Statistical Parsing

What is Statistical Parsing

- Meaning of Parsing:

Breaking a sentence into grammatical structure (NP, VP, PP etc.) and creating a **parse tree**.

Example sentence:
"The boy eats an apple."

Tree:

```mathematica
                S
          ┌─────┴─────┐
         NP           VP
     ┌───┴───┐    ┌───┴───┐
    Det      N    V       NP
   the      boy  eats   an apple
```

✔️ Parsing helps machine understand **who is doing what.**

Why Parsing is Difficult? → Ambiguity

A single sentence may have **many possible parse trees**.

This is called **syntactic ambiguity**.

Example:

“I saw the man with a telescope.”

Two meanings → Two trees:

1. I used the telescope
2. The man has a telescope

Traditional grammar (CFG) **cannot** decide which meaning is correct.

What is Statistical Parsing

- Statistical Parsing =
  Grammar + Probability + Corpus Data
- It uses:
- ✓ Grammar rules
  ✓ Probability of each rule
  ✓ Training on Treebanks
- And selects the parse tree that is **most likely** to be correct.
- "Statistical parsing chooses the most probable parse tree using data and probability, not just rules."

Why Do We Need Statistical Parsing

CFG treats all rules equally

- It cannot prefer one parse tree over another.

Ambiguity becomes unmanageable

- More words → More possible trees.

Statistical parsing solves this

- By learning from real-world data and assigning probabilities.
- Example:
- Rule:
  NP → Det N
  Probability = 0.6 (because it appears often in corpus)
- Rule:
  NP → NP PP
  Probability = 0.1 (appears less often)
- So parser prefers NP → Det N
  This helps remove wrong trees.

Example of Probabilistic Parsing

Sentence: **“The cat chased the mouse.”**

- Suppose the parser finds **3 possible parse trees**:
- Tree A → probability = 0.45
- Tree B → probability = 0.38
- Tree C → probability = 0.17

The parser **chooses Tree A**, even if grammar allows all 3.

- This is the core idea of statistical parsing.

How Do We Get Probabilities?

Probabilities come from **Treebanks** like:

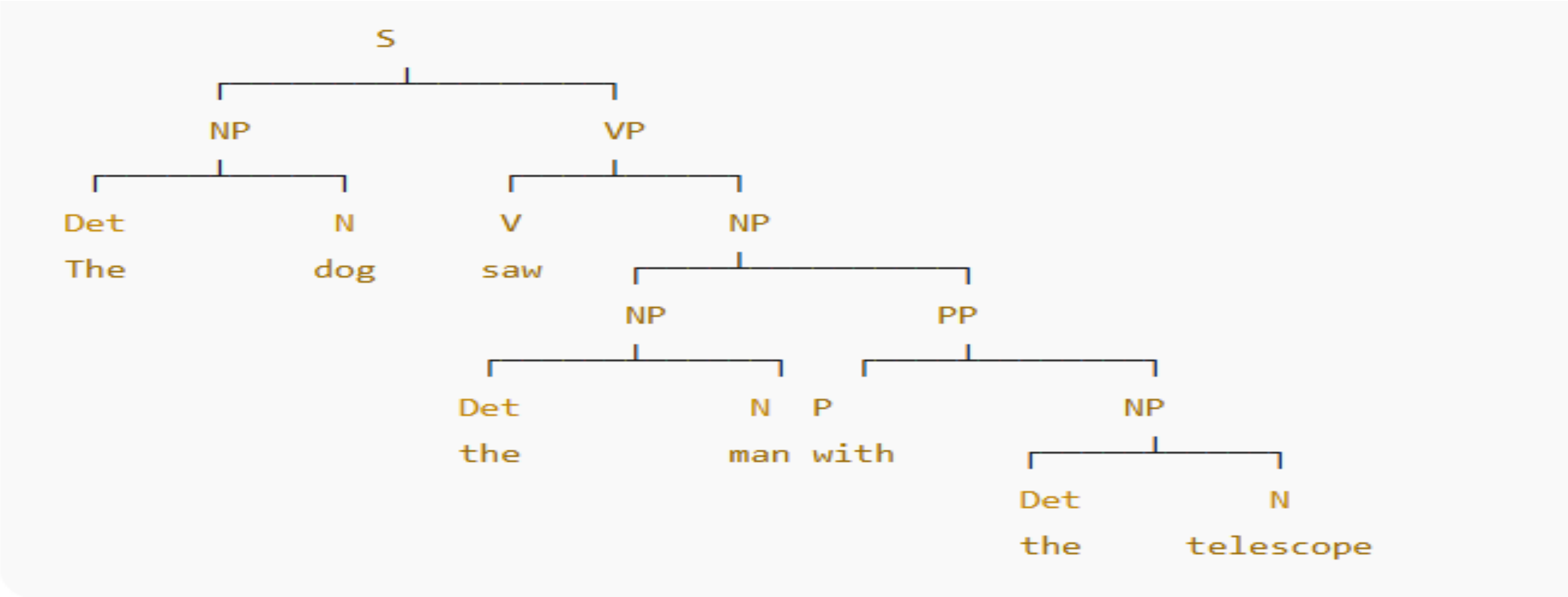
- Penn Treebank
- Brown Corpus
- Google Treebanks
- These are manually annotated datasets with parse trees.

Example from Treebank:

- NP → Det N appears **600 times**
  NP → NP PP appears **100 times**
- Probability = count(rule) ÷ total NP expansions
- = 600 / (600 + 100)
  = 0.857
- So NP → Det N is **highly probable**.

Example Sentence

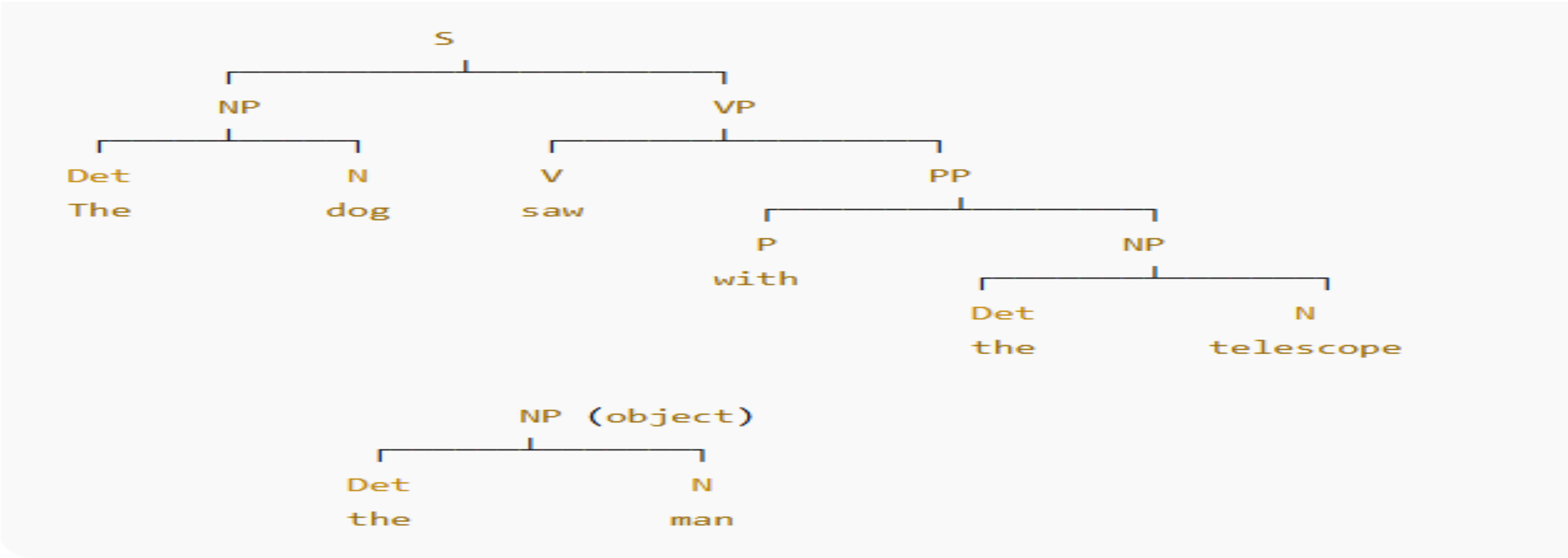
- The dog saw the man with the telescope.
- PP attaches to NP (Meaning: The man has the telescope)
- **Correct Parse Tree – PP under NP**



PP is child of NP
PP is attached to **“the man”**
Meaning: *The man has a telescope*

PP attaches to VP (Meaning: The dog used the telescope to see)

- Correct Parse Tree – PP under VP



PP is child of VP
PP modifies the verb **"saw"**
Meaning: *The dog used a telescope to see the man*

How do we get probabilities? (from Treebank)

- We look at a big Treebank (collection of sentences with correct parse trees) and **count** how often each grammar rule happens.

Example counts from Treebank

For NP:

- **NP → Det N** appears **80 times**
- **NP → NP PP** appears **20 times**

So total NP expansions = 80 + 20 = 100

Rule probabilities

$$P(NP \to Det\ N) = 80/100 = 0.8$$

$$P(NP \to NP\ PP) = 20/100 = 0.2$$

For VP (in our simple example):

- **VP → V NP** appears 100 times

$$P(VP \to V\ NP) = 1.0$$

For PP:

- **PP → P NP** appears 100 times

$$P(PP \to P\ NP) = 1.0$$

Now we also check **where PP usually attaches** in similar sentences:

In Treebank we find 10 similar cases:

- PP attaches to **NP** in 3 sentences
- PP attaches to **VP** in 7 sentences

$$P(\text{PP attaches to NP}) = 3/10 = 0.3$$

$$P(\text{PP attaches to VP}) = 7/10 = 0.7$$

Calculate probability of each tree

◆ Tree A: PP attaches to NP

Rules used:

- subject NP → Det N : **0.8**
- object NP → NP PP : **0.2**
- VP → V NP : **1.0**
- PP → P NP : **1.0**
- PP attached to NP : **0.3**

Multiply all:

$$P(\text{Tree A}) = 0.8 \times 0.2 \times 1.0 \times 1.0 \times 0.3 = 0.048$$

◆ Tree B: PP attaches to VP

Rules used:

- subject NP → Det N : **0.8**
- object NP → Det N : **0.8**
- VP → V NP : **1.0**
- PP → P NP : **1.0**
- PP attached to VP : **0.7**

$$P(\text{Tree B}) = 0.8 \times 0.8 \times 1.0 \times 1.0 \times 0.7 = 0.448$$

Step 4 — Statistical parser decision

- **Tree A** probability = **0.048**
- **Tree B** probability = **0.448**

Tree B has much higher probability,
so the statistical parser chooses **Tree B** as the correct parse.