Wednesday, April 29
Announcements
Announcements

• Homework 9 (4 pts) due Wednesday 4/29 @ 11:59pm
Announcements

• Homework 9 (4 pts) due Wednesday 4/29 @ 11:59pm
• Quiz 4 due Thursday 4/30 @ 11:59pm
Announcements

• Homework 9 (4 pts) due Wednesday 4/29 @ 11:59pm
• Quiz 4 due Thursday 4/30 @ 11:59pm
• No videos on Friday 5/1; Come to lecture (and fill out the HKN course survey at the end)
Announcements

• Homework 9 (4 pts) due Wednesday 4/29 @ 11:59pm
• Quiz 4 due Thursday 4/30 @ 11:59pm
• No videos on Friday 5/1; Come to lecture (and fill out the HKN course survey at the end)
  ▪ If at least 60% of students respond, everyone gets an extra credit point
Announcements

• Homework 9 (4 pts) due Wednesday 4/29 @ 11:59pm
• Quiz 4 due Thursday 4/30 @ 11:59pm
• No videos on Friday 5/1; Come to lecture (and fill out the HKN course survey at the end)
  ▪ If at least 60% of students respond, everyone gets an extra credit point
• Next week: 18 hours of review sessions Monday, Tuesday, & Wednesday 11–5 in 271/273 Soda
Announcements

• Homework 9 (4 pts) due Wednesday 4/29 @ 11:59pm
• Quiz 4 due Thursday 4/30 @ 11:59pm
• No videos on Friday 5/1; Come to lecture (and fill out the HKN course survey at the end)
  ▪ If at least 60% of students respond, everyone gets an extra credit point
• Next week: 18 hours of review sessions Monday, Tuesday, & Wednesday 11–5 in 271/273 Soda
  ▪ Two TAs are available every hour
Announcements

- Homework 9 (4 pts) due Wednesday 4/29 @ 11:59pm
- Quiz 4 due Thursday 4/30 @ 11:59pm
- No videos on Friday 5/1; Come to lecture (and fill out the HKN course survey at the end)
  - If at least 60% of students respond, everyone gets an extra credit point
- Next week: 18 hours of review sessions Monday, Tuesday, & Wednesday 11–5 in 271/273 Soda
  - Two TAs are available every hour
  - One room will be a review session going over topic-specific problems
Announcements

• Homework 9 (4 pts) due Wednesday 4/29 @ 11:59pm
• Quiz 4 due Thursday 4/30 @ 11:59pm
• No videos on Friday 5/1; Come to lecture (and fill out the HKN course survey at the end)
  ▪ If at least 60% of students respond, everyone gets an extra credit point
• Next week: 18 hours of review sessions Monday, Tuesday, & Wednesday 11–5 in 271/273 Soda
  ▪ Two TAs are available every hour
  ▪ One room will be a review session going over topic-specific problems
  ▪ The other room is unstructured; staff will answer any questions you have
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 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

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\(^1\)

\(^1\)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

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 read¹

¹Preface 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

**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
Representing Syntactic Structure
Representing Syntactic Structure

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

COWS
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

Noun

Verb

cows

intimidate

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

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

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

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/

cows
intimidate

cows
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**
    - **Noun**: cows
  - **Verb Phrase**
    - **Verb**: intimidate
  - **Noun Phrase**
    - **Noun**: cows

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

---

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:
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)
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 https://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:

- **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
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

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

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

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

Example:
- **Tagged**: cows = Leaf('N', 'cows')
- **Tagged**: intimidate = Leaf('V', 'intimidate')

Sentence structure:
- $S$, $NP$, $VP$ = 'S', 'NP', 'VP'
- $Tree(S, [Tree(NP, [cows]), Verb(intimidate)])$
Representing Syntactic Structure

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

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

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

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

\[
\begin{align*}
\text{cows} &= \text{Leaf('N', 'cows')} \\
\text{intimidate} &= \text{Leaf('V', 'intimidate')} \\
\text{S, NP, VP} &= 'S', 'NP', 'VP' \\
\text{Tree(S, [Tree(NP, [cows]),)} \\
\text{Tree(VP, [intimidate,)} \\
\text{Tree(NP, [cows]]])}
\end{align*}
\]

(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

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

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

A Sentence ...

\[ S \rightarrow NP \, VP \]
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 ...

\[ S \rightarrow NP \ VP \]
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

\[ S \rightarrow NP \, VP \]
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.

<table>
<thead>
<tr>
<th>Grammar</th>
</tr>
</thead>
<tbody>
<tr>
<td>S → NP VP</td>
</tr>
<tr>
<td>NP → 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>
</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 → buffalo</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

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

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.

\[
\begin{align*}
S & \rightarrow NP \ VP \\
NP & \rightarrow N \\
N & \rightarrow \text{buffalo}
\end{align*}
\]
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 → buffalo</td>
</tr>
<tr>
<td>VP → V NP</td>
</tr>
</tbody>
</table>

buffalo
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 → buffalo
VP → V NP
```

A *Sentence* ...  

... can be expanded as ...  

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

```
NP S VP NP
V
```

**buffalo**
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.

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

- $S \rightarrow NP \ VP$
- $NP \rightarrow N$
- $N \rightarrow \text{buffalo}$
- $VP \rightarrow V \ NP$
- $V \rightarrow \text{buffalo}$
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

\[
\begin{align*}
S & \rightarrow \text{NP} \text{ VP} \\
\text{NP} & \rightarrow \text{N} \\
\text{N} & \rightarrow \text{buffalo} \\
\text{VP} & \rightarrow \text{V} \text{ NP} \\
\text{V} & \rightarrow \text{buffalo}
\end{align*}
\]
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 → buffalo</td>
</tr>
<tr>
<td>VP → V NP</td>
</tr>
<tr>
<td>V → buffalo</td>
</tr>
</tbody>
</table>
```

```text

... 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.

\[
\text{Grammar}
\]

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

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.

<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 → buffalo</td>
</tr>
<tr>
<td>VP → V NP</td>
</tr>
<tr>
<td>V → buffalo</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 → buffalo</td>
</tr>
<tr>
<td>VP → V NP</td>
</tr>
<tr>
<td>V → buffalo</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
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
```

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

Expand all tags recursively, but constrain words to match input

```
NP   S   VP
```

```
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

NP  S  VP

buffalo  buffalo  buffalo  buffalo

0  1  2  3  4
```
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input

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

Expand all tags recursively, but constrain words to match input

```
NP  S  VP
buffalo  buffalo  buffalo  buffalo
0  1  2  3  4
```
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input

Constraint: A Leaf must match the input word

buffalo buffalo buffalo buffalo

0 1 2 3 4
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input

Constraint: A Leaf must match the input word

buffalo  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
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
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
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

\[
\begin{align*}
S & \rightarrow \ NP \ VP \\
NP & \rightarrow \ NN \ NNS \\
VP & \rightarrow \ VB \ NP \\
NP & \rightarrow \ NNS
\end{align*}
\]

\[
\begin{align*}
NN & \rightarrow \ teacher \\
NNS & \rightarrow \ strikes \\
VB & \rightarrow \ idle \\
NNS & \rightarrow \ kids
\end{align*}
\]
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*

```
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 ────┘
```

teacher strikes idle kids

Rule frequency per 100,000 tags

\[
\begin{align*}
S & \rightarrow \text{NP } \text{VP} & 25372 \\
\text{NP} & \rightarrow \text{NN } \text{NNS} \\
\text{VP} & \rightarrow \text{VB } \text{NP} \\
\text{NP} & \rightarrow \text{NNS}
\end{align*}
\]

\[
\begin{align*}
\text{NN} & \rightarrow \text{teacher} \\
\text{NNS} & \rightarrow \text{strikes} \\
\text{VB} & \rightarrow \text{idle} \\
\text{NNS} & \rightarrow \text{kids}
\end{align*}
\]
Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

\[
S \rightarrow NP \ VP \\
NP \rightarrow NN \ NNS \\
VP \rightarrow VB \ NP \\
NP \rightarrow NNS
\]

\[
NN \quad NNS \quad VB \quad NNS
\]

Rule frequency per 100,000 tags

- \( S \rightarrow NP \ VP \): 25372
- \( NP \rightarrow NN \ NNS \): 1335
- \( VP \rightarrow VB \ NP \):
- \( NP \rightarrow NNS \):
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

- \( S \rightarrow \text{NP VP} \) 25372  \( \text{NN} \rightarrow \text{teacher} \)
- \( \text{NP} \rightarrow \text{NN NNS} \) 1335  \( \text{NNS} \rightarrow \text{strikes} \)
- \( \text{VP} \rightarrow \text{VB NP} \) 6679  \( \text{VB} \rightarrow \text{idle} \)
- \( \text{NP} \rightarrow \text{NNS} \)  \( \text{NNS} \rightarrow \text{kids} \)
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

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

25372  NN ➔ teacher  5
1335  NNS ➔ strikes
6679  VB ➔ idle
4282  NNS ➔ kids
```

teacher strikes idle kids
Not all syntactic structures are equally common

teacher strikes idle kids

Rule frequency per 100,000 tags

| Rule            | Frequency | | Rule   | Word      | Count |
|-----------------|-----------|------------------|--------|
| $S \rightarrow \text{ NP VP}$ | 25372     | $\text{ NN } \rightarrow$ teacher | 5      |
| $\text{ NP } \rightarrow \text{ NN NNS}$ | 1335      | $\text{ NNS } \rightarrow$ strikes | 25     |
| $\text{ VP } \rightarrow \text{ VB NP}$ | 6679      | $\text{ VB } \rightarrow$ idle    |        |
| $\text{ NP } \rightarrow \text{ NNS}$ | 4282      | $\text{ NNS } \rightarrow$ kids   |        |
Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

```
S  -->  NP  VP
25372

NP  -->  NN  NNS
1335

VP  -->  VB  NP
6679

NP  -->  NNS
4282
```

Rule frequency per 100,000 tags

```
NN  -->  teacher  5

NNS  -->  strikes  25

VB  -->  idle  26

NNS  -->  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

<table>
<thead>
<tr>
<th>Structure</th>
<th>Frequency</th>
<th>Tag</th>
<th>Word</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>S  ----&gt; NP  VP</td>
<td>25372</td>
<td>NN</td>
<td>teacher</td>
<td>5</td>
</tr>
<tr>
<td>NP  ----&gt; NN  NNS</td>
<td>1335</td>
<td>NNS</td>
<td>strikes</td>
<td>25</td>
</tr>
<tr>
<td>VP  ----&gt; VB  NP</td>
<td>6679</td>
<td>VB</td>
<td>idle</td>
<td>26</td>
</tr>
<tr>
<td>NP  ----&gt; NNS</td>
<td>4282</td>
<td>NNS</td>
<td>kids</td>
<td>32</td>
</tr>
</tbody>
</table>
Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

```
S      NP      VP
teacher strikes idle kids
```

Rule frequency per 100,000 tags

<table>
<thead>
<tr>
<th>Rule</th>
<th>Frequency</th>
<th>Tag 1</th>
<th>Tag 2</th>
<th>Tag 3</th>
<th>Tag 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>S → NP VP</td>
<td>25372</td>
<td>NN</td>
<td>teacher</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>NP → NN</td>
<td>4358</td>
<td>VBZ</td>
<td>strikes</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>VP → VBZ NP</td>
<td>3160</td>
<td>JJ</td>
<td>idle</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>NP → JJ NNS</td>
<td>2526</td>
<td>NNS</td>
<td>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>Rule</th>
<th>Frequency 1</th>
<th>Frequency 2</th>
<th>Rule</th>
<th>Frequency 1</th>
<th>Frequency 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>25372</td>
<td>NN</td>
<td>teacher</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>1335</td>
<td>VBZ</td>
<td>strikes</td>
<td>25</td>
<td>19</td>
</tr>
<tr>
<td>VP</td>
<td>6679</td>
<td>JJ</td>
<td>idle</td>
<td>26</td>
<td>18</td>
</tr>
<tr>
<td>NP</td>
<td>4282</td>
<td>NNS</td>
<td>kids</td>
<td>32</td>
<td></td>
</tr>
</tbody>
</table>

(Demo)