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

Tianze Shi*  Liang Huang†  Lillian Lee*
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

Background

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

She wanted to eat an apple

INPUT

OUTPUT
Transition-based Dependency Parsing

Background

\[ O(n^3) \text{ in theory} \quad O(n^6) \text{ in practice} \quad \text{Back to } O(n^3) \quad \text{Results} \]

Initial state

Transition

\[ \vdots \]

Terminal states

\[ \vdots \]
Goal:
\[
\max \text{ score}( \cdots ) = \max \sum \text{ score}( \cdots )
\]
Exact Decoding

• Dynamic programming (Huang and Sagae, 2010; Kuhlmann, Gómez-Rodríguez and Satta, 2011)

... since transition (sub-)sequences can be reused
Exact Decoding

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

Initial state

Terminal states

Exponential to polynomial

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

Results
## Transition Systems

<table>
<thead>
<tr>
<th></th>
<th>DP Complexity</th>
<th># Transitions</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

Presentational convenience
Arc-hybrid Transition System

Search State

Stack

... \( s_2 \) \( s_1 \) \( s_0 \)

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

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

Transitions

shift

Same as arc-standard
Arc-hybrid Transition System

Transitions

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

Results

Same as arc-standard
Arc-hybrid Transition System

Transitions

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

Same as arc-standard
### Arc-hybrid Transition System

<table>
<thead>
<tr>
<th>Stack</th>
<th>Buffer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ROOT</strong></td>
<td>She</td>
</tr>
<tr>
<td><strong>Shift</strong></td>
<td>ROOT</td>
</tr>
<tr>
<td><strong>Reduce</strong></td>
<td>ROOT</td>
</tr>
<tr>
<td><strong>Shift</strong></td>
<td>ROOT</td>
</tr>
<tr>
<td><strong>Shift</strong></td>
<td>ROOT</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

<table>
<thead>
<tr>
<th>Stack</th>
<th>Buffer</th>
</tr>
</thead>
</table>
| **initial** | **ROOT**  
She wanted to eat an apple |
| **shift** | **ROOT**  
She wanted to eat an apple |
| **shift** | **ROOT**  
She wanted | to eat an apple |
| **reduce** | **ROOT**  
wanted | to eat an apple |
| **shift** | **ROOT**  
wanted  
to | eat an apple |

**Background:** $O(n^3)$ in theory, $O(n^6)$ in practice. Return to $O(n^3)$. **Results:**
Arc-hybrid Transition System

Stack | Buffer

**Initial**

<table>
<thead>
<tr>
<th>Stack</th>
<th>Buffer</th>
</tr>
</thead>
<tbody>
<tr>
<td>She wanted to eat an apple</td>
<td>ROOT</td>
</tr>
</tbody>
</table>

**Shift**

<table>
<thead>
<tr>
<th>Stack</th>
<th>Buffer</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROOT</td>
<td>She wanted to eat an apple</td>
</tr>
</tbody>
</table>

**Shift**

<table>
<thead>
<tr>
<th>Stack</th>
<th>Buffer</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROOT She</td>
<td>wanted to eat an apple</td>
</tr>
</tbody>
</table>

**Reduce**

<table>
<thead>
<tr>
<th>Stack</th>
<th>Buffer</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROOT</td>
<td>wanted to eat an apple</td>
</tr>
</tbody>
</table>

She
Arc-hybrid Transition System

Stack

initial

shift

shift

reduce

Buffer

She wanted to eat an apple

She wanted to eat an apple

wanted to eat an apple

wanted to eat an apple

wanted to eat an apple
Arc-hybrid Transition System

Stack

initial
  ROOT
  She wanted to eat an apple

shift
  ROOT
  She wanted to eat an apple

shift
  ROOT
  She wanted to eat an apple

reduce
  ROOT
  wanted to eat an apple

shift
  ROOT
  wanted to

Buffer

ROOT
She wanted to eat an apple

ROOT
She wanted to eat an apple

ROOT
She wanted to eat an apple

ROOT
She wanted to eat an apple

ROOT
to eat an apple
Arc-hybrid Transition System

Stack

| initial | | ROOT | She | wanted | to | eat | an | apple |

| shift | | ROOT | She | wanted | to | eat | an | apple |

| shift | | ROOT | She | wanted | to | eat | an | apple |

| reduce | | ROOT | | wanted | to | eat | an | apple |

| shift | | ROOT | wanted | | to | | eat | an | apple |

| shift | | ROOT | wanted | to | | | eat | an | apple |
Arc-hybrid Transition System

**Stack**

- **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

---

**Background**

- $O(n^3)$ in theory
- $O(n^6)$ in practice

**Results**

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

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

Stack

reduce

ROOT wanted

shift

ROOT wanted eat

Buffer
to

eat an apple

an apple

reduce

ROOT wanted eat

shift

ROOT wanted eat an

apple

an

shift

ROOT wanted eat apple
Arc-hybrid Transition System

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

\[ \text{Stack} \]

\[ \text{Buffer} \]

\text{reduce}_\uparrow
\begin{array}{|c|c|}
\hline
\text{ROOT} & \text{wanted} \\
\hline
\end{array}

\begin{array}{|c|}
\hline
\text{eat an apple} \\
\hline
\end{array}

to

\text{shift}
\begin{array}{|c|c|c|}
\hline
\text{ROOT} & \text{wanted} & \text{eat} \\
\hline
\end{array}

\begin{array}{|c|c|}
\hline
\text{an apple} \\
\hline
\end{array}

\text{shift}
\begin{array}{|c|c|c|}
\hline
\text{ROOT} & \text{wanted} & \text{eat} & \text{an} \\
\hline
\end{array}

\begin{array}{|c|}
\hline
\text{apple} \\
\hline
\end{array}

\text{reduce}_\uparrow
\begin{array}{|c|c|c|}
\hline
\text{ROOT} & \text{wanted} & \text{eat} \\
\hline
\end{array}

\begin{array}{|c|}
\hline
\text{apple} \\
\hline
\end{array}

\text{shift}
\begin{array}{|c|c|c|c|}
\hline
\text{ROOT} & \text{wanted} & \text{eat} & \text{apple} \\
\hline
\end{array}

\begin{array}{|c|}
\hline
\text{an} \\
\hline
\end{array}
Arc-hybrid Transition System

Background

$O(n^3)$ in theory

$O(n^6)$ in practice

Back to $O(n^3)$

Results

Stack

<table>
<thead>
<tr>
<th>reduce</th>
<th>ROOT</th>
<th>wanted</th>
</tr>
</thead>
<tbody>
<tr>
<td>shift</td>
<td>ROOT</td>
<td>wanted</td>
</tr>
<tr>
<td>shift</td>
<td>ROOT</td>
<td>wanted</td>
</tr>
<tr>
<td>reduce</td>
<td>ROOT</td>
<td>wanted</td>
</tr>
</tbody>
</table>

Buffer

eat | an | apple

to

an

apple

apple
Arc-hybrid Transition System

Background

$O(n^3)$ in theory

$O(n^6)$ in practice

Back to $O(n^3)$

Results

Stack

**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

an | apple

apple

an

an | apple
Arc-hybrid Transition System

Background

$O(n^3)$ in theory

$O(n^6)$ in practice

Back to $O(n^3)$

Results

Stack

Buffer

reduce\(\sim\)

ROOT wanted eat

apple

reduce\(\sim\)

ROOT wanted

eat

reduce\(\sim\)

ROOT wanted

wanted

(terminal)
Arc-hybrid Transition System

Background

$O(n^3)$ in theory

$O(n^6)$ in practice

Back to $O(n^3)$

Results

Stack

reduce\(\sim\)

ROOT wanted eat

apple

reduce\(\sim\)

ROOT wanted eat

Buffer

reduce\(\sim\)

ROOT wanted

eat

reduce\(\sim\)

(ROOT)

wished

wanted

25
Arc-hybrid Transition System

Stack

reduce

ROOT wanted eat

apple

reduce

ROOT wanted

eat

reduce (terminal)

ROOT wanted

Buffer
Deduction System for Arc-hybrid

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

- **Deduction Item**
  
  \[ i, j \]

- **Goal**
  
  \[ 0, n + 1 \]

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

\[ O(n#) \] in theory \[ O(n%) \] in practice
Deduction System for Arc-hybrid

\[
\text{shift} \quad \frac{[i, j]}{[j, j + 1]}
\]
Deduction System for Arc-hybrid

\[ \text{reduce} \quad [k, j] \quad [? , j] \]

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

\[ \ldots ? \quad j \ldots \]
Deduction System for Arc-hybrid

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

... i k

\[ \quad \text{reduce} \quad \]

... i

\[ \quad k \quad \]

j ...

\[ \quad j \quad \]

30
Deduction System for Arc-hybrid

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

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

\[
\text{reduce} \quad [i, k] [k, j] \quad \text{reduce} \quad [i, k] [j, j] \quad \text{reduce} \quad [i, j] [j, j]
\]
In Kuhlmann et al. (2011)'s notation

\[\text{reduce} \quad \frac{[i, k] [k, j]}{[i, j]}\]
Deduction System for Arc-hybrid

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

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

\[ \cdots \quad i \quad \cdots \]

\[ \cdots \quad [i, k] \quad \cdots \]

\[ \cdots \quad i \quad k \quad \cdots \]

\[ \cdots \quad [k, j] \quad \cdots \]

\[ \cdots \quad i \quad k \quad \cdots \]

\[ \cdots \quad i \quad \cdots \]

Results

\[ \cdots \quad i \quad \cdots \]

\[ \cdots \quad [i, j] \quad \cdots \]

\[ \cdots \quad j \quad \cdots \]

\[ \cdots \quad j \quad \cdots \]
Deduction System for Arc-hybrid

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

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

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

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

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

• Compare the following features

\[ s_3 \quad b_0 \quad \ldots \]
\[ \ldots \quad s_1 \quad s_0 \quad b_0 \quad \ldots \]

• Time complexities are different!!!

\[ s_{sh}(i, j) \]
\[ \text{shift} \]
\[ \ldots \quad i \quad j \quad j+1 \quad \ldots \]

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

\[ s_{sh} (?, i, j) \]
\[ \text{shift} \]
\[ \ldots \quad i \quad j \quad j+1 \quad \ldots \]

Information about \( s_1 \) is not available, needs extra bookkeeping
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)
Best-known Time Complexities (recap)

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

**Background**

\[ O(n^3) \]

Theoretical

\[ O(n^6) \]

Practical

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

<table>
<thead>
<tr>
<th>Search Method</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beam search</td>
<td>(Zhang and Clark, 2011; Weiss et al., 2015)</td>
</tr>
<tr>
<td>Best-first search</td>
<td>(Sagae and Lavie, 2006; Sagae and Tsujii, 2007; Zhao et al., 2013)</td>
</tr>
<tr>
<td>Dynamic oracles</td>
<td>(Goldberg and Nivre, 2012, 2013)</td>
</tr>
<tr>
<td>“Global” normalization on the beam</td>
<td>(Zhou et al., 2015; Andor et al., 2016)</td>
</tr>
<tr>
<td>Reinforcement learning</td>
<td>(Lê and Fokkens, 2017)</td>
</tr>
<tr>
<td>Learning to search</td>
<td>(Daumé III and Marcu, 2005; Chang et al., 2016; Wiseman and Rush, 2016)</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
How Many Positional Features Do We Need?

Non-neural (manual engineering)

Chen and Manning (2014)
How Many Positional Features Do We Need?

• Chen and Manning (2014)

Background  \( O(n^3) \) in theory  \( O(n^6) \) in practice  Back to \( O(n^3) \)  Results
How Many Positional Features Do We Need?

Non-neural (manual engineering)

Chen and Manning (2014)

Stack LSTM (Dyer et al., 2016)

Bi-LSTM

Kiperwasser and Goldberg (2016)

Cross and Huang (2016)

More tree-structure information

Exponential DP  Slow DP  Fast DP

Background \( O(n^3) \) in theory \( O(n^6) \) in practice Back to \( O(n^3) \) Results
How Many Positional Features Do We Need?

• LSTMs can be used to encode the entire stack and buffer (Dyer et al., 2016)
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)
Model Architecture

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

Multi-layer perceptron

\[ s_2 \quad s_1 \quad s_0 \quad b_0 \]

Bi-directional LSTM

Word embeddings + POS embeddings

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

Model Architecture

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

Bi-directional LSTM

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

Multi-layer perceptron

Word embeddings + POS embeddings

\( s_1, s_0, b_0 \)
She wanted to eat an apple

Model Architecture

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

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>
</tr>
</thead>
<tbody>
<tr>
<td>{s_2, s_1, s_0, b_0}</td>
<td>94.08 ±0.13</td>
</tr>
<tr>
<td>{s_1, s_0, b_0}</td>
<td>94.08 ±0.05</td>
</tr>
<tr>
<td>{s_0, b_0}</td>
<td>94.03 ±0.12</td>
</tr>
<tr>
<td>{b_0}</td>
<td>52.39 ±0.23</td>
</tr>
</tbody>
</table>

Considered in prior work
• Our minimal feature set works

Stack

\[ \ldots s_0 \]

Buffer

\[ b_0 \ldots \]

• Counter-intuitive, but works for greedy decoding

\[ \ldots s_1 s_0 \]

reduce\(\sim\)

\[ \ldots s_1 \]

\[ b_0 \ldots \]
Implication of Minimal Feature Set

- The bare deduction items already contain enough information to extract features
- We don’t need extra book keeping
- 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., 2016)

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

Our work

More tree-structure information

Exponential DP Slow DP Fast DP Fast(er) DP
Best-known Time Complexities (recap)

- **Theoretical**: $O(n^3)\;\text{in theory}$
- **Practical**: $O(n^6)\;\text{in practice}$

**Gap**: Feature representation
Our contribution

Background

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

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

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

Results

\( O(n^3) \)

Bi-directional LSTMs

\( O(n^3) \)

\( O(n^6) \)

Theoretical

Practical
Decoding

Score of the sub-sequence

\[
\frac{[i, j]: v}{[j, j + 1]: 0}
\]

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

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

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

\[
\max \text{ score}([i \rightarrow \cdots \rightarrow j]) + \text{ cost}([k \rightarrow \cdots \rightarrow i]) - \text{ score}([k \rightarrow \cdots \rightarrow j])
\]

\[
\text{reduce} \quad \frac{[i, k]: v_1 \quad [k, j]: v_2}{[i, j]: v_1 + v_2 + s_{sh}(i, k) + s_{re}(k, j) + 1(head(k) \neq j)}
\]
Comparing with State-of-the-art

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

Chinese
CTB
UAS

English PTB UAS

Local
Global

CTB
UAS
Comparing with State-of-the-art

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

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 93.5 94.0 94.5 95.0 95.5 96.0

Our arc-eager DP
Our arc-hybrid DP

Local
Global
Our Global
Comparing with State-of-the-art
Comparing with State-of-the-art

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

<table>
<thead>
<tr>
<th>Chinese</th>
<th>CTB</th>
<th>UAS</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>English</th>
<th>PTB</th>
<th>UAS</th>
</tr>
</thead>
</table>

| Local | Global | Our Local | Our Global | Ensemble |

| KG16a | KG16b | BGDS16 | DBLMS15 | CFHGD16 | KBKDS16 | WC16 | DM17 |

Our best local

Our arc-eager DP

Our arc-hybrid DP

15 Our all global

5 Our arc-eager DP

5 Our arc-hybrid DP

Results

61
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)
Conclusion

• 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

• 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

Tianze Shi* Liang Huang† Lillian Lee*

* Cornell University † Oregon State University
CKY-style Visualization

**Axiom** $[0, 1]$

**Inference Rules**

\[
\begin{align*}
\text{sh} & \quad \frac{[i, j]}{[j, j + 1]} \\
\text{re}_1 & \quad \frac{[k, i] \ [i, j]}{[k, j]} \\
\text{re}_2 & \quad \frac{[k, i] \ [i, j]}{[k, j]}
\end{align*}
\]

\[
\begin{align*}
\text{sh} & \quad \frac{i \ j}{j, j + 1} \\
\text{re}_1 & \quad \frac{k \ i \ i \ j}{k \ j} \\
\text{re}_2 & \quad \frac{k \ i \ i \ j}{k \ j}
\end{align*}
\]

**Goal** $[0, n + 1]$

(b) Arc-hybrid