Empirical Support for Probabilistic GLR Parsing

Virach Sornlertlamvanich and et al.
Department of Computer Science,
Tokyo Institute of Technology.

JSPS-HITACHI Workshop
25-26 May 1998
Background

- Probabilistic parsing:
  - filter out meaningless parses
  - aid in choosing/ranking for the most likely interpretation

- Probabilistic parsers:
  - Original PCFG: insufficient context
  - Chitrao and Grishman (90): Two-level PCFG
  - Su et al. (91): shift-reduce parsing framework
  - Black et al. (92): History-Based Grammar (HBG)
  - Magerman et al. (95): Chart, CKY, statistical decision-tree
  - etc.

⇒ Originated from PCFG, extended to include more context, modeled independently from the parsing algorithms.
Background

- Probabilistic parsers in the GLR parsing framework:
  - Wright and Wrigley (91): identical to PCFG
  - Goddeau and Zue (92): input symbol prediction
  - Briscoe and Carroll (93): action probability
  - Li et al. (96): pre-terminal bi-gram constraints

⇒ inherit the efficiency of GLR parsing.
⇒ use the provided context of GLR parsing.
Aims of this research

- Verify our newly proposed model, Probabilistic GLR (PGLR) model.
- Evaluate the PGLR model against the existing Briscoe & Carroll (B&C) and Two-level PCFG models.
- Analytical discussion on the results.
- Implementation with a CLR table, compared to an LALR table.
GLR parsing

- A table-driven shift-reduce left-to-right parser for context-free grammars, constructing a rightmost derivation in reverse.

\[ \text{action}_{i+1} = [\text{state}_i, \text{symbol}_{i+1}] \]

- Configuration:
  
  \[
  \begin{array}{c|c}
  \text{stack} & \text{input} \\
  \hline
  (s_0X_1s_1X_2s_2 \cdots X_ms_m, & a_ia_{i+1} \cdots a_n$) \\
  \text{shift action:} & \text{given:} \\
  (s_0X_1s_1X_2s_2 \cdots X_ms_ma_i$s, & a_{i+1} \cdots a_n$) \\
  \text{reduce action:} & \text{and:} \\
  (s_0X_1s_1X_2s_2 \cdots X_{m-r}s_{m-r}A$s, & a_ia_{i+1} \cdots a_n$) \\
  \Rightarrow \text{Stack transitions} & \\
  \end{array}
  \]
GLR parsing

• Grammar:
  (1) S → S S
  (2) S → x

• LR table:

<table>
<thead>
<tr>
<th>state</th>
<th>action</th>
<th>goto</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>x</td>
<td>$</td>
</tr>
<tr>
<td>1</td>
<td>sh1</td>
<td>re2</td>
</tr>
<tr>
<td>2</td>
<td>re2</td>
<td>acc</td>
</tr>
<tr>
<td>3</td>
<td>rel / sh1</td>
<td>rel</td>
</tr>
</tbody>
</table>

⇒ A pair of state and input symbol is the constraint for selecting the parsing action.
Briscoe & Carroll’s model

- A parse tree is regarded as a sequence of state transitions.

- Action probability is the probability of a transition out of a state. Therefore, action probabilities are normalized within each state.

- Probability for a reduce action is subdivided according to the state reached after applying the action, aiming at capturing the left context during the parse.

- Parse probability is the geometric mean of the applied action probabilities, to avoid the bias in favor of parsing involving fewer rules.
### Briscoe & Carroll’s model

<table>
<thead>
<tr>
<th>state</th>
<th>action</th>
<th>goto</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>sh1 (5)</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>S</td>
</tr>
<tr>
<td>1</td>
<td>re2 (10)</td>
<td>re2 (5)</td>
</tr>
<tr>
<td></td>
<td>(0).33; (2).33</td>
<td>(2).26; (3).08</td>
</tr>
<tr>
<td>2</td>
<td>sh1 (9)</td>
<td>acc (5)</td>
</tr>
<tr>
<td></td>
<td>.64</td>
<td>.36</td>
</tr>
<tr>
<td>3</td>
<td>re1 (4) / sh1 (1)</td>
<td>re1 (6)</td>
</tr>
<tr>
<td></td>
<td>(0).36 / .09</td>
<td>(0).45; (2).09</td>
</tr>
</tbody>
</table>

![Diagram](a) [4]  ![Diagram](b) [1]
Briscoe & Carroll’s model

- Advantages:
  - inherit the efficiency of GLR parsing
  - use the provided context by the nature of the GLR parsing
    Left context: parsing state
    Right context: input symbol

- Problematic issues:
  - no probabilistic formalization
  - input symbol after applying a reduce action is not changed
  - stack-top state after stack-pop operation is deterministic
Summary: B&C vs PGLR

- **Normalization**

  **B&C**: within each state.
  **PGLR**: according to state membership, i.e. in $S_s$ or $S_r$.

Transition probability:

$$P(l_i, a_i, \sigma_i|\sigma_{i-1}) \approx \begin{cases} P(l_i, a_i|s_{i-1}) & \text{(for } s_{i-1} \in S_s) \\ P(a_i|s_{i-1}, l_i) & \text{(for } s_{i-1} \in S_r) \end{cases}$$

- $S_s$: $s_0$ and all the states reached after a shift action
- $S_r$: all the states reached after a reduce action

($S_s \cap S_r = \emptyset$)
Summary: B&C vs PGLR

- **Action probability**
  
  **B&C**: reduce actions are subdivided according to the state reached after applying the action.
  
  **PGLR**: one action one probability.

- **Parse probability**
  
  **B&C**: geometric mean of action probabilities applied for a parse.
  
  **PGLR**: product of action probabilities applied for a parse.
## Summary: B&C vs PGLR

<table>
<thead>
<tr>
<th>state</th>
<th>action</th>
<th>goto</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>x</td>
<td>$</td>
</tr>
<tr>
<td>0</td>
<td>sh1 (5)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>re2 (10)</td>
<td>re2</td>
</tr>
<tr>
<td></td>
<td>(0).33;(2).33</td>
<td>(2).26;(3).08</td>
</tr>
<tr>
<td></td>
<td>.67</td>
<td>.33</td>
</tr>
<tr>
<td>2</td>
<td>sh1 (9)</td>
<td>acc</td>
</tr>
<tr>
<td></td>
<td>.64</td>
<td>.36</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>3</td>
<td>rel1 (4) / sh1 (1)</td>
<td>rel1 (6)</td>
</tr>
<tr>
<td></td>
<td>(0).36 / .09</td>
<td>(0).45;(2).09</td>
</tr>
<tr>
<td></td>
<td>.80 / .20</td>
<td>1.0</td>
</tr>
</tbody>
</table>

(a) [4]

(b) [1]
Two-level PCFG

- Two-level PCFG (Chitrao and Grishman, 1990)
- Pseudo Context-sensitive Grammar (Charniak and Carroll, 1994)

\[
P(\text{VP} \rightarrow \text{adverb}, \text{verb} \mid \rho(\text{VP}) = \text{NP})
\]

⇒ Incorporate context for PCFG.
⇒ Accurately reflect the true distribution of English (word based) language string.
⇒ Minimize the model’s per-word (per-tag) cross entropy.
Evaluation

- Morphological and syntactic analysis:
  - Given a string of characters as the input
  - The task includes: word segmentation, POS tagging and parse tree construction

- ATR Japanese corpus

- Grammar:
  - 762 rules of the Japanese phrase structure grammar
  - 137 non-terminal symbols
  - 407 terminal symbols
Model trainability

Parsing accuracy on 510 sentences (open test set) for different proportions of the training set

PGLR
Briscoe & Carroll
2-level PCFG
PCFG
Model analysis

- Grammar:
  
  (1) \( X \rightarrow Uc \)
  
  (2) \( X \rightarrow U \)
  
  (3) \( U \rightarrow a \)
  
  (4) \( U \rightarrow b \)

- Rule probabilities for Two-level PCFG:
  
  (1) \( S ; X \rightarrow Uc \) (1/3)
  
  (2) \( S ; X \rightarrow U \) (2/3)
  
  (3) \( X ; U \rightarrow a \) (1/3)
  
  (4) \( X ; U \rightarrow b \) (2/3)
Comparative results for Two-level PCFG, B&C and PGLR

![Diagram showing three trees with labels (S1)[1], (S2)[2], and (S3)[0].]

<table>
<thead>
<tr>
<th>Models</th>
<th>(S1)</th>
<th>(S2)</th>
<th>(S3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCFG</td>
<td>1/9</td>
<td>4/9</td>
<td>2/9</td>
</tr>
<tr>
<td>Two-level PCFG</td>
<td>1/9</td>
<td>4/9</td>
<td>2/9</td>
</tr>
<tr>
<td>B&amp;C</td>
<td>1/6</td>
<td>1/3</td>
<td>0</td>
</tr>
<tr>
<td>PGLR</td>
<td>1/3</td>
<td>2/3</td>
<td>0</td>
</tr>
</tbody>
</table>
LALR and CLR table-based PGLR

- The degree of context-sensitivity of the states in CLR table is higher than those in LALR table.
- Data sparseness problems in using CLR table.

<table>
<thead>
<tr>
<th></th>
<th>LALR table</th>
<th>CLR table</th>
</tr>
</thead>
<tbody>
<tr>
<td>States</td>
<td>856</td>
<td>3,715</td>
</tr>
<tr>
<td>Shift</td>
<td>11,445</td>
<td>43,833</td>
</tr>
<tr>
<td>Reduce</td>
<td>164,058</td>
<td>756,715</td>
</tr>
<tr>
<td>Goto</td>
<td>4,682</td>
<td>19,733</td>
</tr>
<tr>
<td>States in $S_S$</td>
<td>488</td>
<td>2,539</td>
</tr>
<tr>
<td>States in $S_T$</td>
<td>368</td>
<td>1,176</td>
</tr>
</tbody>
</table>
LALR and CLR table-based PGLR

Distribution of parsing accuracy on 534 sentences (open test set) over the sentence length

Percentages of correct parses

Sentence length (number of words)

PGLR(CLR)  
PGLR(LALR)  

22
LALR and CLR table-based PGLR

Learning curve of the actions in PGLR using an LALR table
(total of 11,445 shift and 164,058 reduce actions)

Learning curve of the actions in PGLR using a CLR table
(total of 43,833 shift and 756,715 reduce actions)
LALR and CLR table-based PGLR

Parsing accuracy on 510 sentences (open test set)
by changing the proportion of training set

Fraction of 10,361 training sentences

Parsing accuracy (%)
Conclusion and future work

- Parse performance:
PGLR > B&C > Two-level PCFG > PCFG

- The PGLR model is able to make effective use of both global and local context provided in the GLR parsing framework.

- No significant distinction between the results of PGLR(LALR) and PGLR(CLR).

⇒ Lexicalize the probabilistic model
⇒ Include long distance constraints
⇒ Verify the PGLR model with a larger corpus