Dependency Parsing

Natural Language Processing

(based on revision of Chris Manning Lectures)

Announcement

- TA announcements (if any)...

Suggested Readings

1. https://web.stanford.edu/~jurafsky/slp3/14.pdf (dependency parsing)
2. Incrementality in Deterministic Dependency Parsing
3. A Fast and Accurate Dependency Parser using Neural Networks
4. Dependency Parsing
5. Globally Normalized Transition-Based Neural Networks
6. Universal Stanford Dependencies: A cross-linguistic typology
7. Universal Dependencies website

Two views of linguistic structure: **Context-free** grammar and **Dependency** grammar

1. Two views of linguistic structure: Context-free grammar

Phrase structure organizes words into nested constituents

Starting unit: words

the, cat, cuddly, by, door

det noun adj prep noun

Words combine into phrases

The cuddly cat, by the door

np: noun phrase; prep np

pp: prepositional phrase

Phrases can combine into bigger phrases

The cuddly cat by the door

np np

pp

np

Noun → *flights* | *breeze* | *trip* | *morning*
Verb → *is* | *prefer* | *like* | *need* | *want* | *fly*
Adjective → *cheapest* | *non-stop* | *first* | *latest* | *other* | *direct*
Pronoun → *me* | *I* | *you* | *it*
Proper-Noun → *Alaska* | *Baltimore* | *Los Angeles* | *Chicago* | *United* | *American*
Determiner → *the* | *a* | *an* | *this* | *these* | *that*
Preposition → *from* | *to* | *on* | *near*
Conjunction → *and* | *or* | *but*

Figure 12.2 The lexicon for $\mathscr{L}_0$.

We called the word on the right (e.g., the) **terminal symbols**, and the grammar rules as **lexicon**

1. Two views of linguistic structure: Context-free grammar

NP -> Det N
e.g., the cat

NP -> Det (Adj) N
e.g., the large cat

PP -> P NP
e.g., by the door

NP -> Det (Adj) N (PP)
e.g., the large cat by the door

NP -> Det (Adj)* N (PP)
e.g., the large cute furry cat by the door

VP -> V PP
e.g., talk to the cat

S -> NP VP
e.g., the cat walked behind the dog

More grammar rules! As much as we want....

1. Two views of linguistic structure: Context-free grammar

Using context-free grammar to parse gives us a **parse tree**

Grammar Rules	Examples
$S \rightarrow NP\ VP$	I + want a morning flight
$NP \rightarrow Pronoun$	I
$\mid\ Proper\text{-}Noun$	Los Angeles
$\mid\ Det\ Nominal$	a + flight
$Nominal \rightarrow Nominal\ Noun$	morning + flight
$\mid\ Noun$	flights
$VP \rightarrow Verb$	do
$\mid\ Verb\ NP$	want + a flight
$\mid\ Verb\ NP\ PP$	leave + Boston + in the morning
$\mid\ Verb\ PP$	leaving + on Thursday
$PP \rightarrow Preposition\ NP$	from + Los Angeles

Figure 12.3 The grammar for $\mathscr{L}_0$, with example phrases for each rule.

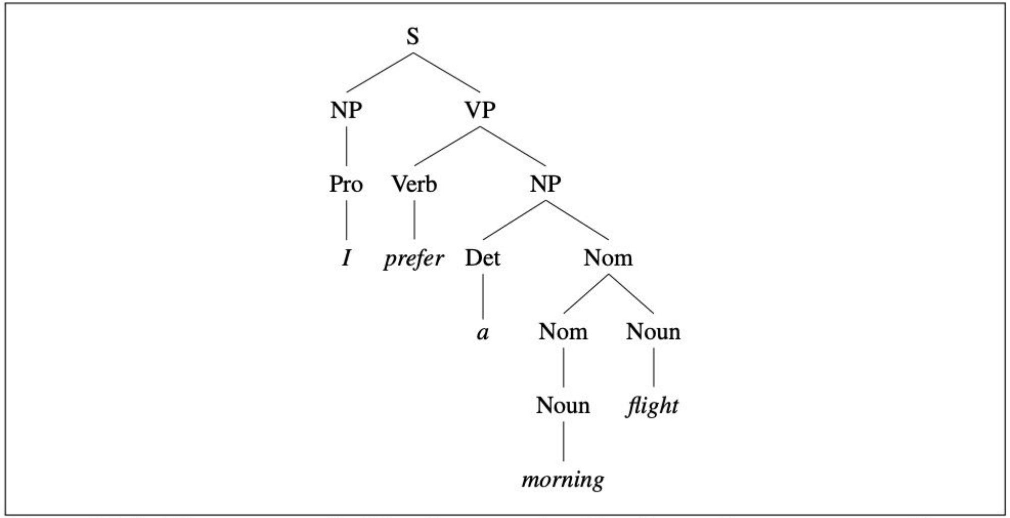
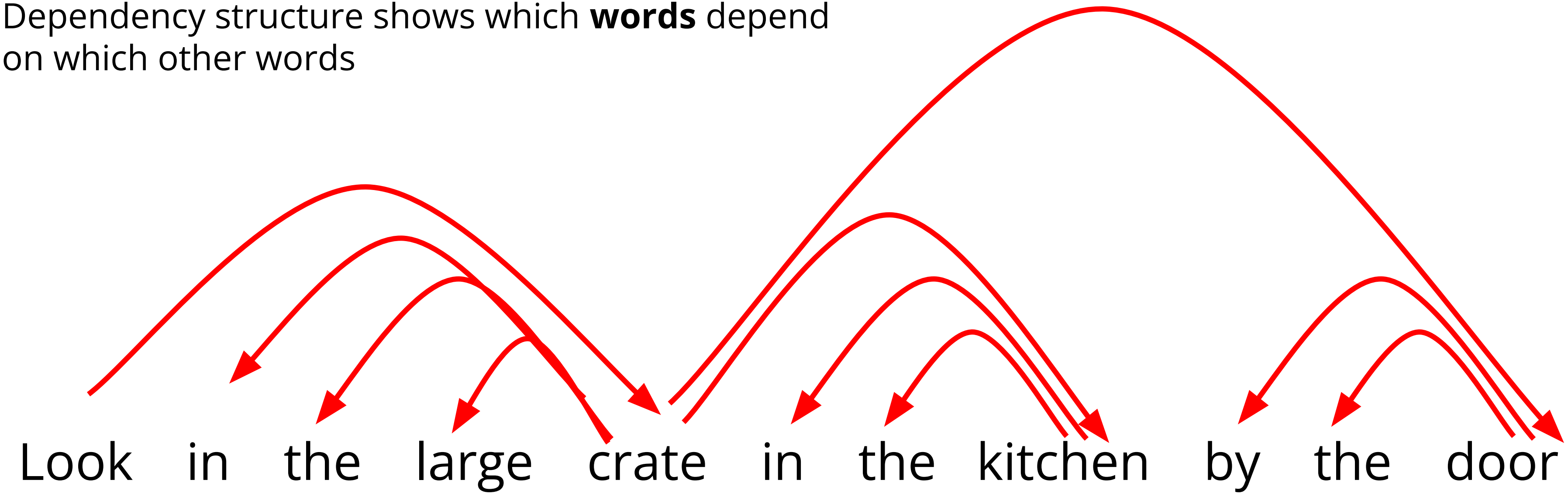


Figure 12.4 The parse tree for "I prefer a morning flight" according to grammar $\mathscr{L}_0$.

CFG is useful for **grammar checking** (check whether the sentence deviates from the lexicon), **entities extraction** (check common pattern), **sentence classification** (e.g., question has certain pattern)

2. Two views of linguistic structure: Dependency grammar

Dependency structure shows which **words** depend on which other words



E.g., crate "depends on" the word "large". Here "crate" (start of the arrow) is called "head" and "large" (where the arrow point to) is called the "dependent"

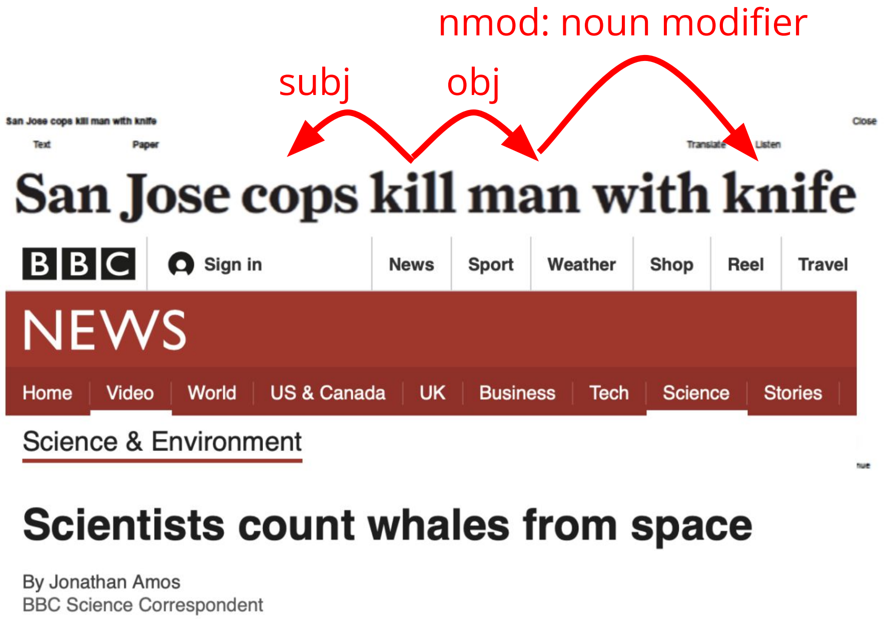
2. Two views of linguistic structure: Dependency grammar

Clausal Argument Relations	Description
NSUBJ	Nominal subject
DOBJ	Direct object
IOBJ	Indirect object
CCOMP	Clausal complement
XCOMP	Open clausal complement
Nominal Modifier Relations	**Description**
NMOD	Nominal modifier
AMOD	Adjectival modifier
NUMMOD	Numeric modifier
APPOS	Appositional modifier
DET	Determiner
CASE	Prepositions, postpositions and other case markers
Other Notable Relations	**Description**
CONJ	Conjunct
CC	Coordinating conjunction

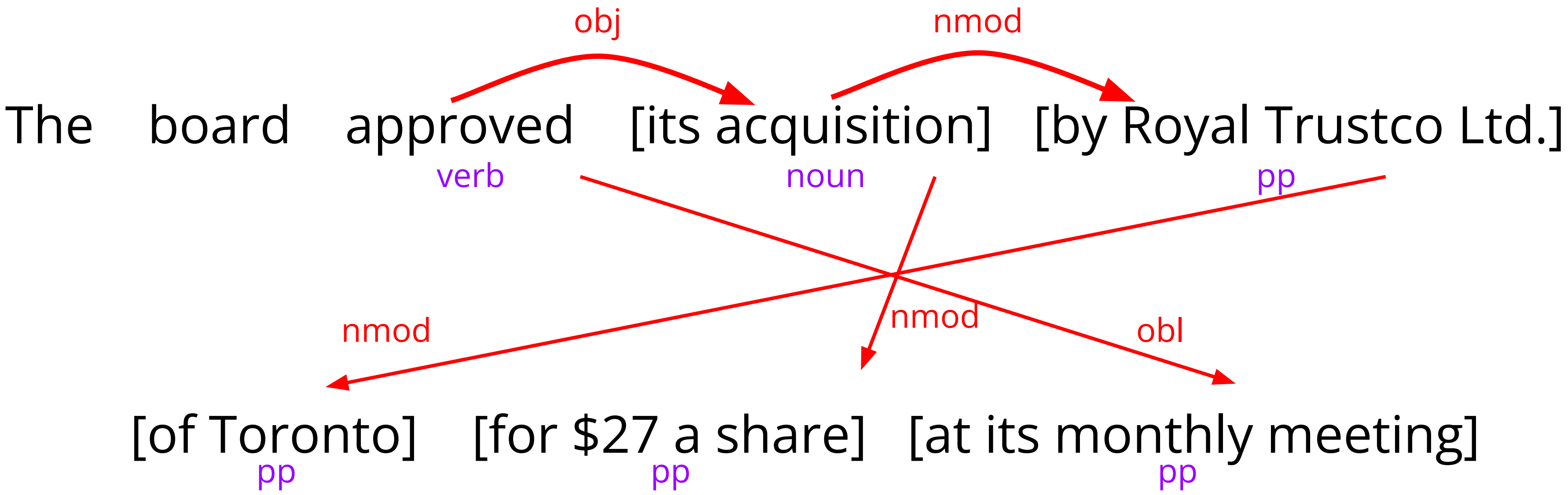
Figure 14.2 Some of the Universal Dependency relations (de Marneffe et al., 2014).

https://universaldependencies.org/u/dep/

Prepositional phrase attachment ambiguity



How to link?



The number of possible parses grows exponentially w.r.t. to the number of prepositional phrases (here n = 4) (i.e., Catalan numbers)

Coordination scope ambiguity

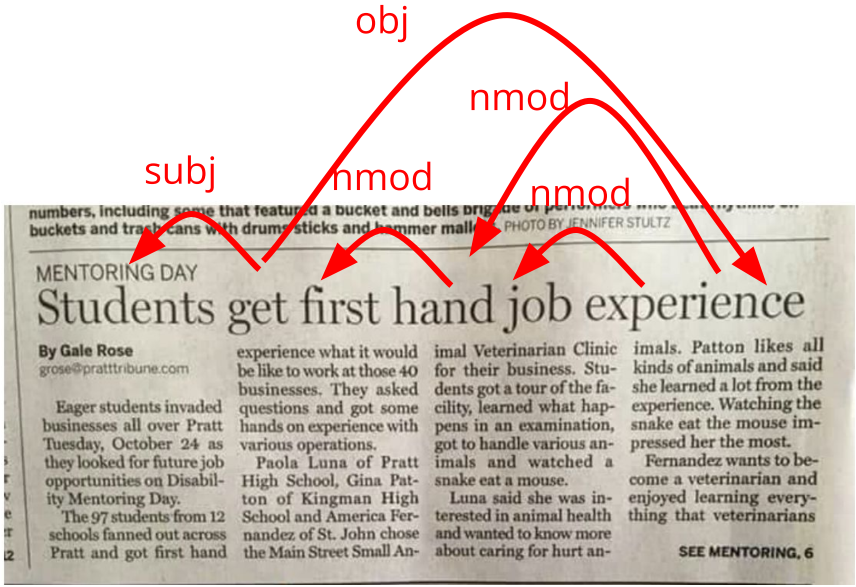
coord nmod

[**Shuttle veteran**] and [**longtime NASA executive Fred Gregory**] appointed to board

nmod

[**Shuttle veteran and longtime NASA executive**] Fred Gregory appointed to board

Adjectival/Adverbial Modifier ambiguity



numbers, including some that featured a bucket and bells brigade of performers ... buckets and trash cans with drums sticks and hammer mallets. PHOTO BY JENNIFER STULTZ

MENTORING DAY

Students get first hand job experience

By Gale Rose
grose@pratttribune.com

Eager students invaded businesses all over Pratt Tuesday, October 24 as they looked for future job opportunities on Disability Mentoring Day.

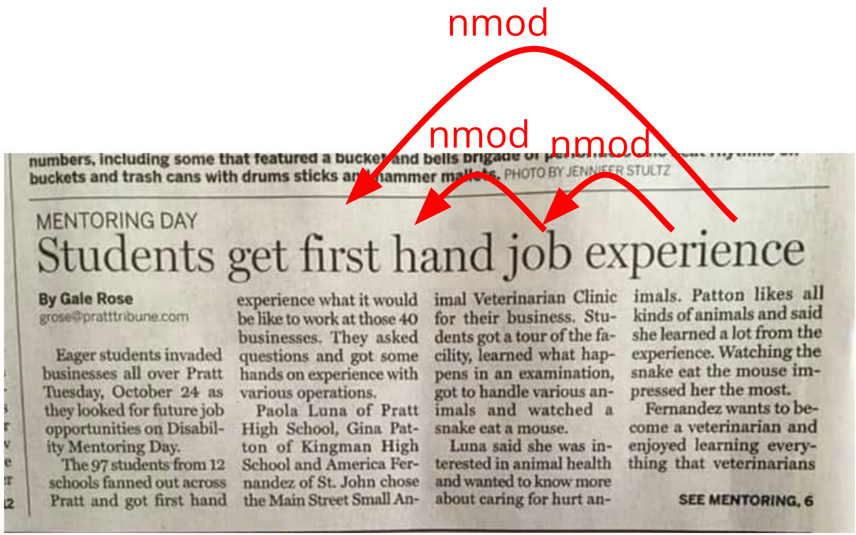
The 97 students from 12 schools fanned out across Pratt and got first hand experience what it would be like to work at those 40 businesses. They asked questions and got some hands on experience with various operations.

Paola Luna of Pratt High School, Gina Patton of Kingman High School and America Fernandez of St. John chose the Main Street Small Animal Veterinarian Clinic for their business. Students got a tour of the facility, learned what happens in an examination, got to handle various animals and watched a snake eat a mouse.

Luna said she was interested in animal health and wanted to know more about caring for hurt animals. Patton likes all kinds of animals and said she learned a lot from the experience. Watching the snake eat the mouse impressed her the most.

Fernandez wants to become a veterinarian and enjoyed learning everything that veterinarians

SEE MENTORING, 6



numbers, including some that featured a bucket and bells brigade of performers ... buckets and trash cans with drums sticks and hammer mallets. PHOTO BY JENNIFER STULTZ
MENTORING DAY
Students get first hand job experience
By Gale Rose
grose@pratttribune.com
Eager students invaded businesses all over Pratt Tuesday, October 24 as they looked for future job opportunities on Disability Mentoring Day.
The 97 students from 12 schools fanned out across Pratt and got first hand experience what it would be like to work at those 40 businesses. They asked questions and got some hands on experience with various operations.
Paola Luna of Pratt High School, Gina Patton of Kingman High School and America Fernandez of St. John chose the Main Street Small Animal Veterinarian Clinic for their business. Students got a tour of the facility, learned what happens in an examination, got to handle various animals and watched a snake eat a mouse.
Luna said she was interested in animal health and wanted to know more about caring for hurt animals. Patton likes all kinds of animals and said she learned a lot from the experience. Watching the snake eat the mouse impressed her the most.
Fernandez wants to become a veterinarian and enjoyed learning everything that veterinarians
SEE MENTORING, 6

Verb phrase (VP) attachment ambiguity

this?

or this?

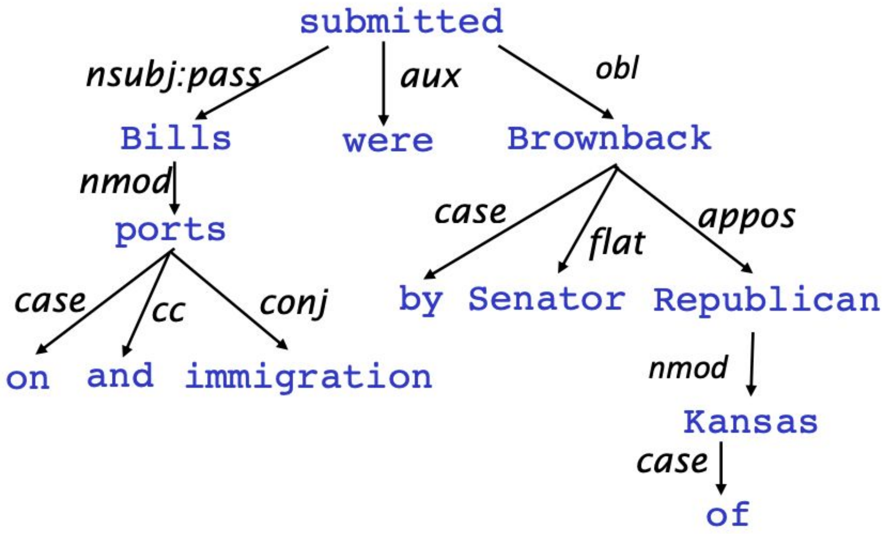
Mutilated body washes up on **Rio beach** **to be used for Olympics beach volleyball**

VP

Dependency as tree

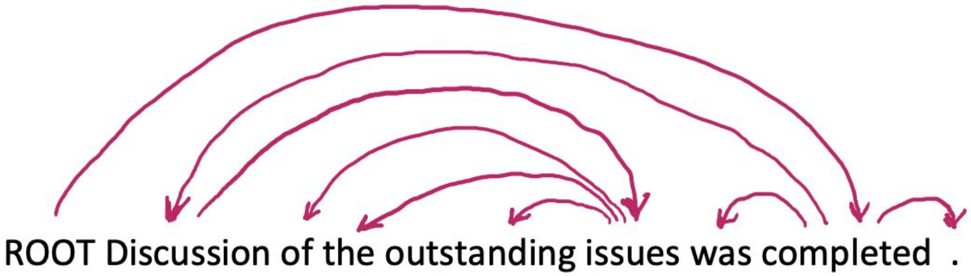
Dependencies can be redrawn as tree (motivated by computer science)

Dependencies usually form a **connected**, **acyclic**, **single-root** graph.



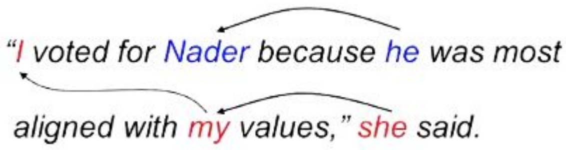
Dependency as tree

- Some people draw the arrows one way; some the other way!
  - **Tesniere (1959)** had them point from head to dependent - we follow that convention
- We usually add a **fake ROOT** so every word is a dependent of precisely 1 other node



Why we need to learn sentence structure?

- Humans need to work out what *depends* what
  - Thus, similarly, a model needs to understand sentence structure in order to interpret language correctly
  - E.g., **conference resolution**

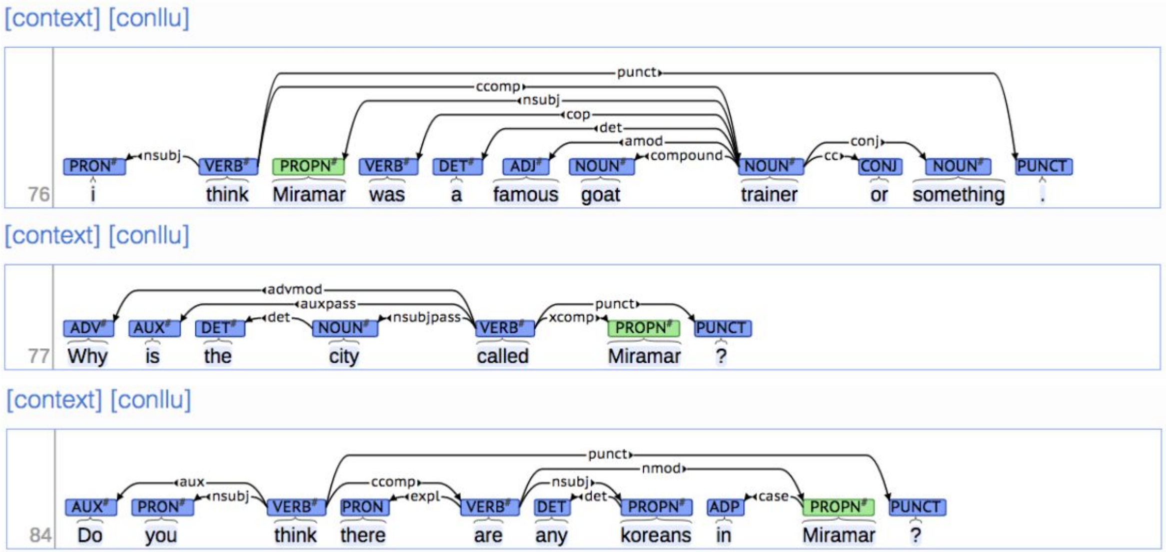


- Human language is **ambiguous** by nature
  - Knowing these ambiguities allow us to watch out for these potential problems in our model

Dependencies Treebanks

The rise of annotated data and Universal Dependencies treebank

Brown corpus (1967; PoS tagged 1979); Lancaster-IBM Treebank (starting late 1980s); Marcus et al. 1993, The Penn Treebank, Computational Linguistics; Universal Dependencies: http://universaldependencies.org/



- Many parsers, part-of-speech taggers, etc. can be built on it
- Broad coverage, not just a few intuitions
- Contain frequencies and distributional information
- Provide a way to evaluate existing NLP systems - turn NLP into actual science

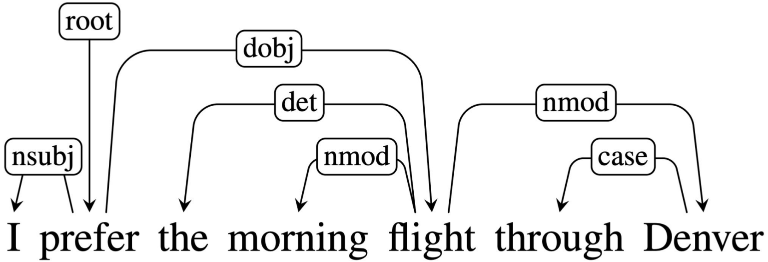
So how do we build a parser once we got these dependencies?

Many information you can use from the treebanks

1. **Bilexical affinities**
   a. Check whether both ends of dependencies seem right
2. **Dependency distance**
   a. Check whether the distance is too far away from usual
3. **Intervening material**
   a. Because sentences are normally organized around verbs, dependencies rarely span intervening verbs
   b. Because punctuation (e.g., commas) indicates segment, it also indicates less plausible spanning intervening punctuations
4. **Valency of heads**
   a. Given a head, what is usually its dependents and on which side?

So how do we build a parser once we got these dependencies?

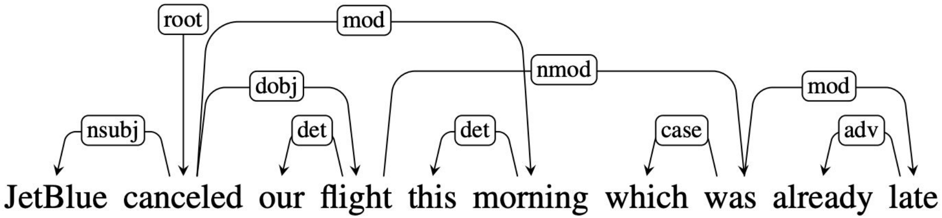
- A sentence is **parsed** by choosing for each word, what other (including ROOT) it is a dependent of
- To make the problem easier to solve, we impose some **constraints**:
    a. There is a single designated root node that has no incoming arcs
    b. With the exception of the root node, each vertex has exactly one incoming arc
    c. There is a unique path from the root node to each vertex in V
    d. Don’t want cycles, e.g., A->B, B->A



Projectivity

Note that some dependency parsing algorithms are **projective**:

Projectivity: An arc from a head to a dependent is said to be projective **if there is a path from the head to every word that lies between the head and the dependent in the sentence**. A dependency tree is then said to be projective if all the arcs that make it up are projective.



In this example, the arc from **flight** to its modifier was is non-projective since there is no path from **flight** to the intervening words **this** and **morning**. A dependency tree is projective if it can be drawn with **no crossing edges**. Here there is no way to link flight to its dependent was without crossing the arc that links morning to its head.

Projectivity

There are computational limitations to the most widely used families of parsing algorithms. The **transition-based approaches** we gonna discuss shortly **can only produce projective trees**, hence any sentences with non-projective structures will necessarily contain some errors. This limitation is one of the motivations for the more flexible graph-based parsing approach.

Dependency Parsing Methods

Greedy transition-based parsing [Nivre IWPT 2003]

- A simple form of greedy discriminative dependency parser
- The parser does a sequence of bottom up actions
  - Roughly like "shift" or "reduce" in a shift-reduce parser, but the "reduce" actions are specialized to create dependencies with head on left or right
- The parser has:
  - **a stack σ**, written with top to the right
    - which starts with the ROOT symbol
  - **a buffer β**, written with top to the left
    - which starts with the input sentence
  - a set of **dependency arcs A**
    - which starts off empty
  - a set of **actions**
    - shift or left arc or right arc

An efficient algorithm for projective dependency parsing, Joakim Nivre, 2003. https://aclanthology.org/W03-3017/

Greedy transition-based parsing

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

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

Finish: $\sigma = [w]$, $\beta = \emptyset$

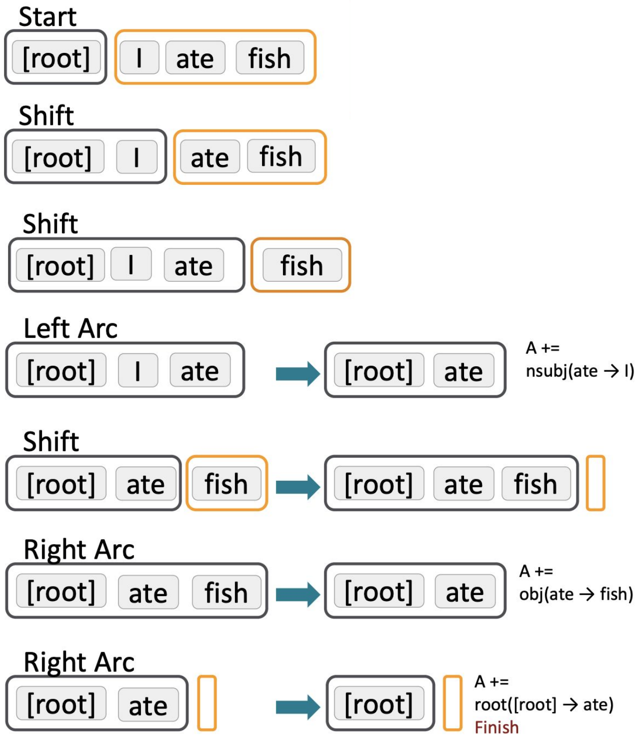
adding the relationship

Start: $\sigma = [\text{ROOT}]$, $\beta = w_1, \ldots, w_n$, $A = \emptyset$

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: $\sigma = [w]$, $\beta = \emptyset$

Greedy transition-based parsing



if the only thing in the stack is ROOT, we can only SHIFT

once we shift, we can either perform dependencies by doing LEFT-ARC or RIGHT-ARC. But here, let's say my model tells me to SHIFT more

similarly, my model gonna decide whether to LEFT-ARC or RIGHT-ARC. It seems to be LEFT ARC, because "I" is the subject of "ate"

the dependent is gone. We add the dependencies to A

We SHIFT again. Nothing left on the buffer. We must do either RIGHT-ARC or LEFT-ARC. Let's say the model tells me to perform RIGHT-ARC.

the dependent is gone. We add the dependencies to A

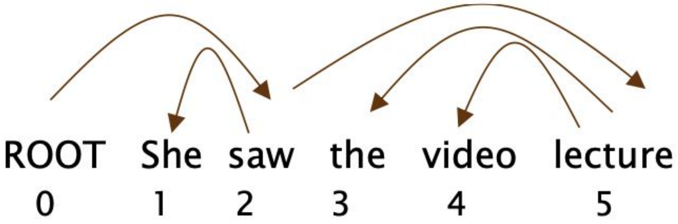
the dependent is gone. We add the dependencies to A. We stop when the buffer is empty and the stack contains only ROOT.

MaltParser [Nivre and Hall 2006]

- **So how to choose the right action??**
  - Answer: Stand back, I know **machine learning**!
- Each action is predicted by a **discriminative classifier** (e.g., SVM) over each legal move
  - Max of 3 untyped choices (shift, left, right); max of |R| × 2 (left, right) + 1 (shift) when typed (consider also the relations)
  - **Features: top of stack word, POS; first in buffer word; etc.**
    - For each word, the features are represented by a one-hot encoding vector. Since there could be millions of features, this vector tends to be very big….(10^6)
  - There is NO search (in the simplest form)
  - But you can profitably do a beam search if you wish (slower but better)
    - You keep k good parse prefixes at each time step
    - The model's accuracy is fractionally below the state of the art in dependency parsing, but it provides very fast linear time parsing, with high accuracy – great for parsing the web

MaltParser: A Data-Driven Parser-Generator for Dependency Parsing, Nivre et al., 2006. https://aclanthology.org/L06-1084/

Evaluation of the parser



$$\text{Acc} = \frac{\#\text{ correct deps}}{\#\text{ of deps}}$$

UAS = 4 / 5 = 80%

LAS = 2 / 5 = 40%

Gold

1	2	She	nsubj
2	0	saw	root
3	5	the	det
4	5	video	nn
5	2	lecture	obj

Parsed

1	2	She	nsubj
2	0	saw	root
3	4	the	det
4	5	video	nsubj
5	2	lecture	ccomp

UAS (unlabeled accuracy score): Just count how many match without considering the relations (e.g., nsubj)

LAS (labeled accuracy score): the label (e.g., nsubj) must also match

Neural Dependency Parsing

A neural transition-based dependency parser [Chen and manning 2014]

- The traditional dependency parser **feature vector is big** and thus computationally costly
- Instead use **word embeddings**!
    - Concatenated along with part of speech tags (POS) and dependency labels (represented as one-hot, but they are much smaller than the full vector anyway)
- Still use the **transition-based approach**

Other research has further improved by adding deeper network, adding beam search, e.g., SyntaxNet and the Parsey McParseFace model (2016)

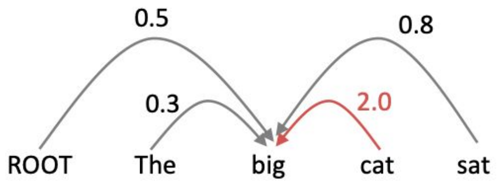
Parser	UAS	LAS	sent. / s
MaltParser	89.8	87.2	469
MSTParser	91.4	88.1	10
TurboParser	**92.3**	89.6	8
C & M 2014	92.0	**89.7**	**654**

MST and TurboParser is graph-based with higher accuracy but is much slower....

A Fast and Accurate Dependency Parser using Neural Networks, Chen and Manning, 2014. https://aclanthology.org/D14-1082 /
Announcing Syntaxnet: The World's Most Accurate Parser Goes Open source https://ai.googleblog.com/2016/05/announcing-syntaxnet-worlds-most.html

A neural graph-based dependency parser [Dozat and manning 2017]

- Can handle **non-projectivity**, unlike transition-based
- Compute a score for **every possible dependency** for each word
  - n^2 possible dependencies in a sentence of length n (this makes parsing slow)
- Repeat the same process for each other word
- Determine the optimal tree by using MST algorithm (in our Data Structure and Algorithms class)



e.g., picking the head for "big"

Method	UAS	LAS (PTB WSJ SD 3.3
Chen & Manning 2014	92.0	89.7
Weiss et al. 2015	93.99	92.05
Andor et al. 2016	94.61	92.79
Dozat & Manning 2017	**95.74**	**94.08**

Stanford's Graph-based Neural Dependency Parser at the CoNLL 2017 Shared Task, Dozat et al.,, 2017. https://aclanthology.org/K17-3002.pdf