Factorization of Multitext Grammars as Multi-dimensional Permutations

Hao Zhang and Daniel Gildea

Computer Science Department
University of Rochester
Outline

- From Multitext Grammars (MTG) to Multi-dimensional Permutations
- Factorizing Multi-dimensional Permutations into Trees
- Enumerating Multi-dimensional Permutation Trees
- Asymptotics
# CFG, Synchronous CFG and Multitext Grammars

<table>
<thead>
<tr>
<th></th>
<th>CFG</th>
<th>SCFG</th>
<th>3-MTG</th>
</tr>
</thead>
<tbody>
<tr>
<td>terminals</td>
<td>words</td>
<td>word pairs</td>
<td>word triples</td>
</tr>
<tr>
<td>nonterminals</td>
<td>phrases</td>
<td>phrase pairs</td>
<td>phrase triples</td>
</tr>
<tr>
<td>productions</td>
<td>sequences</td>
<td>permutations</td>
<td>2-permutations</td>
</tr>
</tbody>
</table>

\[
X \rightarrow X_1 X_2 X_3
\]

- **Parsing:**
  - over spans
  - over cells
  - over cubes

Zhang and Gildea
Multitext Alignment

I go there often

wo

J’

I

gain

there

often

qu

na’er

Chang
Alignment View of Two-dimensional Permutations

\[ \pi = (\text{id}_n = (1, 2, 3, 4), \pi_1 = (1, 3, 2, 4), \pi_2 = (1, 4, 2, 3)) \]
\[ \pi = (\text{id}_4, (2, 1, 3, 4), (2, 3, 1, 4)) \]

\[ \pi' = (\text{id}_4, (2, 1, 3, 4), (3, 4, 2, 1)) \]
Two-dimensional Permutation Trees

ternary

binary
Factorization Problem

- Given a d-permutation, produce a k-ary d-permutation tree with k minimized.

- Basic idea: shift-reduce on the alignment links, make reductions whenever possible, $O(d \cdot n^2)$.

- Optimization: apply Uno & Yagiura (2000) algorithm for eliminating impossible reduction boundaries, $O(d \cdot n)$.

- Independently discovered earlier by Bui-Xuan et al. (2005) using PQ tree as the representation tool. Oh, no...
Factorization Algorithm
Factorization Algorithm
Factorization Algorithm

\[
\begin{array}{cccc}
3 & 4 & 2 & 1 \\
2 & 1 & 3 & 4 \\
1 & 2 & 3 & 4 \\
\end{array}
\]
Factorization Algorithm

\[
\begin{array}{ccc}
3 & 4 & (2) \\
(2) & 1) & 3 & 4 \\
(1) & 2) & 3 & 4 \\
\end{array}
\]
Factorization Algorithm
Factorization Algorithm
Factorization Algorithm

(3 4) (2 1) (3 4) (2 1)

(2 1) (3 4) (2 1) (3 4)

(1 2) (3 4) (1 2) (3 4)
Factorization Algorithm

\[
((3, 4) \quad (2, 1))
\]

\[
((2, 1) \quad (3, 4))
\]

\[
((1, 2) \quad (3, 4))
\]
Enumeration of $k$-arizable $d$-permutations

- $d$-permutation trees are unique upon normalization.
- A normalized $d$-permutation tree has two properties.
  - Minimized: a $k$-ary $d$-permutation is used if and only if it is not $k - 1$ parsable.
    
    $$H_{d,k} = (k!)^d - S_{d,k-1}[k]$$

  - Left-most: if a node is binary, its right child must not be binary with the same orientation.
Generating Function for $S_{d,k}$

$$R_{d,k}(x) = x + xR_{d,k}(x) + (2^d - 1)R_{d,k}^2(x) + \sum_{k'=3}^{k} H_{d,k'} \left( R_{d,k}^{k'}(x) + R_{d,k}^{k'+1}(x) \right)$$

It is algebraic.
**Simple $d$-permutations**

<table>
<thead>
<tr>
<th>$k$</th>
<th>$H_{2,k}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>172</td>
</tr>
<tr>
<td>5</td>
<td>5204</td>
</tr>
<tr>
<td>6</td>
<td>222716</td>
</tr>
<tr>
<td>7</td>
<td>12509188</td>
</tr>
<tr>
<td>8</td>
<td>889421564</td>
</tr>
</tbody>
</table>
## Growth Rate of $S_{d,k}$

$$G_{d,k} = \lim_{n \to \infty} \frac{S_{d,k}[n]}{S_{d,k}[n-1]}$$

<table>
<thead>
<tr>
<th></th>
<th>$G_{2,k}$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>13.93</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>22.08</td>
<td>14</td>
</tr>
<tr>
<td>6</td>
<td>29.97</td>
<td>16</td>
</tr>
<tr>
<td>8</td>
<td>38.19</td>
<td>18</td>
</tr>
<tr>
<td>10</td>
<td>47.31</td>
<td>20</td>
</tr>
</tbody>
</table>

$$G_{d,k} = \left(\frac{k}{e}\right)^d + O(k^{d-1} \cdot \log k)$$
Conclusions and Questions

- If all d-permutations are equally likely, with probability nearly one a d-permutation is not factorizable.

- Can natural language multitext grammars use relatively small k to cover real multitexts? Seems we do need gaps...

- If we introduce a fixed number of gaps, how much more coverage can we achieve, theoretically?

- In terms of introducing gaps, there are still many open questions even for two languages. For example, can we characterize the difference between STAG and SCFG from the perspective of alignment (permutation) coverage?