Announcements
Ambiguity
Syntactic Ambiguity in English

Programs must be written for people to read
Syntactic Ambiguity in English

Programs must be written for people to read\textsuperscript{1}

\textsuperscript{1}Preface of \textit{Structure and Interpretation of Computer Programs}  
by Harold Abelson and Gerald Sussman with Julie Sussman
Syntactic Ambiguity in English

Programs must be written for people to read

1Preface of *Structure and Interpretation of Computer Programs* by Harold Abelson and Gerald Sussman with Julie Sussman
Syntactic Ambiguity in English

Programs must be written for people to read

Preface of *Structure and Interpretation of Computer Programs*
by Harold Abelson and Gerald Sussman with Julie Sussman
Programs must be written for people to read\(^1\)

\(^1\)Preface of *Structure and Interpretation of Computer Programs* by Harold Abelson and Gerald Sussman with Julie Sussman
Programs must be written for people to read

\(^1\)Preface of *Structure and Interpretation of Computer Programs* by Harold Abelson and Gerald Sussman with Julie Sussman
Programs must be written for people to read

Preface of *Structure and Interpretation of Computer Programs* by Harold Abelson and Gerald Sussman with Julie Sussman

---

1Preface of *Structure and Interpretation of Computer Programs* by Harold Abelson and Gerald Sussman with Julie Sussman
Syntactic Ambiguity in English

Programs must be written for people to read

\[1\]Preface of *Structure and Interpretation of Computer Programs* by Harold Abelson and Gerald Sussman with Julie Sussman
Syntactic Ambiguity in English

**Program (noun)**
- a series of coded software instructions

**Program (verb)**
- provide a computer with coded instructions

Programs must be written for people to read

**must (verb)**
- be obliged to

**must (noun)**
- dampness or mold

Definitions from the New Oxford American Dictionary
Syntax Trees
Representing Syntactic Structure
Representing Syntactic Structure

COWS
Representing Syntactic Structure

Noun

COWS
Representing Syntactic Structure

Photo by Vince O'Sullivan licensed under http://creativecommons.org/licenses/by-nc-nd/2.0/
Representing Syntactic Structure

Photo by Vince O'Sullivan licensed under http://creativecommons.org/licenses/by-nc-nd/2.0/
Representing Syntactic Structure

Photo by Vince O'Sullivan licensed under http://creativecommons.org/licenses/by-nc-nd/2.0/

Noun

cows

Verb

intimidate
Representing Syntactic Structure

Photo by Vince O'Sullivan licensed under http://creativecommons.org/licenses/by-nc-nd/2.0/

Noun cows Verb intimidate Noun cows
Representing Syntactic Structure

Photo by Vince O'Sullivan licensed under http://creativecommons.org/licenses/by-nc-nd/2.0/
Representing Syntactic Structure

Sentence

Noun Phrase

Noun Verb Noun

cows intimidate cows

Photo by Vince O'Sullivan licensed under http://creativecommons.org/licenses/by-nc-nd/2.0/
Representing Syntactic Structure

Photo by Vince O'Sullivan licensed under http://creativecommons.org/licenses/by-nc-nd/2.0/
Representing Syntactic Structure

Photo by Vince O'Sullivan licensed under http://creativecommons.org/licenses/by-nc-nd/2.0/
Representing Syntactic Structure

A *Tree* represents a phrase:

cows intimidate cows

Photo by Vince O'Sullivan licensed under http://creativecommons.org/licenses/by-nc-nd/2.0/
Representing Syntactic Structure

A **Tree** represents a phrase:

- **tag** — What kind of phrase (e.g., S, NP, VP)

![Diagram of a tree structure representing a sentence with tags for Noun Phrase and Verb Phrase]
Representing Syntactic Structure

A **Tree** represents a phrase:

- **tag** — What kind of phrase (e.g., *S*, *NP*, *VP*)
- **branches** — Sequence of **Tree** or **Leaf** components

Photo by Vince O'Sullivan licensed under [http://creativecommons.org/licenses/by-nc-nd/2.0/](http://creativecommons.org/licenses/by-nc-nd/2.0/)
A **Tree** represents a phrase:

- **tag** — What kind of phrase (e.g., S, NP, VP)
- **branches** — Sequence of **Tree** or **Leaf** components

A **Leaf** represents a single word:
Representing Syntactic Structure

A **Tree** represents a phrase:
- **tag** — What kind of phrase (e.g., S, NP, VP)
- **branches** — Sequence of **Tree** or **Leaf** components

A **Leaf** represents a single word:
- **tag** — What kind of word (e.g., N, V)
Representing Syntactic Structure

A **Tree** represents a phrase:

- **tag** — What kind of phrase (e.g., S, NP, VP)
- **branches** — Sequence of Tree or Leaf components

A **Leaf** represents a single word:

- **tag** — What kind of word (e.g., N, V)
- **word** — The word

Photo by [Vince O'Sullivan](http://creativecommons.org/licenses/by-nc-nd/2.0/) licensed under [Creative Commons](http://creativecommons.org/licenses/by-nc-nd/2.0/)
Representing Syntactic Structure

A **Tree** represents a phrase:
- **tag** — What kind of phrase (e.g., S, NP, VP)
- **branches** — Sequence of Tree or Leaf components

A **Leaf** represents a single word:
- **tag** — What kind of word (e.g., N, V)
- **word** — The word

Photo by Vince O'Sullivan licensed under [http://creativecommons.org/licenses/by-nc-nd/2.0/]
Representing Syntactic Structure

A **Tree** represents a phrase:
- **tag** — What kind of phrase (e.g., S, NP, VP)
- **branches** — Sequence of **Tree** or **Leaf** components

A **Leaf** represents a single word:
- **tag** — What kind of word (e.g., N, V)
- **word** — The word

Photo by Vince O'Sullivan licensed under
http://creativecommons.org/licenses/by-nc-nd/2.0/
Representing Syntactic Structure

A **Tree** represents a phrase:

- **tag** -- What kind of phrase (e.g., S, NP, VP)
- **branches** -- Sequence of Tree or Leaf components

A **Leaf** represents a single word:

- **tag** -- What kind of word (e.g., N, V)
- **word** -- The word
Representing Syntactic Structure

A **Tree** represents a phrase:
- **tag** — What kind of phrase (e.g., S, NP, VP)
- **branches** — Sequence of Tree or Leaf components

A **Leaf** represents a single word:
- **tag** — What kind of word (e.g., N, V)
- **word** — The word

Photo by Vince O'Sullivan licensed under http://creativecommons.org/licenses/by-nc-nd/2.0/
Representing Syntactic Structure

A **Tree** represents a phrase:

- **tag** — What kind of phrase (e.g., S, NP, VP)
- **branches** — Sequence of Tree or Leaf components

A **Leaf** represents a single word:

- **tag** — What kind of word (e.g., N, V)
- **word** — The word
Representing Syntactic Structure

A **Tree** represents a phrase:

- **tag** — What kind of phrase (e.g., *S*, *NP*, *VP*)
- **branches** — Sequence of **Tree** or **Leaf** components

A **Leaf** represents a single word:

- **tag** — What kind of word (e.g., *N*, *V*)
- **word** — The word
Representing Syntactic Structure

A **Tree** represents a phrase:

- **tag** — What kind of phrase (e.g., *S*, *NP*, *VP*)
- **branches** — Sequence of **Tree** or **Leaf** components

A **Leaf** represents a single word:

- **tag** — What kind of word (e.g., *N*, *V*)
- **word** — The word
Representing Syntactic Structure

A **Tree** represents a phrase:

- **tag** — What kind of phrase (e.g., S, NP, VP)
- **branches** — Sequence of Tree or Leaf components

A **Leaf** represents a single word:

- **tag** — What kind of word (e.g., N, V)
- **word** — The word

\[
\text{cows} = \text{Leaf}('N', \text{'cows'})
\]
Representing Syntactic Structure

A **Tree** represents a phrase:
- **tag** — What kind of phrase (e.g., S, NP, VP)
- **branches** — Sequence of **Tree** or **Leaf** components

A **Leaf** represents a single word:
- **tag** — What kind of word (e.g., N, V)
- **word** — The word

```
cows = Leaf('N', 'cows')
intimidate = Leaf('V', 'intimidate')
```

Photo by Vince O'Sullivan licensed under http://creativecommons.org/licenses/by-nc-nd/2.0/
Representing Syntactic Structure

A **Tree** represents a phrase:

- **tag** -- What kind of phrase (e.g., S, NP, VP)
- **branches** -- Sequence of Tree or Leaf components

A **Leaf** represents a single word:

- **tag** -- What kind of word (e.g., N, V)
- **word** -- The word

\[
\text{cows} = \text{Leaf}(\text{'N'}, \text{'cows'})
\]
\[
\text{intimidate} = \text{Leaf}(\text{'V'}, \text{'intimidate'})
\]
\[
\text{S, NP, VP} = \text{'S'}, \text{'NP'}, \text{'VP'}
\]
Representing Syntactic Structure

A **Tree** represents a phrase:
- **tag** --- What kind of phrase (e.g., S, NP, VP)
- **branches** --- Sequence of **Tree** or **Leaf** components

A **Leaf** represents a single word:
- **tag** --- What kind of word (e.g., N, V)
- **word** --- The word

\[
cows = \text{Leaf}(\text{'N'}, \text{'cows'})
\]
\[
\text{intimidate} = \text{Leaf}(\text{'V'}, \text{'intimidate'})
\]
\[
S, \ NP, \ VP = \text{'S'}, \text{'NP'}, \text{'VP'}
\]
\[
\text{Tree}(S, \text{[Tree(NP, [cows])]},
\]
Representing Syntactic Structure

A **Tree** represents a phrase:
- **tag** — What kind of phrase (e.g., S, NP, VP)
- **branches** — Sequence of Tree or Leaf components

A **Leaf** represents a single word:
- **tag** — What kind of word (e.g., N, V)
- **word** — The word

cows = Leaf('N', 'cows')
intimidate = Leaf('V', 'intimidate')
S, NP, VP = 'S', 'NP', 'VP'
Tree(S, [Tree(NP, [cows]),
      Tree(VP, [intimidate])])
Representing Syntactic Structure

A **Tree** represents a phrase:
- **tag** — What kind of phrase (e.g., S, NP, VP)
- **branches** — Sequence of Tree or Leaf components

A **Leaf** represents a single word:
- **tag** — What kind of word (e.g., N, V)
- **word** — The word

```
cows = Leaf('N', 'cows')
intimidate = Leaf('V', 'intimidate')
S, NP, VP = 'S', 'NP', 'VP'
Tree(S, [Tree(NP, [cows]),
         Tree(VP, [intimidate,
                     Tree(NP, [cows])])])
```
Representing Syntactic Structure

A **Tree** represents a phrase:
- **tag** -- What kind of phrase (e.g., S, NP, VP)
- **branches** -- Sequence of Tree or Leaf components

A **Leaf** represents a single word:
- **tag** -- What kind of word (e.g., N, V)
- **word** -- The word

```
cows = Leaf('N', 'cows')
intimidate = Leaf('V', 'intimidate')
S, NP, VP = 'S', 'NP', 'VP'
Tree(S, [Tree(NP, [cows]),
       Tree(VP, [intimidate,
                Tree(NP, [cows])]))])
```

(Demo)
Grammars
Context-Free Grammar Rules
Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words.
Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words

\[ S \rightarrow NP \quad VP \]
A grammar rule describes how a tag can be expanded as a sequence of tags or words:

A \textit{Sentence} ...
A grammar rule describes how a tag can be expanded as a sequence of tags or words.

A sentence ...

... can be expanded as ...

Context-Free Grammar Rules
A grammar rule describes how a tag can be expanded as a sequence of tags or words...
Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words.

A Sentence ...

... can be expanded as ...

... a Noun Phrase then a Verb Phrase.
Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words.

A Sentence ...

... can be expanded as ...

... a Noun Phrase then a Verb Phrase.

Grammar

\[ S \rightarrow NP \ VP \]
Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words.

A Sentence ...

... can be expanded as ...

... a Noun Phrase then a Verb Phrase.

Grammar

\[ S \rightarrow NP \ VP \]
Grammar

A grammar rule describes how a tag can be expanded as a sequence of tags or words.

A Sentence ...

... can be expanded as ...

... a Noun Phrase then a Verb Phrase.

Grammar

\[ S \rightarrow NP \; VP \]
Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words.

A Sentence ...

... can be expanded as ...

... a Noun Phrase then a Verb Phrase.

Grammar

\[
S \rightarrow NP \ VP \\
NP \rightarrow N
\]
A grammar rule describes how a tag can be expanded as a sequence of tags or words.

A Sentence ...

... can be expanded as ...

... a Noun Phrase then a Verb Phrase.

Grammar:

- S → NP VP
- NP → N
A grammar rule describes how a tag can be expanded as a sequence of tags or words.

**Context-Free Grammar Rules**

A sentence ... can be expanded as ...

... a Noun Phrase then a Verb Phrase.

<table>
<thead>
<tr>
<th>Grammar</th>
</tr>
</thead>
<tbody>
<tr>
<td>S → NP VP</td>
</tr>
<tr>
<td>NP → N</td>
</tr>
<tr>
<td>N → COWS</td>
</tr>
</tbody>
</table>
Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words.

**Grammar**

- S → NP VP
- NP → N
- N → COWS

A *Sentence* ...

... can be expanded as ...

... a *Noun Phrase* then a *Verb Phrase*.
Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words.

A Sentence ...

... can be expanded as ...

... a Noun Phrase then a Verb Phrase.

Grammar

| S       | → | NP  | VP |
| NP      | → | N   |
| N       | → | COWS |

COWS
Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words.

A Sentence ...

... can be expanded as ...

... a Noun Phrase then a Verb Phrase.

Grammar

<table>
<thead>
<tr>
<th>Rule</th>
<th>TAGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S →</td>
<td>NP</td>
</tr>
<tr>
<td>NP →</td>
<td>N</td>
</tr>
<tr>
<td>N</td>
<td>cows</td>
</tr>
<tr>
<td>VP →</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td>NP</td>
</tr>
</tbody>
</table>

Example:

S → NP VP
NP → N
cows VP → V NP

COWS
A grammar rule describes how a tag can be expanded as a sequence of tags or words.

A Sentence ...

... can be expanded as ...

... a Noun Phrase then a Verb Phrase.

<table>
<thead>
<tr>
<th>Grammar</th>
</tr>
</thead>
<tbody>
<tr>
<td>S → NP VP</td>
</tr>
<tr>
<td>NP → N</td>
</tr>
<tr>
<td>N → COWS</td>
</tr>
<tr>
<td>VP → V NP</td>
</tr>
</tbody>
</table>
**Context-Free Grammar Rules**

A grammar rule describes how a tag can be expanded as a sequence of tags or words.

A *Sentence* ...

... can be expanded as ...

... a *Noun Phrase* then a *Verb Phrase*.

**Grammar**

- **S** → **NP** **VP**
- **NP** → **N**
- **N** → *cows*
- **VP** → **V** **NP**
- **V** → *intimidate*
Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words.

A *Sentence* ...

... can be expanded as ...

... a *Noun Phrase* then a *Verb Phrase*.

<table>
<thead>
<tr>
<th>Grammar</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>S</strong> → <strong>NP</strong> <strong>VP</strong></td>
</tr>
<tr>
<td><strong>NP</strong> → <strong>N</strong></td>
</tr>
<tr>
<td><strong>N</strong> → <em>cows</em></td>
</tr>
<tr>
<td><strong>VP</strong> → <strong>V</strong> <strong>NP</strong></td>
</tr>
<tr>
<td><strong>V</strong> → <em>intimidate</em></td>
</tr>
</tbody>
</table>
Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words.

A Sentence ...

... can be expanded as ...

... a Noun Phrase then a Verb Phrase.

### Grammar

- **S** → **NP** **VP**
- **NP** → **N**
- **N** → *cows*
- **VP** → **V** **NP**
- **V** → *intimidate*
Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words.

\[ S \rightarrow NP \ VP \]
\[ NP \rightarrow N \]
\[ N \rightarrow \text{cows} \]
\[ VP \rightarrow V \ NP \]
\[ V \rightarrow \text{intimidate} \]

A Sentence ...

... can be expanded as ...

... a Noun Phrase then a Verb Phrase.

cows intimidate
Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words.

A **Sentence** ...

... can be expanded as ...

... a **Noun Phrase** then a **Verb Phrase**.

**Grammar**

\[
\begin{align*}
S & \rightarrow NP \ VP \\
NP & \rightarrow N \\
N & \rightarrow cows \\
VP & \rightarrow V \ NP \\
V & \rightarrow intimidate
\end{align*}
\]
Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words.

A Sentence ...

... can be expanded as ...

... a Noun Phrase then a Verb Phrase.

<table>
<thead>
<tr>
<th>Grammar</th>
</tr>
</thead>
<tbody>
<tr>
<td>S ⟷ NP VP</td>
</tr>
<tr>
<td>NP ⟷ N</td>
</tr>
<tr>
<td>N ⟷ cows</td>
</tr>
<tr>
<td>VP ⟷ V NP</td>
</tr>
<tr>
<td>V ⟷ intimidate</td>
</tr>
</tbody>
</table>
Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words.

A Sentence ...

... can be expanded as ...

... a Noun Phrase then a Verb Phrase.

Grammar

<table>
<thead>
<tr>
<th>Rule</th>
<th>Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>S → NP VP</td>
<td></td>
</tr>
<tr>
<td>NP → N</td>
<td></td>
</tr>
<tr>
<td>N → cows</td>
<td></td>
</tr>
<tr>
<td>VP → V NP</td>
<td></td>
</tr>
<tr>
<td>V → intimidate</td>
<td></td>
</tr>
</tbody>
</table>

(Demo)
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input

buffalo  buffalo  buffalo  buffalo
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input

buffalo  buffalo  buffalo  buffalo
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input

\[ S \]

buffalo  buffalo  buffalo  buffalo
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input

---

0 1 2 3 4

buffalo buffalo buffalo buffalo

---
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input

```
0 1 2 3 4
buffalo buffalo buffalo buffalo
```

```
NP   S   VP
```
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input

\[
\text{NP} \quad \text{S} \quad \text{VP}
\]
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input

```
S
  NP
  VP
  buffalo
  buffalo
  buffalo
  buffalo
```
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input

Constraint: A **Leaf** must match the input word
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input

Constraint:
A **Leaf** must match the input word
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input

Constraint: A Leaf must match the input word
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input

```
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td></td>
<td>VP</td>
<td></td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>buffalo</td>
<td></td>
<td>buffalo</td>
<td></td>
</tr>
</tbody>
</table>
```
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input

```
<table>
<thead>
<tr>
<th></th>
<th></th>
<th>S</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NP</td>
<td></td>
<td></td>
<td>VP</td>
<td></td>
</tr>
</tbody>
</table>
```

```
0     buffalo  1     buffalo  2     buffalo  3     buffalo  4
```
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input.
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input.
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input.
Learning

(Demo)
Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common
Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

teacher strikes idle kids
Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

```
S
```

teacher strikes idle kids
Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

teacher strikes idle kids
Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

```
teacher strikes idle kids
```
Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

```
S    NP    VP
NP    NN    NNS
VP    VB    NP
NP    NNS
NN    teacher
NNS   strikes
VB    idle
NNS   kids
```
Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

```
S       NP       VP
   |       |       |
  |       |       |
|       |       |
  |       |       |
|       |       |
NN   NNS   VB   NNS
```

*teacher* strikes *idle* *kids*

**Rule frequency per 100,000 tags**

- $S \rightarrow NP \ VP$
- $NP \rightarrow NN \ NNS$
- $VP \rightarrow VB \ NP$
- $NP \rightarrow NNS$
- $NN \rightarrow teacher$
- $NNS \rightarrow strikes$
- $VB \rightarrow idle$
- $NNS \rightarrow kids$
Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

```
S      VP
   NP
     NN NNS VB NNS
```

teacher strikes idle kids

Rule frequency per 100,000 tags

- $S \rightarrow NP \ VP$: 25372
- $NN \rightarrow$ teacher
- $NP \rightarrow NN \ NNS$
- $NNS \rightarrow$ strikes
- $VP \rightarrow VB \ NP$
- $VB \rightarrow$ idle
- $NP \rightarrow NNS$
- $NNS \rightarrow$ kids
Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

```
S → NP VP
NP → NN NNS
VP → VB NP
NP → NNS

teacher strikes idle kids

Rule frequency per 100,000 tags

S → NP VP 25372
NP → NN NNS 1335
VP → VB NP
NP → NNS

NN → teacher
NNS → strikes
VB → idle
NNS → kids
```
Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

```
S   NP   VP

NP  NN  NNS  VB  NNS
```

teacher strikes idle kids

Rule frequency per 100,000 tags

<table>
<thead>
<tr>
<th>Rule</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>S → NP VP</td>
<td>25372</td>
</tr>
<tr>
<td>NP → NN NNS</td>
<td>1335</td>
</tr>
<tr>
<td>VP → VB NP</td>
<td>6679</td>
</tr>
<tr>
<td>NP → NNS</td>
<td></td>
</tr>
<tr>
<td>NN → teacher</td>
<td></td>
</tr>
<tr>
<td>NNS → strikes</td>
<td></td>
</tr>
<tr>
<td>VB → idle</td>
<td></td>
</tr>
<tr>
<td>NNS → kids</td>
<td></td>
</tr>
</tbody>
</table>
Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

```
S  --->  NP  VP
NP --->  NN  NNS
VP --->  VB  NP
NP --->  NNS
```

Rule frequency per 100,000 tags

- $S \rightarrow NP \ VP$: 25372
- $NP \rightarrow NN \ NNS$: 1335
- $VP \rightarrow VB \ NP$: 6679
- $NP \rightarrow NNS$: 4282
- $NN \rightarrow teacher$: 25372
- $NNS \rightarrow strikes$: 1335
- $VB \rightarrow idle$: 6679
- $NNS \rightarrow kids$: 4282
Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

```
S ───> NP  VP
NP ───> NN  NNS
NP ───> NNS
VP ───> VB  NP

teacher strikes idle kids
```

Rule frequency per 100,000 tags

<table>
<thead>
<tr>
<th>Rule</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>S → NP VP</td>
<td>25,372</td>
</tr>
<tr>
<td>NP → NN NNS</td>
<td>1,335</td>
</tr>
<tr>
<td>VP → VB NP</td>
<td>6,679</td>
</tr>
<tr>
<td>NP → NNS</td>
<td>4,282</td>
</tr>
<tr>
<td>NN → teacher</td>
<td>5</td>
</tr>
<tr>
<td>NNS → strikes</td>
<td></td>
</tr>
<tr>
<td>VB → idle</td>
<td></td>
</tr>
<tr>
<td>NNS → kids</td>
<td></td>
</tr>
</tbody>
</table>
Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

```
S ---- NP ---- VP
   |      |      |
   |      |      |
   |      |      |
  NN   NNS   VB   NNS

teacher strikes idle kids
```

Rule frequency per 100,000 tags

<table>
<thead>
<tr>
<th>Rule</th>
<th>Frequency</th>
<th>Headword</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>S → NP VP</td>
<td>25372</td>
<td>teacher</td>
<td>5</td>
</tr>
<tr>
<td>NP → NN NNS</td>
<td>1335</td>
<td>strikes</td>
<td>25</td>
</tr>
<tr>
<td>VP → VB NP</td>
<td>6679</td>
<td>idle</td>
<td></td>
</tr>
<tr>
<td>NP → NNS</td>
<td>4282</td>
<td>kids</td>
<td></td>
</tr>
</tbody>
</table>
Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

```
S  -->  NP  VP
NN  -->  teacher  5
NP  -->  NN  NNS
1335  NNS  -->  strikes  25
VP  -->  VB  NP
6679  VB  -->  idle  26
NP  -->  NNS
4282  NNS  -->  kids
```
Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

teacher strikes idle kids

Rule frequency per 100,000 tags

S → NP VP  25372  NN → teacher  5
NP → NN NNS  1335  NNS → strikes  25
VP → VB NP  6679  VB → idle  26
NP → NNS  4282  NNS → kids  32
Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

```
S

NP  VP

NN  VBZ  JJ  NNS
```

teacher strikes idle kids

**Rule frequency per 100,000 tags**

<table>
<thead>
<tr>
<th>Grammar</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>S → NP  VP</td>
<td>25372</td>
</tr>
<tr>
<td>NP → NN</td>
<td>1335 4358</td>
</tr>
<tr>
<td>VP → VBZ  NP</td>
<td>6679 3160</td>
</tr>
<tr>
<td>NP → JJ  NNS</td>
<td>4282 2526</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Grammar</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN → teacher</td>
<td>5</td>
</tr>
<tr>
<td>VBZ → strikes</td>
<td>25 19</td>
</tr>
<tr>
<td>JJ → idle</td>
<td>26 18</td>
</tr>
<tr>
<td>NNS → kids</td>
<td>32</td>
</tr>
</tbody>
</table>
Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

```
S  ---  NP  VP
NP  ---  VP  NP
  NN  VBZ  JJ  NNS
```

teacher strikes idle kids

**Rule frequency per 100,000 tags**

<table>
<thead>
<tr>
<th>Rule</th>
<th>Frequency</th>
<th>Tag</th>
<th>Frequency</th>
<th>Tag</th>
<th>Frequency</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>S → NP VP</td>
<td>25372</td>
<td>teacher</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP → NN</td>
<td>1335</td>
<td>4358</td>
<td>VBZ → strikes</td>
<td>25</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>VP → VBZ NP</td>
<td>6679</td>
<td>3160</td>
<td>JJ → idle</td>
<td>26</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>NP → JJ NNS</td>
<td>4282</td>
<td>2526</td>
<td>NNS → kids</td>
<td>32</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Translation
Syntactic Reordering
Syntactic Reordering

English
Syntactic Reordering

English → Yoda-English
Syntactic Reordering

English → Yoda–English

Help you, I can!
Yes! Mm!
Syntactic Reordering

English → Yoda-English

Help you, I can!
Yes! Mm!

When 900 years old you reach,
look as good, you will not. Hm.
Syntactic Reordering

English   $\rightarrow$   Yoda-English

S

NP     VP

PRP   MD   VP

I    can    VB    PRP

help    you

Help you, I can! Yes! Mm!

When 900 years old you reach, look as good, you will not. Hm.
Syntactic Reordering

English → Yoda-English

```
S
  /\   \\
 VP /  \\
  /   \\
 NP /    \\
  /     \\
 PRP /      \\
   /        \\
 I      can
```

Help you, I can! Yes! Mm!

When 900 years old you reach, look as good, you will not. Hm.
Syntactic Reordering

English → Yoda–English

```
S  
|  
| VP  NP  VP
|     |     |
| VB  PRP  PRP  MD
| help  you  I  can
```

Help you, I can! Yes! Mm!

When 900 years old you reach, look as good, you will not. Hm.
Help you, I can!
Yes! Mm!

When 900 years old you reach, look as good, you will not. Hm.
Syntactic Reordering

English  \rightarrow  Yoda-English

S

VP

VB  PRP

help  you

NP

PRP

I

VP

MD

can

Help you, I can!
Yes! Mm!

When 900 years old you reach,
look as good, you will not. Hm.

(Demo)