Carnegie Mellon University
Language
Technologies
Institute

11-411/11-611 Natural Language Processing

Treebanks and Probabilistic Parsing

David R. Mortensen and Lori Levin

March 28, 2023

Language Technologies Institute

Learning Objectives

- Understand what a treebank is
- Be able to name three important treebanks and what types of treebanks they are
- Express the reasons that treebanks should be treated with respect as well as caution
- Describe how a PCFG can be trained, give a constituency treebank like the Penn Treebank
- Implement parsing and recognition with PCFGs
- Describe how a dependency parser can be trained, give a dependency treebank (e.g., one of the UD treebanks)

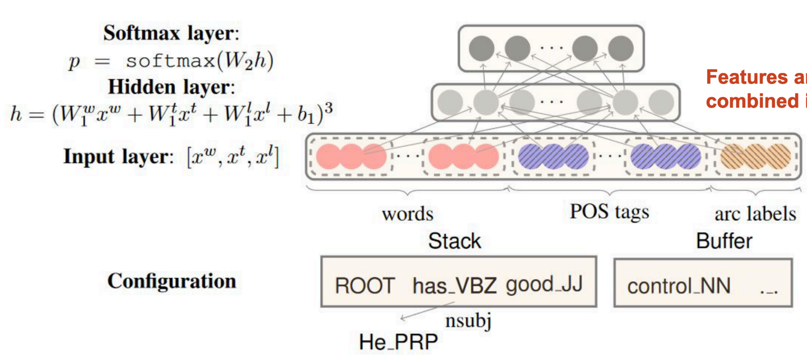
Training a Dependency Parser with a UD Treebank

Universal Dependency Treebanks Are Designed for Cross-Lingual Training

- Train a dependency parser in over 100 languages
    - One at a time
    - Cross-lingually

Training a Transition-Based Parser Means Training a Classifier

A popular kind of classifier for transition-based parsing is a FFNN, as in Chen and Manning (2014):



Features are automatically combined inside NN.

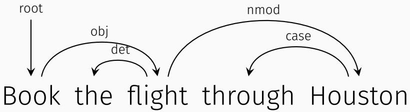
Only atomic inputs are fine, no longer manual feature combinations.

Example of Features: Feed-Forward Neural Transition Parser

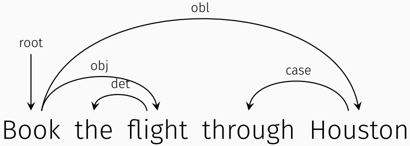
Here are the features extracted by Chen and Manning's (2014) feed-forward neural model for transition parsing:

- The top three words on S and B (6 features)
  $s_1, s_2, s_3, b_1, b_2, b_3$
- The two leftmost/rightmost children of the top two words on S (8 features)
  $lc_1(s_i), lc_2(s_i), rc_1(si), rc_2(s_i)\ i = 1, 2$
- The leftmost and rightmost grandchildren (4 features)
  $lc_1(lc_1(s_i)), rc_1(rc_1(s_i))\ i = 1, 2$
- POS tags for all words invoked above (18 features)
- Arc labels of all children/grandchildren invoked above (12 features)

Book the flight through Houston



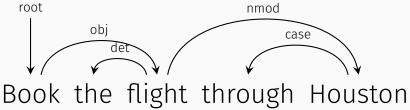
The flight is going through Houston.



Houston is the travel agent.

Creating Training Oracle from UD Treeback

Reference



Stack (Features)

ROOT

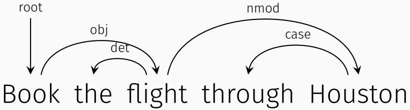
Buffer (Features)

book
the
flight
through
Houston

History (Features)

Creating Training Oracle from UD Treeback

Reference



Stack (Features)

book
ROOT

Buffer (Features)

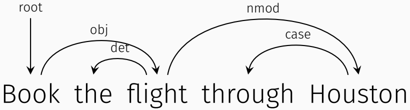
the
flight
through
Houston

History (Features)

SHIFT

Creating Training Oracle from UD Treeback

Reference



Stack (Features)

the
book
ROOT

Buffer (Features)

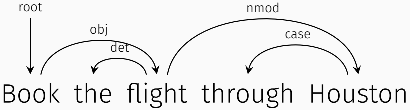
flight
through
Houston

History (Features)

SHIFT SHIFT

Creating Training Oracle from UD Treeback

Reference



Stack (Features)

flight
the
book
ROOT

Buffer (Features)

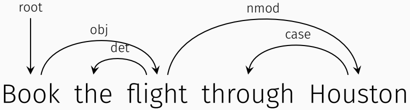
through
Houston

History (Features)

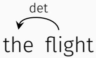
SHIFT SHIFT SHIFT

Creating Training Oracle from UD Treeback

Reference



Stack (Features)



book

ROOT

Buffer (Features)

through

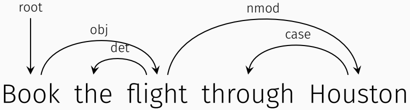
Houston

History (Features)

SHIFT SHIFT SHIFT LEFT-ARC

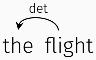
Creating Training Oracle from UD Treeback

Reference



Stack (Features)

through



book
ROOT

Buffer (Features)

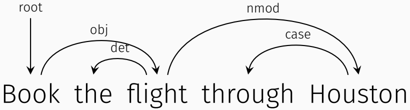
Houston

History (Features)

SHIFT SHIFT SHIFT LEFT-ARC SHIFT

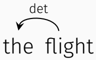
Creating Training Oracle from UD Treeback

Reference



Stack (Features)

Houston
through



book
ROOT

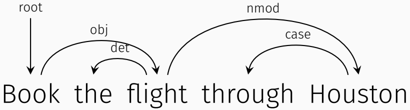
Buffer (Features)

History (Features)

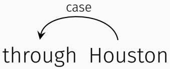
SHIFT SHIFT SHIFT LEFT-ARC SHIFT SHIFT

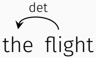
Creating Training Oracle from UD Treeback

Reference



Stack (Features)





book

ROOT

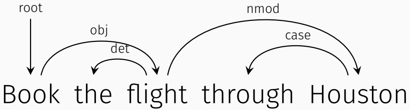
Buffer (Features)

History (Features)

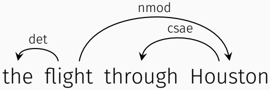
SHIFT SHIFT SHIFT LEFT-ARC SHIFT SHIFT LEFT-ARC

Creating Training Oracle from UD Treeback

Reference



Stack (Features)



book

ROOT

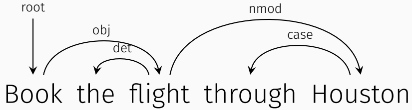
Buffer (Features)

History (Features)

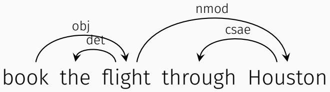
SHIFT SHIFT SHIFT LEFT-ARC SHIFT SHIFT LEFT-ARC RIGHT-ARC

Creating Training Oracle from UD Treeback

Reference



Stack (Features)



ROOT

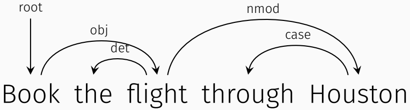
Buffer (Features)

History (Features)

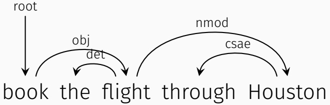
SHIFT SHIFT SHIFT LEFT-ARC SHIFT SHIFT LEFT-ARC RIGHT-ARC RIGHT-ARC

Creating Training Oracle from UD Treeback

Reference



Stack (Features)



Buffer (Features)

History (Features)

SHIFT SHIFT SHIFT LEFT-ARC SHIFT SHIFT LEFT-ARC RIGHT-ARC RIGHT-ARC

Phrase Structure Treebanks

Grammars Can Be Encoded Explicitly or Implicitly

Explicit

S $\rightarrow$ NP VP

NP $\rightarrow$ Det N

VP $\rightarrow$ V NP

Det $\rightarrow$ a | the

N $\rightarrow$ professor | students

V $\rightarrow$ delighted | annoyed

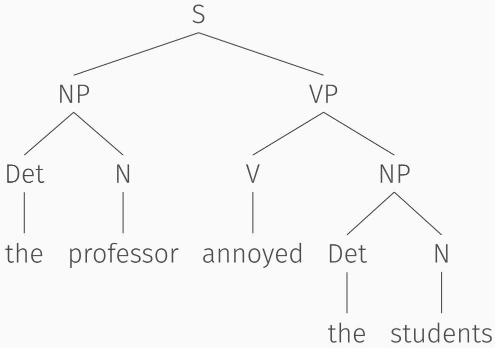
As in a hand-crafted grammar.

Implicit

```
(S
  (NP
    (Det the)
    (N professor))
  (VP
    (V annoyed)
    (NP
      (Det the)
      (N students))))
```

As in a treebank.

New notation for constituency trees




```
(S
  (NP
    (Det the)
    (N professor))
  (VP
    (V annoyed)
    (NP
      (Det the)
      (N students))))
```

Example Sentence from PTB

```
( (S
    (NP-SBJ
      (NP (NNP Pierre) (NNP Vinken) )
      (, ,)
      (ADJP
        (NP (CD 61) (NNS years) )
        (JJ old) )
      (, ,) )
    (VP (MD will)
      (VP (VB join)
        (NP (DT the) (NN board) )
        (PP-CLR (IN as)
          (NP (DT a) (JJ nonexecutive) (NN director) ) )
        (NP-TMP (NNP Nov.) (CD 29) ) ) )
    (. .) ) )
```

The First Big Treebank Was the Penn TreeBank

Contents: (about 3 million words)

- Brown corpus:
  - a slice of life from 1967
  - several genres of text: magazine, news, fiction, non-fiction
  - no speech
- ATIS (Air Travel Information Service corpus):
  - people make travel plans with a human travel agent
  - transcribed speech
  - task-oriented dialogue
- Switchboard Corpus:
  - transcribed speech
  - people talk on the phone with strangers about assigned topics like recycling
- Wall Street Journal corpus:
  - Around 1989 to 1991

How was PTB created?

Highly trained human linguists used a 300-page instruction manual.

What is in the instruction manual? Hundreds of details:

- What to do with appositives:
  Pierre Vinken, 61 years old, ...
- What to do with ages:
  61 years old
- What to do with auxiliary verbs:
  will join the board
- What to do with first and last names:
  Pierre Vinken

Remember that a tree can be described by rules

```
( (S
(NP-SBJ
  (NP (NNP Pierre) (NNP Vinken) )
  (, ,)
  (ADJP
    (NP (CD 61) (NNS years) )
    (JJ old) )
  (, ,) )
(VP (MD will)
  (VP (VB join)
    (NP (DT the) (NN board) ) ))))
```

```
S --> NP-SBJ  VP
NP-SBJ --> NP , ADJP ,
NP --> NNP  NNP
ADJP --> NP  JJ
NP --> CD NNS
VP --> MD VP
VP --> VB NP
NP --> DT NN
```

If you turn PTB into rules

More than 30,000 rule types

Many types with only one token (rules that are only used once)

Some PTB Rules by Frequency

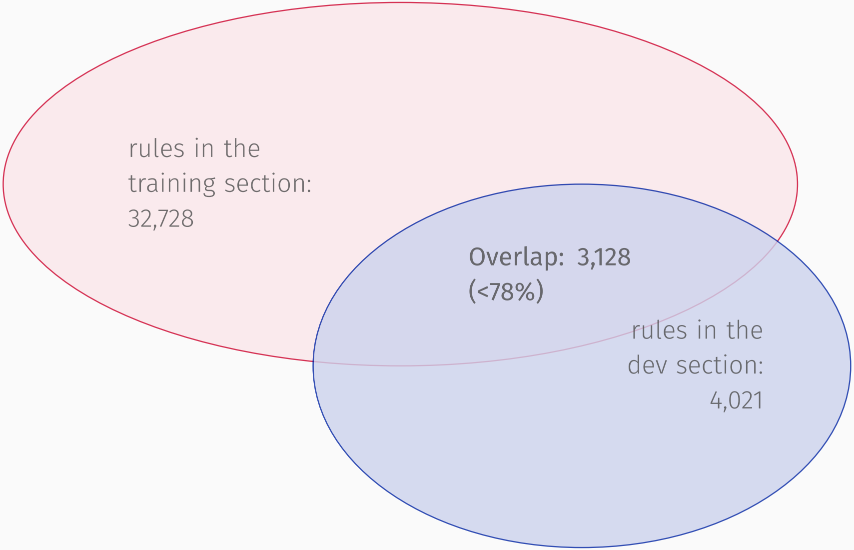
40717 PP → IN NP
33803 S → NP-SBJ VP
22513 NP-SBJ → -NONE-
21877 NP → NP PP
20740 NP → DT NN
14153 S → NP-SBJ VP .
12922 VP → TO VP
11881 PP-LOC → IN NP
11467 NP-SBJ → PRP
11378 NP → -NONE-
11291 NP → NN
...
989 VP → VBG S
985 NP-SBJ → NN
983 PP-MNR → IN NP
983 NP-SBJ → DT
969 VP → VBN VP
...

100 VP → VBD PP-PRD
100 PRN → : NP :
100 NP → DT JJS
100 NP-CLR → NN
99 NP-SBJ-1 → DT NNP
98 VP → VBN NP PP-DIR
98 VP → VBD PP-TMP
98 PP-TMP → VBG NP
97 VP → VBD ADVP-TMP VP
...
10 WHNP-1 → WRB JJ
10 VP → VP CC VP PP-TMP
10 VP → VP CC VP ADVP-MNR
10 VP → VBZ S , SBAR-ADV
10 VP → VBZ S ADVP-TMP

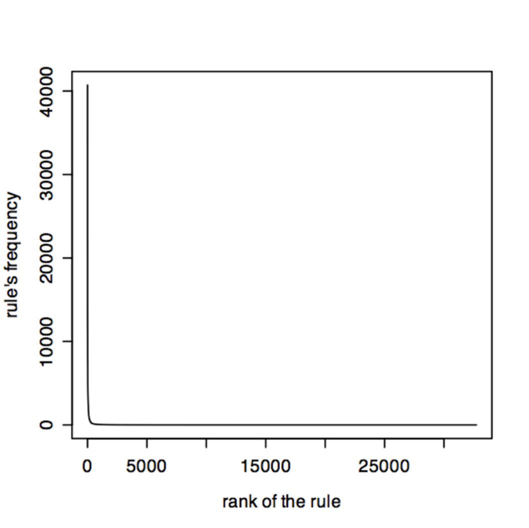
Some PTB Rules by Frequency

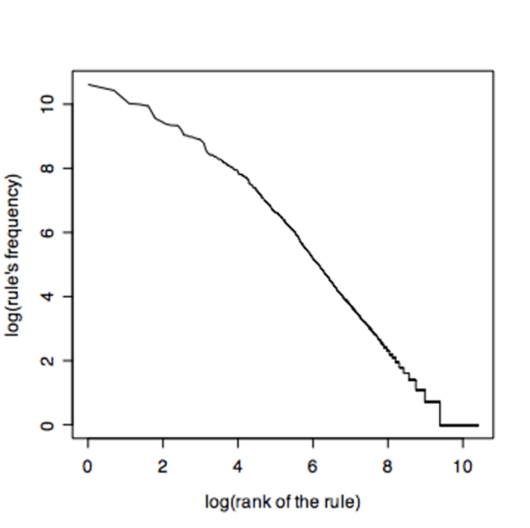
- 40,717: PP $\rightarrow$ IN PP
    - IN means *preposition* in this treebank.
    - *of the company*, *up three points*, *about linguistics*
    - *Of* is the second most frequent word in English
- 22,513: NP-SBJ $\rightarrow$ NONE
    - I want to NONE go
    - The person that NONE met her
- 21,877: NP $\rightarrow$ NP PP
    - the final hours of campaigning
    - a new heavyweight in providing emergency funds

Many Rules in the PTB Are Sparsely Attested



The Most Frequent Rules are Very Frequent and the Rest Are Infrequent





The Promise and Peril of Treebanks

Proper Ambivalence toward Treebanks

Why you should have great respect for treebanks

Why you should be cautious around treebanks

The Making of a Treebank

- Develop initial coding manual (hundreds of pages long)
  - Linguists define categories and tests
  - Try to foresee as many complications as possible

  Develop annotation tools (annotation UI, pre-parser) Collect data (corpora)
  - Composition depends on the purpose of the corpus
  - Must also be pre-processed
- Automatically parse the corpus/corpora
- Train annotators ("coders")
- Manually correct the automatic annotations ("code")
  - Generally done by non-experts under the direction of linguists
  - When cases are encountered that are not in the coding manual…
    - Revise the coding manual to include them
    - Check that already-annotated sections of the corpus are consistent with the new standard

This is expensive and time-consuming!

Why You Should Respect Treebanks

Treebanks require great skill

- Expert linguists make thousands of decisions
- Many annotators must remember all of the decisions and use them consistently, including knowing which decision to use
- The "coding manual" containing all of the decisions is hundreds of pages long

Treebanks take many years to make

- Writing the coding manual, training coders, building user-interface tools, etc., all take a lot of time
- So does the actual coding of the data and quality assurance

Treebanks are expensive

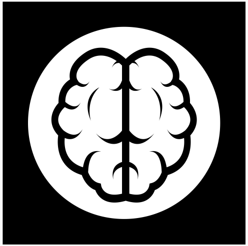
Somebody has to secure funding for these projects

You Should Be Cautious around Treebanks

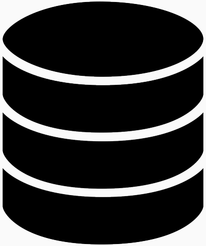
- They are **too big to fail**
- The are **produced under pressure** of time and funding
- Although most of the decisions are made by experts, **most of the coding is done by non-experts**

To make a good model, you should understand what you’re modeling

There Are Two Sources of Improvement in Machine Learning



Better models



Better data

A Good ML Practitioner Cares about Models and Data

Naïve practitioners of NLP often make the assumptions “data is data” and “more data is always better that better data.”

For treebanks, neither of these are true.

The structure of annotations greatly affects the way in which they can be used

A constituency treebank is not good for the same things as a dependency treebank—and to some extent, vice versa.

The nature of the data matters very much

The fact that PTB trees have a very flat structure and that so many of the rules occur in only one tree has many implications for its use.

Crowdsourcing is often not the way out

Crowdsourced data, while cheap and convenient, is not useful when complex judgements are involved

Things that are made possible by treebanks

- Probabilistic Context-Free Parsing
- Creating an oracle for dependency parsing

Probabilistic Context Free Parsing

Recognition and Parsing Are Related Problems

Input: sentence $\mathbf{w} = (w_1, \ldots, w_n)$ and grammar G

Output (recognition): true iff $\mathbf{w} \in L(G)$

Output (parsing): one or more derivations for $\mathbf{w}$, under G

What if, instead, we were interested in the probability that w is a sentence in $L(G)$? Or to know the probability of any given derivation for w given G? **We can train such a model with a treebank**.

Probabilistic Context Free Grammars

N a set of non-terminal symbols

Σ a set of terminal symbols (disjoint from N)

R a set of rules or productions of the form $A \rightarrow \beta$ $[p]$, where

$A \in N$

$\beta \in (\Sigma \cup N)*$

p is a number between 0 and 1 expressing the probability of A being rewritten as β, i.e., $P(\beta|A)$

S a designated start symbols in N

A Sample PCFG

S	$\rightarrow$	$NP\ VP$	[.80]	Det	$\rightarrow$	$that$ [.05] | the [.80] | a [.15]	
S	$\rightarrow$	$Aux\ NP\ VP$	[.15]	$Noun$	$\rightarrow$	$book$	[.10]
S	$\rightarrow$	VP	[.05]	$Noun$	$\rightarrow$	$flights$	[.50]
NP	$\rightarrow$	$Det\ Nom$	[.20]	$Noun$	$\rightarrow$	$meal$	[.40]
NP	$\rightarrow$	$Proper\text{-}Noun$	[.35]	$Verb$	$\rightarrow$	$book$	[.30]
NP	$\rightarrow$	Nom	[.05]	$Verb$	$\rightarrow$	$include$	[.30]
NP	$\rightarrow$	$Pronoun$	[.40]	$Verb$	$\rightarrow$	$want$	[.40]
Nom	$\rightarrow$	$Noun$	[.75]	Aux	$\rightarrow$	can	[.40]
Nom	$\rightarrow$	$Noun\ Nom$	[.20]	Aux	$\rightarrow$	$does$	[.30]
Nom	$\rightarrow$	$Proper\text{-}Noun\ Nom$	[.05]	Aux	$\rightarrow$	do	[.30]
VP	$\rightarrow$	$Verb$	[.55]	$Proper\text{-}Noun$	$\rightarrow$	TWA	[.40]
VP	$\rightarrow$	$Verb\ NP$	[.40]	$Proper\text{-}Noun$	$\rightarrow$	$Denver$	[.40]
VP	$\rightarrow$	$Verb\ NP\ NP$	[.05]	$Pronoun$	$\rightarrow$	you [.40] | I [.60]	

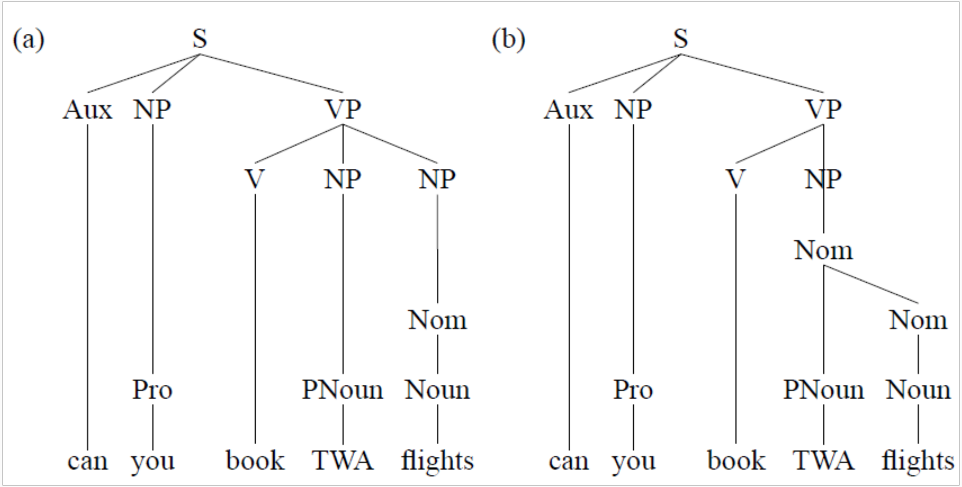
Figure 12.1 A PCFG; a probabilistic augmentation of the miniature English grammar and lexicon in Figure 10.2. These probabilities are not based on a corpus; they were made up merely for expository purposes.

The Probability of a Parse Tree under a PCFG

$$P(T, \mathrm{w}) = \prod_{n \in T} P(r(n)) \qquad (1)$$

The joint probability of a particular parse T and sentence w, is defined as the product of the probabilities of all the rules r used to expand each node n in the parse tree.

A Sentence with Two Parses



Comparing the Two Parses of the Example Sentence

Rules			P	Rules			P
S	→	Aux NP VP	.15	S	→	Aux NP VP	.15
NP	→	Pro	.40	NP	→	Pro	.40
VP	→	V NP NP	.05	VP	→	V NP	.40
NP	→	Nom	.05	NP	→	Nom	.05
NP	→	PNoun	.35	Nom	→	PNoun Nom	.05
Nom	→	Noun	.75	Nom	→	Noun	.75
Aux	→	Can	.40	Aux	→	Can	.40
NP	→	Pro	.40	NP	→	Pro	.40
Pro	→	you	.40	Pro	→	you	.40
Verb	→	book	.30	Verb	→	book	.30
PNoun	→	TWA	.40	Pnoun	→	TWA	.40
Noun	→	flights	.50	Noun	→	flights	.50

Disambiguating with Probabilities

Left parse: book flights for (on behalf of) TWA.

$$
\begin{aligned}
P(w_L) =& 0.15 \times 0.40 \times 0.05 \times 0.05 \times 0.35 \times 0.75 \times 0.40 \times 0.40 \times \\
& 0.40 \times 0.30 \times 0.40 \times 0.50 \\
=& 1.5 \times 10^{-6}
\end{aligned}
$$

Right parse: book flights that are on TWA.

$$
\begin{aligned}
p(w_R) =& 0.15 \times 0.40 \times 0.40 \times 0.05 \times 0.05 \times 0.75 \times 0.40 \times 0.40 \\
& \times 0.40 \times 0.30 \times 0.40 \times 0.05 \\
=& 1.7 \times 10^{-6}
\end{aligned}
$$

Right parse wins!

Training a PCFG with a Treebank

Given a constituency treebank, training a PCFG can be as simple as cataloging all of the rules and assigning probabilities based on maximum likelihood estimation.

$$P(\alpha \to \beta | \alpha) = \frac{\text{Count}(\alpha \to \beta)}{\sum_\gamma \text{Count}(\alpha \to \gamma)} = \frac{\text{Count}(\alpha \to \beta)}{\text{Count}(\alpha)} \tag{2}$$

Suppose that the (small) treebank has 100 instances of nodes labelled VP, with four ways to expand VP as shown below:

Rule	Count in Treebank	Probability
$VP \to V$	30	0.30
$VP \to V\ NP$	30	0.30
$VP \to V\ NP\ NP$	15	0.15
$VP \to V\ PP$	25	0.25

Questions?