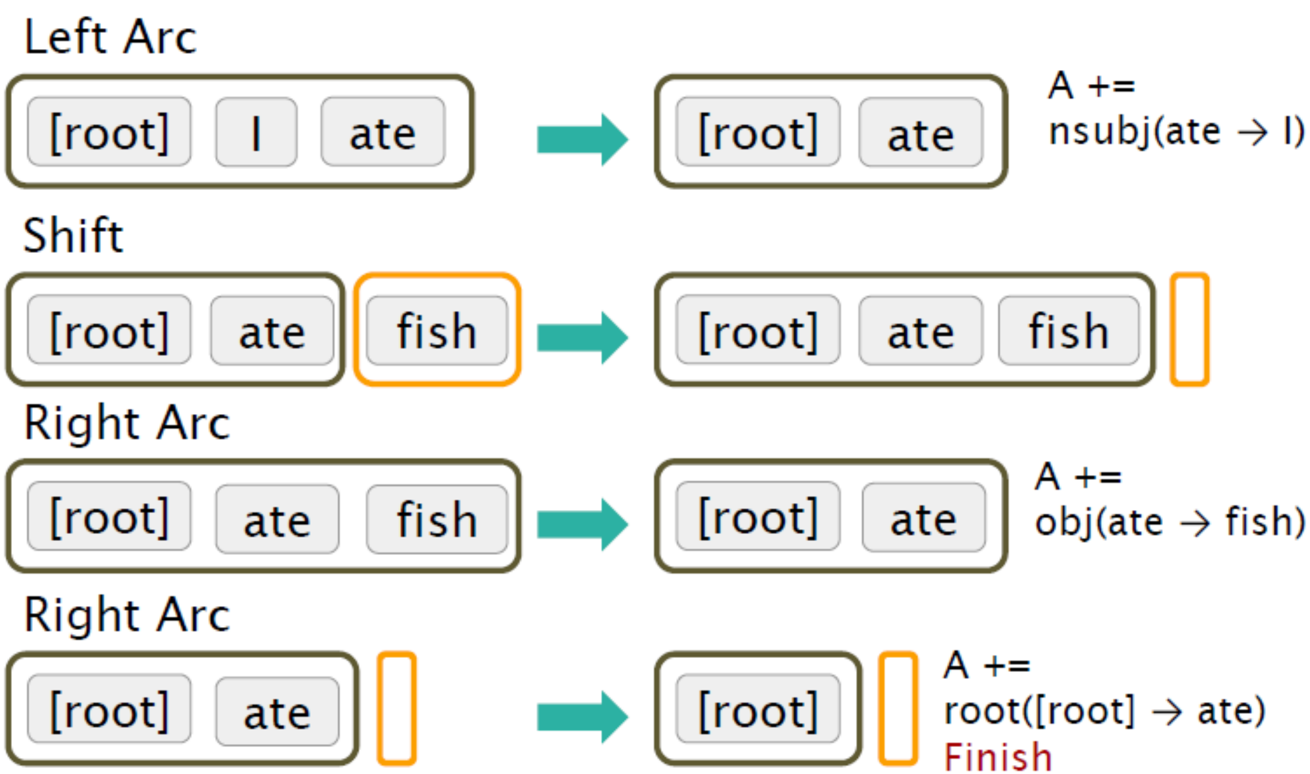
Parsing Example

TECHNISCHE UNIVERSITÄT DARMSTADT

Christopher Manning

Arc-standard transition-based parser

Analysis of “I ate fish”



Neural Model for Parsing

Model Architecture

Softmax probabilities

Output layer y

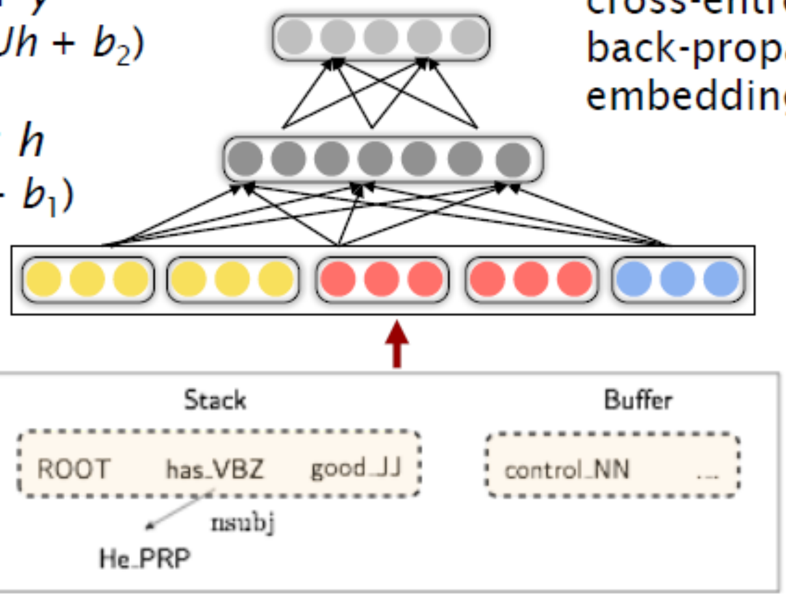
$y = \text{softmax}(Uh + b_2)$

Hidden layer h

$h = \text{ReLU}(Wx + b_1)$

Input layer x

lookup + concat



cross-entropy error will be back-propagated to the embeddings.

Question 18

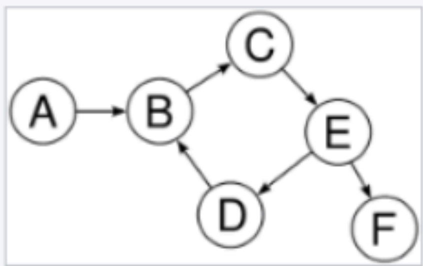
- Which two measures are used to evaluate dependency parsing?

Answer 18

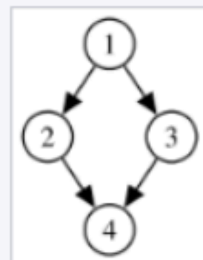
- LAS and UAS
  - UAS: only count correct relations
  - LAS: also include the types

Tree structure violations

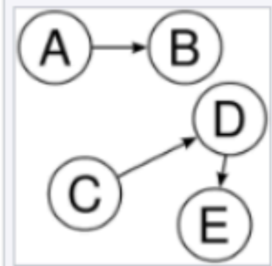
- No dependent can have two heads



Not a tree: cycle B→C→E→D→B. B has more than one parent (inbound edge).



Not a tree: undirected cycle 1-2-4-3. 4 has more than one parent (inbound edge).



Not a tree: two non-connected parts, A→B and C→D→E. There is more than one root.

How to deal with conjunctions?

- “I laugh and say”

Problem: „I“ is subject of „laugh“ and „say“

Index	Word	Head (index)
1	I	2
2	laugh	0
3	and	4
4	say	2

- How does the corresponding tree look like in visual representations