research = re- + *kikro-
Revisit the Development of Probabilistic Symbol-Refined Grammars

Weiwei Sun
Wangxuan Institute of Computer Technology
Peking University

@StudentWorkshop.NLPCC2019
The etymological meaning of *research*

```
research
  └── re-

search
  └── Latin
      └── circus
          └── PIE
              └── *kikro-
                  2×*sker-(cut)
```

from http://www.etymonline.com/
The etymological meaning of *research*

**research**

- **re-**
- **search**
  - Latin
  - *circus*
  - PIE
  - *kikro-*
    - 2×*sker-* (cut)

This talk

A historical introduction to one type of parsing model

1. Parsing: An (Unnecessary?) Introduction
2. PCFG: Early Failure
3. Lexicalization: A Breakthrough Technique
4. Alternatives? Simple PCFG Again
This talk

A historical introduction to one type of parsing model

1. Parsing: An (Unnecessary?) Introduction
2. PCFG: Early Failure
3. Lexicalization: A Breakthrough Technique
4. Alternatives? Simple PCFG Again

some thoughts
Outline

1. Parsing: An (Unnecessary?) Introduction
2. PCFG: Early Failure
3. Lexicalization: A Breakthrough Technique
4. Alternatives? Simple PCFG Again
Let’s start from *Friends*
Let’s start from *Friends*

Well, ya know how I always wanted to **go out with Chip Matthews in high school**? Well, tonight I actually **went out with Chip Matthews in high school**.

- **go out with Chip Matthews in high school**.
- **go out in high school**.
- **Chip Matthews in high school**.
Let’s start from *Friends*

Well, ya know how I always wanted to go out with Chip Matthews in high school? Well, tonight I actually went out with Chip Matthews in high school.

*go out with Chip Matthews in high school.*

- go out in high school.
- Chip Matthews in high school.

With syntax, we can distinguish between them.
NL sentences could be really long.
NL sentences could be really long.

With syntax, we can understand very long sentences.
Structuring a Sentence

0101010101010101010101010101010101010101010101010101010101010
0110101000001001111001100110011111110011101111001100100100001
Structuring a Sentence

\[ \sqrt{2} - 1 \]
Structuring a Sentence

\[ \sqrt{2} - 1 \]

市发展改革委关于转发省发展改革委关于转发国家发展改革委办公厅关于对真抓实干成效明显地方加大中央预算内投资激励支持力度的通知的通知的通知
Structuring a Sentence

\[ \sqrt{2} - 1 \]

市发展改革委关于转发省发展改革委关于转发国家发展改革委办公厅关于对真抓实干成效明显地方加大中央预算内投资激励支持力度的通知的通知的通知
Structuring a Sentence

\[ \sqrt{2} - 1 \]

市发展改革委关于转发省发展改革委关于转发国家发展改革委办公厅关于对真抓实干成效明显地方加大中央预算内投资激励支持力度的通知的通知的通知
Structuring a Sentence

\[ \sqrt{2} - 1 \]

市发展改革委关于转发省发展改革委关于转发国家发展改革委办公厅关于对真抓实干成效明显地方加大中央预算内投资激励支持力度的通知的通知的通知
Phrase structure

The basic idea
Phrase structure organizes words into nested constituents, which can be represented as a tree.
Phrase structure

The basic idea
Phrase structure organizes words into nested constituents, which can be represented as a tree.

Example
Different structures; different meaning

VP

VP
  go out

PP
  P
  with
  NP
  C. M.

PP
  in high school
Different structures; different meaning

Parsing

To get trees automatically
Different structures; different meaning

Parsing
To get trees automatically
Outline

1 Parsing: An (Unnecessary?) Introduction
2 PCFG: Early Failure
3 Lexicalization: A Breakthrough Technique
4 Alternatives? Simple PCFG Again
search

research

re-

search

Latin

circus

PIE

*kikro-
cut, cut
Probabilistic Context-Free Grammar

1. \( V = \{S, NP, VP, AdjP, AdvP\} \cup \{N, V, Adj, Adv\} \)

2. \( T = \{\text{colorless, green, ideas, sleep, furiously}\} \)

3. \( P \)

<table>
<thead>
<tr>
<th>Rule</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S \rightarrow NP \ VP )</td>
<td>1.0</td>
</tr>
<tr>
<td>( VP \rightarrow V \ AdvP )</td>
<td>1.0</td>
</tr>
<tr>
<td>( AdvP \rightarrow Adv )</td>
<td>1.0</td>
</tr>
<tr>
<td>( NP \rightarrow AdjP \ NP )</td>
<td>0.5</td>
</tr>
<tr>
<td>( NP \rightarrow N )</td>
<td>0.5</td>
</tr>
<tr>
<td>( AdjP \rightarrow Adj )</td>
<td>1.0</td>
</tr>
<tr>
<td>( Adj \rightarrow \text{colorless} )</td>
<td>0.5</td>
</tr>
<tr>
<td>( N \rightarrow \text{ideas} )</td>
<td>1.0</td>
</tr>
<tr>
<td>( Adv \rightarrow \text{furiously} )</td>
<td>1.0</td>
</tr>
<tr>
<td>( Adj \rightarrow \text{green} )</td>
<td>0.5</td>
</tr>
<tr>
<td>( V \rightarrow \text{sleep} )</td>
<td>1.0</td>
</tr>
</tbody>
</table>

4. \( S \)
Probabilistic Context-Free Grammar

1. $V = \{S, \text{NP}, \text{VP}, \text{AdjP}, \text{AdvP}\} \cup \{\text{N}, \text{V}, \text{Adj}, \text{Adv}\}$

2. $T = \{\text{colorless, green, ideas, sleep, furiously}\}$

3. $P$

<table>
<thead>
<tr>
<th>Rule</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S \rightarrow \text{NP} \ \text{VP}$</td>
<td>1.0</td>
</tr>
<tr>
<td>$\text{VP} \rightarrow \text{V} \ \text{AdvP}$</td>
<td>1.0</td>
</tr>
<tr>
<td>$\text{AdvP} \rightarrow \text{Adv}$</td>
<td>1.0</td>
</tr>
<tr>
<td>$\text{AdjP} \rightarrow \text{Adj}$</td>
<td>1.0</td>
</tr>
<tr>
<td>$\text{Adj} \rightarrow \text{colorless}$</td>
<td>0.5</td>
</tr>
<tr>
<td>$\text{N} \rightarrow \text{ideas}$</td>
<td>1.0</td>
</tr>
<tr>
<td>$\text{Adv} \rightarrow \text{furiously}$</td>
<td>1.0</td>
</tr>
<tr>
<td>$\text{NP} \rightarrow \text{AdjP} \ \text{NP}$</td>
<td>0.5</td>
</tr>
<tr>
<td>$\text{NP} \rightarrow \text{N}$</td>
<td>0.5</td>
</tr>
<tr>
<td>$\text{V} \rightarrow \text{sleep}$</td>
<td>1.0</td>
</tr>
</tbody>
</table>

4. $S$

We can derive the structure of a string.

- $S$
Probabilistic Context-Free Grammar

1. \( V = \{S, NP, VP, AdjP, AdvP\} \cup \{N, V, Adj, Adv\} \)

2. \( T = \{\text{colorless, green, ideas, sleep, furiously}\} \)

3. \( P \)

<table>
<thead>
<tr>
<th>Rule</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S \rightarrow NP \ VP )</td>
<td>1.0</td>
</tr>
<tr>
<td>( VP \rightarrow V \ AdvP )</td>
<td>1.0</td>
</tr>
<tr>
<td>( AdvP \rightarrow Adv )</td>
<td>1.0</td>
</tr>
<tr>
<td>( Adj \rightarrow \text{colorless} )</td>
<td>0.5</td>
</tr>
<tr>
<td>( N \rightarrow \text{ideas} )</td>
<td>1.0</td>
</tr>
<tr>
<td>( Adv \rightarrow \text{furiously} )</td>
<td>1.0</td>
</tr>
<tr>
<td>( NP \rightarrow AdjP \ NP )</td>
<td>0.5</td>
</tr>
<tr>
<td>( NP \rightarrow N )</td>
<td>0.5</td>
</tr>
<tr>
<td>( AdjP \rightarrow Adj )</td>
<td>1.0</td>
</tr>
<tr>
<td>( Adj \rightarrow \text{green} )</td>
<td>0.5</td>
</tr>
<tr>
<td>( V \rightarrow \text{sleep} )</td>
<td>1.0</td>
</tr>
</tbody>
</table>

4. We can derive the structure of a string.

- \( S \Rightarrow NP \ VP \) 1.0
Probabilistic Context-Free Grammar

1. \( V = \{S, NP, VP, AdjP, AdvP\} \cup \{N, V, Adj, Adv\} \)

2. \( T = \{\text{colorless, green, ideas, sleep, furiously}\} \)

3. \( P \)

\[
\begin{array}{c|c|c}
\text{S} \rightarrow \text{NP} \text{ VP} & 1.0 & \text{NP} \rightarrow \text{AdjP} \text{ NP} & 0.5 \\
\text{VP} \rightarrow \text{V} \text{ AdvP} & 1.0 & \text{NP} \rightarrow \text{N} & 0.5 \\
\text{AdvP} \rightarrow \text{Adv} & 1.0 & \text{AdjP} \rightarrow \text{Adj} & 1.0 \\
\text{Adj} \rightarrow \text{colorless} & 0.5 & \text{Adj} \rightarrow \text{green} & 0.5 \\
\text{N} \rightarrow \text{ideas} & 1.0 & \text{V} \rightarrow \text{sleep} & 1.0 \\
\text{Adv} \rightarrow \text{furiously} & 1.0 & & \\
\end{array}
\]

4. \( S \)

We can derive the structure of a string.

- \( S \Rightarrow \) NP VP \hspace{1cm} 1.0
  \( \Rightarrow \) N VP \hspace{1cm} 0.5
Probabilistic Context-Free Grammar

1. \( V = \{S, \text{NP}, \text{VP}, \text{AdjP}, \text{AdvP}\} \cup \{\text{N, V, Adj, Adv}\} \)

2. \( T = \{\text{colorless, green, ideas, sleep, furiously}\} \)

3. \( P \)

<table>
<thead>
<tr>
<th>Rule</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S \rightarrow \text{NP VP} )</td>
<td>1.0</td>
</tr>
<tr>
<td>( \text{VP} \rightarrow \text{V AdvP} )</td>
<td>1.0</td>
</tr>
<tr>
<td>( \text{AdvP} \rightarrow \text{Adv} )</td>
<td>1.0</td>
</tr>
<tr>
<td>( \text{Adj} \rightarrow \text{colorless} )</td>
<td>0.5</td>
</tr>
<tr>
<td>( \text{N} \rightarrow \text{ideas} )</td>
<td>1.0</td>
</tr>
<tr>
<td>( \text{Adv} \rightarrow \text{furiously} )</td>
<td>1.0</td>
</tr>
<tr>
<td>( \text{NP} \rightarrow \text{AdjP NP} )</td>
<td>0.5</td>
</tr>
<tr>
<td>( \text{NP} \rightarrow \text{N} )</td>
<td>0.5</td>
</tr>
<tr>
<td>( \text{AdjP} \rightarrow \text{Adj} )</td>
<td>1.0</td>
</tr>
<tr>
<td>( \text{Adj} \rightarrow \text{green} )</td>
<td>0.5</td>
</tr>
<tr>
<td>( \text{V} \rightarrow \text{sleep} )</td>
<td>1.0</td>
</tr>
</tbody>
</table>

4. \( S \)

We can derive the structure of a string.

- \( S \Rightarrow \text{NP VP} \) 1.0
  - \( \Rightarrow \text{N VP} \) 0.5
  - \( \Rightarrow \text{ideas VP} \) 1.0
  - \( \Rightarrow \text{ideas sleep AdvP} \) 1.0
  - \( \Rightarrow \text{ideas sleep Adv} \) 1.0
  - \( \Rightarrow \text{ideas sleep furiously} \) 1.0
Probabilistic Context-Free Grammar

1. \[ V = \{S, \text{NP}, \text{VP}, \text{AdjP}, \text{AdvP}\} \cup \{\text{N, V, Adj, Adv}\} \]

2. \[ T = \{\text{colorless, green, ideas, sleep, furiously}\} \]

3. \[ P \]

<table>
<thead>
<tr>
<th>Rule</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>S \rightarrow \text{NP VP}</td>
<td>1.0</td>
</tr>
<tr>
<td>VP \rightarrow \text{V AdvP}</td>
<td>1.0</td>
</tr>
<tr>
<td>\text{AdvP} \rightarrow \text{Adv}</td>
<td>1.0</td>
</tr>
<tr>
<td>\text{Adj} \rightarrow \text{colorless}</td>
<td>0.5</td>
</tr>
<tr>
<td>\text{N} \rightarrow \text{ideas}</td>
<td>1.0</td>
</tr>
<tr>
<td>\text{Adv} \rightarrow \text{furiously}</td>
<td>1.0</td>
</tr>
<tr>
<td>\text{NP} \rightarrow \text{AdjP NP}</td>
<td>0.5</td>
</tr>
<tr>
<td>\text{NP} \rightarrow \text{N}</td>
<td>0.5</td>
</tr>
<tr>
<td>\text{AdjP} \rightarrow \text{Adj}</td>
<td>1.0</td>
</tr>
<tr>
<td>\text{Adj} \rightarrow \text{green}</td>
<td>0.5</td>
</tr>
<tr>
<td>\text{V} \rightarrow \text{sleep}</td>
<td>1.0</td>
</tr>
</tbody>
</table>

4. We can derive the structure of a string.
   - \[ S \Rightarrow \text{NP VP} \quad 1.0 \]
     \[ \Rightarrow \text{N VP} \quad 0.5 \]
     \[ \Rightarrow \text{ideas VP} \quad 1.0 \]
     \[ \Rightarrow \text{ideas V AdvP} \quad 1.0 \]
Probabilistic Context-Free Grammar

1. \( V = \{S, \text{NP}, \text{VP}, \text{AdjP}, \text{AdvP}\} \cup \{N, V, \text{Adj}, \text{Adv}\} \)

2. \( T = \{\text{colorless, green, ideas, sleep, furiously}\} \)

3. \( P \)

<table>
<thead>
<tr>
<th></th>
<th>S \rightarrow \text{NP VP}</th>
<th>1.0</th>
<th>NP \rightarrow \text{AdjP NP}</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>VP \rightarrow V AdvP</td>
<td>1.0</td>
<td>NP \rightarrow N</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>AdvP \rightarrow Adv</td>
<td>1.0</td>
<td>AdjP \rightarrow Adj</td>
<td>1.0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Adj \rightarrow \text{colorless}</th>
<th>0.5</th>
<th>Adj \rightarrow \text{green}</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>N \rightarrow ideas</td>
<td>1.0</td>
<td>V \rightarrow sleep</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>Adv \rightarrow \text{furiously}</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4. S

We can derive the structure of a string.

- S \Rightarrow \text{NP VP} \quad 1.0
  \Rightarrow \text{N VP} \quad 0.5
  \Rightarrow \text{ideas VP} \quad 1.0
  \Rightarrow \text{ideas V AdvP} \quad 1.0
  \Rightarrow \text{ideas sleep AdvP} \quad 1.0

\[\text{score} = 0\]
Probabilistic Context-Free Grammar

1. \( V = \{S, \text{NP}, \text{VP}, \text{AdjP}, \text{AdvP}\} \cup \{N, V, \text{Adj}, \text{Adv}\} \)

2. \( T = \{\text{colorless, green, ideas, sleep, furiously}\} \)

3. \( P \)

<table>
<thead>
<tr>
<th>Production</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S \rightarrow \text{NP} \ \text{VP} )</td>
<td>1.0</td>
</tr>
<tr>
<td>( \text{VP} \rightarrow V \ \text{AdvP} )</td>
<td>1.0</td>
</tr>
<tr>
<td>( \text{AdvP} \rightarrow \text{Adv} )</td>
<td>1.0</td>
</tr>
<tr>
<td>( \text{Adj} \rightarrow \text{colorless} )</td>
<td>0.5</td>
</tr>
<tr>
<td>( \text{N} \rightarrow \text{ideas} )</td>
<td>1.0</td>
</tr>
<tr>
<td>( \text{Adv} \rightarrow \text{furiously} )</td>
<td>1.0</td>
</tr>
<tr>
<td>( \text{NP} \rightarrow \text{AdjP} \ \text{NP} )</td>
<td>0.5</td>
</tr>
<tr>
<td>( \text{NP} \rightarrow \text{N} )</td>
<td>0.5</td>
</tr>
<tr>
<td>( \text{AdjP} \rightarrow \text{Adj} )</td>
<td>1.0</td>
</tr>
<tr>
<td>( \text{Adj} \rightarrow \text{green} )</td>
<td>0.5</td>
</tr>
<tr>
<td>( V \rightarrow \text{sleep} )</td>
<td>1.0</td>
</tr>
</tbody>
</table>

4. We can derive the structure of a string.

- \( S \Rightarrow \text{NP} \ \text{VP} \) 1.0
- \( \Rightarrow \text{N} \ \text{VP} \) 0.5
- \( \Rightarrow \text{ideas} \ \text{VP} \) 1.0
- \( \Rightarrow \text{ideas} \ \text{V} \ \text{AdvP} \) 1.0
- \( \Rightarrow \text{ideas} \ \text{sleep} \ \text{AdvP} \) 1.0
- \( \Rightarrow \text{ideas} \ \text{sleep} \ \text{Adv} \) 1.0

score = 0.
Probabilistic Context-Free Grammar

1. \( V = \{ S, NP, VP, AdjP, AdvP \} \cup \{ N, V, Adj, Adv \} \)
2. \( T = \{ \text{colorless, green, ideas, sleep, furiously} \} \)
3. \( P \)

<table>
<thead>
<tr>
<th>Production</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S \rightarrow NP \ VP )</td>
<td>1.0</td>
</tr>
<tr>
<td>( VP \rightarrow V \ AdvP )</td>
<td>1.0</td>
</tr>
<tr>
<td>( AdvP \rightarrow Adv )</td>
<td>1.0</td>
</tr>
<tr>
<td>( Adj \rightarrow colorless )</td>
<td>0.5</td>
</tr>
<tr>
<td>( N \rightarrow ideas )</td>
<td>1.0</td>
</tr>
<tr>
<td>( Adv \rightarrow furiously )</td>
<td>1.0</td>
</tr>
<tr>
<td>( NP \rightarrow AdjP \ NP )</td>
<td>0.5</td>
</tr>
<tr>
<td>( NP \rightarrow N )</td>
<td>0.5</td>
</tr>
<tr>
<td>( AdjP \rightarrow Adj )</td>
<td>1.0</td>
</tr>
<tr>
<td>( Adj \rightarrow green )</td>
<td>0.5</td>
</tr>
<tr>
<td>( V \rightarrow sleep )</td>
<td>1.0</td>
</tr>
</tbody>
</table>

4. We can derive the structure of a string.

- \( S \Rightarrow NP \ VP \) 1.0
- \( \Rightarrow NP \ VP \) 0.5
- \( \Rightarrow ideas VP \) 1.0
- \( \Rightarrow ideas V \ AdvP \) 1.0
- \( \Rightarrow ideas sleep AdvP \) 1.0
- \( \Rightarrow ideas sleep Adv \) 1.0
- \( \Rightarrow ideas sleep furiously \) 1.0

\[ \text{score} = 0 \]
Probabilistic Context-Free Grammar

1. \( V = \{S, \text{NP}, \text{VP}, \text{AdjP}, \text{AdvP}\} \cup \{\text{N, V, Adj, Adv}\} \)

2. \( T = \{\text{colorless, green, ideas, sleep, furiously}\} \)

3. \( P \)

\[
\begin{array}{|c|c|}
\hline
\text{S} \rightarrow \text{NP} \ \text{VP} & 1.0 \\
\text{VP} \rightarrow \text{V} \ \text{AdvP} & 1.0 \\
\text{AdvP} \rightarrow \text{Adv} & 1.0 \\
\text{Adj} \rightarrow \text{colorless} & 0.5 \\
\text{N} \rightarrow \text{ideas} & 1.0 \\
\text{Adv} \rightarrow \text{furiously} & 1.0 \\
\text{NP} \rightarrow \text{AdjP} \ \text{NP} & 0.5 \\
\text{NP} \rightarrow \text{N} & 0.5 \\
\text{AdjP} \rightarrow \text{Adj} & 1.0 \\
\text{Adj} \rightarrow \text{green} & 0.5 \\
\text{V} \rightarrow \text{sleep} & 1.0 \\
\hline
\end{array}
\]

4. \( S \)

We can derive the structure of a string.

- \( S \Rightarrow \text{NP} \ \text{VP} \)
- \( \Rightarrow \text{N} \ \text{VP} \)
- \( \Rightarrow \text{ideas} \ \text{VP} \)
- \( \Rightarrow \text{ideas} \ \text{V} \ \text{AdvP} \)
- \( \Rightarrow \text{ideas} \ \text{sleep} \ \text{AdvP} \)
- \( \Rightarrow \text{ideas} \ \text{sleep} \ \text{Adv} \)
- \( \Rightarrow \text{ideas} \ \text{sleep} \ \text{furiously} \)

\[
\begin{array}{c}
S \\
\Downarrow \\
\text{NP} \quad \text{VP} \\
\Downarrow \\
\text{N} \quad \text{V} \quad \text{AdvP} \\
\Downarrow \\
\text{Ideas} \quad \text{sleep} \quad \text{Adv} \\
\Downarrow \\
\text{furiously}
\end{array}
\]
We can derive the structure of a string.

- $S \Rightarrow NP \ VP$ 1.0
  $\Rightarrow N \ VP$ 0.5
  $\Rightarrow$ ideas $VP$ 1.0
  $\Rightarrow$ ideas $V \ AdvP$ 1.0
  $\Rightarrow$ ideas sleep $AdvP$ 1.0
  $\Rightarrow$ ideas sleep $Adv$ 1.0
  $\Rightarrow$ ideas sleep furiously 1.0

score = 0.5
Naive application is not successful

- \( q(\text{NP} \rightarrow \text{NP PP}) > q(\text{VP} \rightarrow \text{VP PP}) \): The first one
- \( q(\text{NP} \rightarrow \text{NP PP}) < q(\text{VP} \rightarrow \text{VP PP}) \): The second one

F-score = 72.64

Naive application is not successful

- $q(NP \rightarrow NP\ PP) > q(VP \rightarrow VP\ PP)$: The first one
- $q(NP \rightarrow NP\ PP) < q(VP \rightarrow VP\ PP)$: The second one
Naive application is not successful

- \( q(NP \rightarrow NP \text{ PP}) > q(VP \rightarrow VP \text{ PP}) \): The first one
- \( q(NP \rightarrow NP \text{ PP}) < q(VP \rightarrow VP \text{ PP}) \): The second one
Naive application is not successful

- $q(NP \rightarrow NP \ PP) > q(VP \rightarrow VP \ PP)$: The first one
- $q(NP \rightarrow NP \ PP) < q(VP \rightarrow VP \ PP)$: The second one
Naive application is not successful

- $q(\text{NP} \rightarrow \text{NP PP}) > q(\text{VP} \rightarrow \text{VP PP})$: The first one
- $q(\text{NP} \rightarrow \text{NP PP}) < q(\text{VP} \rightarrow \text{VP PP})$: The second one

F-score = 72.64

Outline

1 Parsing: An (Unnecessary?) Introduction
2 PCFG: Early Failure
3 Lexicalization: A Breakthrough Technique
4 Alternatives? Simple PCFG Again
Lexicalized PCFG

Solution (Collins, 1999)

Add annotations specifying richer information for each rule
Lexicalized CFG

Rewrite rules

- $X(h) \rightarrow Y(h) \ Z(m)$ for $X, Y, Z \in N$, and $h, w \in T$
- $X(h) \rightarrow Y(m) \ Z(h)$ for $X, Y, Z \in N$, and $h, w \in T$
- $X(h) \rightarrow h$ for $X \in N$, and $h \in T$

<table>
<thead>
<tr>
<th>Rule</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(oil-wrestles) → NP(Penny) VP(oil-wrestles)</td>
<td>1.0</td>
</tr>
<tr>
<td>VP(oil-wrestles) → VP(oil-wrestles) PP(in)</td>
<td>0.5</td>
</tr>
<tr>
<td>VP(oil-wrestles) → V(oil-wrestles) NP(orangutan)</td>
<td>0.25</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>V(oil-wrestle) → oil-wrestle</td>
<td>1.0</td>
</tr>
<tr>
<td>Det(an) → an</td>
<td>0.4</td>
</tr>
<tr>
<td>Det(a) → a</td>
<td>0.6</td>
</tr>
<tr>
<td>N(orangutan) → orangutan</td>
<td>0.5</td>
</tr>
<tr>
<td>N(bikini) → bikini</td>
<td>0.5</td>
</tr>
<tr>
<td>P(in) → in</td>
<td>1.0</td>
</tr>
</tbody>
</table>
Outline

1. Parsing: An (Unnecessary?) Introduction
2. PCFG: Early Failure
3. Lexicalization: A Breakthrough Technique
4. Alternatives? (Simple) PCFG Again
Cutting-edge research

re-

search

| Latin

| PIE

circus

*kikro-cut, cut
### Alternatives?

<table>
<thead>
<tr>
<th>System name</th>
<th>Short description</th>
<th>Main publications</th>
<th>Software</th>
<th>Results (PARSEVAL)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Charniak &amp; Johnson's Parser</td>
<td>Lexicalized N-Best PCFG + Discriminative reranking</td>
<td>Johnson and Charniak (2005)</td>
<td>Download</td>
<td>91.4%</td>
<td>also works well on Brown</td>
</tr>
<tr>
<td>Self-trained Charniak &amp; Johnson</td>
<td>Above + self-training on ~2 million raw sentences from</td>
<td>McClosky, Charniak, and Johnson (2006)</td>
<td>Download</td>
<td>92.1%</td>
<td>also works well on Brown</td>
</tr>
<tr>
<td>Berkeley Parser</td>
<td>Automatically induced PCFG</td>
<td>Petrov et al. (2006), Petrov and Klein (2007)</td>
<td>Berkeley Parser</td>
<td>90.1%</td>
<td>works well also for Chinese and German</td>
</tr>
<tr>
<td>Link Grammar</td>
<td>Dependency grammar</td>
<td>Temperley, Sleator, Lafferty, others (1995-2006)</td>
<td>Actively supported project</td>
<td>?</td>
<td>Persian, Arabic, Chinese, German, Russian dictionaries have been developed.</td>
</tr>
</tbody>
</table>
## Alternatives?

<table>
<thead>
<tr>
<th>System name</th>
<th>Short description</th>
<th>Main publications</th>
<th>Software</th>
<th>Results (PARSEVAL)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Charniak &amp; Johnson's Parser</td>
<td>Lexicalized N-Best PCFG + Discriminative reranking</td>
<td>Johnson and Charniak (2005)</td>
<td>Download [link]</td>
<td>91.4%</td>
<td>also works well on Brown</td>
</tr>
<tr>
<td>Berkeley Parser</td>
<td>Automatically induced PCFG</td>
<td>Petrov et al. (2006), Petrov and Klein (2007)</td>
<td>Berkeley Parser [link]</td>
<td>90.1%</td>
<td>works well also for Chinese and German</td>
</tr>
<tr>
<td>Link Grammar</td>
<td>Dependency grammar</td>
<td>Temperley, Sleator, Lafferty, others (1995-2006)</td>
<td>Actively supported project [link]</td>
<td>?</td>
<td>Persian, Arabic, Chinese, German, Russian dictionaries have been developed.</td>
</tr>
</tbody>
</table>
Enriching representations

Parent annotation (Johnson, 1998)
Add annotations specifying richer information for each production rule of a given PCFG.
F-score $= 72.64 \rightarrow 76.81$

Enriching representations

Parent annotation (Johnson, 1998)

Add annotations specifying richer information for each production rule of a given PCFG.

F-score $= 72.64 \rightarrow 76.81$

# Vertical and horizontal Markovization

<table>
<thead>
<tr>
<th>Vertical Order</th>
<th>$v = 1$</th>
<th>$v \leq 2$</th>
<th>$v = 2$</th>
<th>$v \leq 3$</th>
<th>$v = 3$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No annotation</td>
<td>Sel. Parents</td>
<td>All Parents</td>
<td>Sel. GParents</td>
<td>All GParents</td>
</tr>
<tr>
<td>$h = 0$</td>
<td>71.27</td>
<td>74.75</td>
<td>74.68</td>
<td>76.50</td>
<td>76.74</td>
</tr>
<tr>
<td></td>
<td>854</td>
<td>2285</td>
<td>2984</td>
<td>4943</td>
<td>7797</td>
</tr>
<tr>
<td>$h = 1$</td>
<td>72.5</td>
<td>77.42</td>
<td>77.42</td>
<td>78.59</td>
<td>79.18</td>
</tr>
<tr>
<td></td>
<td>3119</td>
<td>6564</td>
<td>7312</td>
<td>12374</td>
<td>15740</td>
</tr>
<tr>
<td>$h \leq 2$</td>
<td>73.46</td>
<td>77.77</td>
<td>77.81</td>
<td>79.07</td>
<td>79.74</td>
</tr>
<tr>
<td></td>
<td>3863</td>
<td>7619</td>
<td>8367</td>
<td>13627</td>
<td>16994</td>
</tr>
<tr>
<td>$h = 2$</td>
<td>72.96</td>
<td>77.50</td>
<td>77.50</td>
<td>78.97</td>
<td>79.07</td>
</tr>
<tr>
<td></td>
<td>6207</td>
<td>11398</td>
<td>12132</td>
<td>19545</td>
<td>22886</td>
</tr>
<tr>
<td>$h = \infty$</td>
<td>72.62</td>
<td>76.91</td>
<td>76.81</td>
<td>78.54</td>
<td>78.72</td>
</tr>
<tr>
<td></td>
<td>9657</td>
<td>14247</td>
<td>14666</td>
<td>20123</td>
<td>22002</td>
</tr>
</tbody>
</table>

Figure 2: Markovizations: $F_1$ and grammar size.

Further annotation enrichment

<table>
<thead>
<tr>
<th>Annotation</th>
<th>Cumulative</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Size</td>
<td>F₁</td>
<td>Δ F₁</td>
<td>Δ F₁</td>
</tr>
<tr>
<td>Baseline ($v \leq 2, h \leq 2$)</td>
<td>7619</td>
<td>77.77</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNARY-INTERNAL</td>
<td>8065</td>
<td>78.32</td>
<td>0.55</td>
<td>0.55</td>
</tr>
<tr>
<td>UNARY-DT</td>
<td>8066</td>
<td>78.48</td>
<td>0.71</td>
<td>0.17</td>
</tr>
<tr>
<td>UNARY-RB</td>
<td>8069</td>
<td>78.86</td>
<td>1.09</td>
<td>0.43</td>
</tr>
<tr>
<td>TAG-PA</td>
<td>8520</td>
<td>80.62</td>
<td>2.85</td>
<td>2.52</td>
</tr>
<tr>
<td>SPLIT-IN</td>
<td>8541</td>
<td>81.19</td>
<td>3.42</td>
<td>2.12</td>
</tr>
<tr>
<td>SPLIT-AUX</td>
<td>9034</td>
<td>81.66</td>
<td>3.89</td>
<td>0.57</td>
</tr>
<tr>
<td>SPLIT-CC</td>
<td>9190</td>
<td>81.69</td>
<td>3.92</td>
<td>0.12</td>
</tr>
<tr>
<td>SPLIT-%</td>
<td>9255</td>
<td>81.81</td>
<td>4.04</td>
<td>0.15</td>
</tr>
<tr>
<td>TMP-NP</td>
<td>9594</td>
<td>82.25</td>
<td>4.48</td>
<td>1.07</td>
</tr>
<tr>
<td>GAPPED-S</td>
<td>9741</td>
<td>82.28</td>
<td>4.51</td>
<td>0.17</td>
</tr>
<tr>
<td>POSS-NP</td>
<td>9820</td>
<td>83.06</td>
<td>5.29</td>
<td>0.28</td>
</tr>
<tr>
<td>SPLIT-VP</td>
<td>10499</td>
<td>85.72</td>
<td>7.95</td>
<td>1.36</td>
</tr>
<tr>
<td>BASE-NP</td>
<td>11660</td>
<td>86.04</td>
<td>8.27</td>
<td>0.73</td>
</tr>
<tr>
<td>DOMINATES-V</td>
<td>14097</td>
<td>86.91</td>
<td>9.14</td>
<td>1.42</td>
</tr>
<tr>
<td>RIGHT-REC-NP</td>
<td>15276</td>
<td>87.04</td>
<td>9.27</td>
<td>1.94</td>
</tr>
</tbody>
</table>

Latent annotations

Additional information as latent variables (Matsuzaki et al., 2005)

\[
\begin{align*}
S_x & \rightarrow NP_x \rightarrow VP_x \\
          & \quad \rightarrow V_x \rightarrow NP_x \\
          & \quad \rightarrow NP_x \rightarrow PP_x \\
          & \quad \rightarrow DET_x \rightarrow N_x \rightarrow P_x \\
          & \quad \rightarrow DET_x \rightarrow N_x
\end{align*}
\]
Latent annotations

Additional information as latent variables (Matsuzaki et al., 2005)

Cool idea; many many challenges

- How to induce $x$?
- How to find a best parse?
Latent annotations

Additional information as latent variables (Matsuzaki et al., 2005)

Cool idea; many many challenges

- How to induce $x$?
- How to find a best parse?
research

re-

search

| Latin
| circus
| PIE
| *kikro-
cut, cut
<table>
<thead>
<tr>
<th>System name</th>
<th>Short description</th>
<th>Main publications</th>
<th>Software</th>
<th>Result (PARSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Charniak &amp; Johnson's Parser</td>
<td>Lexicalized N-Best PCFG + Discriminative reranking</td>
<td>Johnson and Charniak (2005)</td>
<td>Download</td>
<td>91.4%</td>
</tr>
<tr>
<td>Collins' Parser</td>
<td>Lexicalized PCFG</td>
<td>Collins (1999), Bikel (2004)</td>
<td>Dan Bikel's implementation</td>
<td>?</td>
</tr>
<tr>
<td>Berkeley Parser</td>
<td>Automatically induced PCFG</td>
<td>Petrov et al. (2006), Petrov and Klein (2007)</td>
<td>Berkeley Parser</td>
<td>90.1%</td>
</tr>
<tr>
<td>Link Grammar</td>
<td>Dependency grammar</td>
<td>Temperley, Sleator, Lafferty, others (1995-2006)</td>
<td>Actively supported project</td>
<td>?</td>
</tr>
</tbody>
</table>

**Model**

<table>
<thead>
<tr>
<th>Model</th>
<th>LP</th>
<th>LR</th>
<th>LF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Klein &amp; Manning (2003)</td>
<td>86.9</td>
<td>85.7</td>
<td>86.3</td>
</tr>
<tr>
<td>Matsuzaki et al. (2005)</td>
<td>86.1</td>
<td>86.0</td>
<td></td>
</tr>
<tr>
<td>Petrov et al. (2006)</td>
<td>89.8</td>
<td>89.6</td>
<td></td>
</tr>
<tr>
<td>Shindo et al. (2012)</td>
<td></td>
<td></td>
<td>91.1</td>
</tr>
<tr>
<td>Shindo et al. (2012)</td>
<td></td>
<td></td>
<td>92.4</td>
</tr>
</tbody>
</table>
Lessons learned

1998 • PCFG Models of Linguistic Tree Representations

2003 • Accurate Unlexicalized Parsing (ACL best paper)

2005 • Probabilistic CFG with Latent Annotations

2006 • Learning Accurate, Compact, and Interpretable Tree Annotation

2012 • Bayesian Symbol-Refined Tree Substitution Grammars for Syntactic Parsing (ACL best paper)

Too many; a short list:
• Diversity
• Interpretability
• Revisit old things

Research

Research

Research

Research

Research

Research

Research

Research
Lessons learned

1998 · · · PCFG Models of Linguistic Tree Representations

2003 · · · Accurate Unlexicalized Parsing (ACL best paper)

2005 · · · Probabilistic CFG with Latent Annotations

2006 · · · Learning Accurate, Compact, and Interpretable Tree Annotation

2012 · · · Bayesian Symbol-Refined Tree Substitution Grammars for Syntactic Parsing (ACL best paper)
Lessons learned

<table>
<thead>
<tr>
<th>Year</th>
<th>Research Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>1998</td>
<td>PCFG Models of Linguistic Tree Representations</td>
</tr>
<tr>
<td>2003</td>
<td>Accurate Unlexicalized Parsing (ACL best paper)</td>
</tr>
<tr>
<td>2005</td>
<td>Probabilistic CFG with Latent Annotations</td>
</tr>
<tr>
<td>2006</td>
<td>Learning Accurate, Compact, and Interpretable Tree Annotation</td>
</tr>
<tr>
<td>2012</td>
<td>Bayesian Symbol-Refined Tree Substitution Grammars for Syntactic Parsing (ACL best paper)</td>
</tr>
</tbody>
</table>

Lessons that I've learned

- Diversity
- Interpretability
- Revisit old things...
Lessons learned

1998 · · · PCFG Models of Linguistic Tree Representations

2003 · · · Accurate Unlexicalized Parsing (ACL best paper)

2005 · · · Probabilistic CFG with Latent Annotations

2006 · · · Learning Accurate, Compact, and Interpretable Tree Annotation

2012 · · · Bayesian Symbol-Refined Tree Substitution Grammars for Syntactic Parsing (ACL best paper)
Lessons learned

1998 • PCFG Models of Linguistic Tree Representations

2003 • Accurate Unlexicalized Parsing (ACL best paper)

2005 • Probabilistic CFG with Latent Annotations

2006 • Learning Accurate, Compact, and Interpretable Tree Annotation

2012 • Bayesian Symbol-Refined Tree Substitution Grammars for Syntactic Parsing (ACL best paper)
Lessons learned

1998 • PCFG Models of Linguistic Tree Representations

2003 • Accurate Unlexicalized Parsing (ACL best paper)

2005 • Probabilistic CFG with Latent Annotations

2006 • Learning Accurate, Compact, and Interpretable Tree Annotation

2010 • Bayesian Symbol-Refined Tree Substitution Grammars for Syntactic Parsing (ACL best paper)
Lessons learned

1998 • PCFG Models of Linguistic Tree Representations

2003 • Accurate Unlexicalized Parsing (ACL best paper)

2005 • Probabilistic CFG with Latent Annotations

2006 • Learning Accurate, Compact, and Interpretable Tree Annotation

2012 • Bayesian Symbol-Refined Tree Substitution Grammars for Syntactic Parsing (ACL best paper)

Lessons that I’ve learned

Too many; a short list:

- Diversity
- Interpretability
- Revisit old things
Thank You


References II

