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

$\Rightarrow$ inherit the efficiency of GLR parsing.

$\Rightarrow$ 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.

$$action_{i+1} = [state_i, symbol_{i+1}]$$

- Configuration:

	stack	input
given:		
	$(s_0 X_1 s_1 X_2 s_2 \cdots X_m s_m,$	$a_i a_{i+1} \cdots a_n \$)$
shift action:		
	$(s_0 X_1 s_1 X_2 s_2 \cdots X_m s_m a_i s,$	$a_{i+1} \cdots a_n \$)$
reduce action:		
	$(s_0 X_1 s_1 X_2 s_2 \cdots X_{m-r} s_{m-r} A s,$	$a_i a_{i+1} \cdots a_n \$)$

$\Rightarrow$ Stack transitions

GLR parsing

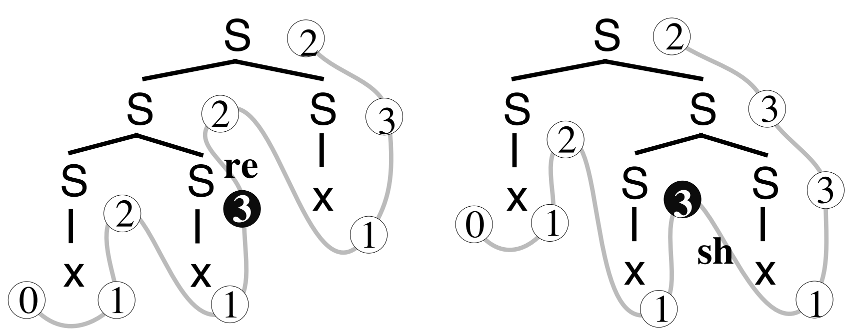
- Grammar:

  (1) S → S S

  (2) S → x

- LR table:

state	action		goto
	x	$	S
0	sh1		2
1	re2	re2	
2	sh1	acc	3
3	**re1 / sh1**	re1	3



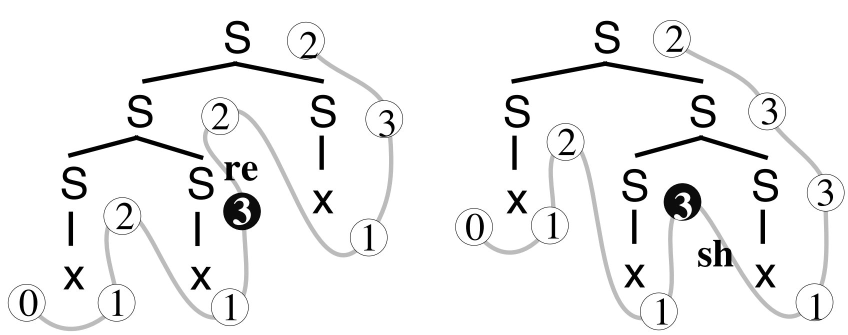
⇒ A pair of state and input symbol is the constraint for selecting the parsing action.

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

state	action		goto
	x	$	S
0	**sh1** (5) 1.0		2
1	**re2** (10) $^{(0)}$.33;$^{(2)}$.33	**re2** (5) $^{(2)}$.26;$^{(3)}$.08	
2	**sh1** (9) .64	**acc** (5) .36	3
3	**re1** (4) / **sh1** (1) $^{(0)}$.36 / .09	**re1** (6) $^{(0)}$.45;$^{(2)}$.09	3



(a) [4] (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 $\boldsymbol{S_s}$ or $\boldsymbol{S_r}$.

Transition probability:

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

$\boldsymbol{S_s}$: s_0 and all the states reached after a shift action

$\boldsymbol{S_r}$: all the states reached after a reduce action

$$(\boldsymbol{S_s} \cap \boldsymbol{S_r} = \emptyset)$$

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.

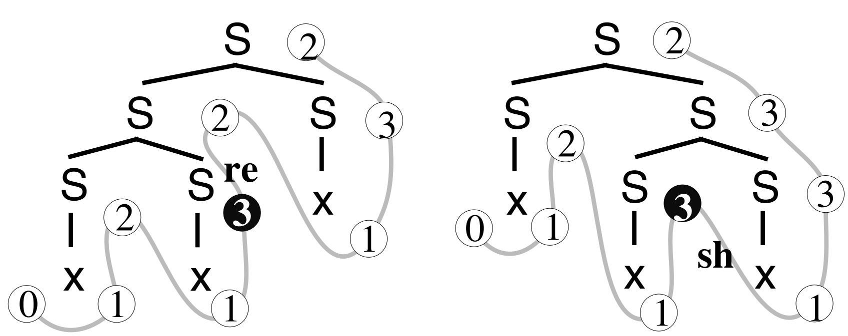
- Parse probability

  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

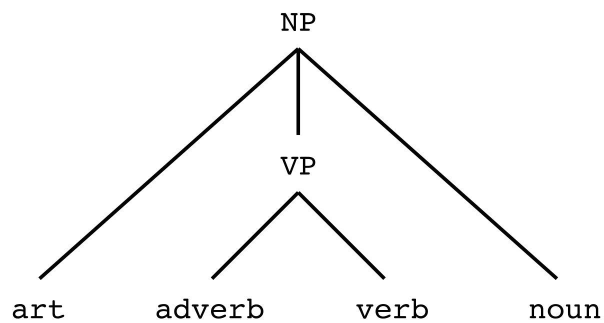
state	action		goto
	x	$	S
0	**sh1** (5) 1.0 1.0		2
1	**re2** (10) $^{(0)}.33;^{(2)}.33$ .67	**re2** (5) $^{(2)}.26;^{(3)}.08$ .33	
2	**sh1** (9) .64 1.0	**acc** (5) .36 1.0	3
3	**re1** (4) / **sh1** (1) $^{(0)}.36$ / .09 .80 / .20	**re1** (6) $^{(0)}.45;^{(2)}.09$ 1.0	3



(a) [4] (b) [1]

Two-level PCFG

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



$$P(VP \rightarrow adverb,\ verb \mid \rho(VP) = NP)$$

$\Rightarrow$ Incorporate context for PCFG.

$\Rightarrow$ Accurately reflect the true distribution of English (word based) language string.

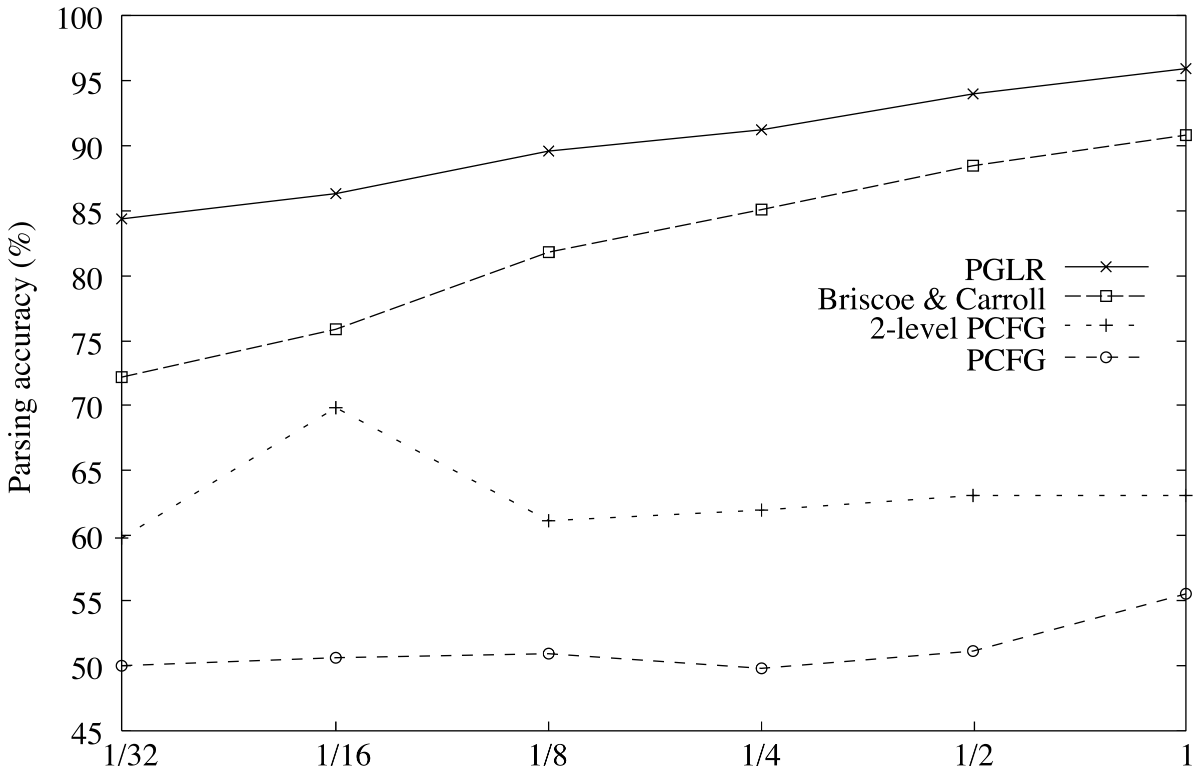
$\Rightarrow$ Minimize the model's per-word (per-tag) cross entropy.

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

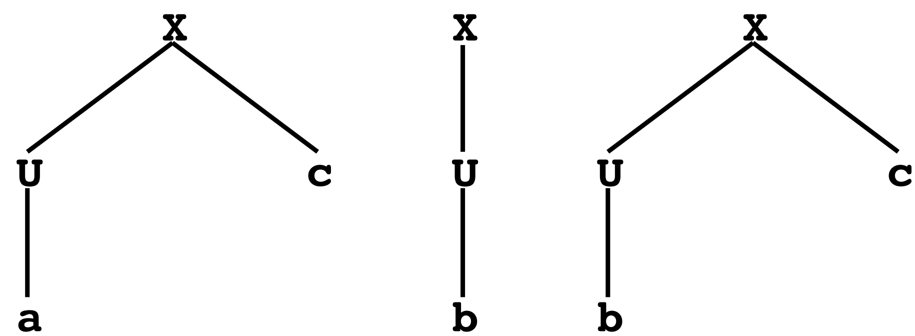


Fraction of 10,361 training sentences

Model analysis

- Grammar:

(1)	X	$\rightarrow$	U	c
(2)	X	$\rightarrow$	U	
(3)	U	$\rightarrow$	a	
(4)	U	$\rightarrow$	b	

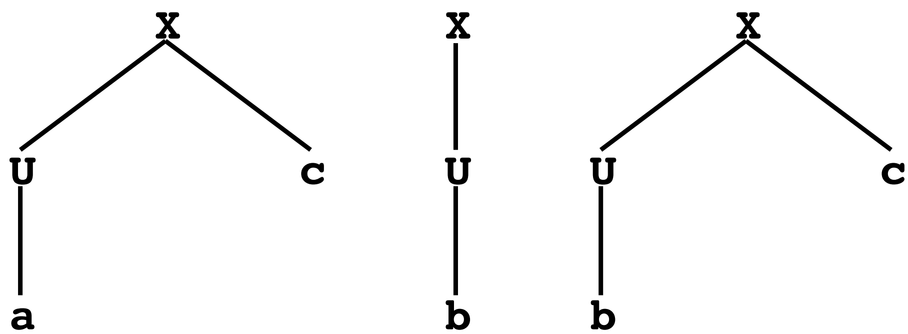


(S1)[1] **(S2)[2]** **(S3)[0]**

- Rule probabilities for Two-level PCFG:

(1)	S	; X	$\rightarrow$	U	c	(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



(S1)[1] (S2)[2] (S3)[0]

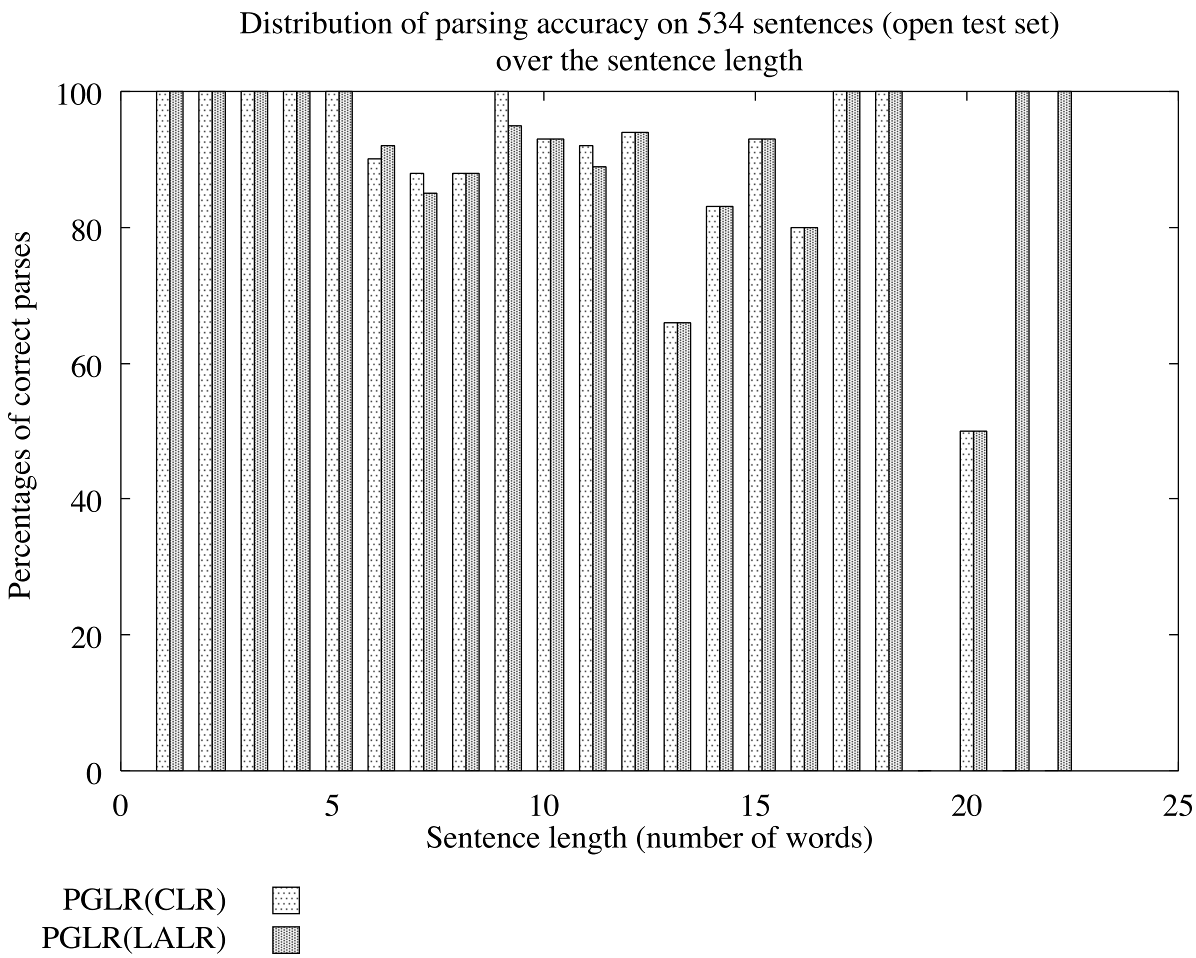
Models	(S1)	(S2)	(S3)
PCFG	1/9	4/9	2/9
Two-level PCFG	1/9	4/9	2/9
B&C	1/6	1/3	0
PGLR	1/3	2/3	0

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.

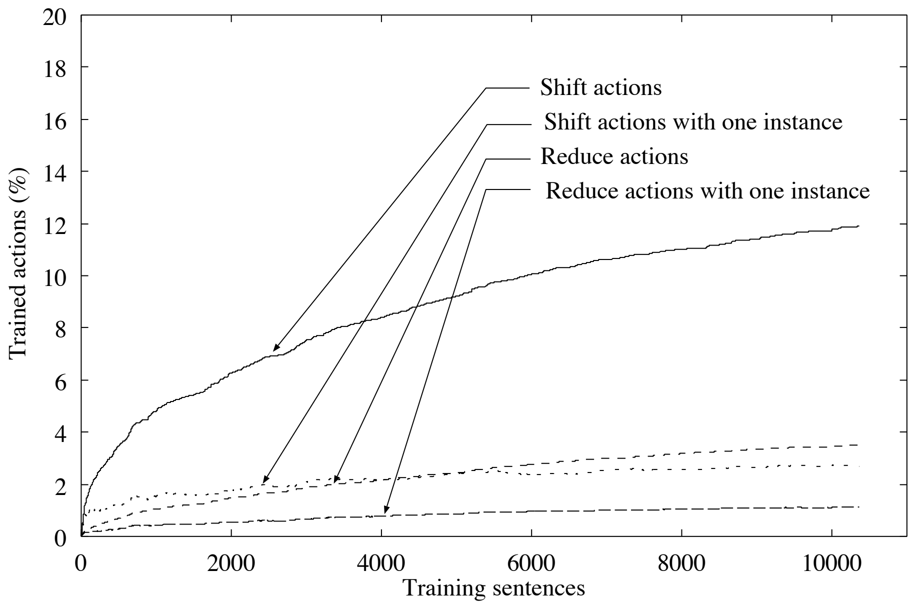
	LALR table	**CLR table**
States	856	3,715
Shift	11,445	43,833
Reduce	164,058	756,715
Goto	4,682	19,733
States in S_s	488	2,539
States in S_r	368	1,176

LALR and CLR table-based PGLR

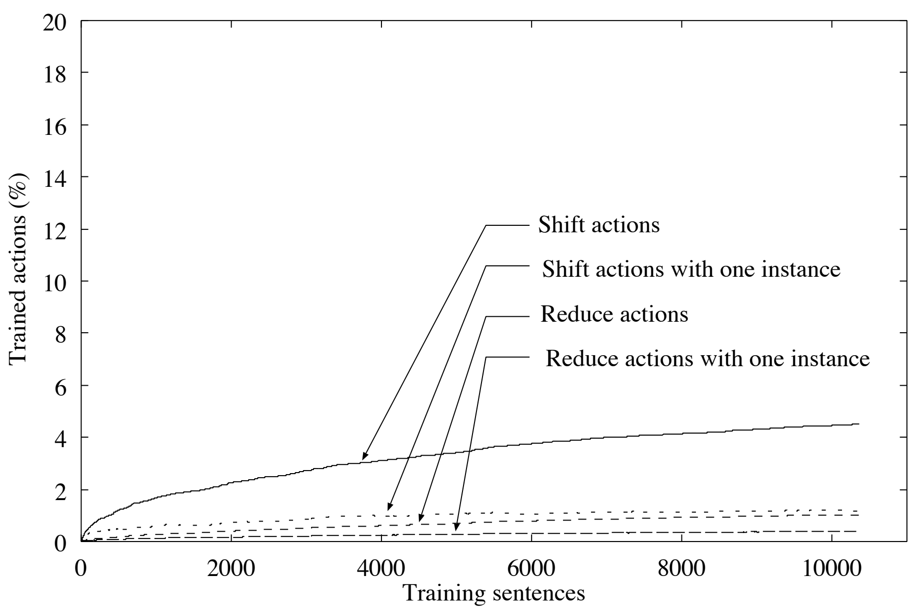


LALR and CLR table-based PGLR

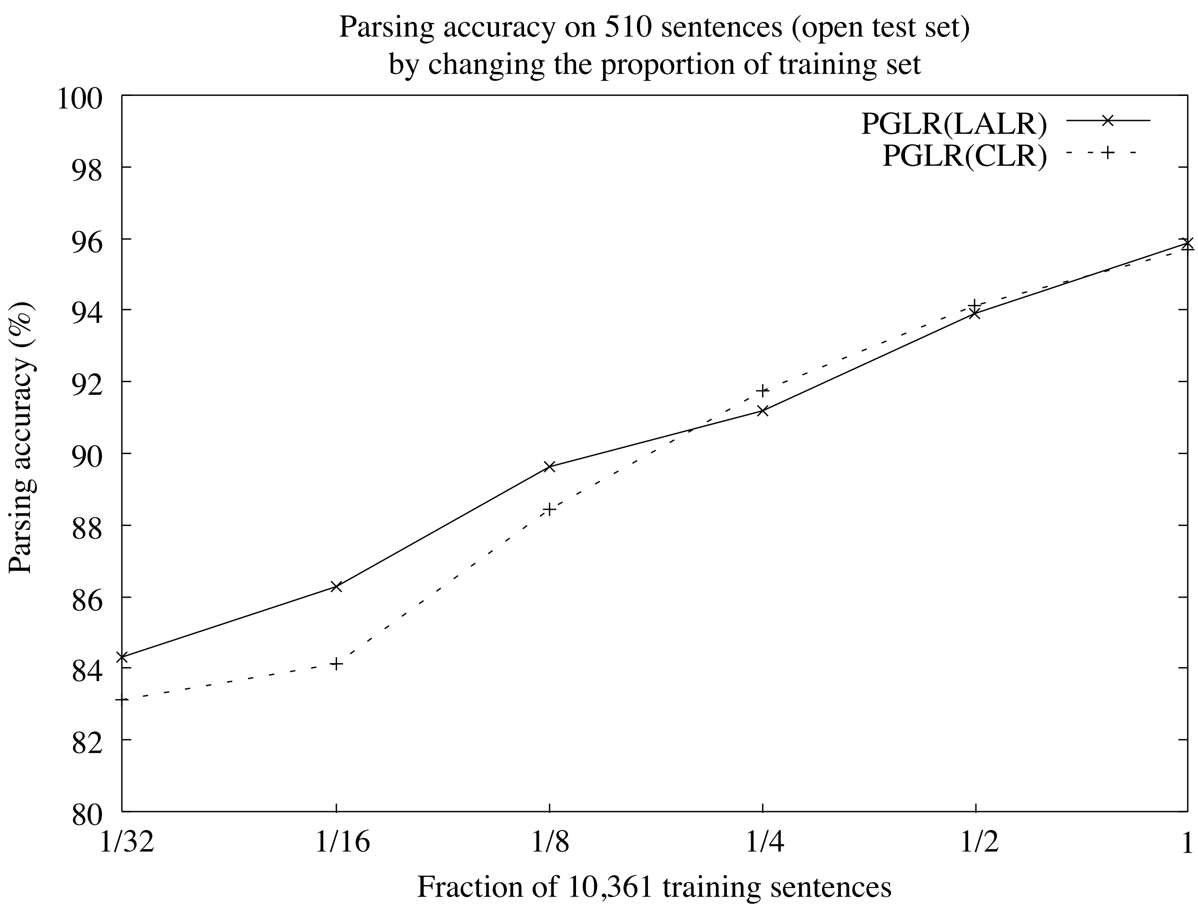
Learning curve of the actions in PGLR using an LALR table
(total of 11,445 shift and 164,058 reduce actions)



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



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).

$\Rightarrow$ Lexicalize the probabilistic model

$\Rightarrow$ Include long distance constraints

$\Rightarrow$ Verify the PGLR model with a larger corpus