Natural Language Processing

Info 159/259

Lecture 12: Syntax (Mar 4, 2023)

Many slides & instruction ideas borrowed from:

David Bamman, Greg Durrett & Dan Jurafsky

Logistics

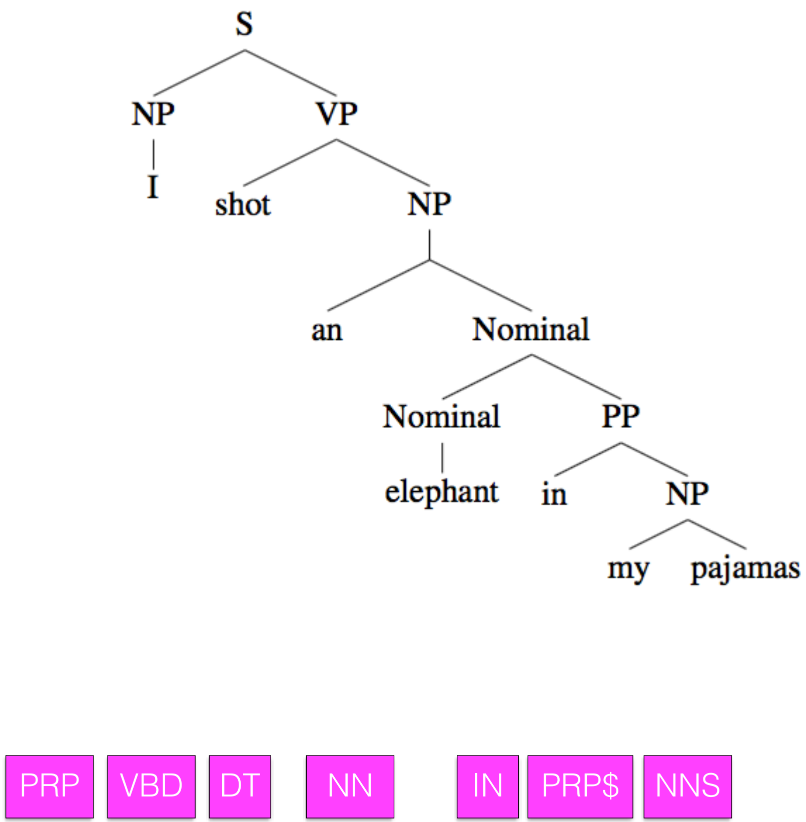
- Quiz 5 is due tonight (Mon 3/4).
- 259 Project Proposals due Tues. 3/5.
- Homework 4 is due this Friday 3/8 (start now if you haven’t already)
- This week: Syntax & Parsing

Syntax

- With syntax, we're moving from labels for discrete items — documents (sentiment analysis), tokens (POS tagging, NER) — to the structure between items.

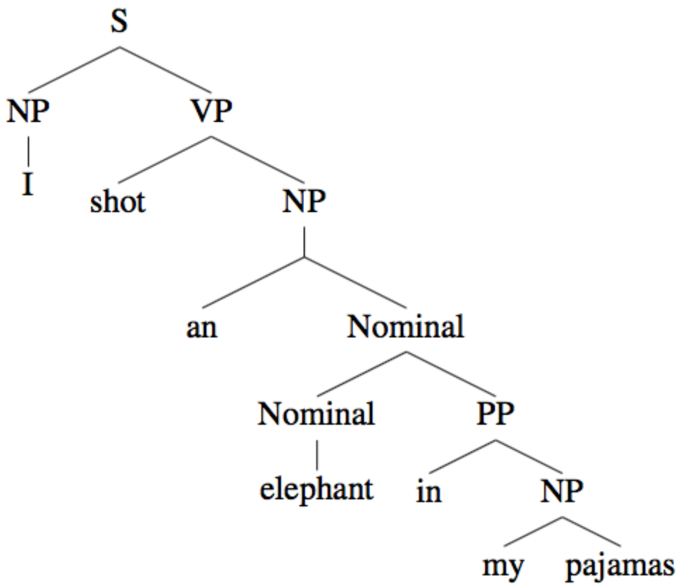
PRP VBD DT NN IN PRP$ NNS

I shot an elephant in my pajamas



I shot an elephant in my pajamas

Why is syntax important?

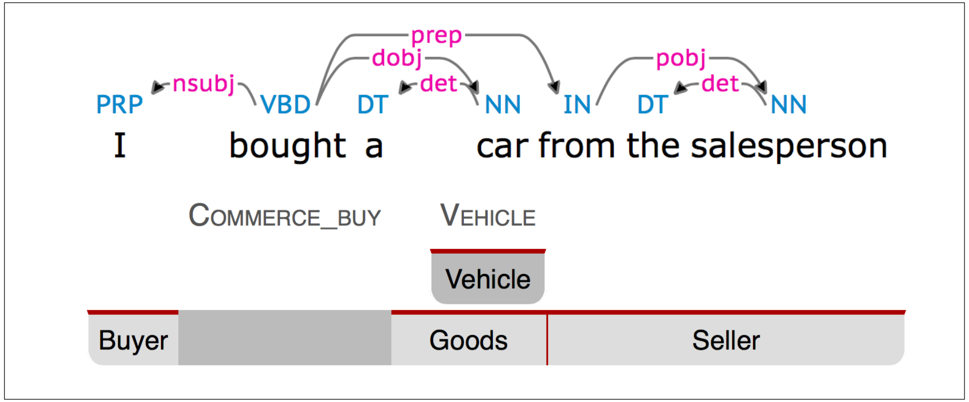


Why is POS important?

- POS tags are indicative of syntax
- POS = cheap multiword expressions [(JJ|NN)+ NN]
- POS tags are indicative of pronunciation ("I contest the ticket" vs "I won the contest"

Why is syntax important?

- Foundation for semantic analysis (on many levels of representation: semantic roles, compositional semantics, frame semantics)



Why is syntax important?

- Strong representation for discourse analysis (e.g., coreference resolution)

Bill VBD Jon; he was having a good day.

- Many factors contribute to pronominal coreference (including the specific verb above), but syntactic subjects > objects > objects of prepositions are more likely to be antecedents

Why is syntax important?

Linguistic typology; relative positions of subjects (S), objects (O) and verbs (V)

SVO	English, Mandarin	I grabbed the chair
SOV	Latin, Japanese	I the chair grabbed
VSO	Hawaiian	Grabbed I the chair
OSV	Yoda	Patience you must have
…	…	…

Data

- NELL SVO triples (604 million nsubj+dobj relations from 230B words on the web

police	found	five .030 bullets	1
police	found	seven dead rebels	3
police	found	two hidden cameras	2
police	found	wanders lover	1
police	found	211 pounds	4
police	found	Marcia	3
police	found	bank draft	1
police	found	diskette	2
police	found	five marijuana plants	3
police	found	items used	1
police	found	judge	5

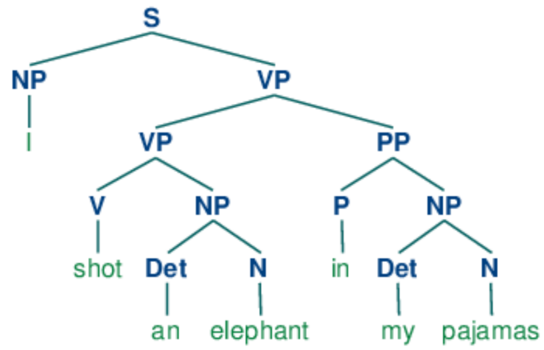
Syntax

- Syntax is fundamentally about the hierarchical structure of language and (in some theories) which sentences are grammatical in a language

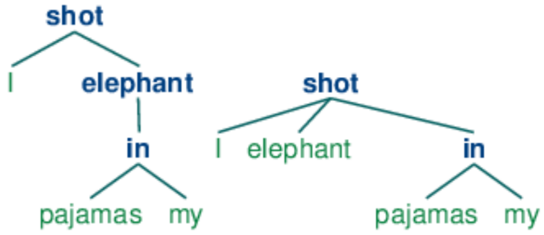
words → phrases → clauses → sentences

Formalisms

Phrase structure grammar
(Chomsky 1957)

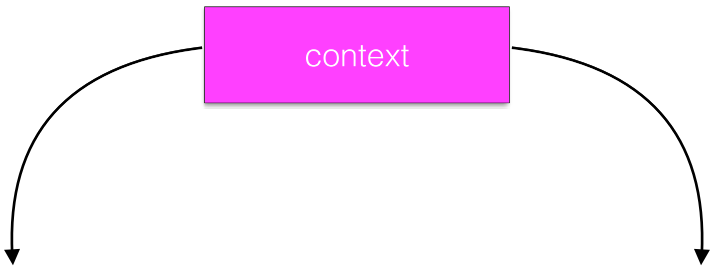


Dependency grammar
(Mel'čuk 1988; Tesnière 1959; Pāṇini)



Constituency

- Groups of words (“constituents”) behave as single units
- “Behave” = show up in the same distributional environments



everyone likes	______________	
a bottle of	______________	is on the table
	______________	makes you drunk
a cocktail with	______________	and seltzer

Parts of speech

- Parts of speech are categories of words defined distributionally by the morphological and syntactic contexts a word appears in.

Syntactic distribution

- Substitution test (for POS): if a word is replaced by another word, does the sentence remain grammatical?

Kim saw the	elephant	before we did
	dog	
	idea	
	*of	
	*goes	

Bender 2013

Syntactic distributions

three parties from Brooklyn	arrive
a high-class spot such as Mindy’s	attracts
the Broadway coppers	love
they	sit

Jurafsky and Martin 2017

Syntactic distributions

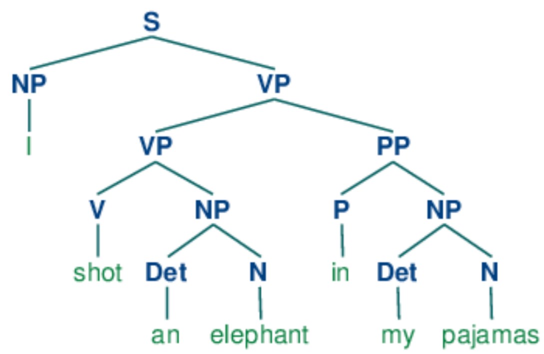
I'd like to fly from Atlanta to Denver

^ ^ ^ ^

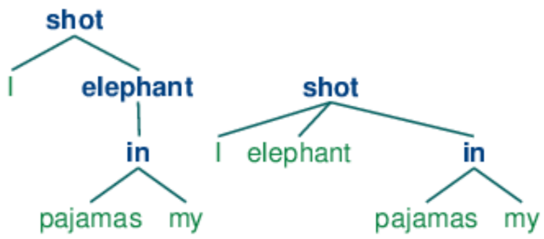
on September seventeenth

Formalisms

Phrase structure grammar
(Chomsky 1957)

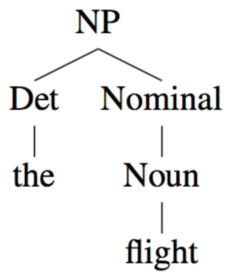


Dependency grammar
(Mel'čuk 1988; Tesnière 1959; Pāṇini)

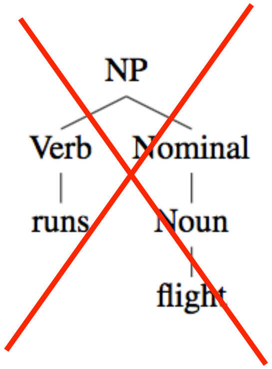


Context-free grammar

- A CFG gives a formal way to define what meaningful constituents are and exactly how a constituent is formed out of other constituents (or words). It defines valid structure in a language.



NP → Det Nominal



NP → Verb Nominal

Context-free grammar

A context-free grammar defines how symbols in a language combine to form valid structures

NP	→	Det Nominal
NP	→	ProperNoun
Nominal	→	Noun | Nominal Noun
Det	→	a | the
Noun	→	flight

non-terminals

lexicon/
terminals

Context-free grammar

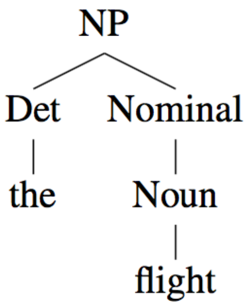
N	Finite set of non-terminal symbols	NP, VP, S
Σ	Finite alphabet of terminal symbols	the, dog, a
R	Set of production rules, each $A \rightarrow \beta$ $\beta \in (\Sigma, N)$	S → NP VP Noun → dog
S	Start symbol	

Language

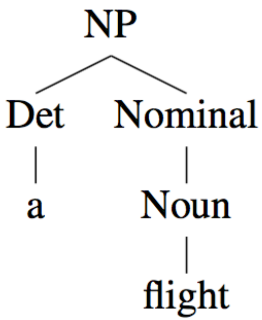
The formal language defined by a CFG is the set of strings derivable from S (start symbol)

Derivation

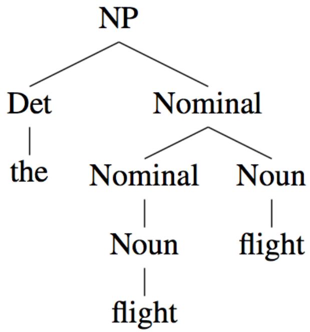
Given a CFG, a derivation is the sequence of productions used to generate a string of words (e.g., a sentence), often visualized as a parse tree.



the flight



a flight



the flight flight

Verb phrases

VP	→	Verb	disappear
VP	→	Verb NP	prefer a morning flight
VP	→	Verb NP PP	prefer a morning flight on Tuesday
VP	→	Verb PP	leave on Tuesday
VP	→	Verb S	I think [$_S$ I want a new flight]
VP	→	Verb VP	want [$_{VP}$ to fly today]

Not every verb can appear in each of these productions

Verb phrases

VP	→	Verb	*I filled
VP	→	Verb NP	*I exist the morning flight
VP	→	Verb NP PP	*I exist the morning flight on Tuesday
VP	→	Verb PP	*I filled on Tuesday
VP	→	Verb S	*I exist [$_S$ I want a new flight]
VP	→	Verb VP	* I fill [$_{VP}$ to fly today]

Not every verb can appear in each of these productions

CFGs

- Building a CFG by hand is really hard
- To capture all (and only) grammatical sentences, need to exponentially increase the number of categories (e.g., detailed subcategorization info)

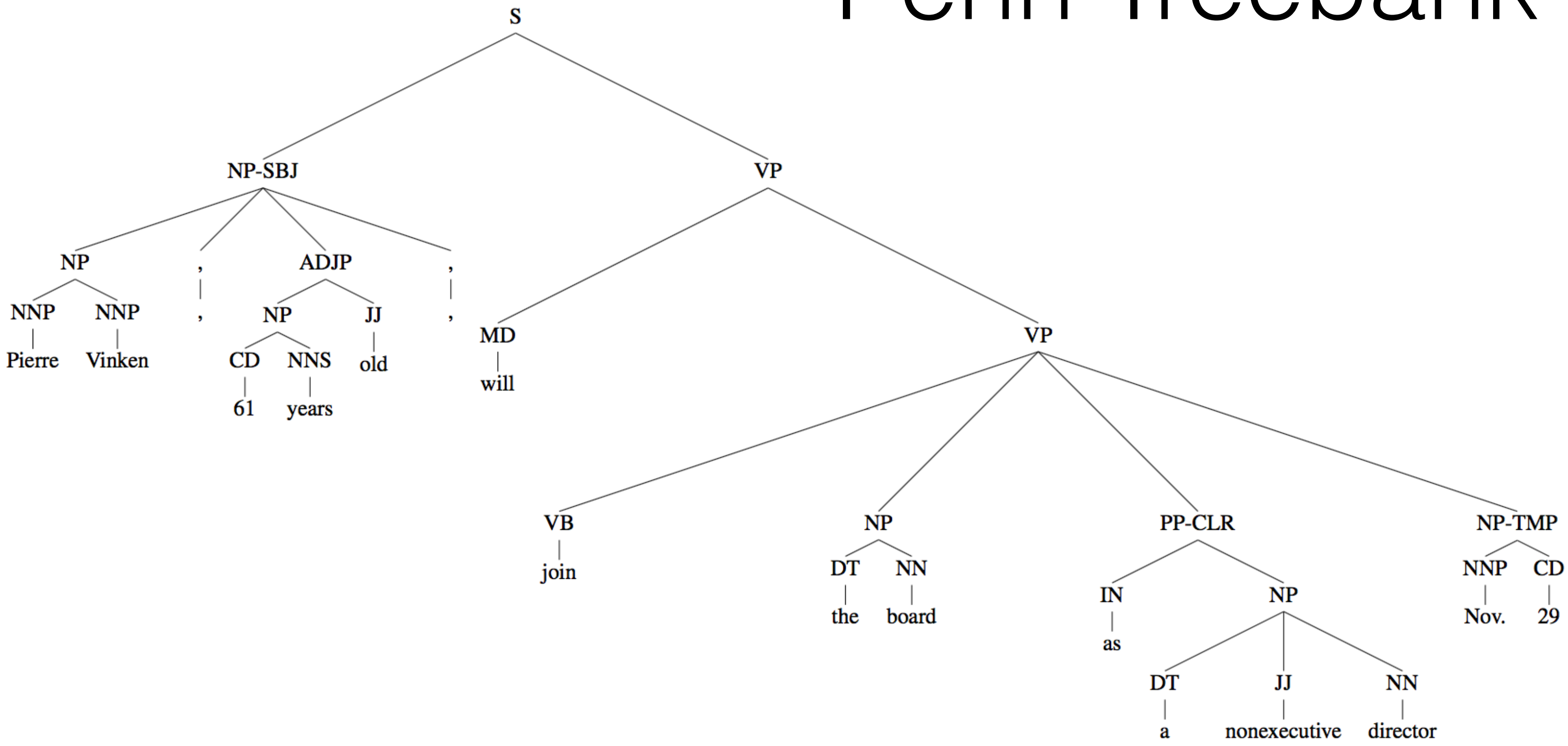
Verb-with-no-complement	→	disappear
Verb-with-S-complement	→	said
VP	→	Verb-with-no-complement
VP	→	Verb-with-S-complement S

Treebanks

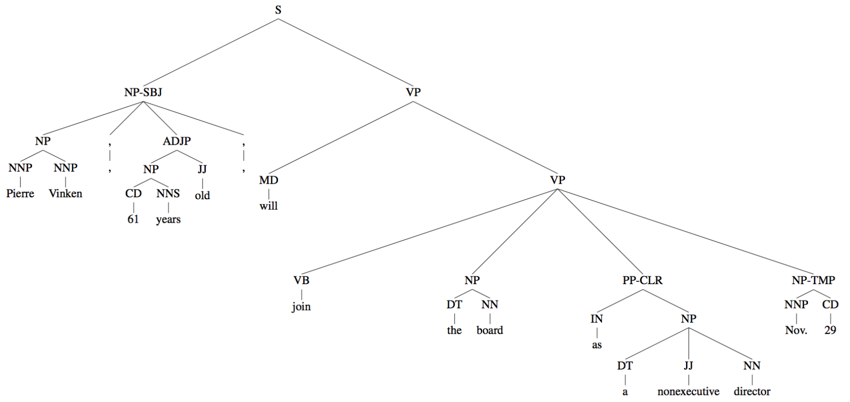
- Rather than create the rules by hand, we can annotate sentences with their syntactic structure and then extract the rules from the annotations

- Treebanks: collections of sentences annotated with syntactic structure

Penn Treebank



Penn Treebank



NP	→	NNP NNP
NP-SBJ	→	NP , ADJP ,
S	→	NP-SBJ VP
VP	→	VB NP PP-CLR NP-TMP

Example rules extracted from this single annotation

Penn Treebank

$$
\begin{aligned}
NP &\rightarrow DT\ JJ\ NN \\
NP &\rightarrow DT\ JJ\ NNS \\
NP &\rightarrow DT\ JJ\ NN\ NN \\
NP &\rightarrow DT\ JJ\ JJ\ NN \\
NP &\rightarrow DT\ JJ\ CD\ NNS \\
NP &\rightarrow RB\ DT\ JJ\ NN\ NN \\
NP &\rightarrow RB\ DT\ JJ\ JJ\ NNS \\
NP &\rightarrow DT\ JJ\ JJ\ NNP\ NNS \\
NP &\rightarrow DT\ NNP\ NNP\ NNP\ NNP\ JJ\ NN \\
NP &\rightarrow DT\ JJ\ NNP\ CC\ JJ\ JJ\ NN\ NNS \\
NP &\rightarrow RB\ DT\ JJS\ NN\ NN\ SBAR \\
NP &\rightarrow DT\ VBG\ JJ\ NNP\ NNP\ CC\ NNP \\
NP &\rightarrow DT\ JJ\ NNS\ ,\ NNS\ CC\ NN\ NNS\ NN \\
NP &\rightarrow DT\ JJ\ JJ\ VBG\ NN\ NNP\ NNP\ FW\ NNP \\
NP &\rightarrow NP\ JJ\ ,\ JJ\ \text{``}\ SBAR\ \text{''}\ NNS
\end{aligned}
$$

Jurafsky and Martin 2017

Using CFG

- A basic CFG allows us to check whether a sentence is grammatical in the language it defines
- Binary decision: a sentence is either in the language (a series of productions yields the words we see) or it is not.
- Where would this be useful?

PCFG

- Probabilistic context-free grammar: each production is also associated with a probability.

- This lets us calculate the probability of a parse for a given sentence; for a given parse tree T for sentence S comprised of n rules from R (each A → β):

$$P(T, S) = \prod_{i}^{n} P(\beta \mid A)$$

PCFG

N	Finite set of non-terminal symbols	NP, VP, S
Σ	Finite alphabet of terminal symbols	the, dog, a
R	Set of production rules, each $A \rightarrow \beta$ [p] $p = P(\beta \mid A)$	$S \rightarrow NP\ VP$ Noun → dog
S	Start symbol	

PCFG

$$\sum_{\beta} P(A \rightarrow \beta) = 1$$

(equivalently)

$$\sum_{\beta} P(\beta \mid A) = 1$$

Estimating PCFGs

How do we calculate $P(A \rightarrow \beta)$?

Estimating PCFGs

$$\sum_{\beta} P(\beta \mid A) = \frac{C(A \to \beta)}{\sum_{\gamma} C(A \to \gamma)}$$

(equivalently)

$$\sum_{\beta} P(\beta \mid A) = \frac{C(A \to \beta)}{C(A)}$$

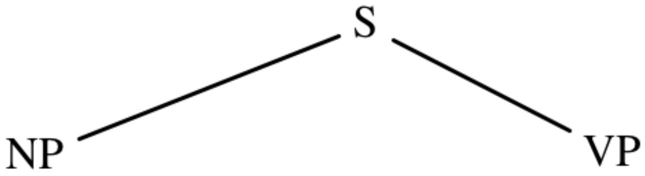
A		β	P(β | NP)
NP	→	NP PP	0.092
NP	→	DT NN	0.087
NP	→	NN	0.047
NP	→	NNS	0.042
NP	→	DT JJ NN	0.035
NP	→	NNP	0.034
NP	→	NNP NNP	0.029
NP	→	JJ NNS	0.027
NP	→	QP -NONE-	0.018
NP	→	NP SBAR	0.017
NP	→	NP PP-LOC	0.017
NP	→	JJ NN	0.015
NP	→	DT NNS	0.014
NP	→	CD	0.014
NP	→	NN NNS	0.013
NP	→	DT NN NN	0.013
NP	→	NP CC NP	0.013

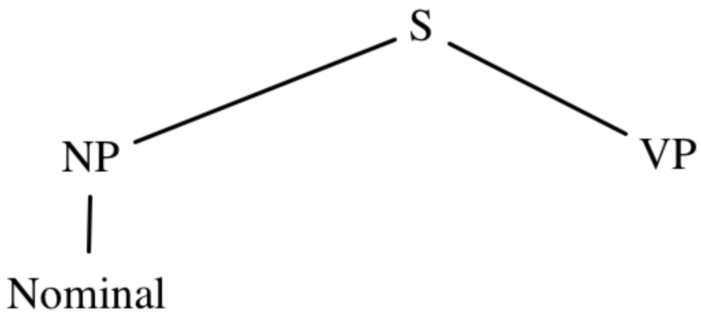
PCFGs

- A CFG tells us whether a sentence is in the language it defines
- A PCFG gives us a mechanism for assigning scores (here, probabilities) to different parses for the same sentence.

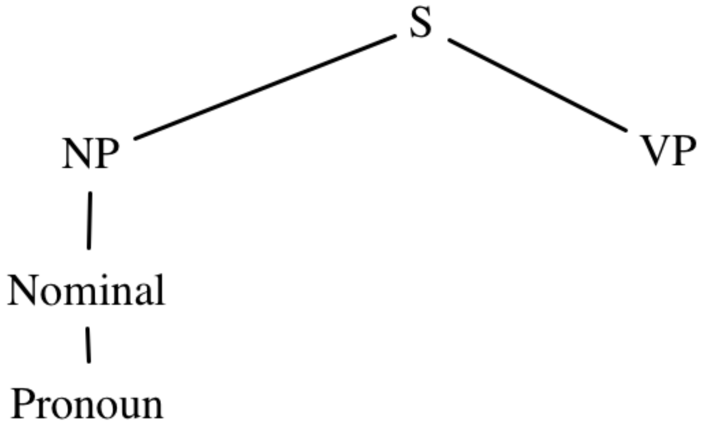
S

$$P(\text{NP VP} \mid \text{S})$$

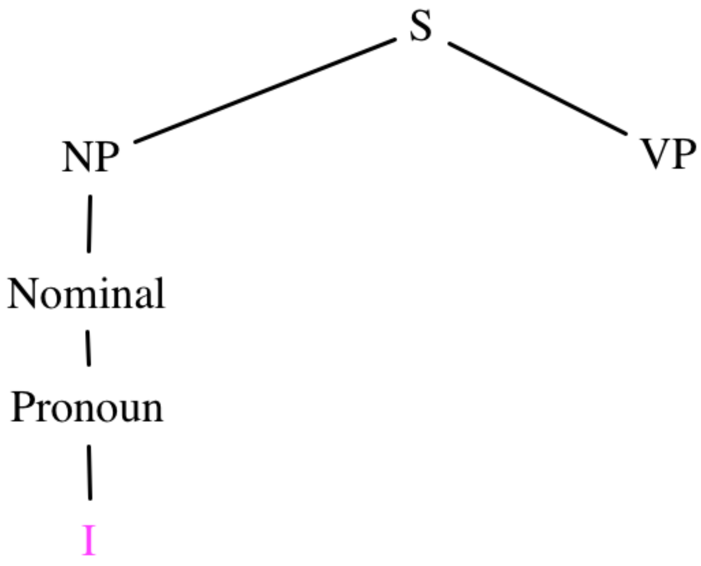




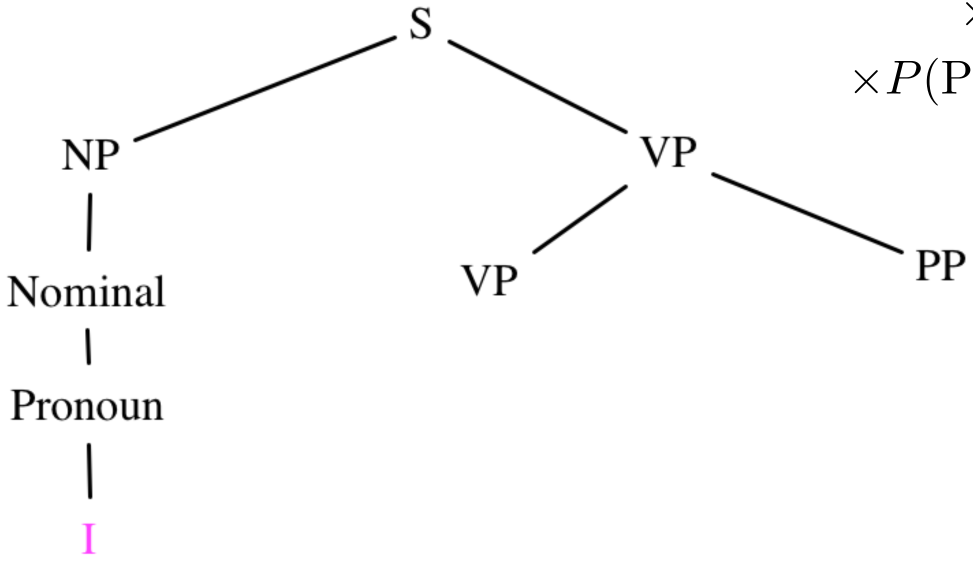
$$P(\text{NP VP} \mid \text{S})$$
$$\times P(\text{Nominal} \mid \text{NP})$$



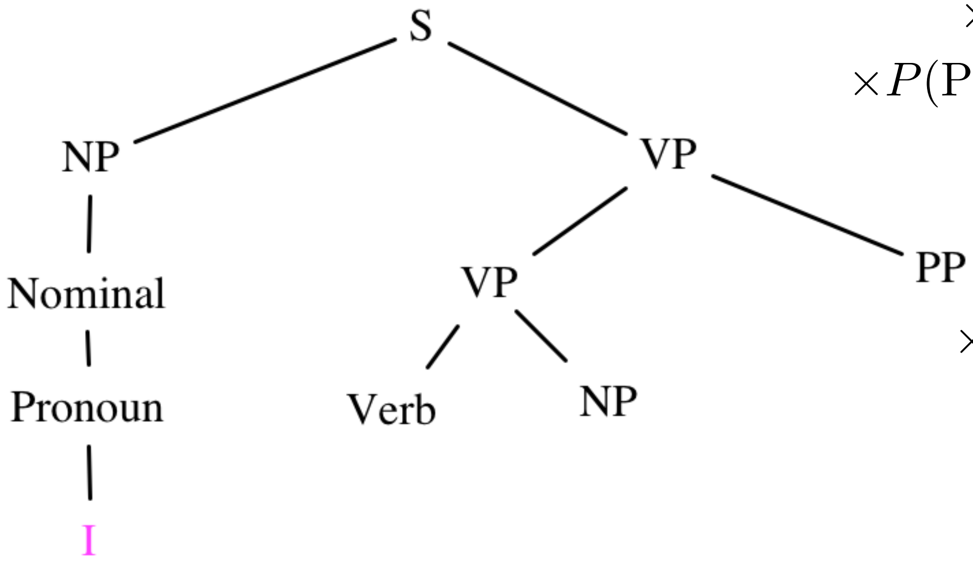
$$P(\text{NP VP} \mid \text{S})$$
$$\times P(\text{Nominal} \mid \text{NP})$$
$$\times P(\text{Pronoun} \mid \text{Nominal})$$



$$P(\text{NP VP} \mid \text{S}) \\ \times P(\text{Nominal} \mid \text{NP}) \\ \times P(\text{Pronoun} \mid \text{Nominal}) \\ \times P(\text{I} \mid \text{Pronoun})$$

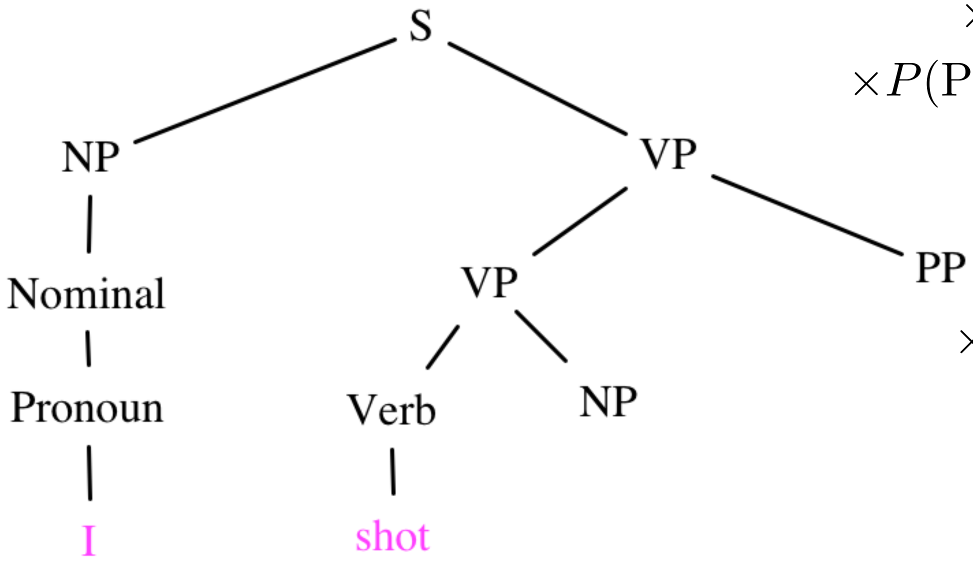


$$P(\text{NP VP} \mid \text{S})$$
$$\times P(\text{Nominal} \mid \text{NP})$$
$$\times P(\text{Pronoun} \mid \text{Nominal})$$
$$\times P(\text{I} \mid \text{Pronoun})$$
$$\times P(\text{VP PP} \mid \text{VP})$$



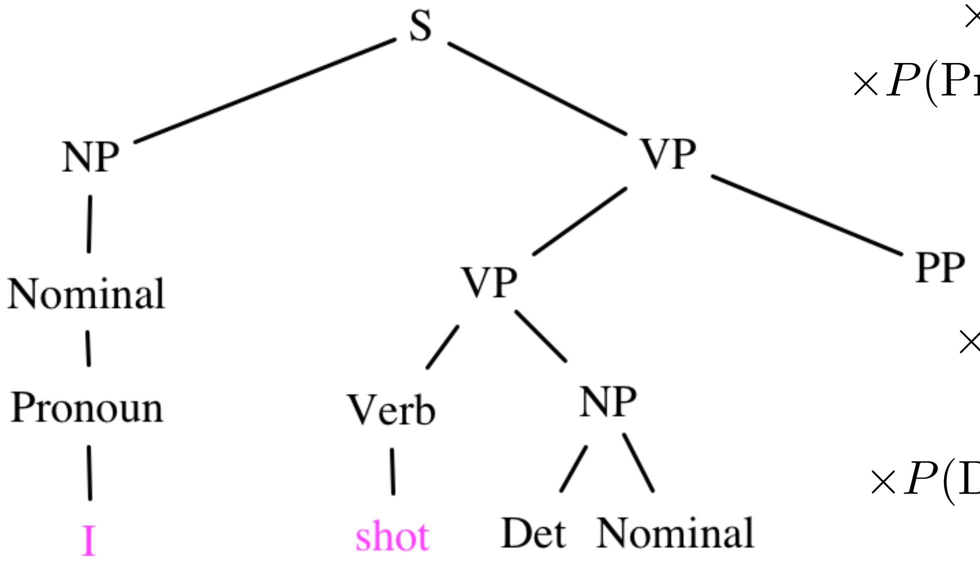
$$P(\text{NP VP} \mid \text{S})$$
$$\times P(\text{Nominal} \mid \text{NP})$$
$$\times P(\text{Pronoun} \mid \text{Nominal})$$
$$\times P(\text{I} \mid \text{Pronoun})$$
$$\times P(\text{VP PP} \mid \text{VP})$$

$$\times P(\text{Verb NP} \mid \text{VP})$$

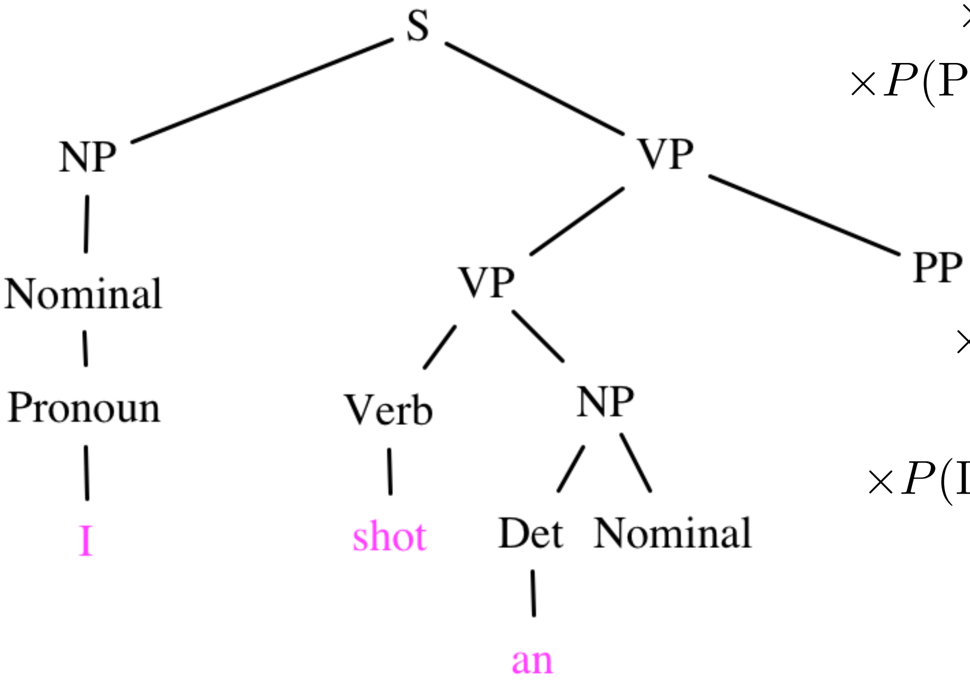


$$P(\text{NP VP} \mid \text{S})$$
$$\times P(\text{Nominal} \mid \text{NP})$$
$$\times P(\text{Pronoun} \mid \text{Nominal})$$
$$\times P(\text{I} \mid \text{Pronoun})$$
$$\times P(\text{VP PP} \mid \text{VP})$$

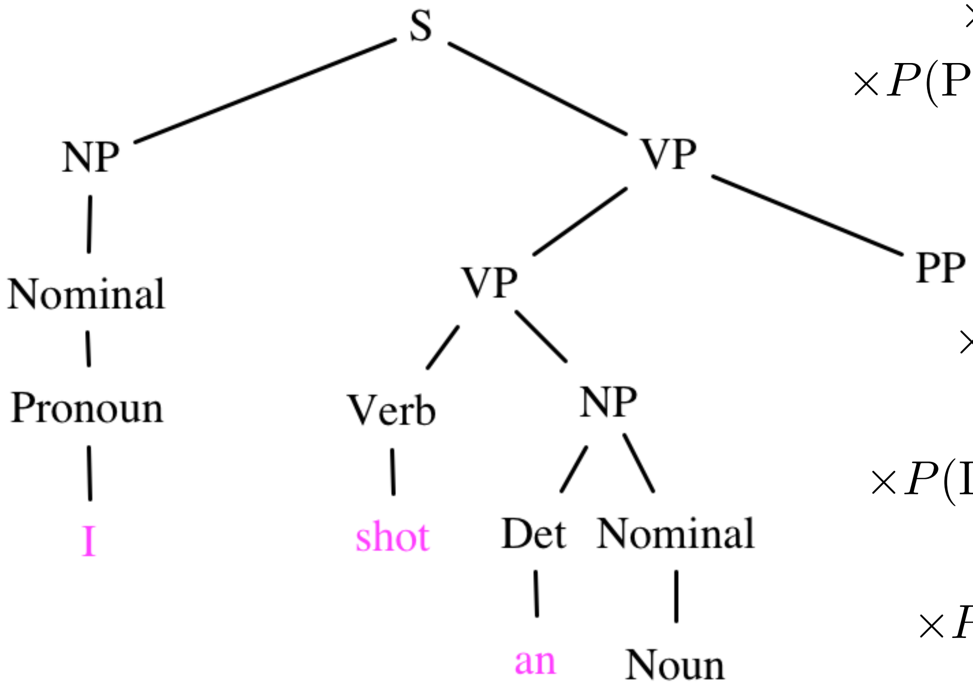
$$\times P(\text{Verb NP} \mid \text{VP})$$
$$\times P(\text{shot} \mid \text{Verb})$$



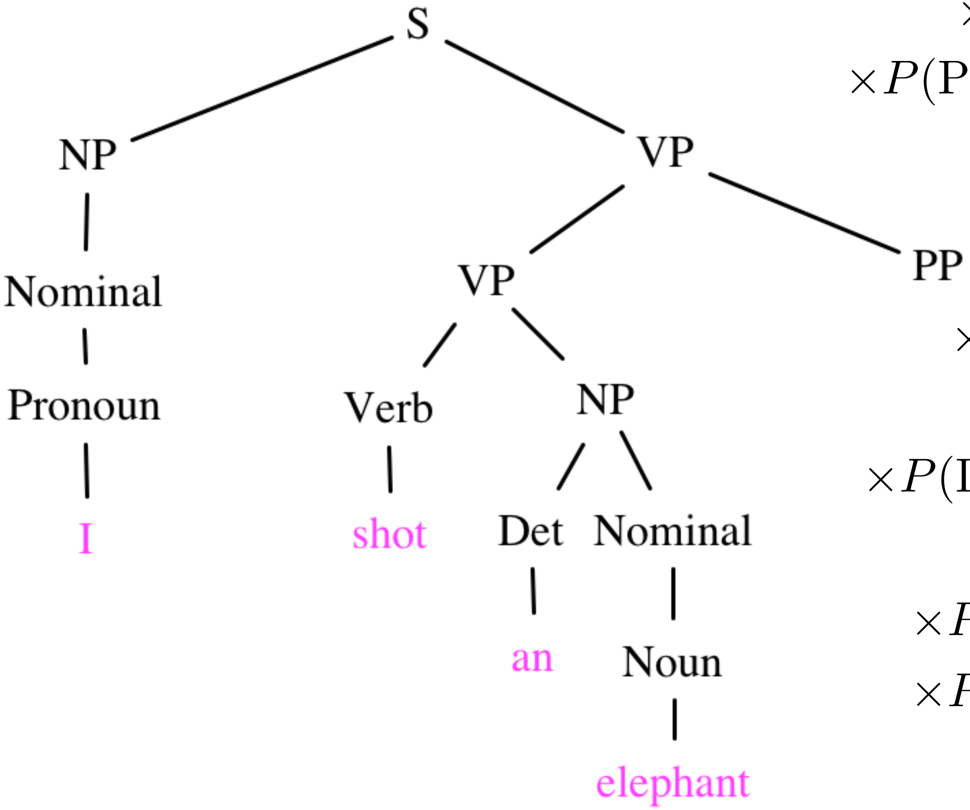
$$P(\text{NP VP} \mid \text{S})$$
$$\times P(\text{Nominal} \mid \text{NP})$$
$$\times P(\text{Pronoun} \mid \text{Nominal})$$
$$\times P(\text{I} \mid \text{Pronoun})$$
$$\times P(\text{VP PP} \mid \text{VP})$$
$$\times P(\text{Verb NP} \mid \text{VP})$$
$$\times P(\text{shot} \mid \text{Verb})$$
$$\times P(\text{Det Nominal} \mid \text{NP})$$



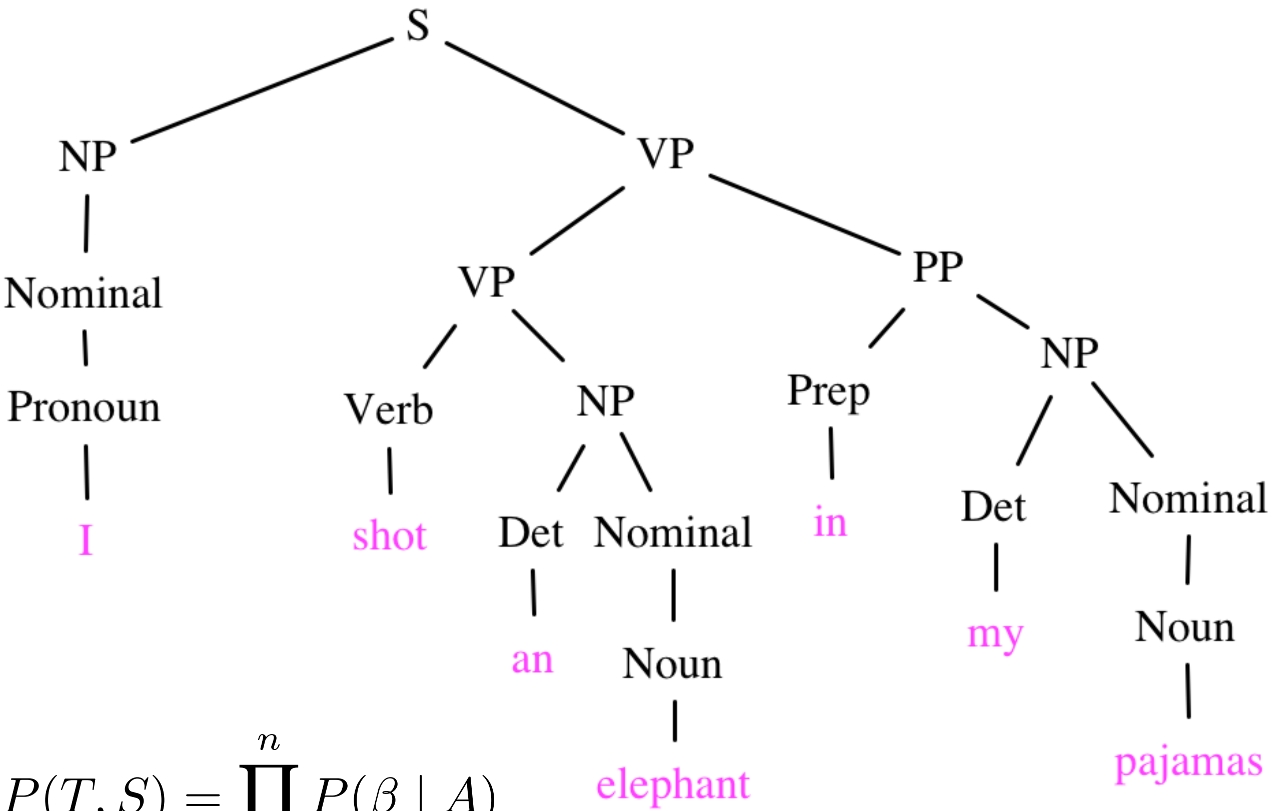
$$P(\text{NP VP} \mid \text{S})$$
$$\times P(\text{Nominal} \mid \text{NP})$$
$$\times P(\text{Pronoun} \mid \text{Nominal})$$
$$\times P(\text{I} \mid \text{Pronoun})$$
$$\times P(\text{VP PP} \mid \text{VP})$$
$$\times P(\text{Verb NP} \mid \text{VP})$$
$$\times P(\text{shot} \mid \text{Verb})$$
$$\times P(\text{Det Nominal} \mid \text{NP})$$
$$\times P(\text{an} \mid \text{Det})$$



$$P(\text{NP VP} \mid \text{S})$$
$$\times P(\text{Nominal} \mid \text{NP})$$
$$\times P(\text{Pronoun} \mid \text{Nominal})$$
$$\times P(\text{I} \mid \text{Pronoun})$$
$$\times P(\text{VP PP} \mid \text{VP})$$
$$\times P(\text{Verb NP} \mid \text{VP})$$
$$\times P(\text{shot} \mid \text{Verb})$$
$$\times P(\text{Det Nominal} \mid \text{NP})$$
$$\times P(\text{an} \mid \text{Det})$$
$$\times P(\text{Noun} \mid \text{Nominal})$$



$$
\begin{aligned}
& P(\text{NP VP} \mid \text{S}) \\
& \times P(\text{Nominal} \mid \text{NP}) \\
& \times P(\text{Pronoun} \mid \text{Nominal}) \\
& \times P(\text{I} \mid \text{Pronoun}) \\
& \times P(\text{VP PP} \mid \text{VP}) \\
& \times P(\text{Verb NP} \mid \text{VP}) \\
& \times P(\text{shot} \mid \text{Verb}) \\
& \times P(\text{Det Nominal} \mid \text{NP}) \\
& \times P(\text{an} \mid \text{Det}) \\
& \times P(\text{Noun} \mid \text{Nominal}) \\
& \times P(\text{elephant} \mid \text{Noun})
\end{aligned}
$$

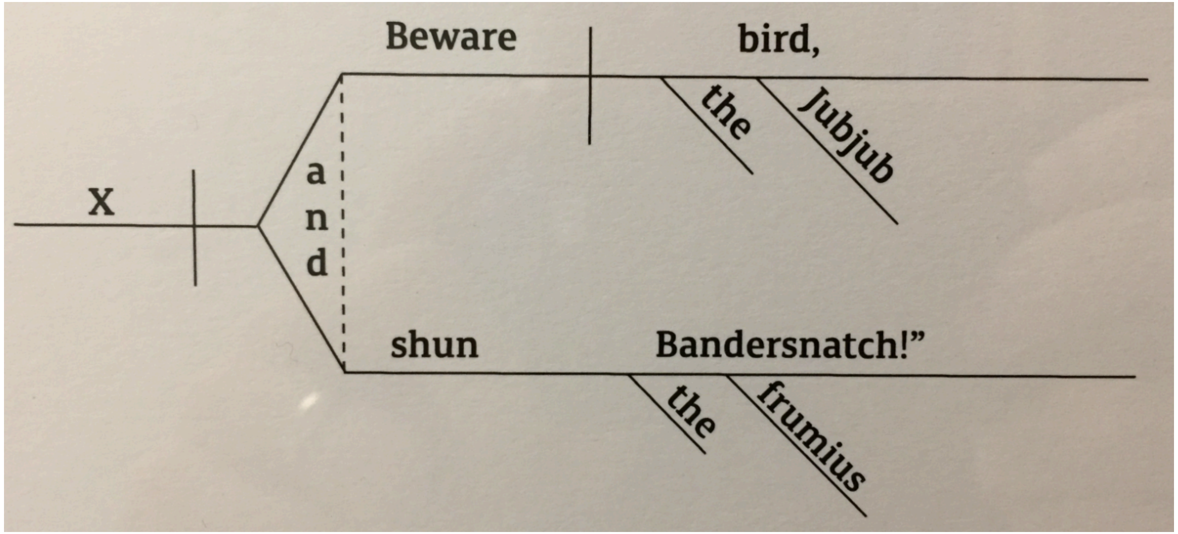


$$P(T, S) = \prod_{i}^{n} P(\beta \mid A)$$

Dependency syntax

- Sgall, Dependency-based formal description of language (1994)
- Mel’čuk, Dependency Syntax: Theory and Practice (1988)
- Tesnière, Éléments de syntaxe structurale (1959)
- Medieval theories of grammar (Covington 1984)
- Pānini grammar of Sanskrit (ca. 5th-century BCE)

Dependency syntax



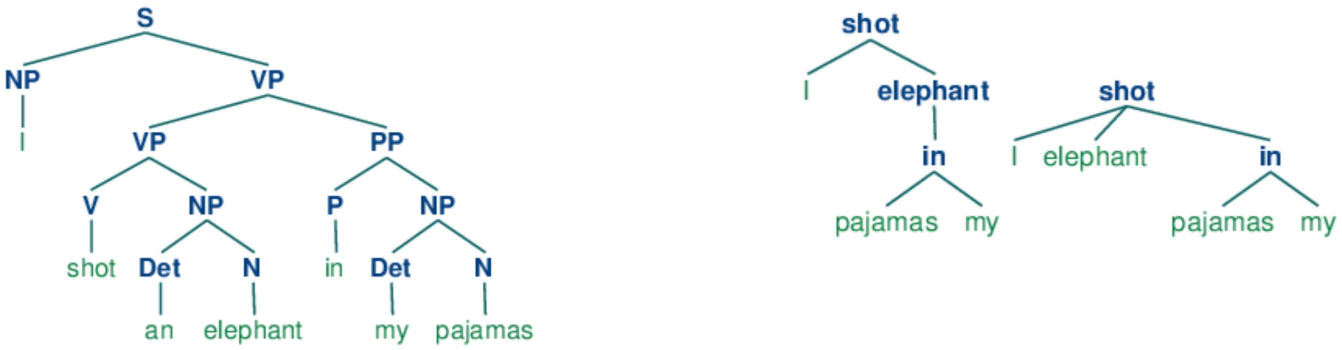
"Sentence diagramming"

Dependency syntax

- “Between the word and its neighbors, the mind perceives connections, the totality of which forms the structure of the sentence. The structural connections establish dependency relations between the words. Each connection in principle unites a superior and an inferior term.”

Tesnier 1959; Nivre 2005

Dependency syntax

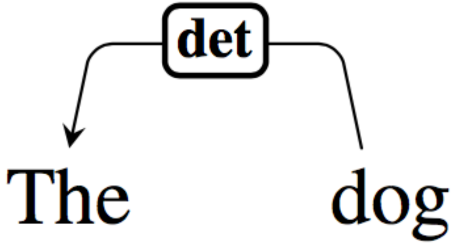


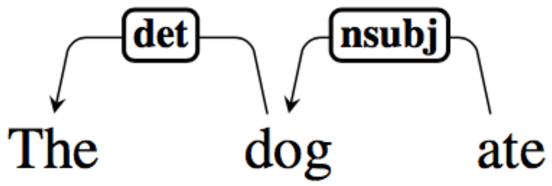
- Dependency syntax doesn't have non-terminal structure like a CFG; words are directly linked to each other.

Dependency syntax

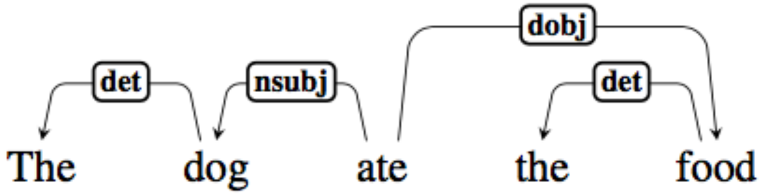
- Syntactic structure = asymmetric, binary relations between words.

Tesnier 1959; Nivre 2005


det
The
dog



det
nsubj
The
dog
ate


dobj
det
nsubj
det
The
dog
ate
the
food

Trees

- A dependency structure is a directed graph G = (V,A) consisting of a set of vertices *V* and arcs *A* between them. Typically constrained to form a tree:
  - Single root vertex with no incoming arcs
  - Every vertex has exactly one incoming arc except root (single head constraint)
  - There is a unique path from the root to each vertex in V (acyclic constraint)

Trees

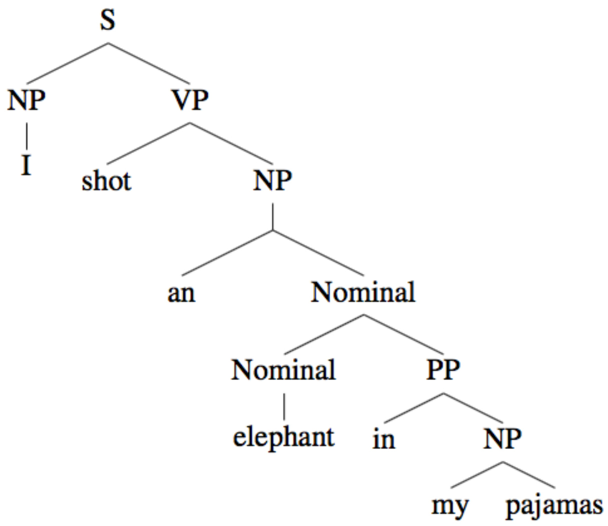
- Unlike phrase-structure trees, dependency trees aren’t tied to the linear order of the words in a sentence.

- Adding a constraint derived from the linear order of words in a sentence allows for more efficient parsing algorithms (as we’ll see next class).

Word order

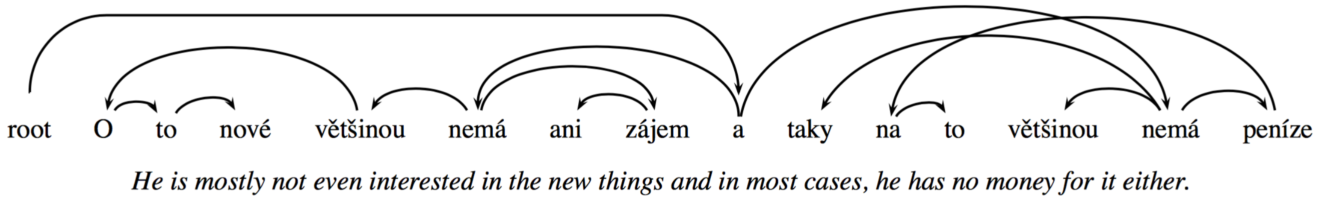
- Dependency relations belong to the structural order of a sentence, not the linear order.

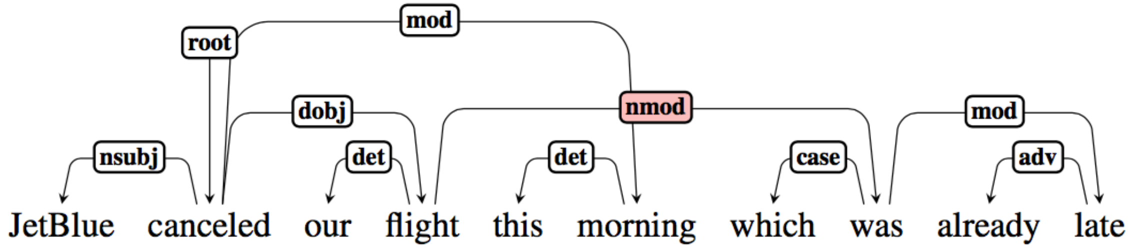
- This is different from a phrase-structure tree, where the syntax is constrained by the linear order of the sentence (a different linear order yields a different parse tree).



Tesnière 1959

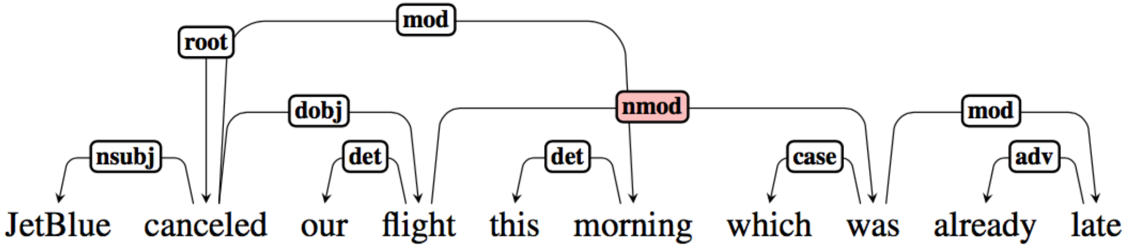
Free word order





Projectivity

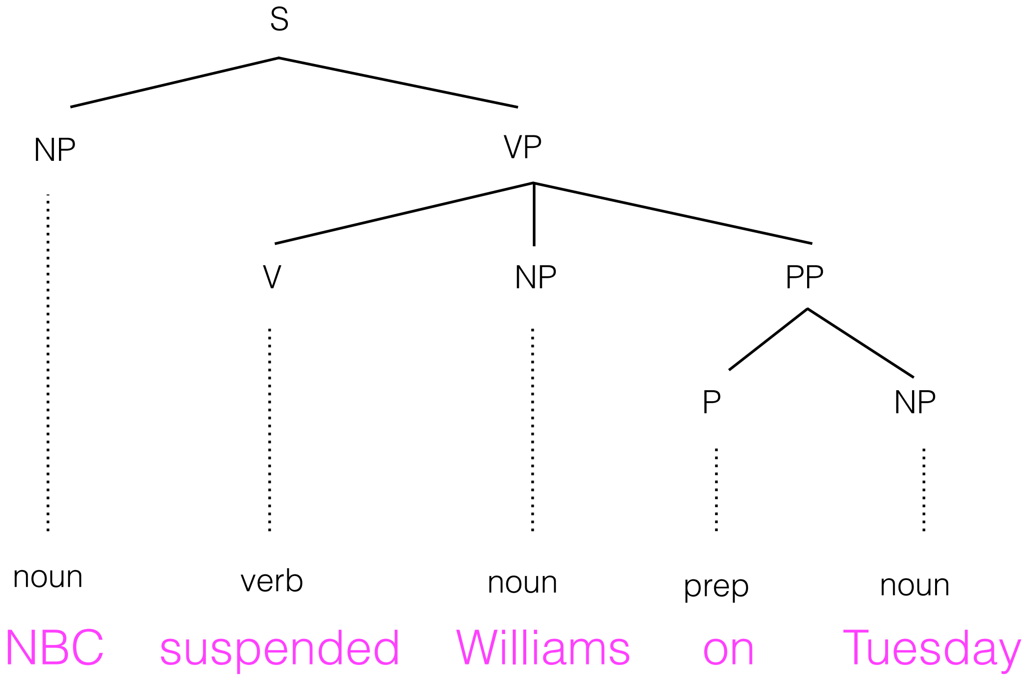
- An arc between a head and dependent is projective if there is a path from the head to every word between the head and dependent.



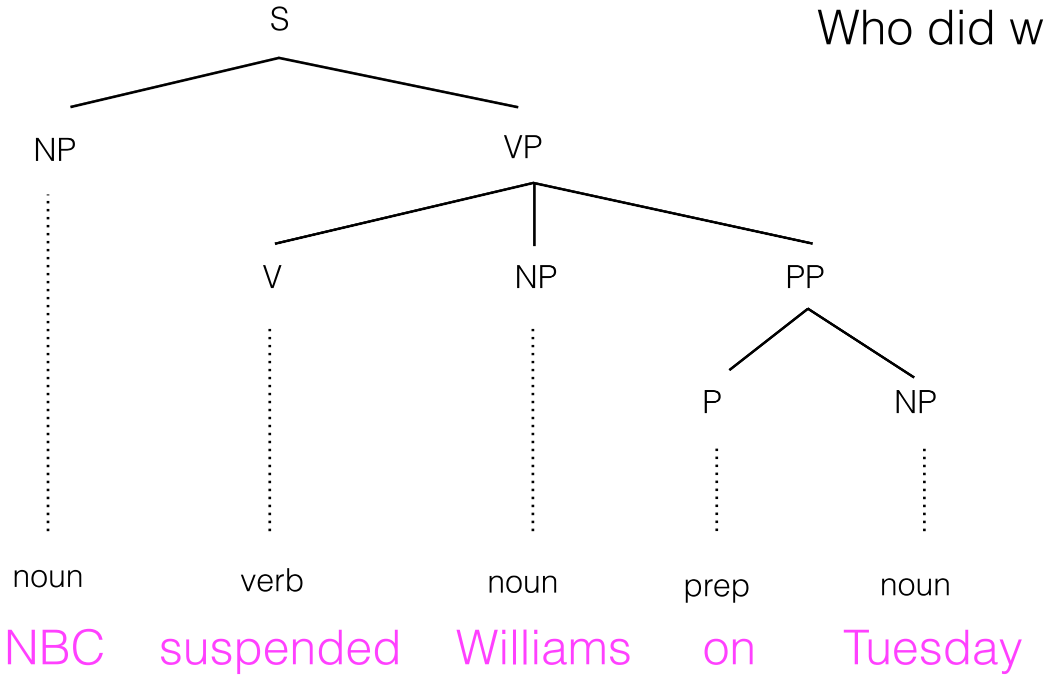
Dependencies vs constituents

- Dependency links are closer to semantic relationships; no need to infer the relationships from the structure of a tree
- A dependency tree contains one edge for each word, no intermediate hidden structures that also need to be learned for parsing.
- Easier to represent languages with free word order.

Covington 2001

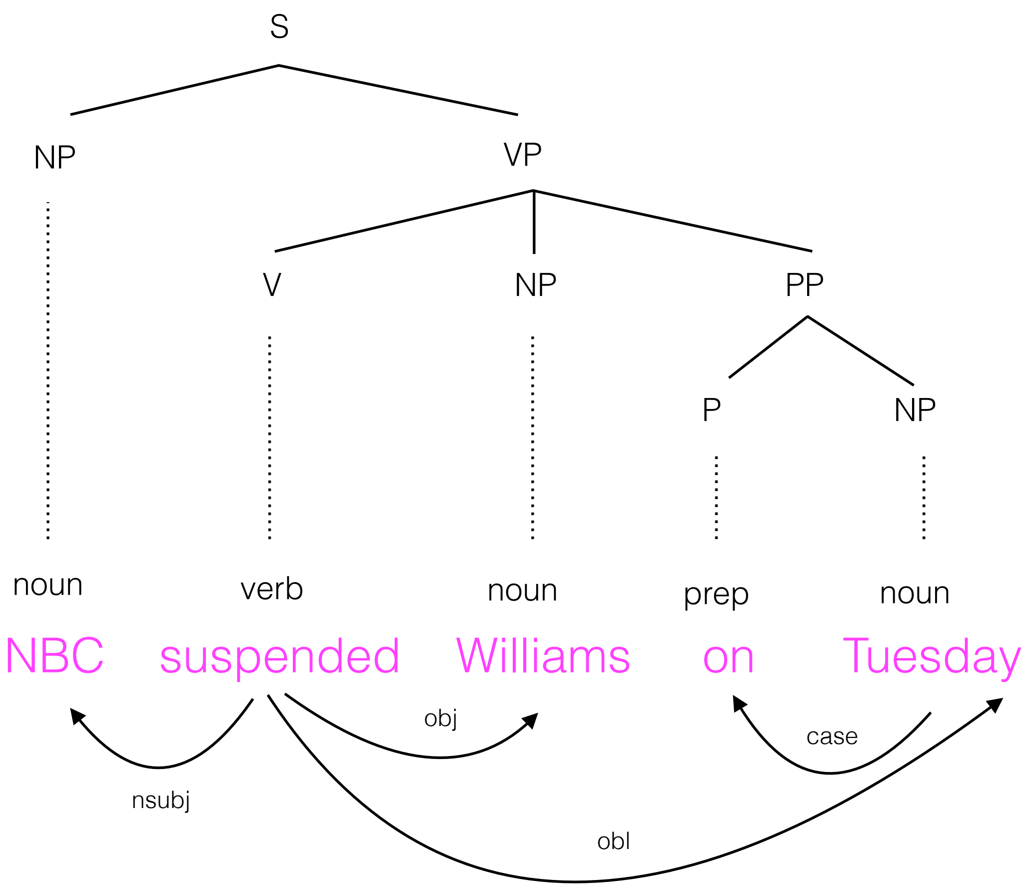

S
NP
VP
V
NP
PP
P
NP
noun
verb
noun
prep
noun
NBC
suspended
Williams
on
Tuesday

Who did what to whom?



subject: S → NP VP

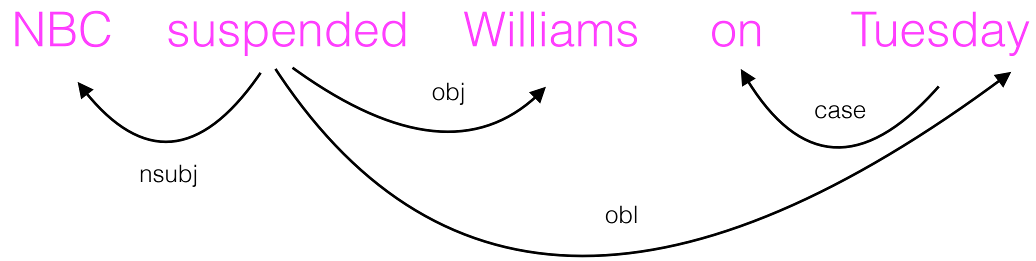
direct object: S → NP (VP → … NP …)


S
NP
VP
V
NP
PP
P
NP
noun
verb
noun
prep
noun
NBC
suspended
Williams
on
Tuesday
obj
case
nsubj
obl

Dependency grammar

Captures binary relations between words

- nsubj(NBC, suspended)
- obj(Williams, suspended)



Universal Dependencies

UD Treebanks

Afrikaans	49K	ⓁⒻ	–				
Ancient Greek	202K	ⓁⒻ			✓		
Ancient Greek-PROIEL	211K	ⓁⒻ	–		✓		
Arabic	242K	ⓁⒻ	–		✓		
Arabic-NYUAD	629K	ⓁⒻ	–		✓		
Arabic-PUD	20K	ⓁⒻ	–				W
Basque	121K	ⓁⒻ			✓		
Belarusian	8K	ⓁⒻ	–		✓		
Bulgarian	156K	ⓁⒻ		✓	✓		
Buryat	10K	ⓁⒻ	–				
Catalan	530K	ⓁⒻ		✓	✓	GPL	
Chinese	123K	ⓁⒻ		✓	✓		W
Chinese-CFL	7K	Ⓛ					
Chinese-PUD	21K	Ⓕ	–				W
Coptic	4K	ⓁⒻ			✓		
Croatian	197K	ⓁⒻ	–	✓	✓		W
Czech	1,503K	ⓁⒻ		✓	✓		

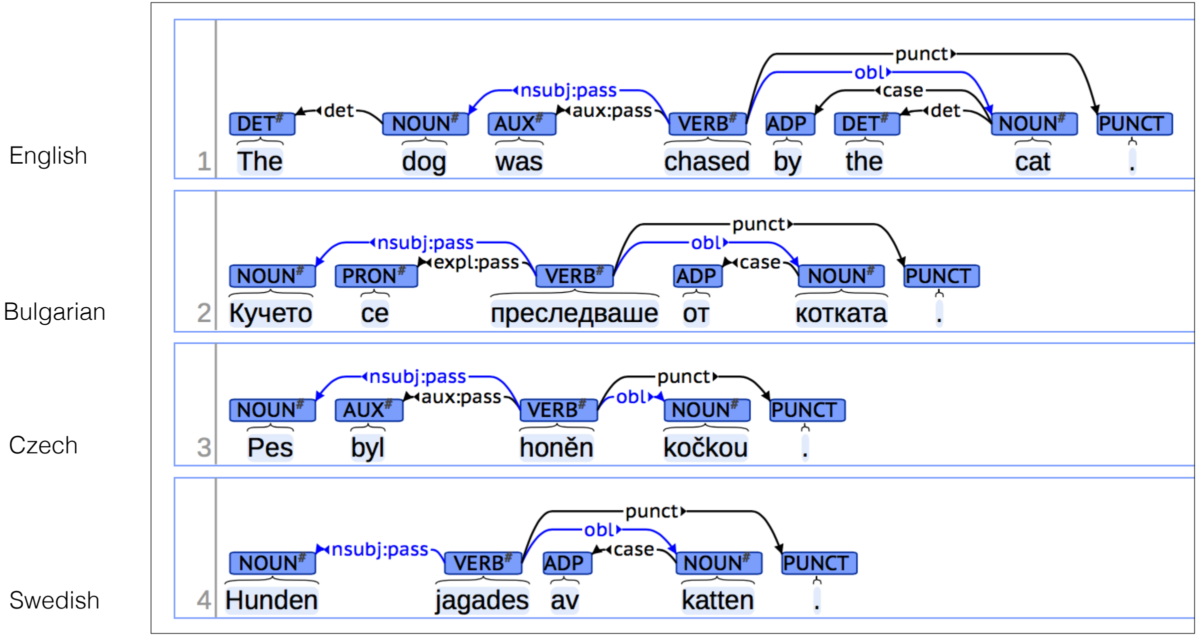
http://universaldependencies.org

Universal Dependencies

- Developing cross-linguistically consistent treebank annotation for many languages
- Goals:
  - Facilitating multilingual parser development
  - Cross-lingual learning
  - Parsing research from a language typology perspective.

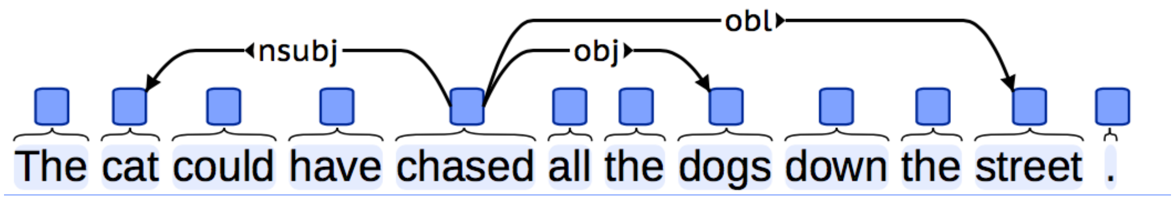
http://universaldependencies.org

Universal Dependencies



http://universaldependencies.org

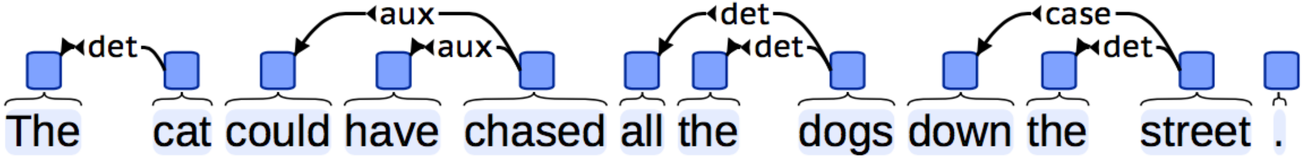
UD Principles



Dependency relations mainly hold between content words.

http://universaldependencies.org/u/overview/syntax.html

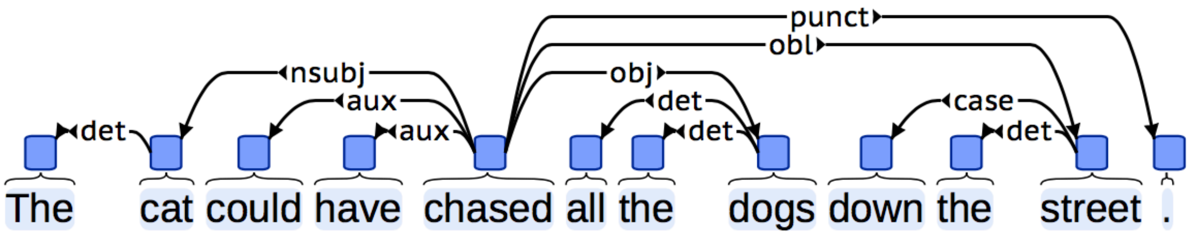
UD Principles



Function words dependent on closest related content word

http://universaldependencies.org/u/overview/syntax.html

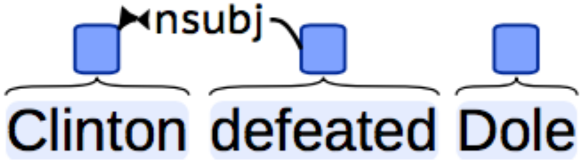
UD Principles



http://universaldependencies.org/u/overview/syntax.html

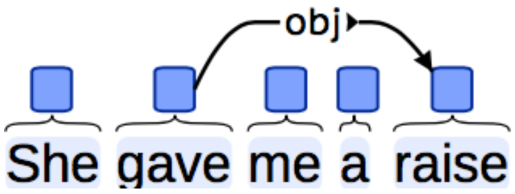
nsubj

- Syntactic subject of active verbs



obj

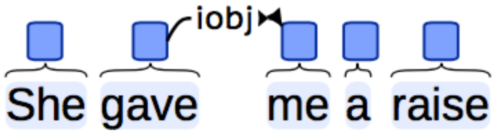
- Generally, the entity that is acted upon as the direct object of the predicate.



- Note the term for a direct object in older versions of Universal Dependencies (e.g., described in SLP3) is "dobj".

iobj

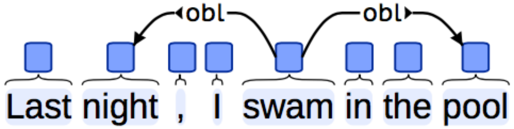
- Indirect object: recipients of ditransitive verbs of exchange (verbs requiring two objects)

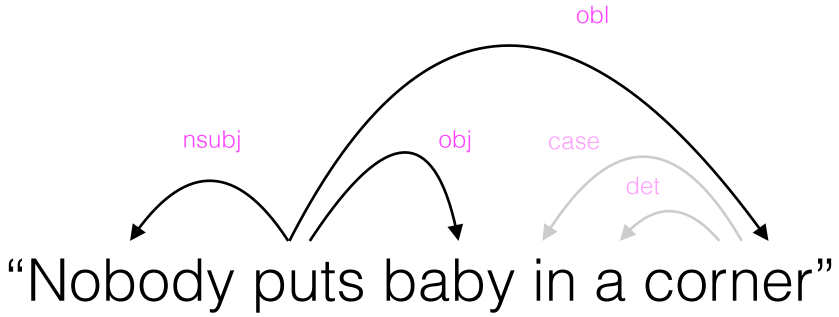


nsubj		iobj	obj
She	teaches	her daughters	math
She	told	her daughters	a story

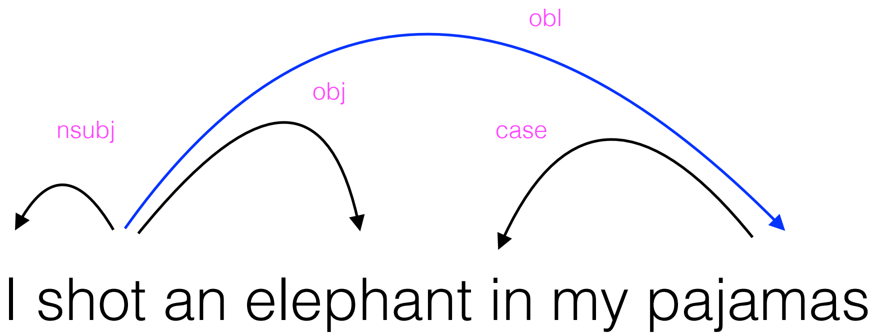
obl

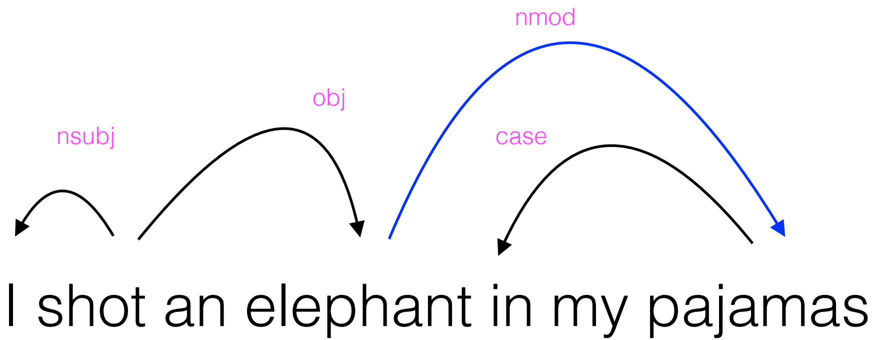
- Any nominal functioning as non-required argument or adjunct of a verb, including temporal and locational nominal modifiers and agents of passive verbs





[*Dirty Dancing*]


obl
obj
nsubj
case
I shot an elephant in my pajamas


nmod
obj
nsubj
case
I shot an elephant in my pajamas

Heads up: Summer NLP Course

- **Social Aspects of NLP**
- Instructor: Lucy Li
- https://classes.berkeley.edu/content/2024-summer-info-290-001-lec-001

Logistics

- Quiz 5 is due tonight (Mon 3/4).
- 259 Project Proposals due Tues. 3/5.
- Homework 4 is due this Friday 3/8 (start if haven’t already)
- This week: Syntax & Parsing