Handling Complexity of Synchronous Grammars for Machine Translation

Hao Zhang

Computer Science Department
University of Rochester
Introduction
Synchronous CFG Example

Boaweier yu Shalong juxing le huitan
Powell held a meeting with Sharon
Notions for SCFG

• A generic n-ary SCFG rule is written as

\[ X \rightarrow X_1^{(1)} \ldots X_n^{(n)}, \ X_{\pi(1)}^{(\pi(1))} \ldots X_{\pi(n)}^{(\pi(n))} \]

where each \( X_i \) is a variable which can take the value of any nonterminal in the grammar.

• For example, the 3-ary rule \( S \rightarrow \begin{bmatrix} PP \\ VP \\ NP \end{bmatrix} \) can be written as

\[ S \rightarrow \text{NP}^{(1)} \text{PP}^{(2)} \text{VP}^{(3)}, \ \text{NP}^{(1)} \text{VP}^{(3)} \text{PP}^{(2)} \]

where \( \pi = (1, 3, 2) \).
Problems in Practice

- Learning
  - Grammar induction (obtaining rules)
  - Parameter estimation (obtaining probabilities)

- Search
  - Synchronous parsing (alignment)
  - Decoding (translation)
Publications

- **Efficient Multi-pass Decoding for Synchronous Context Free Grammars**. Hao Zhang and Daniel Gildea. *In ACL-08:HLT*.

- **Bayesian Learning of Non-compositional Phrases with Synchronous Parsing**. Hao Zhang, Chris Quirk, Robert C. Moore and Daniel Gildea. *In ACL-08:HLT*.


- **Efficient Search for Inversion Transduction Grammar**. Hao Zhang and Daniel Gildea. *In EMNLP-06*.

- **Factoring Synchronous Grammars by Sorting**. Daniel Gildea, Giorgio Satta, and Hao Zhang. *In COLING/ACL-06*.

- **Inducing Word Alignments with Bilexical Synchronous Trees**. Hao Zhang and Daniel Gildea. *In COLING/ACL-06*.

- **Synchronous Binarization for Machine Translation**. Hao Zhang, Liang Huang, Daniel Gildea and Kevin Knight. *In HLT/NAACL-06*.

- **Machine Translation as Lexicalized Parsing with Hooks**. Liang Huang, Hao Zhang and Daniel Gildea. *In Proceedings of the 9th International Workshop on Parsing Technologies (IWPT-05)*.

- **Stochastic Lexicalized Inversion Transduction Grammar for Alignment**. Hao Zhang and Daniel Gildea. *In ACL-05*.

- **Syntax-Based Alignment: Supervised or Unsupervised?**. Hao Zhang and Daniel Gildea. *In COLING-04*.
Outline

- **Part I**: Reducing SCFG Complexity through Factorization
- **Part II**: Efficient Multi-pass Decoding for SCFG
- **Part III**: Bayesian Learning of Phrases with Synchronous Parsing
Part I: Grammar Factorization
Synchronous Binarization

Part I: Grammar Factorization
Synchronous Binarization

Sharon held a meeting with Powell, and Baoweier yu Shalong juxing le huitan.
Synchronous Binarization

Sharon held a meeting with Powell.

Powell held a meeting with Sharon.

Bobweier yu Shalong juxing le huitan

Part I: Grammar Factorization
Parsing Complexity of n-ary SCFG

Assume the sentence length (on either side) is $O(|w|)$.

- $O(|w|^{2n+2})$, without any factorization.
- $O(|w|^{n+3}) = O(|w|^{n+4})$, with binarization on one side.
- $O(|w|^{3+3}) = O(|w|^{6})$, with synchronous binarization maintaining continuous spans on both sides.
Example from a Real System

(Galley et al., 2006)
Binarization Example for Tree Transducer

\[
\begin{align*}
\text{ADJP} & \rightarrow \ T^{(1)}, & \ T^{(1)} \\
\ T & \rightarrow \ RB^{(1)} \ fuze \ PP^{(2)} \ de \ NN^{(3)}, & \ RB^{(1)} \ \text{resp. for the} \ NN^{(3)} \ PP^{(2)} \\
\text{ADJP} & \rightarrow \ T^{(1)}, & \ T^{(1)} \\
\ T & \rightarrow \ V^{(1)}_{1} \ V^{(2)}_{2}, & \ V^{(1)}_{1} \ V^{(2)}_{2} \\
\ V^{(1)}_{1} & \rightarrow \ RB^{(1)}, & \ RB^{(1)} \ fuze \\
\ V^{(2)}_{2} & \rightarrow \ \text{resp. for the} \ NN^{(1)} \ V^{(2)}_{3}, & \ V^{(2)}_{3} \ NN^{(1)} \\
\ V^{(1)}_{3} & \rightarrow \ PP^{(1)}, & \ PP^{(1)} \ de
\end{align*}
\]
How Many Rules Are Binarizable?

- 50,879,242 rules, among which 99.7% are binarizable.
- The solid-line: the distribution of all rules against permutation lengths.
- The dashed-line: the percentage of non-binarizable rules at varying lengths.
The synchronous binarizer is a part of the ISI system which ranked top among all participants in the Chinese-to-English track of the 2006 NIST Machine Translation Evaluation.
Beyond Binarizability

For example, (5, 7, 4, 6, 1, 2, 3) is decomposable:

So we can reduce long SCFG rules into shorter ones. If the longest SCFG rule after such factorization is $k$, we can parse sentences with the grammar in $O(|w|^{k+4})$. 
Asymptotic Results

- $H_k(n)$, the number of $(n - k)$-ary permutations, grows factorially.
  
  $H_k(n)/n! \approx \frac{2^k}{e^2 \cdot k!}$.

  - Percentages: 13.5%, 27.1%, 27.1%, 18.0%, 9.0%, 3.6%, ...

- The number of $k$-ary permutations grows roughly exponentially.
  
  - Bases: 5.83, 5.83, 6.87, 7.33, 7.82, 8.26 ...

(Zhang and Gildea, 2007)
K-arization and Optimal Parsing

O(|w|^{11})


\[\begin{array}{c}
\downarrow \\
O(|w|^8)
\end{array}\]
Part II: Efficient Decoding
Language Model Integrated Decoding for SCFG

- LM-integrated states are represented as $X[i, j, u_{1,...,n-1}, v_{1,...,n-1}]$.
  - A trigram state: $X[3, 6, \text{held}, a, a, \text{meeting}]$
  - A bigram state: $X[3, 6, \text{held}, \text{meeting}]$

- Complexity of decoding is determined by the number of variables at each DP step.
  - $X_1[i, j, u, v] + X_2[j, k, u', v']$

- $O(|w|^{3+4(n-1)})$ for binary SCFG with n-gram language model.

- Can be optimized to $O(|w|^{3+3(n-1)})$, using the “hook” trick.
  (Huang, Zhang and Gildea, 2005)
Progressive LM-Integrated Decoding

- A bigram-integrated state \([X, i, j, u, v]\) is said to be a coarse-level state of a trigram-integrated state \([X, i, j, u', v', v]\).
- Gradually augment the LM-integrated states from lower orders to higher orders.
- Coarse-to-fine decoding in the A* framework.
- The key: using the actual outside cost of a bigram state as a heuristic estimate of the outside cost of a refined trigram state.
Inside/Outside Parsing for Coarse-level Decoding

- An algorithm similar to inside/outside parsing is applied to get the outside costs of the coarse-level LM-integrated states.
- Bottom-up first, then top-down.
- Many algorithmic choices for the coarse-level bottom-up pass, such as CKY, agenda-based.
Heuristics for Fine-grained Decoding

We prioritize decoding states according to

\[ \beta(X[i, j, u_1, u_2, v_1, v_2]) \]
\[ + \alpha(X[i, j, u_1, v_2]) \]
\[ + h_{\text{BestBorder}}(u_1, u_2, v_1, v_2) \]

where

\[ h_{\text{BestBorder}}(u_1, u_2, v_1, v_2) \]
\[ = \max_{s} P_{lm}(u_2 \mid s, u_1) \]
\[ \cdot \max_{s} P_{lm}(s \mid v_1, v_2) \]
Experimental Setup

- Data set: LDC 2002 MT evaluation data set, with a length limit of 20 on the Chinese sentences. 371 sentences, each having 10 references.

- We vary the decoding strategies and beam settings and keep the model unchanged in the experiments.
Decoding Strategies

- **CYK**: standard bottom-up ITG decoder (Wu, 1996; Chiang, 2005), with standard beam pruning.

- **Agenda**: agenda-based ITG decoder (Zhang and Gildea, 2006), with standard beam pruning.

- **Lazy_kbest**: top-down ITG decoder, treating beam pruning as k-best selection (Huang and Chiang, 2007).

- **Bitri_cyk**: two-pass ITG decoder, first pass as cyk with beam pruning, second pass as agenda (Zhang and Gildea, 2008).
### Bigram-pass Outside Cost as Trigram-pass Outside Estimate

<table>
<thead>
<tr>
<th>Decoding Method</th>
<th>Avg. Hyper-edges</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bigram Pass</td>
<td>167K</td>
<td>21.77</td>
</tr>
<tr>
<td>Trigram Pass</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uniform Outside Cost</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Bigram Outside Cost(BO)</td>
<td>+ 629.7K = 796.7K</td>
<td>23.56</td>
</tr>
<tr>
<td>BO+Best Border Heuristic</td>
<td>+ 2.7K = 169.7K</td>
<td>23.46</td>
</tr>
<tr>
<td>Trigram One-pass, with Beam</td>
<td>6401K</td>
<td>23.47</td>
</tr>
</tbody>
</table>

(Zhang and Gildea, 2008)
Two-pass decoding versus One-pass decoding
Two-pass decoding versus One-pass decoding

Part II: Efficient Decoding
Decoding to Maximize BLEU

• Model scores and BLEU scores do not linearly correlate.

• Solution: another decoding pass that maximizes the expected count of translation hypotheses.

• This approximately maximizes the number of matching n-grams.

\[
EC([X, i, j, u, u', v, v]) = \alpha([X, i, j, u, u', v, v]) \cdot \beta([X, i, j, u, u', v, v])
\]

• Similar to Goodman (1996)’s parsing algorithm for maximizing the expected labeled recall.

\[
\max_T \sum_{[X, i, j, u, u', v, v]} EC([X, i, j, u, u', v, v])
\]
Maximizing BLEU

Part II: Efficient Decoding
### Summarization of Decoding Results

<table>
<thead>
<tr>
<th>Decoder</th>
<th>Time</th>
<th>BLEU</th>
<th>Model Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-pass agenda</td>
<td>4317s</td>
<td>22.25</td>
<td>-208.849</td>
</tr>
<tr>
<td>One-pass CYK</td>
<td>3793s</td>
<td>22.89</td>
<td>-207.309</td>
</tr>
</tbody>
</table>

**Multi-pass, CYK first**

| agenda second pass            | 3689s    | 23.56| -205.344    |
| MEC third pass                | 3749s    | 24.07| -203.878    |
| Lazy-cube-pruning             | 3746s    | 22.16| -208.575    |

Summarization of different trigram decoding strategies, using about the same time (10 seconds per sentence).
Part III: Learning Phrases
Pipeline of Phrase-based Systems

Sentence-aligned Corpus

Word-aligned Corpus

Phrase Table

N-gram Table

Decoder

Input

Output

Part III: Learning Phrases
Difficulty of Learning Phrases

- Phrase pairs are substring pairs appearing in sentence pairs.
- The space of all phrase pairs is prohibitively large.
- Traditional EM faces overfitting: data memorization.
Dirichlet Prior for Phrasal Inversion Transduction Grammar

\[\theta_x \mid \alpha_x \sim \text{Dir}(\alpha_x),\]
\[\theta_c \mid \alpha_c \sim \text{Dir}(\alpha_c),\]
\[[X \ X] \mid X \sim \text{Multi}(\theta_x).\]
\[\langle X \ X \rangle \mid C \sim \text{Multi}(\theta_c).\]
\[e/f \mid C \sim \text{Multi}(\theta_c).\]
Variational Bayesian Inference for ITG

We iteratively update the variational parameters to indirectly maximize the posterior probability given the data. The update rule for the phrase pairs is:

\[ \tilde{P}^{(l+1)}(e/f) = \frac{\exp(\psi( E(e/f) + \alpha_C ))}{\exp(\psi( E(C) + m\alpha_C ))}, \]

- Raw fractional counts \( c \) are replaced by \( \exp(\psi( c + \alpha )) \).
- The effect is to penalize infrequent pairs more.
Hard Constraint: Non-compositional Phrases

(Cherry, 2007)
## End-to-end Evaluation

<table>
<thead>
<tr>
<th></th>
<th>Development</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIZA++</td>
<td>37.46</td>
<td>28.24</td>
</tr>
<tr>
<td>ITG-word</td>
<td>35.47</td>
<td>26.55</td>
</tr>
<tr>
<td>ITG-mwm (VB)</td>
<td>39.21</td>
<td>29.02</td>
</tr>
<tr>
<td>ITG-mwm (EM)</td>
<td>39.15</td>
<td>28.47</td>
</tr>
</tbody>
</table>

Translation results on Chinese-English, using the subset of training data (141K sentence pairs) that have length limit 35 on both sides. (No length limit in translation.)

(Zhang, Quirk, Moore and Gildea, 2008)
Conclusions and Future Work
Conclusions

- Binarizable SCFGs are expressive and efficient.
- The marriage of coarse-to-fine and A* decoding produces fast SCFG decoding.
- Dealing with metrics of MT evaluation in decoding achieves good translations.
- The combination of Bayesian learning and ITG constraint guides learning of phrases.
Future Work

- Micro-scoping the space of Binarizable SCFGs to look for SCFGs that are optimal in trading off efficiency and expressiveness.
- Even faster decoding algorithms.
- Efficient unsupervised learning for SCFGs.
- More mathematically principled way of learning phrase pairs.
- An end-to-end SCFG system.
Thank You