Fast(er) Exact Decoding and Global Training for Transition-Based Dependency Parsing via a Minimal Feature Set

Tianze Shi*  Liang Huang†  Lillian Lee*

* Cornell University  † Oregon State University

$O(n^3)$ Theoretical

Minimal Feature Set

$O(n^3)$ Practical
Short Version

• Transition-based dependency parsing has an exponentially-large search space

• $O(n^3)$ exact solutions exist 😃

• In practice, however, we needed rich features $\Rightarrow O(n^6)$ 😞

• (This work) with bi-LSTMs, now we can do $O(n^3)$! 😊

• And we get state-of-the-art results
Short Version

• Transition-based dependency parsing has an exponentially-large search space

• \( O(n^3) \) exact solutions exist 😃

• In practice, however, we needed rich features \( \Rightarrow O(n^6) \) 😞

• (This work) with bi-LSTMs, now we can do \( O(n^3)! \) 😃

• And we get state-of-the-art results
Short Version

• Transition-based dependency parsing has an exponentially-large search space

• $O(n^3)$ exact solutions exist 😊

• In practice, however, we needed rich features $\Rightarrow O(n^6)$ 😞

• (This work) with bi-LSTMs, now we can do $O(n^3)$! 😏

• And we get state-of-the-art results
• Transition-based dependency parsing has an exponentially-large search space

• $O(n^3)$ exact solutions exist 😃

• In practice, however, we needed rich features $\Rightarrow O(n^6)$ 😞

• (This work) with bi-LSTMs, now we can do $O(n^3)$! 😊

• And we get state-of-the-art results
Short Version

• Transition-based dependency parsing has an exponentially-large search space

• $O(n^3)$ exact solutions exist 😃

• In practice, however, we needed rich features $\Rightarrow O(n^6)$ 😞

• (This work) with bi-LSTMs, now we can do $O(n^3)$! 😊

• And we get state-of-the-art results
Dependency Parsing

INPUT: She wanted to eat an apple

OUTPUT: root

xcomp

nsubj

mark

obj
det

Background: $O(n^3)$ in theory, $O(n^6)$ in practice

Results
Transition-based Dependency Parsing

Initial state

Transition

Terminal states

Background $O(n^3)$ in theory $O(n^6)$ in practice Back to $O(n^3)$ Results
Goal:
\[ \max \ \text{score}(\bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet) \]
\[ = \max \ \sum \ \text{score}(\bullet \rightarrow \bullet) \]
Exact Decoding with Dynamic Programming

Goal:
\[
\max \text{ score}(\bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \ldots \rightarrow \bullet)
\]
\[
= \max \sum \text{ score}(\bullet \rightarrow \bullet)
\]

(Huang and Sagae, 2010; Kuhlmann, Gómez-Rodríguez and Satta, 2011)
Transition Systems

<table>
<thead>
<tr>
<th>Arc Type</th>
<th>DP Complexity</th>
<th># Action Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arc-standard</td>
<td>$O(n^4)$</td>
<td>3</td>
</tr>
<tr>
<td>Arc-eager</td>
<td>$O(n^3)$</td>
<td>4</td>
</tr>
<tr>
<td>Arc-hybrid</td>
<td>$O(n^3)$</td>
<td>3</td>
</tr>
</tbody>
</table>

In our paper, we presentational convenience

Background: $O(n^3)$ in theory, $O(n^6)$ in practice

Back to $O(n^3)$

Results
Arc-hybrid Transition System

State

... $s_2$ $s_1$ $s_0$

Stack

Buffer

... $b_0$ $b_1$ ...

Initial State

ROOT She wanted ...

Terminal State

ROOT

(Yamada and Matsumoto, 2003)
(Gómez-Rodríguez et al., 2008)
(Kuhlmann et al., 2011)
Arc-hybrid Transition System

Transitions

\[ \ldots b_0 \ldots \rightarrow \ldots s_1 s_0 \ldots \rightarrow \ldots s_0 \ldots \rightarrow \ldots b_0 \ldots \]

Same as arc-standard

Background $O(n^3)$ in theory $O(n^6)$ in practice Back to $O(n^3)$ Results
Arc-hybrid Transition System

Transitions

\[ \vdash b_0 \vdash \vdash \]

\[ \vdash s_1 s_0 \vdash \vdash \]

\[ \vdash b_0 \vdash \vdash \]

\[ \vdash s_0 \vdash \vdash \]

\[ \vdash b_0 \vdash \vdash \]

\[ \vdash s_0 \vdash \vdash \]

Same as arc-standard

Background \( O(n^3) \) in theory \( O(n^6) \) in practice Back to \( O(n^3) \) Results
Arc-hybrid Transition System

Transitions

- **shift**: ...
  - $b_0$
  - ...

- **reduce** $\leadsto$
  - ...
  - $s_1$
  - $s_0$
  - ...

- **reduce** $\leadsto$
  - ...
  - $s_0$
  - $b_0$
  - ...

Same as arc-standard
Arc-hybrid Transition System

**Stack**

- **initial**: ROOT
- **shift**: ROOT  She  wanted  to  eat  an  apple

**Buffer**

- **shift**: She  wanted  to  eat  an  apple
- **reduce**: ROOT
- **shift**: wanted  to  eat  an  apple
- **shift**: wanted  to  eat  an  apple
# Arc-hybrid Transition System

<table>
<thead>
<tr>
<th>Stack</th>
<th>Buffer</th>
</tr>
</thead>
<tbody>
<tr>
<td>initial</td>
<td><strong>ROOT</strong></td>
</tr>
<tr>
<td>shift</td>
<td><strong>ROOT</strong></td>
</tr>
<tr>
<td>shift</td>
<td><strong>ROOT</strong></td>
</tr>
<tr>
<td>shift</td>
<td><strong>ROOT</strong></td>
</tr>
</tbody>
</table>

**Background**

$O(n^3)$ in theory

$O(n^6)$ in practice

Back to $O(n^3)$

**Results**

She wanted to eat an apple
Arc-hybrid Transition System

**Stack**

<table>
<thead>
<tr>
<th>initial</th>
<th></th>
<th>ROOT</th>
<th>She</th>
<th>wanted</th>
<th>to</th>
<th>eat</th>
<th>an</th>
<th>apple</th>
</tr>
</thead>
</table>

**Buffer**

<table>
<thead>
<tr>
<th>shift</th>
<th>ROOT</th>
<th>She</th>
<th>wanted</th>
<th>to</th>
<th>eat</th>
<th>an</th>
<th>apple</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>shift</th>
<th>ROOT</th>
<th>She</th>
<th>wanted</th>
<th>to</th>
<th>eat</th>
<th>an</th>
<th>apple</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>shift</th>
<th>ROOT</th>
<th>wanted</th>
<th>to</th>
<th>eat</th>
<th>an</th>
<th>apple</th>
</tr>
</thead>
</table>

**Background**

$O(n^3)$ in theory

$O(n^6)$ in practice

Results

Back to $O(n^3)$
Arc-hybrid Transition System

Stack

initial

Buffer

ROOT She wanted to eat an apple

shift

ROOT

wanted to eat an apple

shift

ROOT She

wanted to eat an apple

reduce

ROOT She

wanted to eat an apple

Shift

ROOT wanted

to eat an apple

shift

ROOT wanted to

eat an apple

Background $O(n^3)$ in theory $O(n^6)$ in practice Back to $O(n^3)$ Results
Arc-hybrid Transition System

Stack

initial

shift

reduce

shift

shift

Buffer

ROOT

She

wanted

to

eat

an

apple

ROOT

She

wanted

to

eat

an

apple

wanted

to

eat

an

apple

ROOT

She

wanted

to

eat

an

apple

ROOT

wanted

ROOT

wanted

to

eat

an

apple

ROOT

wanted

to

eat

an

apple

ROOT

wanted

to

eat

an

apple

ROOT

wanted

to

eat

an

apple
# Arc-hybrid Transition System

<table>
<thead>
<tr>
<th>Stack</th>
<th>Buffer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>initial</strong></td>
<td><strong>ROOT</strong></td>
</tr>
<tr>
<td><strong>shift</strong></td>
<td><strong>ROOT</strong></td>
</tr>
<tr>
<td><strong>shift</strong></td>
<td><strong>ROOT</strong></td>
</tr>
<tr>
<td><strong>reduce</strong></td>
<td><strong>ROOT</strong></td>
</tr>
<tr>
<td><strong>shift</strong></td>
<td><strong>ROOT</strong></td>
</tr>
<tr>
<td><strong>shift</strong></td>
<td><strong>ROOT</strong></td>
</tr>
</tbody>
</table>
Arc-hybrid Transition System

**Stack**

ROOT  wanted  to

reduce

ROOT  wanted

shift

ROOT  wanted  eat

shift

ROOT  wanted  eat  an

reduce

ROOT  wanted  eat

shift

ROOT  wanted  eat  apple

**Buffer**

eat  an  apple
to

eat  an  apple

an  apple

an

apple

apple

Background  \(O(n^3)\) in theory  \(O(n^6)\) in practice  Back to \(O(n^3)\)  Results
Arc-hybrid Transition System

Background \( O(n^3) \) in theory \( O(n^6) \) in practice Back to \( O(n^3) \) Results

\[
\begin{align*}
\text{Stack} & \quad \text{Buffer} \\
\text{ROOT} & \quad \text{eat} \quad \text{an} \quad \text{apple} \\
\text{ROOT} & \quad \text{wanted} \quad \text{to} \\
\text{ROOT} & \quad \text{wanted} \\
\text{ROOT} & \quad \text{wanted} \quad \text{eat} \quad \text{an} \\
\text{ROOT} & \quad \text{wanted} \quad \text{eat} \quad \text{an} \\
\text{ROOT} & \quad \text{wanted} \quad \text{eat} \quad \text{an} \quad \text{apple} \\
\end{align*}
\]
Arc-hybrid Transition System

Stack

<table>
<thead>
<tr>
<th>ROOT</th>
<th>wanted</th>
<th>to</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROOT</td>
<td>wanted</td>
<td></td>
</tr>
<tr>
<td>ROOT</td>
<td>wanted</td>
<td>eat</td>
</tr>
<tr>
<td>ROOT</td>
<td>wanted</td>
<td>eat</td>
</tr>
<tr>
<td>ROOT</td>
<td>wanted</td>
<td>eat</td>
</tr>
</tbody>
</table>

Buffer

<table>
<thead>
<tr>
<th>eat</th>
<th>an</th>
<th>apple</th>
</tr>
</thead>
<tbody>
<tr>
<td>eat</td>
<td>an</td>
<td>apple</td>
</tr>
<tr>
<td>an</td>
<td>apple</td>
<td></td>
</tr>
<tr>
<td>an</td>
<td>apple</td>
<td></td>
</tr>
</tbody>
</table>

Background $O(n^3)$ in theory $O(n^6)$ in practice Back to $O(n^3)$ Results
Arc-hybrid Transition System

Background \(O(n^3)\) in theory \(O(n^6)\) in practice

Results

Stack

Reduce

ROOT wanted to

ROOT wanted

Shift

ROOT wanted eat

ROOT wanted eat an

Reduce

ROOT wanted eat an

Shift

ROOT wanted eat an

Buffer

eat an apple

eat an apple

to

an apple

an apple

apple

apple

Back to \(O(n^3)\)
Arc-hybrid Transition System

**Stack**
- **reduce**
  - ROOT wanted to
- **shift**
  - ROOT wanted
- **shift**
  - ROOT wanted eat
- **reduce**
  - ROOT wanted eat

**Buffer**
- eat an apple
- eat an apple
- to
- an apple
- apple

Background $O(n^3)$ in theory $O(n^6)$ in practice Back to $O(n^3)$ Results
Arc-hybrid Transition System

Stack

ROOT wanted to

ROOT wanted

ROOT wanted eat

ROOT wanted eat an

ROOT wanted eat

ROOT wanted eat apple

Buffer

eat an apple

eat an apple
to

eat an apple

an apple

an apple

an

Background $O(n^3)$ in theory $O(n^6)$ in practice Back to $O(n^3)$ Results
Arc-hybrid Transition System

Stack

Buffer

ROOT wanted eat apple

reduce

ROOT wanted eat

reduce

ROOT wanted

terminal

ROOT wanted
Arc-hybrid Transition System

Stack

Buffer

ROOT wanted eat apple

reduce

ROOT wanted eat

reduce

apple

reduce

eat

terminal

ROOT wanted
Arc-hybrid Transition System

Stack

ROOT  wanted  eat  apple

reduce

ROOT  wanted  eat

reduce

ROOT  wanted

reduce

ROOT

terminal

wanted
Arc-hybrid Transition System

Stack

reduce

reduce

reduce

terminal

Buffer

Background $O(n^3)$ in theory $O(n^6)$ in practice Back to $O(n^3)$ Results

ROOT wanted eat apple

ROOT wanted eat

ROOT wanted

ROOT wanted
eat

wanted
Dynamic Programming for Arc-hybrid

Stack

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6 (n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ROOT</td>
</tr>
<tr>
<td>j</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>She</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>wanted</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>to</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>eat</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>an</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>apple</td>
</tr>
</tbody>
</table>

Buffer

• Deduction Item

\[ [i, j] \]

• Goal

\[ [0, n + 1] \]
Dynamic Programming for Arc-hybrid

**Background**

\( O(n^3) \) in theory

\( O(n^6) \) in practice

- Back to \( O(n^3) \)

- **Results**

\[
\text{shift } \frac{[i,j]}{[j,j+1]}
\]

\[
\ldots \quad i \quad j \quad \ldots
\]

\[
\ldots \quad i \quad j \quad j+1 \quad \ldots
\]
Dynamic Programming for Arc-hybrid

reduce:\n\[ [k, j] \rightarrow [?, j] \]

\[ \ldots \quad k \quad \ldots \quad j \quad \ldots \]

\[ \ldots \quad ? \quad \ldots \quad j \quad \ldots \]

\[ k \]
Dynamic Programming for Arc-hybrid

\[ \text{reduce} \begin{bmatrix} i, j \end{bmatrix} \]

\[ \begin{bmatrix} k, j \end{bmatrix} \]

\[ \text{reduce} \begin{bmatrix} i, j \end{bmatrix} \]

\[ \begin{bmatrix} ... & i & k \end{bmatrix} \]

\[ \begin{bmatrix} j & ... \end{bmatrix} \]

\[ \begin{bmatrix} ... & i \end{bmatrix} \]

\[ \begin{bmatrix} j & ... \end{bmatrix} \]
Dynamic Programming for Arc-hybrid

\[
\text{reduce} \prec \frac{[i, k] [k, j]}{[i, j]}
\]

\[
\begin{array}{ccc}
  \cdots & i & k \\
  \cdots & i & k \\
  j & \cdots \\
  j & \cdots \\
\end{array}
\]
Dynamic Programming for Arc-hybrid

In Kuhlmann, Gómez-Rodríguez and Satta (2011)’s notation

\[
\text{reduce} \leftarrow \frac{[i, k] [k, j]}{[i, j]}
\]

\[
[\ast \rightarrow [i, k] [\ast \rightarrow [i, j] [\ast \rightarrow [k, j] [\ast \rightarrow [i, j]]]
\]

\[
\text{reduce} \leftarrow \frac{[i, k]}{[k, j]}
\]

\[
[\ast \rightarrow [i, k] [\ast \rightarrow [j, j] [\ast \rightarrow [i, j]]]
\]

Background $O(n^3)$ in theory $O(n^6)$ in practice Back to $O(n^3)$ Results
Dynamic Programming for Arc-hybrid

\[
\text{reduce} \sim \frac{[i, k] [k, j]}{[i, j]}
\]

\[
\text{reduce} \sim \frac{[i, k]}{[i, j]}
\]

\[
\text{reduce} \sim \frac{[k, j]}{[i, j]}
\]

\[
\text{reduce} \sim \frac{[i, k]}{[i, j]}
\]

Background \( O(n^3) \) in theory \( O(n^6) \) in practice Back to \( O(n^3) \) Results
Dynamic Programming for Arc-hybrid

\[
\text{shift} \quad \frac{[i, j]}{[j, j + 1]}
\]

\[
\text{reduce} \overset{\sim}{\quad} \frac{[i, k] [k, j]}{[i, j]} \quad k \sim j
\]

\[
\text{reduce} \overset{\sim}{\quad} \frac{[i, k] [k, j]}{[i, j]} \quad i \sim k
\]

Goal: \[ [0, n + 1] \]

\[ O(n^3) \]
Time Complexity in Practice

- Complexity depends on feature representation!

- Typical feature representation:
  - Feature templates look at specific *positions* in the stack and in the buffer
Time Complexity in Practice

• Compare the following features

\[
\begin{array}{ccc}
\ldots & s_0 & b_0 & \ldots \\
\end{array}
\]

\[
\begin{array}{ccc}
\ldots & s_1 & s_0 & b_0 & \ldots \\
\end{array}
\]

• Time complexities are different!!!

\[
\begin{array}{ccc}
\ldots & i & j & \ldots \\
\end{array}
\]

\[
\begin{array}{ccc}
\ldots & k & i & j & \ldots \\
\end{array}
\]

\[
\begin{array}{ccc}
\ldots & j & j+1 & \ldots \\
\end{array}
\]

\[
\begin{array}{ccc}
\ldots & i & j & j+1 & \ldots \\
\end{array}
\]

\[s_{sh}(i,j)\xrightarrow{\text{shift}}\]

\[s_{sh}(k,i,j)\xrightarrow{\text{shift}}\]
Time Complexity in Practice

• Complexity depends on feature representation!

• Typical feature representation:
  • Feature templates look at specific positions in the stack and in the buffer

• Best-known complexity in practice: $O(n^6)$
  (Huang and Sagae, 2010)

Stack

Buffer

$\cdots \ s_2 \ \ s_1 \ \ s_0 \ \ \ b_0 \ b_1 \ \ \cdots$

$s_1.\ lc \ \ \ s_1.\ rc \ \ s_0.\ lc \ \ s_0.\ rc$
Best-known Time Complexities (recap)

\[ O(n^3) \]

Theoretical

\[ O(n^6) \]

Practical

Gap: Feature representation
In Practice, Instead of Exact Decoding …

- **Greedy search** (Nivre, 2003, 2004, 2008; Chen and Manning, 2014)
- **Beam search** (Zhang and Clark, 2011; Weiss et al., 2015)
- **Best-first search** (Sagae and Lavie, 2006; Sagae and Tsujii, 2007; Zhao et al., 2013)
- **Dynamic oracles** (Goldberg and Nivre, 2012, 2013)
- **“Global” normalization on the beam** (Zhou et al., 2015; Andor et al., 2016)
- **Reinforcement learning** (Lê and Fokkens, 2017)
- **Learning to search** (Daumé III and Marcu, 2005; Chang et al., 2016; Wiseman and Rush, 2016)
- ...
How Many Positional Features Do We Need?

Non-neural (manual engineering)

☞ Chen and Manning (2014)

Stack

Buffer

\[ s_1.lc_i \quad \ldots \quad s_1.rc_i \quad s_0.lc_i \quad \ldots \quad s_0.rc_i \]

\[ s_1.lc_0.lc_0 \quad s_1.rc_0.rc_0 \quad s_0.lc_0.lc_0 \quad s_0.rc_0.rc_0 \]
How Many Positional Features Do We Need?

Non-neural (manual engineering)

☞ Chen and Manning (2014)

Stack LSTM
☞ Dyer et al. (2015)

Bi-LSTM
☞ Kiperwasser and Goldberg (2016)
☞ Cross and Huang (2016)

Background $O(n^3)$ in theory $O(n^6)$ in practice Back to $O(n^3)$ Results

Stack

$s_0$, $s_1$, $s_2$, ..., $s_{|\sigma|-1}$
How Many Positional Features Do We Need?

• Bi-LSTMs give compact feature representations (Kiperwasser and Goldberg, 2016; Cross and Huang, 2016)

• Features used in Kiperwasser and Goldberg (2016)

• Features used in Cross and Huang (2016)
How Many Positional Features Do We Need?

Non-neural (manual engineering)

- Chen and Manning (2014)

Stack LSTM
- Dyer et al. (2015)

Bi-LSTM
- Kiperwasser and Goldberg (2016)
- Cross and Huang (2016)

Background \( O(n^3) \) in theory \( O(n^6) \) in practice Back to \( O(n^3) \) Results

Summarizing trees on stack

Exponential DP Enables Slow DP

Summarizing input

Enables Fast DP
Model Architecture

\[ S_{sh}, S_{re\leftarrow}, S_{re\rightarrow} \]

- Multi-layer perceptron
- Bi-directional LSTM
- Word embeddings + POS embeddings

She wanted to eat an apple
Model Architecture

\[ S_{sh}, S_{re}, S_{re} \]

Multi-layer perceptron

\[ s_1, s_0, b_0 \]

Bi-directional LSTM

Word embeddings + POS embeddings

She wanted to eat an apple
She wanted to eat an apple.

Model Architecture

$S_{sh}, S_{re\leftarrow}, S_{re\rightarrow}$

Multi-layer perceptron

$s_0, b_0$

Bi-directional LSTM

Word embeddings + POS embeddings
Model Architecture

\[ S_{sh}, S_{re\leftarrow}, S_{re\rightarrow} \]

Multi-layer perceptron

\[ b_0 \]

Bi-directional LSTM

Word embeddings + POS embeddings

She wanted to eat an apple
How Many Positional Features Do We Need?

• We answer the question empirically
  ... experimented with greedy decoding
• Two positional feature vectors are enough!

<table>
<thead>
<tr>
<th>Feature Set</th>
<th>UAS (%)</th>
<th>Std Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>{s_2, s_1, s_0, b_0}</td>
<td>94.08</td>
<td>±0.13</td>
</tr>
<tr>
<td>{s_1, s_0, b_0}</td>
<td>94.08</td>
<td>±0.05</td>
</tr>
<tr>
<td>{s_0, b_0}</td>
<td>94.03</td>
<td>±0.12</td>
</tr>
<tr>
<td>{b_0}</td>
<td>52.39</td>
<td>±0.23</td>
</tr>
</tbody>
</table>

Considered in prior work
How Many Positional Features Do We Need?

- Our minimal feature set

\[
\begin{align*}
\text{Stack} & \quad \text{Buffer} \\
... & \quad s_0 & b_0 & \ldots \\
& & s_0 & b_0 & \ldots \\
& & s_0 & b_0 & \ldots
\end{align*}
\]

- Counter-intuitive, but works for *greedy decoding*

\[
\begin{align*}
\text{Stack} & \quad \text{Buffer} \\
... & \quad s_1 & s_0 & \ldots & b_0 & \ldots \\
& & s_1 & s_0 & b_0 & \ldots \\
& & s_1 & s_0 & b_0 & \ldots
\end{align*}
\]

\[
\text{reduce} \sim
\]

\[
\begin{align*}
& & s_1 & \ldots & b_0 & \ldots \\
& & s_1 & \ldots & b_0 & \ldots
\end{align*}
\]
How Many Positional Features Do We Need?

- Our minimal feature set

  \[ \text{Stack} \quad \text{Buffer} \]

  \[
  \vdots \quad s_0 \quad b_0 \quad \vdots
  \]

- Counter-intuitive, but works for *greedy decoding*

- The bare deduction items already contain enough information to extract features for DP

- Leads to the first \( O(n^3) \) implementation of global decoders!
How Many Positional Features Do We Need?

Non-neural (manual engineering)

- Chen and Manning (2014)

- Stack LSTM: Dyer et al. (2015)

- Bi-LSTM: Kiperwasser and Goldberg (2016)
  - Cross and Huang (2016)

- Our work

Summarizing trees on stack

- Exponential DP
- Enables Slow DP
- Enables Fast DP

Back to $O(n^3)$

Results

$O(n^3)$ in theory $O(n^6)$ in practice
Best-known Time Complexities (recap)

$O(n^3)$ Theoretical

Gap: Feature representation

$O(n^6)$ Practical

Background $O(n^3)$ in theory $O(n^6)$ in practice Back to $O(n^3)$ Results
Our contribution

$$O(n^3)$$ in theory

$$O(n^6)$$ in practice

Back to $$O(n^3)$$

Results
Decoding

Score of the sub-sequence

\[
\begin{align*}
\text{shift} & \quad [i,j] : \nu \\
\text{shift} & \quad [j, j+1] : 0
\end{align*}
\]
Decoding

\[ \Delta = s_{sh}(i, k) + s_{re}(k, j) \]

\[ \text{reduce} \quad \begin{array}{c}
\frac{[i, k] : v_1 \quad [k, j] : v_2}{[i, j] : v_1 + v_2 + \Delta}
\end{array} \]

\[ \text{reduce} \quad \begin{array}{c}
[i, j]
\end{array} \]
Training

- Separate incorrect from correct by a margin

\[
\max \, \text{score}(\text{incorrect}) - \text{score}(\text{correct}) + \text{cost}(\text{incorrect})
\]

- Cost-augmented decoding (Taskar et al., 2005)

\[
\begin{align*}
\text{reduce} \ (i, k) & : v_1 \\
\text{reduce} \ (k, j) & : v_2 \\
\text{reduce} \ (i, j) & : v_1 + v_2 + s_{sh}(i, k) + s_{re}(k, j) + 1(\text{head}\star(k) \neq j)
\end{align*}
\]
Comparing with State-of-the-art

| Background | $O(n^3)$ in theory | $O(n^6)$ in practice | Back to $O(n^3)$ | Results |

CTB UAS

English PTB UAS

- Local
- Global

Data points:
- BGDS16
- CH16
- DBLMS15
- KG16a
- KG16b
- CFHGD16
- KBKDS16
- WC16
- DM17

Scores:
- 86.0
- 86.5
- 87.0
- 87.5
- 88.0
- 88.5
- 89.0
- 89.5
- 90.0
- 90.5
- 91.0
- 91.5
- 92.0
- 92.5
- 93.0
- 93.5
- 94.0
- 94.5
- 95.0
- 95.5
- 96.0
Comparing with State-of-the-art

Results

Background $O(n^3)$ in theory $O(n^6)$ in practice Back to $O(n^3)$

Our Global

Our arc-eager DP

Our arc-hybrid DP

Local

Global

Chinese

CTB

UAS

English

PTB

UAS

86.0

86.5

87.0

87.5

88.0

88.5

89.0

89.5

90.0

90.5

93.0

94.0

95.0

96.0

BGDS15

DBLMS15

KG16b

KG16a

CH16

CFHGD16

DBKDS16

KG16a

DM17

WC16
Comparing with State-of-the-art

Background: $O(n^3)$ in theory, $O(n^6)$ in practice

Results

- **Our best local**
- **Our arc-eager DP**
- **Our arc-hybrid DP**
Comparing with State-of-the-art

Results

Background

\[ O(n^3) \text{ in theory} \quad O(n^6) \text{ in practice} \quad \text{Back to } O(n^3) \]
Results – CoNLL’17 Shared Task

- Macro-average of 81 treebanks in 49 languages
- 2\textsuperscript{nd}–highest overall performance

(Shi, Wu, Chen and Cheng, 2017; Zeman et al., 2017)
• Bi-LSTM feature set is minimal yet highly effective
• First $O(n^3)$ implementation of exact decoders
• Global training and decoding gave high performance
More in Our Paper

• Description and analysis of three transition systems (arc-standard, arc-hybrid, arc-eager)

• CKY-style representations of the deduction systems

\[
\begin{align*}
\frac{k \quad i \quad i \quad j}{k} \quad k \rightarrow i \\
\frac{k}{k} \quad j
\end{align*}
\]

\[
\begin{align*}
\frac{i \quad k \quad k \quad j}{i \quad j}
\end{align*}
\]

\[
\begin{align*}
\frac{k \quad i \quad i \quad j}{k} \quad i \rightarrow j +
\end{align*}
\]

• Theoretical analysis of the global methods
  • Arc-eager models can “simulate” arc-hybrid models
  • Arc-eager models can “simulate” edge-factored models
Fast(er) **Exact Decoding** and **Global Training** for Transition-Based Dependency Parsing via a Minimal Feature Set

[https://github.com/tzshi/dp-parser-emnlp17](https://github.com/tzshi/dp-parser-emnlp17)

Tianze Shi*  Liang Huang†  Lillian Lee*

* Cornell University  † Oregon State University
CKY-style Visualization

Axiom  $[0, 1]$  $0 \ 1$  $0^0 \ 1$

Inference Rules

$sh \quad \frac{[i, j]}{[j, j+1]} \quad j \leq n$

$re_\sim \quad \frac{[k, i] \ [i, j]}{[k, j]}$  $k \sim i$

$re_\sim \quad \frac{[k, i] \ [i, j]}{[k, j]}$  $i \sim j$

Goal  $[0, n + 1]$  $0 \ n + 1$  $[0^0, n + 1]$  $0^0 \ n + 1$
CKY-style Visualization

**Axiom** \([0, 1]\)

**Inference Rules**

\[
\text{sh} \quad \frac{[i, j]}{[j, j + 1]} \quad j \leq n
\]

\[
\text{re}_- \quad \frac{[k, i] \quad [i, j]}{[k, j]} \quad k \rightarrow i
\]

\[
\text{re}_- \quad \frac{[k, i] \quad [i, j]}{[k, j]} \quad i \rightarrow j
\]

**Goal** \([0, n + 1]\)

(b) Arc-hybrid

**Axioms**

\[
\quad \frac{i \quad i + 1}{j \quad j} \quad 0 \leq i, j \leq n
\]

**Inference Rules**

- **right-attach**
- **right-reduce**
- **left-attach**
- **left-reduce**

(d) Edge-factored graph-based parsing.
### Results with Arc-eager and Arc-standard

<table>
<thead>
<tr>
<th>Features</th>
<th>Arc-standard</th>
<th>Arc-hybrid</th>
<th>Arc-eager</th>
</tr>
</thead>
<tbody>
<tr>
<td>${ \overleftarrow{s_2}, \overleftarrow{s_1}, \overleftarrow{s_0}, \overleftarrow{b_0} }$</td>
<td>93.95±0.12</td>
<td>94.08±0.13</td>
<td>93.92±0.04</td>
</tr>
<tr>
<td>${ \overleftarrow{s_1}, \overleftarrow{s_0}, \overleftarrow{b_0} }$</td>
<td>94.13±0.06</td>
<td>94.08±0.05</td>
<td>93.91±0.07</td>
</tr>
<tr>
<td>${ \overleftarrow{s_0}, \overleftarrow{b_0} }$</td>
<td>54.47±0.36</td>
<td>94.03±0.12</td>
<td>93.92±0.07</td>
</tr>
<tr>
<td>${ \overleftarrow{b_0} }$</td>
<td>47.11±0.44</td>
<td>52.39±0.23</td>
<td>79.15±0.06</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Min positions</th>
<th>Arc-standard</th>
<th>Arc-hybrid</th>
<th>Arc-eager</th>
</tr>
</thead>
<tbody>
<tr>
<td>K&amp;G 2016a</td>
<td>-</td>
<td>4</td>
<td>-</td>
</tr>
<tr>
<td>C&amp;H 2016a</td>
<td>3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>our work</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
# Results with Arc-eager and Arc-standard

<table>
<thead>
<tr>
<th>Model</th>
<th>Training</th>
<th>Features</th>
<th>PTB</th>
<th>CTB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>UAS (%)</td>
<td>UEM (%)</td>
</tr>
<tr>
<td>Arc-standard</td>
<td>Local</td>
<td>${s_2, s_1, s_0, b_0}$</td>
<td>93.95±0.12</td>
<td>52.29±0.66</td>
</tr>
<tr>
<td>Arc-hybrid</td>
<td>Local</td>
<td>${s_2, s_1, s_0, b_0}$</td>
<td>93.89±0.10</td>
<td>50.82±0.75</td>
</tr>
<tr>
<td></td>
<td>Local</td>
<td>${s_0, b_0}$</td>
<td>93.80±0.12</td>
<td>49.66±0.43</td>
</tr>
<tr>
<td></td>
<td>Global</td>
<td>${s_0, b_0}$</td>
<td>94.43±0.08</td>
<td>53.03±0.71</td>
</tr>
<tr>
<td>Arc-eager</td>
<td>Local</td>
<td>${s_2, s_1, s_0, b_0}$</td>
<td>93.80±0.12</td>
<td>49.66±0.43</td>
</tr>
<tr>
<td></td>
<td>Local</td>
<td>${s_0, b_0}$</td>
<td>93.77±0.08</td>
<td>49.71±0.24</td>
</tr>
<tr>
<td></td>
<td>Global</td>
<td>${s_0, b_0}$</td>
<td>94.53±0.05</td>
<td>53.77±0.46</td>
</tr>
<tr>
<td>Edge-factored</td>
<td>Global</td>
<td>${h, m}$</td>
<td>94.50±0.13</td>
<td>53.86±0.78</td>
</tr>
</tbody>
</table>