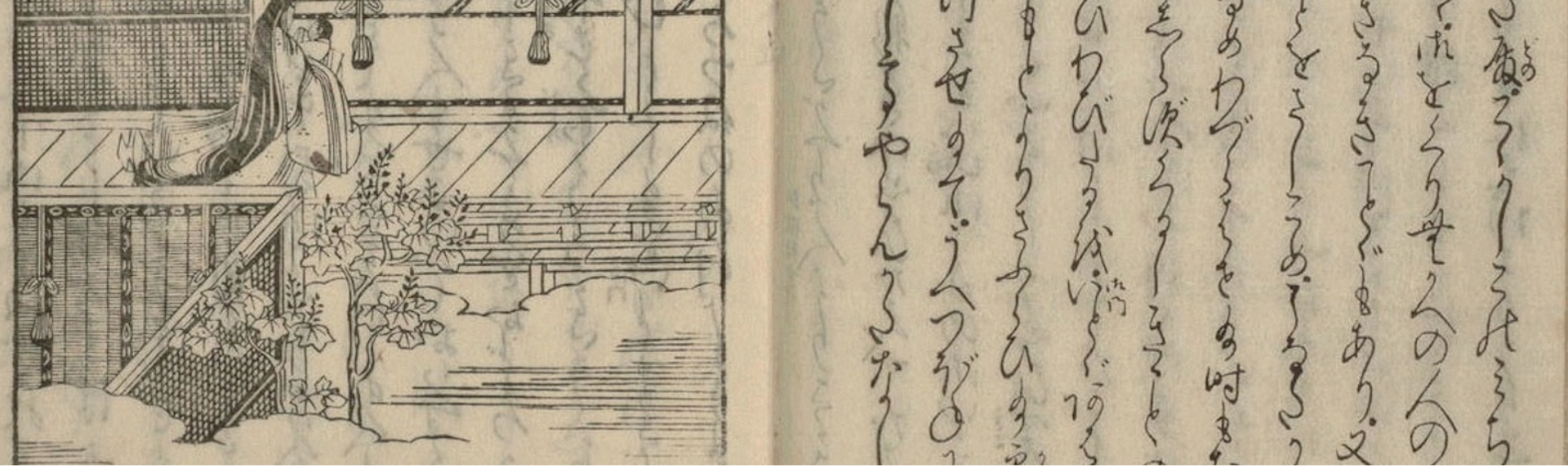


# Natural Language Processing

Info 159/259

Lecture 13: Constituency syntax (March 2, 2021)

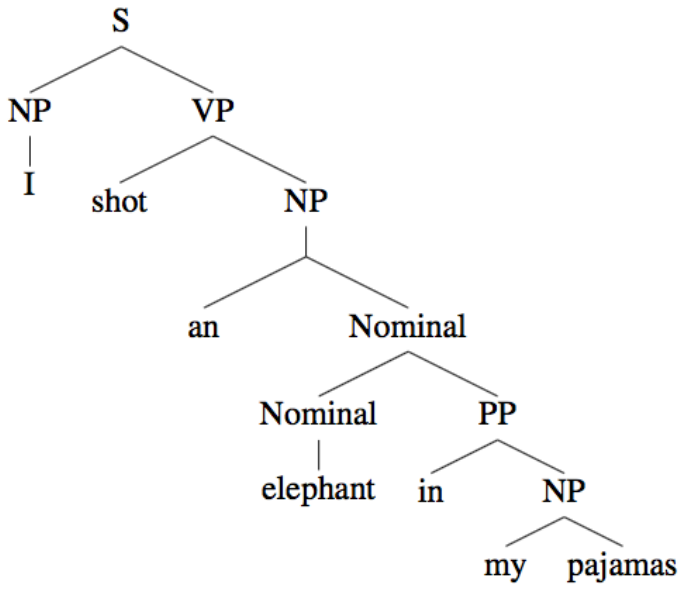
David Bamman, UC Berkeley

# 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 |
|---|---|---|---|---|---|---|

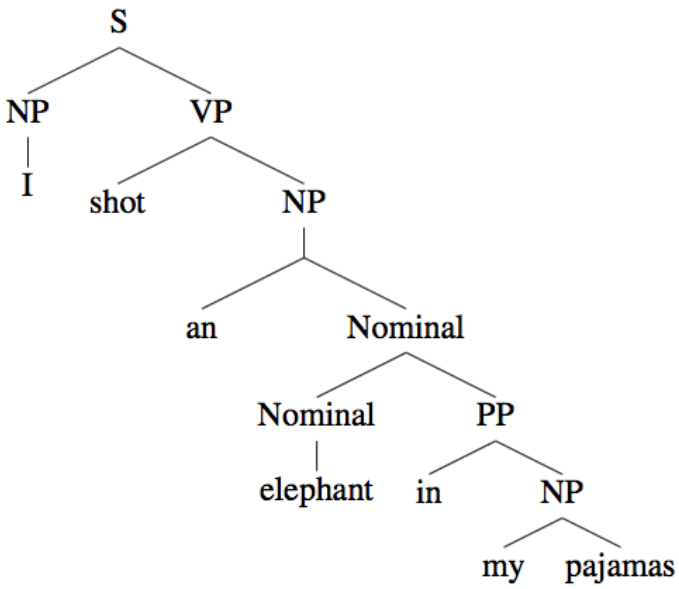
I shot an elephant in my pajamas




PRP VBD DT NN IN PRP$ NNS

I shot an elephant in my pajamas

# Why is syntax important?

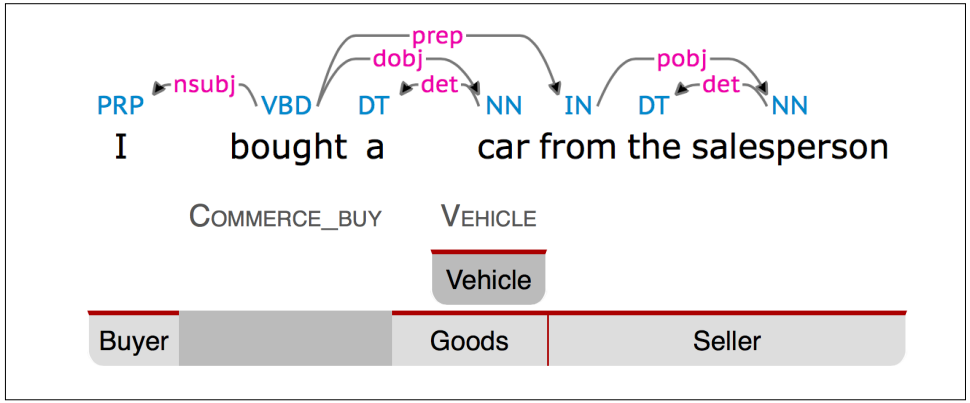


# Why is POS important?

- 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 |
| … | … | … |

# Sentiment analysis

"Unfortunately I already had this exact picture tattooed on my chest, but this shirt is very useful in colder weather."
[overlook1977]

# Question answering

What did Barack Obama teach?

**Barack Hussein Obama II** (born August 4, 1961) is the 44th and current President of the United States, and the first African American to hold the office. Born in Honolulu, Hawaii, Obama is a graduate of Columbia University and Harvard Law School, where he served as president of the *Harvard Law Review*. He was a community organizer in Chicago before earning his law degree. He worked as a civil rights attorney and taught constitutional law at the University of Chicago Law School between 1992 and 2004.

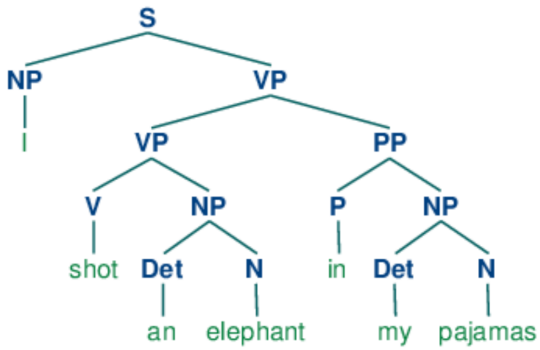
# Syntax

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

words → phrases → clauses → sentences

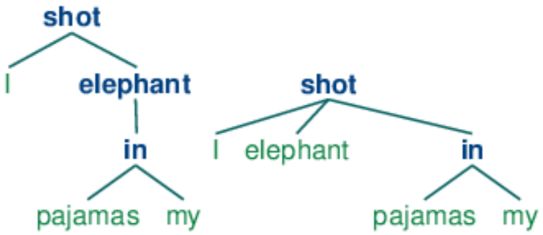
# Formalisms

## Phrase structure grammar
(Chomsky 1957)




today

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




Mar 16

# Constituency

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

context

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

from POS 2/18

# Parts of speech

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

from POS 2/18

# Syntactic distribution

- Substitution test: 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

from POS 2/18

# 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

grammatical only when the entire phrase is present, not an individual word in isolation

| | |
|---|---|
| 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

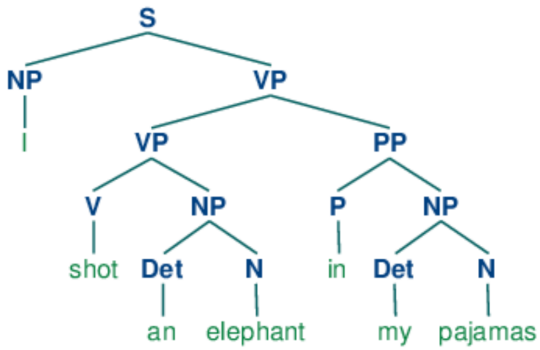
I'd like to fly from Atlanta to Denver

^ ^ ^ ^

on September seventeenth

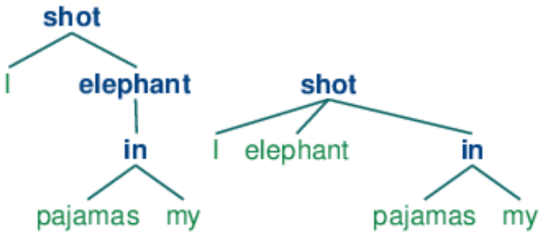
# Formalisms

## Phrase structure grammar
(Chomsky 1957)




today

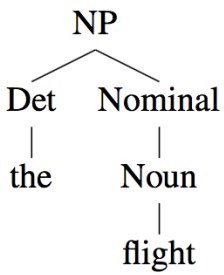
## Dependency grammar
(Mel'čuk 1988; Tesnière 1959; Pāṇini)



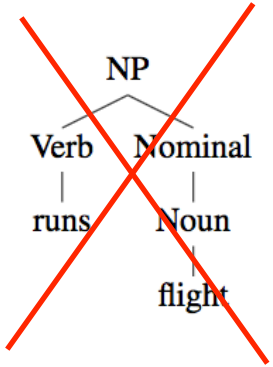

Mar 16

# 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 |
| $\Sigma$ | Finite alphabet of terminal symbols | the, dog, a |
| $R$ | Set of production rules, each<br>$A \rightarrow \beta$<br>$\beta \in (\Sigma, N)$ | S → NP VP<br>Noun → dog |
| $S$ | Start symbol | |

# Infinite strings with finite productions

Some sentences go on

Bender 2016

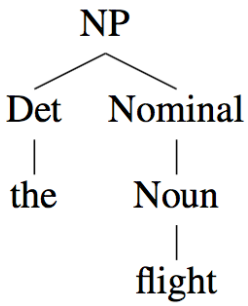
# Infinite strings with finite productions

- This is the house
- This is the house that Jack built
- This is the cat that lives in the house that Jack built
- This is the dog that chased the cat that lives in the house that Jack built
- This is the flea that bit the dog that chased the cat that lives in the house the Jack built
- This is the virus that infected the flea that bit the dog that chased the cat that lives in the house that Jack built

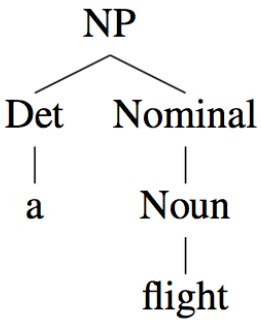
Smith 2017

# Derivation

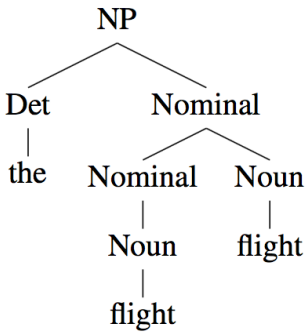
Given a CFG, a derivation is the sequence of productions used to generate a string of words (e.g., a sentence), often visualized as a parse tree.




the flight




a flight




the flight flight

# Language

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

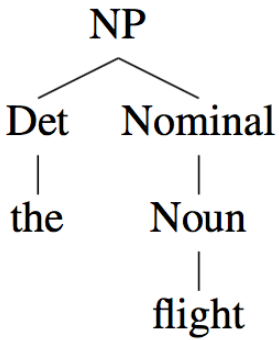
| | | |
|---|---|---|
| *Noun* | → | *flights* \| *breeze* \| *trip* \| *morning* |
| *Verb* | → | *is* \| *prefer* \| *like* \| *need* \| *want* \| *fly* |
| *Adjective* | → | *cheapest* \| *non-stop* \| *first* \| *latest* |
| | | \| *other* \| *direct* |
| *Pronoun* | → | *me* \| *I* \| *you* \| *it* |
| *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 11.2** The lexicon for $\mathscr{L}_0$.

| Grammar Rules | | | Examples |
|---|---|---|---|
| *S* | → | *NP VP* | I + want a morning flight |
| *NP* | → | *Pronoun* | I |
| | \| | *Proper-Noun* | Los Angeles |
| | \| | *Det Nominal* | a + flight |
| *Nominal* | → | *Nominal Noun* | morning + flight |
| | \| | *Noun* | flights |
| *VP* | → | *Verb* | do |
| | \| | *Verb NP* | want + a flight |
| | \| | *Verb NP PP* | leave + Boston + in the morning |
| | \| | *Verb PP* | leaving + on Thursday |
| *PP* | → | *Preposition NP* | from + Los Angeles |

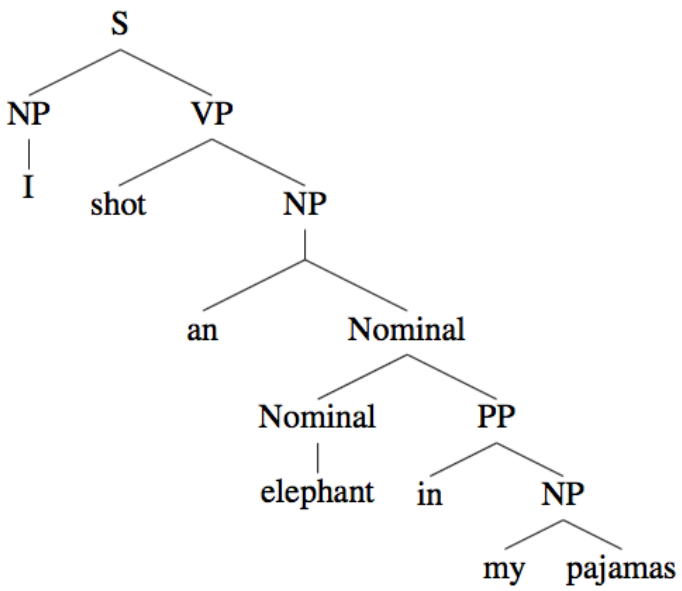
**Figure 11.3** The grammar for $\mathscr{L}_0$, with example phrases for each rule.

# Bracketed notation




[NP [Det the] [Nominal [Noun flight]]]

# Constituents




*Every* internal node is a phrase

- my pajamas
- in my pajamas
- elephant in my pajamas
- an elephant in my pajamas
- shot an elephant in my pajamas
- I shot an elephant in my pajamas

Each phrase could be replaced by another of the same type of constituent

| Production | Meaning | Example |
|---|---|---|
| S → VP | Imperatives | “Show me the right way” |
| S → NP VP | Declaratives | “The dog barks” |
| S → Aux NP VP | Yes/no questions | “Will you show me the right way?” |
| S → Wh-NP VP | Wh-subject question | “Which flights serve breakfast?” |

# 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

# Subcategorization

- Verbs are compatible with different complements
    - Transitive verbs take direct object NP (“I filled the tank”)
    - Intransitive verbs don’t (“I exist”)

# Subcategorization

- The set of possible complements of a verb is its subcategorization frame.

| | | | |
|---|---|---|---|
| VP | → | Verb VP | * I fill [$_{VP}$ to fly today] |
| VP | → | Verb VP | I want [$_{VP}$ to fly today] |

# Coordination

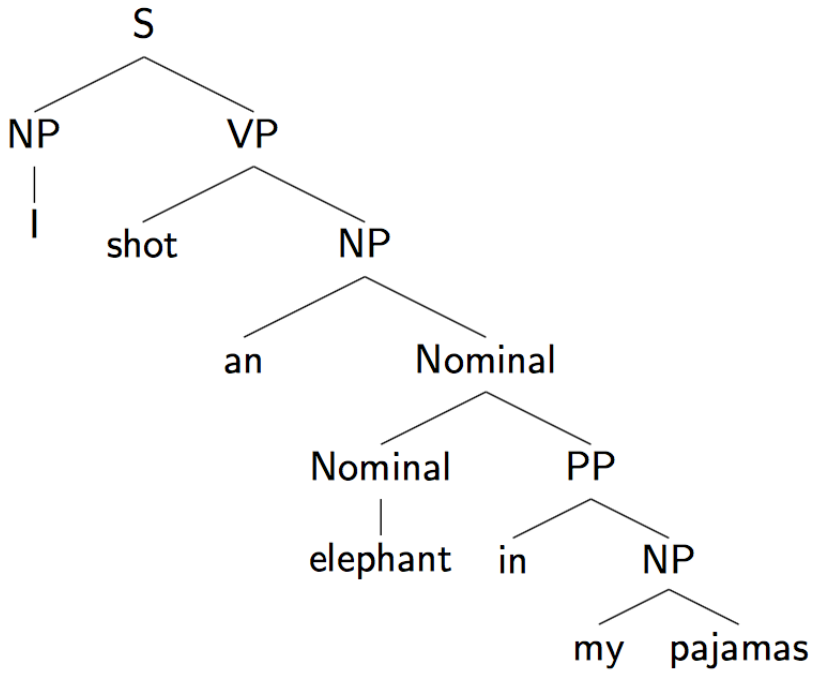
| | | | |
|---|---|---|---|
| NP | → | NP and NP | the dogs and the cats |
| Nominal | → | Nominal and Nominal | dogs and cats |
| VP | → | VP and VP | I came and saw and conquered |
| JJ | → | JJ and JJ | beautiful and red |
| S | → | S and S | I came and I saw and I conquered |

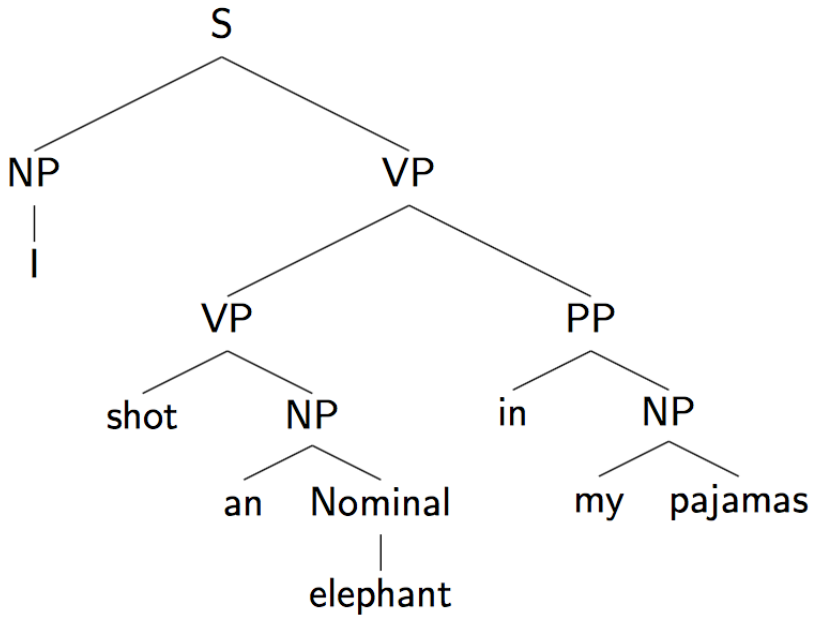
Coordination here also helps us establish whether a group of words forms a constituent

| | | |
|---|---|---|
| S | → | NP VP |
| VP | → | Verb NP |
| VP | → | VP PP |
| Nominal | → | Nominal PP |
| Nominal | → | Noun |
| Nominal | → | Pronoun |
| PP | → | Prep NP |
| NP | → | Det Nominal |
| NP | → | Nominal |
| NP | → | PossPronoun Nominal |

| | | |
|---|---|---|
| Verb | → | shot |
| Det | → | an \| my |
| Noun | → | pajamas \| elephant |
| Pronoun | → | I |
| PossPronoun | → | my |

I shot an elephant in my pajamas


S
NP
VP
I
shot
NP
an
Nominal
Nominal
PP
elephant
in
NP
my
pajamas

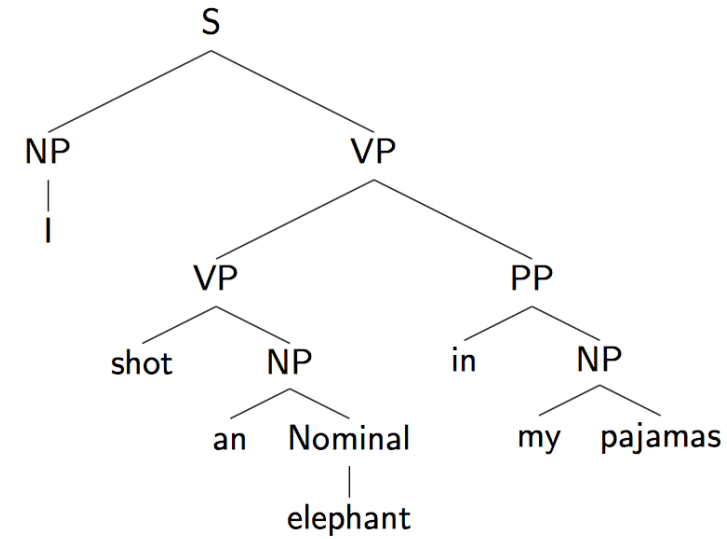

S
NP
VP
I
VP
PP
shot
NP
in
NP
an
Nominal
my
pajamas
elephant

# Evaluation

Parseval (1991): Represent each tree as a collection of tuples:

$<l_1, i_1, j_1>, \ldots, <l_n, i_n, j_n>$

- $l_k$ = label for kth phrase
- $i_k$ = index for first word in kth phrase
- $j_k$ = index for last word in kth phrase

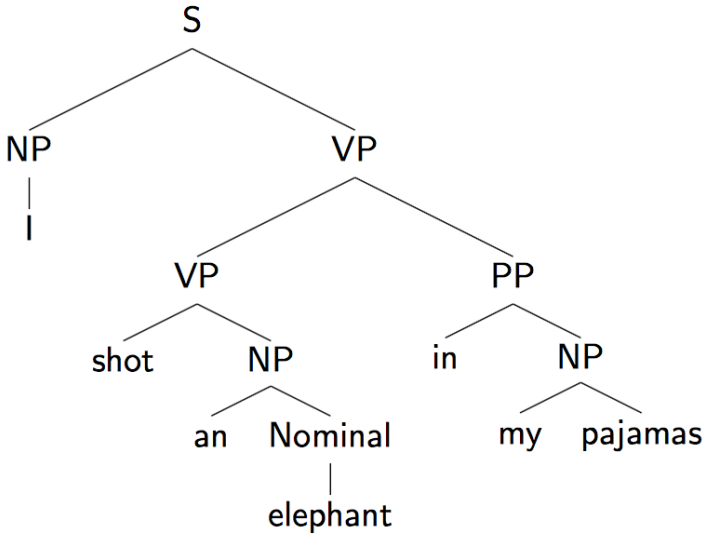



Smith 2017

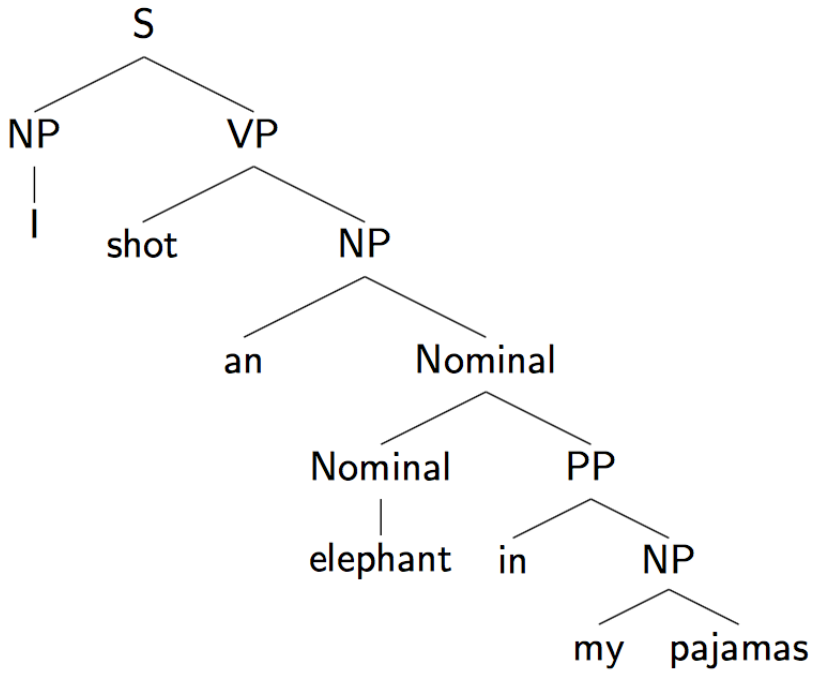
# Evaluation

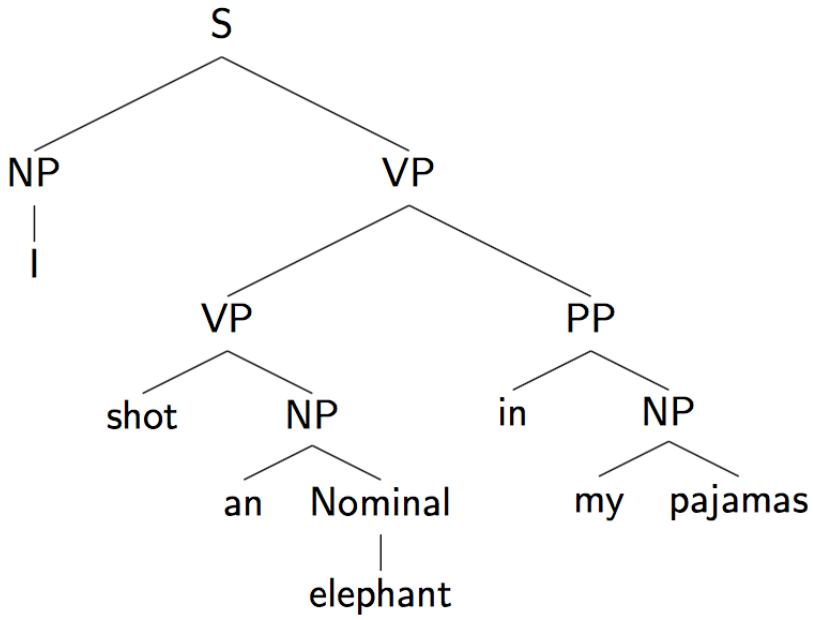
$I_1$ $shot_2$ $an_3$ $elephant_4$ $in_5$ $my_6$ $pajamas_7$

- <S, 1, 7>
- <NP, 1,1>
- <VP, 2, 7>
- <VP, 2, 4>
- <NP, 3, 4>
- <Nominal, 4, 4>
- <PP, 5, 7>
- <NP, 6, 7>




Smith 2017


S
NP
VP
I
shot
NP
an
Nominal
Nominal
PP
elephant
in
NP
my
pajamas


S
NP
VP
I
VP
PP
shot
NP
in
NP
an
Nominal
my
pajamas
elephant

# Evaluation

$I_1$ $shot_2$ $an_3$ $elephant_4$ $in_5$ $my_6$ $pajamas_7$

- <S, 1, 7>
- <NP, 1,1>
- <VP, 2, 7>
- <VP, 2, 4>
- <NP, 3, 4>
- <Nominal, 4, 4>
- <PP, 5, 7>
- <NP, 6, 7>

- <S, 1, 7>
- <NP, 1,1>
- <VP, 2, 7>
- <NP, 3, 7>
- <Nominal, 4, 7>
- <Nominal, 4, 4>
- <PP, 5, 7>
- <NP, 6, 7>

Smith 2017

# Evaluation

Calculate precision, recall, F1 from these collections of tuples

- Precision: number of tuples in tree 1 also in tree 2, divided by number of tuples in tree 1
- Recall: number of tuples in tree 1 also in tree 2, divided by number of tuples in tree 2

Smith 2017

# Evaluation

$I_1$ $shot_2$ $an_3$ $elephant_4$ $in_5$ $my_6$ $pajamas_7$

- <S, 1, 7>
- <NP, 1,1>
- <VP, 2, 7>
- <VP, 2, 4>
- <NP, 3, 4>
- <Nominal, 4, 4>
- <PP, 5, 7>
- <NP, 6, 7>

- <S, 1, 7>
- <NP, 1,1>
- <VP, 2, 7>
- <NP, 3, 7>
- <Nominal, 4, 7>
- <Nominal, 4, 4>
- <PP, 5, 7>
- <NP, 6, 7>

Smith 2017

# 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 |

# CFGs

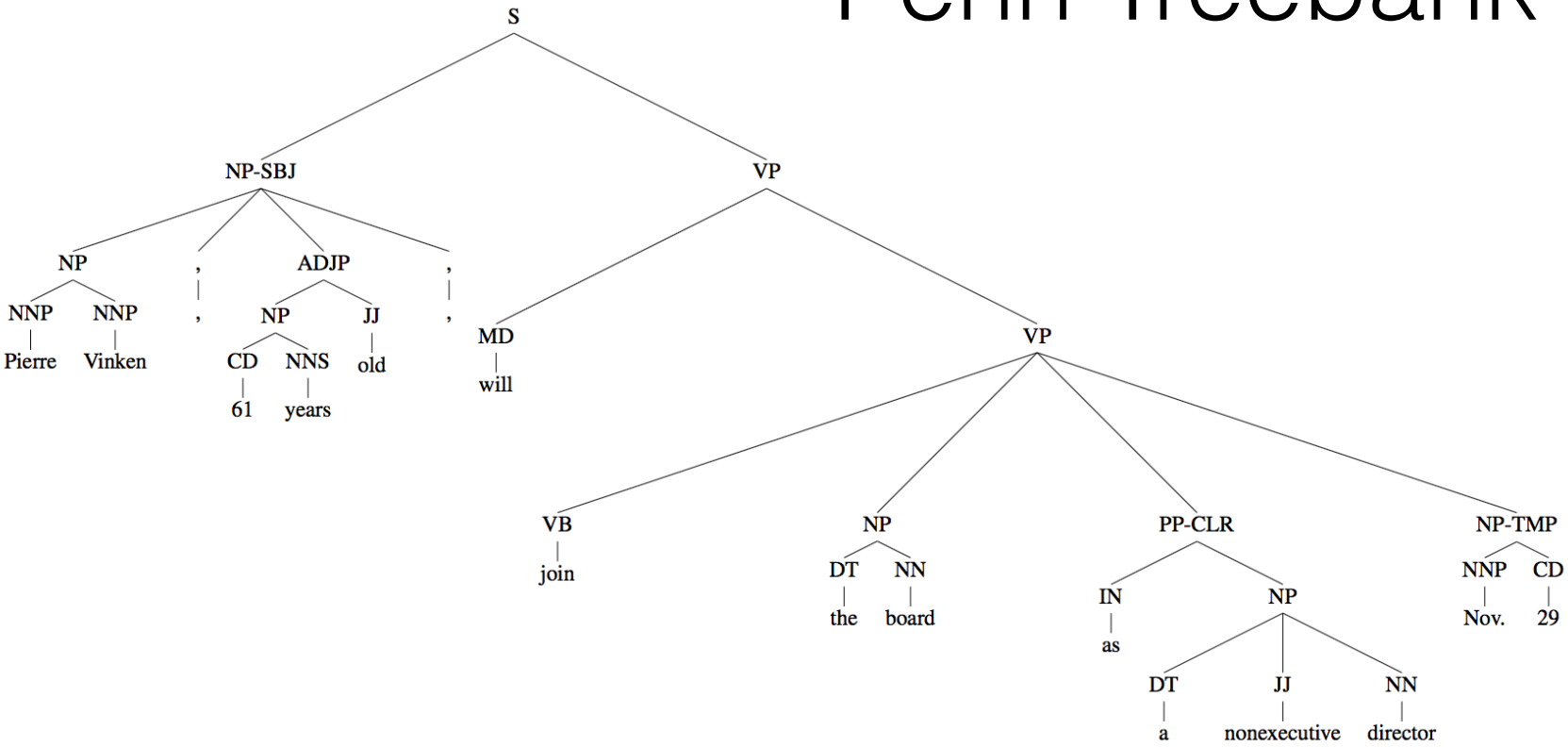
| | | |
|---|---|---|
| Verb-with-no-complement | → | disappear |
| Verb-with-S-complement | → | said |
| VP | → | Verb-with-no-complement |
| VP | → | Verb-with-S-complement S |

- disappear
- said he is going to the airport
- *disappear he is going to the airport

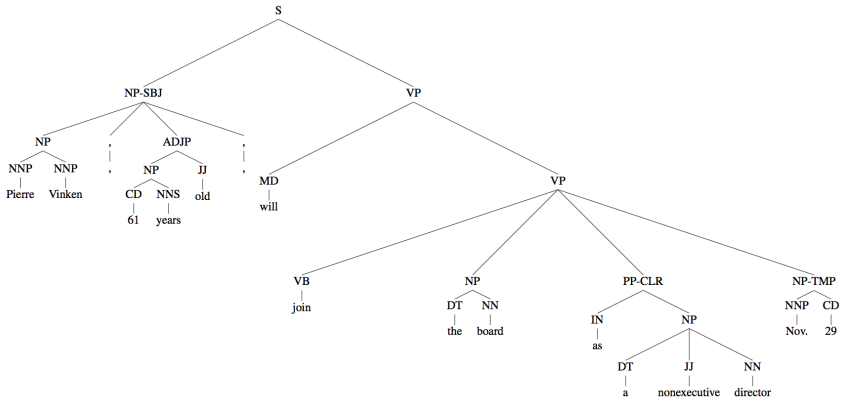
# Treebanks

- 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

```
NP → DT JJ NN
NP → DT JJ NNS
NP → DT JJ NN NN
NP → DT JJ JJ NN
NP → DT JJ CD NNS
NP → RB DT JJ NN NN
NP → RB DT JJ JJ NNS
NP → DT JJ JJ NNP NNS
NP → DT NNP NNP NNP NNP JJ NN
NP → DT JJ NNP CC JJ JJ NN NNS
NP → RB DT JJS NN NN SBAR
NP → DT VBG JJ NNP NNP CC NNP
NP → DT JJ NNS , NNS CC NN NNS NN
NP → DT JJ JJ VBG NN NNP NNP FW NNP
NP → NP JJ , JJ ‘‘ SBAR ’’ NNS
```

Jurafsky and Martin 2017

# 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 |
| $\Sigma$ | Finite alphabet of terminal symbols | the, dog, a |
| $R$ | Set of production rules, each<br>$A \rightarrow \beta$ [p]<br>$p = P(\beta \mid A)$ | S → NP VP<br>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)

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

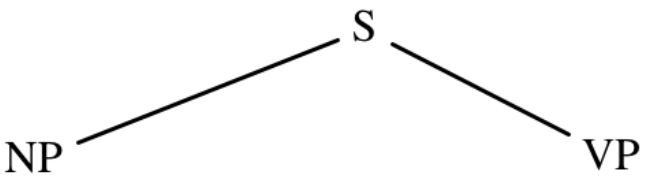
| A | | β | P(β \| NP) |
|---|---|---|---|
| NP | → | NP PP | 0.092 |
| NP | → | DT NN | 0.087 |
| NP | → | NN | 0.047 |
| NP | → | NNS | 0.042 |
| NP | → | DT JJ NN | 0.035 |
| NP | → | NNP | 0.034 |
| NP | → | NNP NNP | 0.029 |
| NP | → | JJ NNS | 0.027 |
| NP | → | QP -NONE- | 0.018 |
| NP | → | NP 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 |

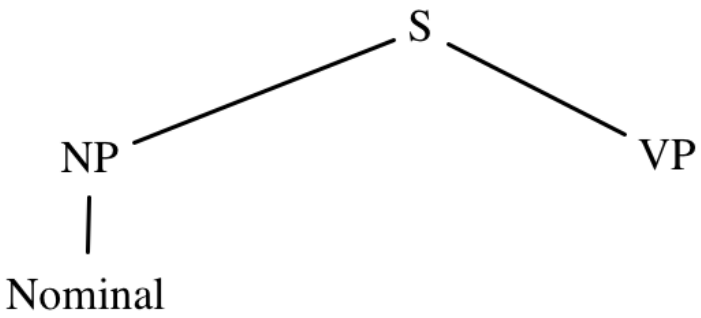
# PCFGs

- 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.

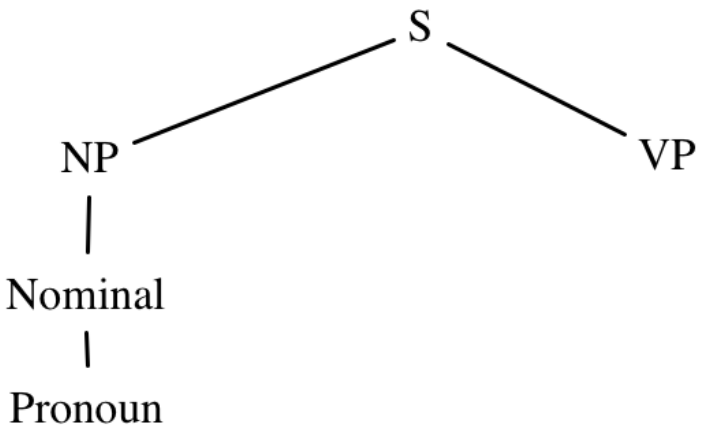
S

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

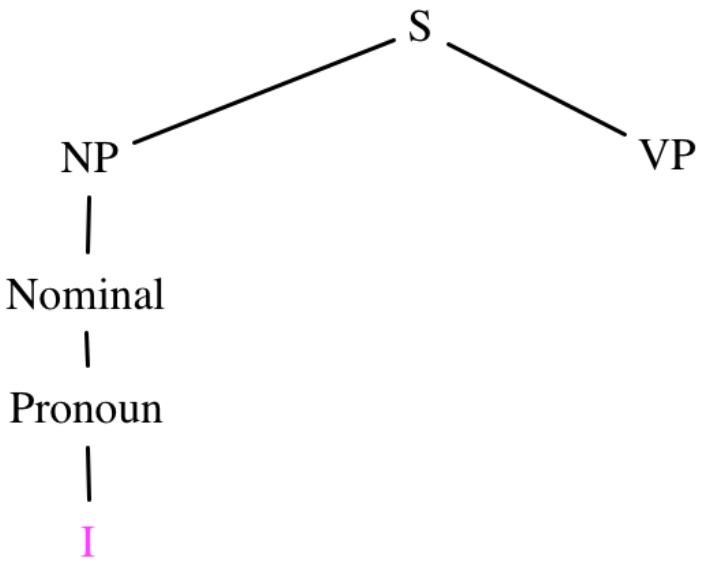




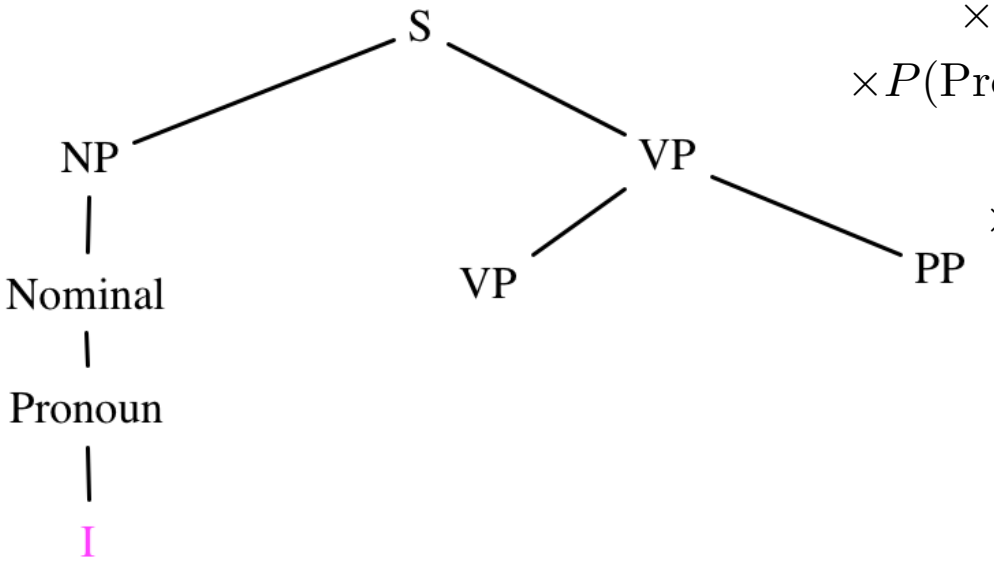

$$P(\text{NP VP} \mid \text{S})$$
$$\times P(\text{Nominal} \mid \text{NP})$$



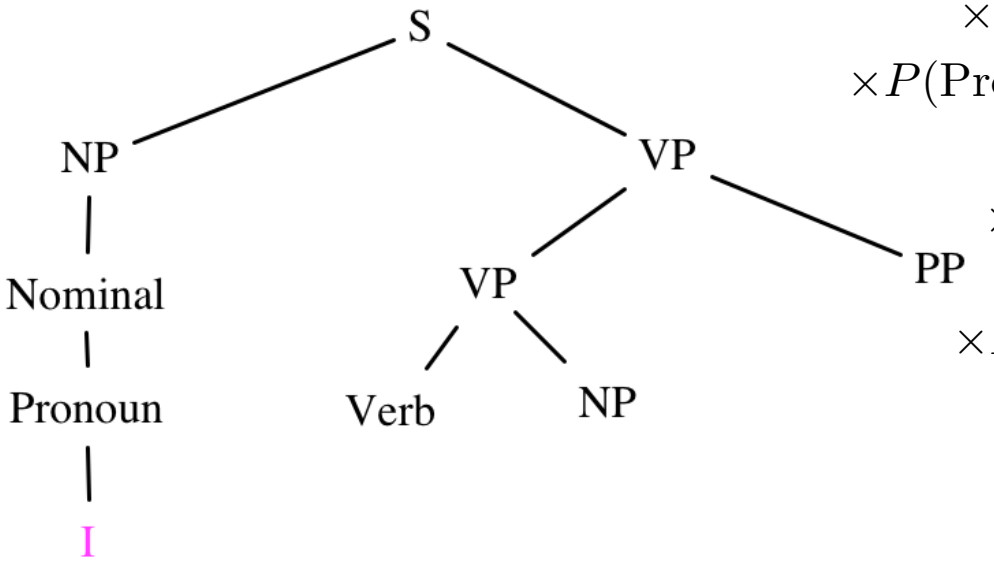

$$P(\text{NP VP} \mid \text{S})$$
$$\times P(\text{Nominal} \mid \text{NP})$$
$$\times P(\text{Pronoun} \mid \text{Nominal})$$



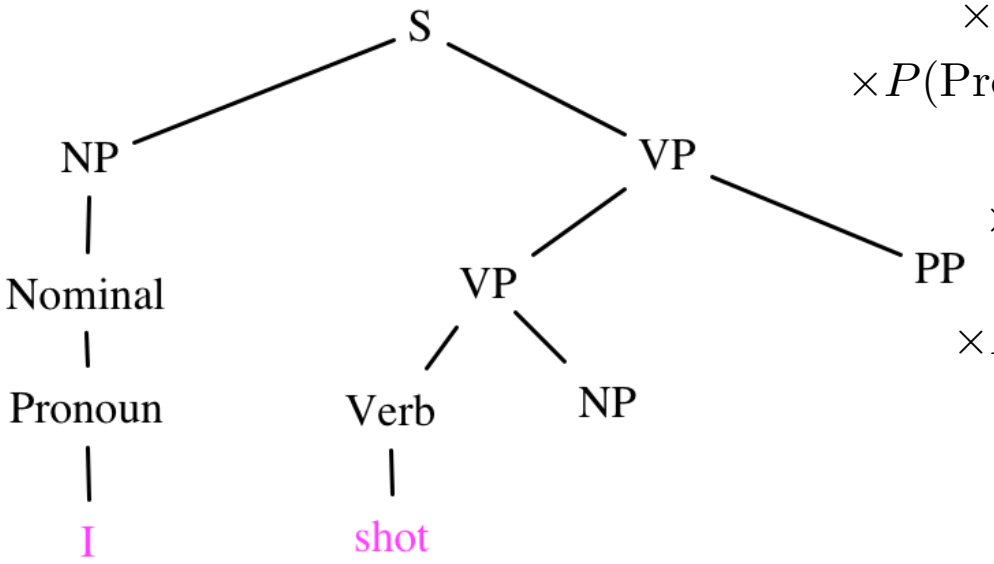

$$P(\text{NP VP} \mid \text{S})$$
$$\times P(\text{Nominal} \mid \text{NP})$$
$$\times P(\text{Pronoun} \mid \text{Nominal})$$
$$\times P(\text{I} \mid \text{Pronoun})$$



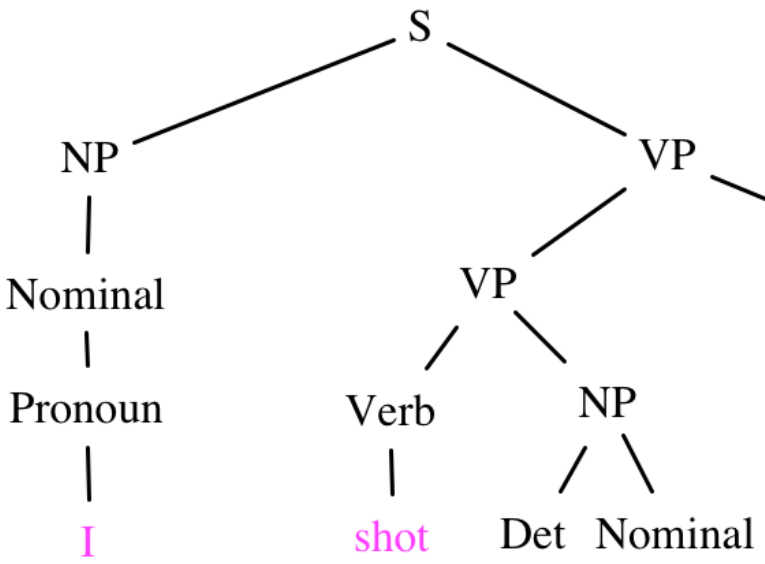

$$P(\text{NP VP} \mid \text{S})$$
$$\times P(\text{Nominal} \mid \text{NP})$$
$$\times P(\text{Pronoun} \mid \text{Nominal})$$
$$\times P(\text{I} \mid \text{Pronoun})$$
$$\times P(\text{VP PP} \mid \text{VP})$$



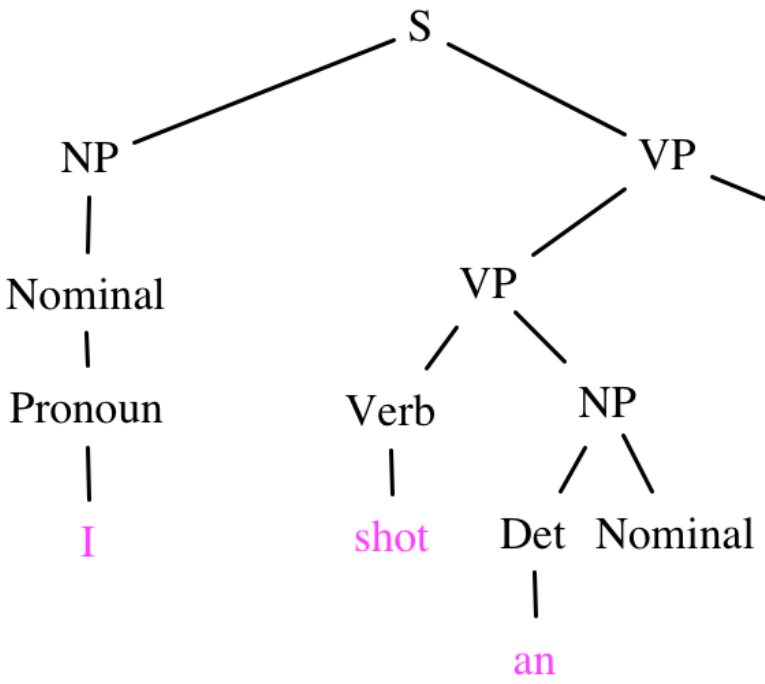

$$P(\text{NP VP} \mid \text{S})$$
$$\times P(\text{Nominal} \mid \text{NP})$$
$$\times P(\text{Pronoun} \mid \text{Nominal})$$
$$\times P(\text{I} \mid \text{Pronoun})$$
$$\times P(\text{VP PP} \mid \text{VP})$$
$$\times P(\text{Verb NP} \mid \text{VP})$$



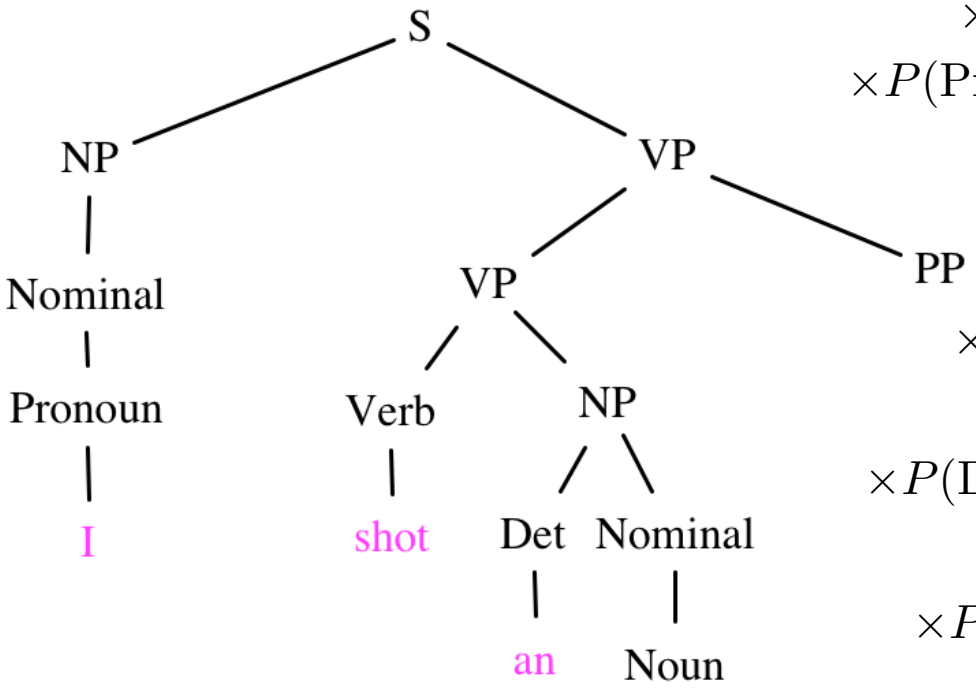

$$P(\text{NP VP} \mid \text{S})$$
$$\times P(\text{Nominal} \mid \text{NP})$$
$$\times P(\text{Pronoun} \mid \text{Nominal})$$
$$\times P(\text{I} \mid \text{Pronoun})$$
$$\times P(\text{VP PP} \mid \text{VP})$$
$$\times P(\text{Verb NP} \mid \text{VP})$$
$$\times P(\text{shot} \mid \text{Verb})$$



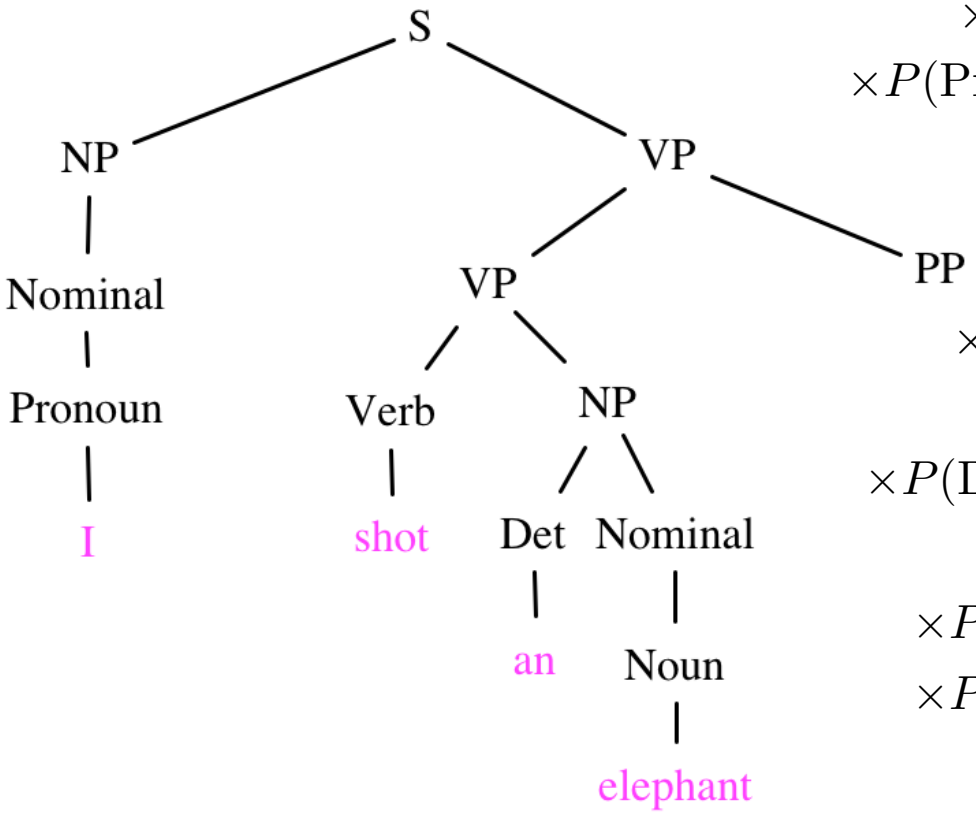

$$
\begin{aligned}
&P(\text{NP VP} \mid \text{S}) \\
&\times P(\text{Nominal} \mid \text{NP}) \\
&\times P(\text{Pronoun} \mid \text{Nominal}) \\
&\times P(\text{I} \mid \text{Pronoun}) \\
&\times P(\text{VP PP} \mid \text{VP}) \\
&\times P(\text{Verb NP} \mid \text{VP}) \\
&\times P(\text{shot} \mid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



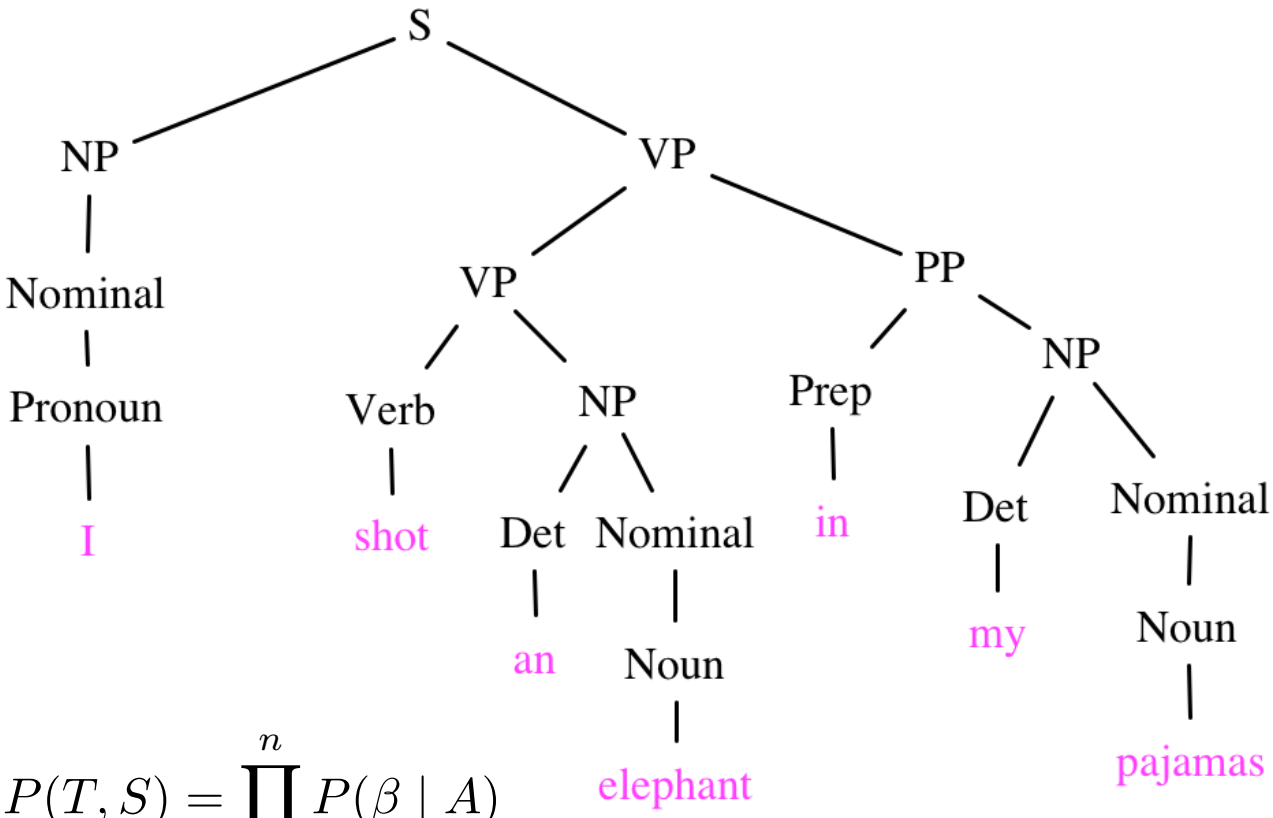

$$
\begin{aligned}
&P(\text{NP VP} \mid \text{S}) \\
&\times P(\text{Nominal} \mid \text{NP}) \\
&\times P(\text{Pronoun} \mid \text{Nominal}) \\
&\times P(\text{I} \mid \text{Pronoun}) \\
&\times P(\text{VP PP} \mid \text{VP}) \\
&\times P(\text{Verb NP} \mid \text{VP}) \\
&\times P(\text{shot} \mid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




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




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




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

# PCFGs

- A PCFG gives us a mechanism for assigning scores (here, probabilities) to different parses for the same sentence.
- But we often care about is finding the single best parse with the highest probability.