Announcements
Ambiguity
Syntactic Ambiguity in English

Programs must be written for people to read
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

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

Sentence

Noun Phrase

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

Preface 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

1Preface of *Structure and Interpretation of Computer Programs*
in 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
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
Syntactic Ambiguity in English

**pro•gram** (noun)
a series of coded software instructions

**pro•gram** (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
Representing Syntactic Structure

Noun

COWS
Representing Syntactic Structure
Representing Syntactic Structure

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

Noun

cows

intimidate
Representing Syntactic Structure

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

cows intimidate
Representing Syntactic Structure

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

Sentence

Noun cows Verb intimidate Noun cows

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

cows intimidate cows

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

Sentence

Noun Phrase

Verb Phrase

Noun: cows
Verb: intimidate
Noun: 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

A Tree represents a phrase:

Sentence
Noun Phrase
Verb Phrase
Noun Phrase
Noun
Verb
Noun

cows
intimidate
cows
Representing Syntactic Structure

A Tree represents a phrase:

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

![Image](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:
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 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
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](http://creativecommons.org/licenses/by-nc-nd/2.0/) licensed under [http://creativecommons.org/licenses/by-nc-nd/2.0/](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

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

\[
cows = \text{Leaf('N', 'cows')}
\]
\[
\text{intimidate} = \text{Leaf('V', 'intimidate')}
\]
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'

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

$$\text{cows} = \text{Leaf('N', 'cows')}$$
$$\text{intimidate} = \text{Leaf('V', 'intimidate')}$$
$$\text{S, NP, VP} = 'S', 'NP', 'VP$$

$$\text{Tree(S, [Tree(NP, [cows])]}$$
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]),
  Tree(VP, [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

```
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

```plaintext
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

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

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

A Sentence ...
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 ...

S → 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**.
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

\[ S \rightarrow NP \ VP \]
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
```
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

\[
\begin{align*}
S & \rightarrow \text{NP VP} \\
\text{NP} & \rightarrow \text{N}
\end{align*}
\]
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.

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>
</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.

<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>

Diagram:

- S → NP VP
- NP → N
- N → COWS

Diagram:

- S
- NP
- VP
- N

Diagram:

- S → NP VP
- NP
- VP
- N
- 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>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>NP VP</td>
</tr>
<tr>
<td>NP</td>
<td>N</td>
</tr>
<tr>
<td>N</td>
<td>COWS</td>
</tr>
</tbody>
</table>

Diagram:

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

**Grammar**

- \( S \rightarrow NP \ VP \)
- \( NP \rightarrow N \)
- \( N \rightarrow COWS \)

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

<table>
<thead>
<tr>
<th>Tag</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>NP VP</td>
</tr>
<tr>
<td>NP</td>
<td>N</td>
</tr>
<tr>
<td>N</td>
<td>COWS</td>
</tr>
<tr>
<td>VP</td>
<td>V NP</td>
</tr>
</tbody>
</table>

NP 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**

- $S \rightarrow NP \ VP$
- $NP \rightarrow N$
- $N \rightarrow \text{cows}$
- $VP \rightarrow V \ NP$
Context-Free Grammar Rules

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

\[ S \rightarrow NP \ VP \]

A *Sentence* ...

... can be expanded as ...

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

**Grammar**

\[
\begin{align*}
S & \rightarrow NP \ VP \\
NP & \rightarrow N \\
N & \rightarrow \text{cows} \\
VP & \rightarrow V \ NP \\
V & \rightarrow \text{intimidate}
\end{align*}
\]
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.

Grammar

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

S → NP VP
NP → N
N → cows
VP → V NP
V → intimidate
A grammar rule describes how a tag can be expanded as a sequence of tags or words:

\[ S \rightarrow NP \quad VP \]

A **Sentence** ...

... can be expanded as ...

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

**Grammar**

- \( S \rightarrow NP \quad VP \)
- \( NP \rightarrow N \)
- \( N \rightarrow cows \)
- \( VP \rightarrow V \quad NP \)
- \( V \rightarrow intimidate \)
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>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

\[
\begin{align*}
S & \rightarrow \text{NP} \ \text{VP} \\
\text{NP} & \rightarrow N \\
N & \rightarrow \text{cows} \\
\text{VP} & \rightarrow V \ \text{NP} \\
V & \rightarrow \text{intimidate}
\end{align*}
\]
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*.

Grammar

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

Example: cows intimidate 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

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>→</td>
</tr>
<tr>
<td>NP</td>
<td>→</td>
</tr>
<tr>
<td>N</td>
<td>→</td>
</tr>
<tr>
<td>VP</td>
<td>→</td>
</tr>
<tr>
<td>V</td>
<td>→</td>
</tr>
</tbody>
</table>

(Demo)
Parsing
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

S

buffalo  buffalo  buffalo  buffalo
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input

buffero  buffero  buffero  buffero
Exhaustive Parsing

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

Expand all tags recursively, but constrain words to match input

```
NP
```

```
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

```
S
  NP
  VP

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
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input

Constraint: A Leaf must match the input word

buffalo buffalo buffalo buffalo
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input

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

```
0  1  2  3  4
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

S

NP

VP

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
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

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

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

teacher strikes idle kids

Rule frequency per 100,000 tags

- $S \rightarrow NP\ VP$  
  - $NN \rightarrow$ teacher
  - $NNS \rightarrow$ strikes
  - $VB \rightarrow$ idle
  - $NNS \rightarrow$ kids

- $NP \rightarrow NN\ NNS$
Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

```
S ---------
NP     VP

NP     NN     NNS

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></td>
</tr>
<tr>
<td>NP → NNS</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rule</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<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

teacher  strikes  idle  kids
```

**Rule frequency per 100,000 tags**

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

Not all syntactic structures are equally common

```
S ——— NNS ——— NN ——— NP ——— VP ——— 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>4282</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

teacher strikes idle kids

Rule frequency per 100,000 tags

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

Not all syntactic structures are equally common

teacher strikes idle kids

Rule frequency per 100,000 tags

<table>
<thead>
<tr>
<th>Rule</th>
<th>Frequency</th>
<th>Word</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

teacher strikes idle kids

Rule frequency per 100,000 tags

<table>
<thead>
<tr>
<th>Rule</th>
<th>Frequency</th>
<th>Word</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>26</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   NNS  VB  NNS
```

teacher strikes idle kids

Rule frequency per 100,000 tags

<table>
<thead>
<tr>
<th>Rule</th>
<th>Frequency</th>
<th>Word</th>
<th>Count</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>26</td>
</tr>
<tr>
<td>NP  →  NNS</td>
<td>4282</td>
<td>kids</td>
<td>32</td>
</tr>
</tbody>
</table>
Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

teacher strikes idle kids

Rule frequency per 100,000 tags

<table>
<thead>
<tr>
<th>Rule</th>
<th>Frequency 1</th>
<th>Frequency 2</th>
<th>Rule</th>
<th>Frequency 3</th>
<th>Frequency 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>S → NP VP</td>
<td>25372</td>
<td></td>
<td>NN → teacher</td>
<td>5</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>
</tr>
<tr>
<td>VP → VBZ NP</td>
<td>6679</td>
<td>3160</td>
<td>JJ → idle</td>
<td>26</td>
<td>18</td>
</tr>
<tr>
<td>NP → JJ NNS</td>
<td>4282</td>
<td>2526</td>
<td>NNS → kids</td>
<td>32</td>
<td></td>
</tr>
</tbody>
</table>
Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

teacher strikes idle kids

Rule frequency per 100,000 tags

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

(Demo)
Translation
Syntactic Reordering
Syntactic Reordering

English
Syntactic Reordering

English \rightarrow Yoda-English
Syntactic Reordering

English \hspace{1cm} \rightarrow \hspace{1cm} 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.
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 → Yoda-English

S
 /     \   
VP    NP  VP
 /    /    |
VB  PRP  PRP  MD
   you    I    can

Help you, I can! Yes! Mm!

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

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

English → Yoda-English

S

VP

VP

np

PRP

PRP

MD

I can

English sentence:

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

Yoda-English:

Help you, I can! Yes! Mm!

Demo

(Demo)

Demo image: Yoda from Star Wars.