HC–search for Incremental Parsing

Yijia Liu, Wanxiang Che, Bing Qin, and Ting Liu

哈爾濱工業大學

Standard Incremental Parsing

Goal: learning a function ***S*** to give ***oracle sequence*** highest score
Learning scheme:

- do beam search on input
- if an error is made (***highest sequence*** != ***oracle sequence***):
  - increase weight for oracle sequence
  - decrease weight for highest sequence

HC-search for Incremental Parsing

HC-search: Doppa et al., 2014:

- structure prediction error is decomposed into two parts
  - $\mathcal{H}$euristic part: the gold structure not included in the set of outputs
  - $\mathcal{C}$ost part: the gold structure not ranked as the highest output

$$\mathcal{E}_{\mathcal{HC}} = \underbrace{L(x, y^*_{\mathcal{H}}, y^*)}_{\epsilon_{\mathcal{H}}} + \underbrace{L(x, \hat{y}, y^*) - L(x, y^*_{\mathcal{H}}, y^*)}_{\epsilon_{\mathcal{C}|\mathcal{H}}},$$

Our Method

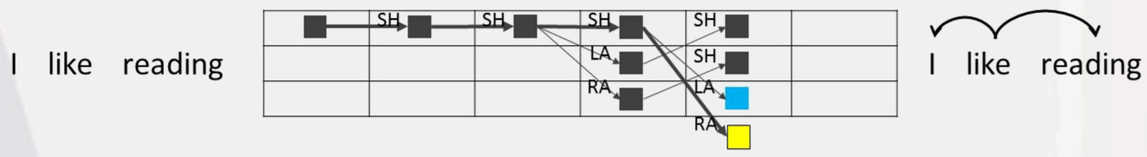
- Decompose S into two functions $\mathcal{H}$ and $\mathcal{C}$
  - Goal of $\mathcal{H}$: include oracle sequence in the output
  - Goal of $\mathcal{C}$: rank the gold tree highest
- Handling the ambiguous problem
  - $\mathcal{H}$ not necessarily rank oracle sequence highest

H-step learning scheme

- do beam search on input
- if oracle sequence falls out beam:
  - increase weight for oracle sequence
  - pick a sequence from beam and decrease its weight
  - We tried pick the *BEST* scored and *WORST* scored

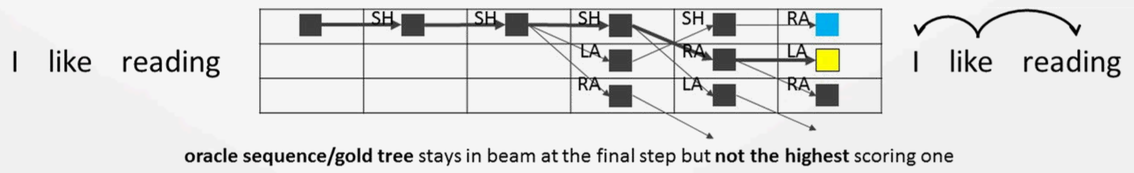
C-step learning scheme

- a typical ranking problem
  - *COARSE* grain ranking: rank the smallest loss outputs higher than the rest
  - *FINE* grain ranking: rank the smaller loss outputs higher

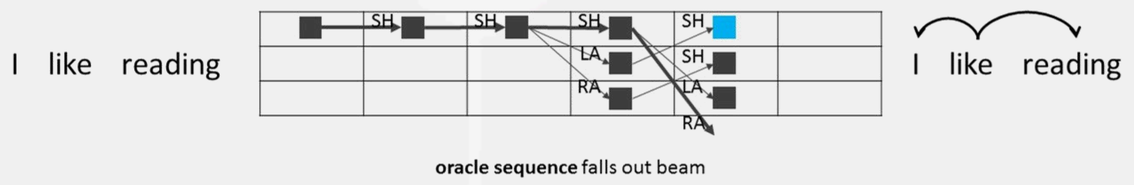


Conclusion: We proposed a new approach for incremental parsing based on the HC-search framework. H-step uncovers high-quality candidate outputs and C-step selects the best loss output with a ranking model.

Two Types of Errors



oracle sequence/gold tree stays in beam at the final step but **not the highest** scoring one

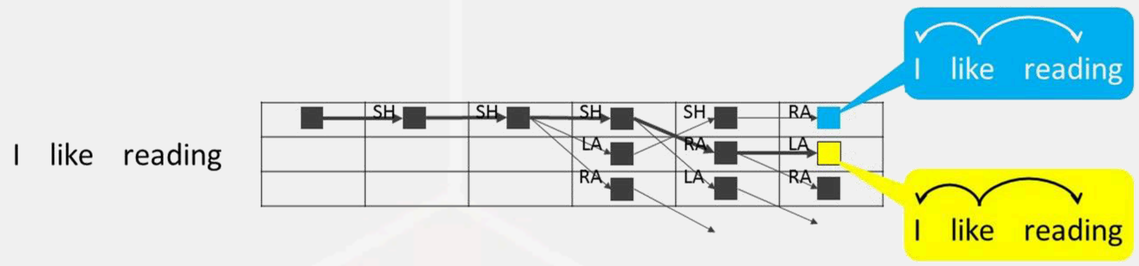


oracle sequence falls out beam

a function ***S*** services a ***dual-role***:

- keeping oracle sequence in beam (reduce second type errors)
- scoring gold tree highest (reduce first type errors)

Ambiguity



Two roles sometimes conflict, serving them with single function ***S*** is problematic

Experiments

Parser	PTB			CTB5		
	Dev	Test	SPD	Dev	Test	SPD
BASELINE	92.95	92.48	1x	86.76	86.44	1x
BEST+FINE	**93.13**	**92.76** (+0.28)	1.25x	87.25	87.04 (+0.60)	1.08x
BEST+COARSE	92.94	92.44 (-0.04)	1.30x	86.61	86.51 (+0.07)	1.07x
WORST+FINE	93.12	92.73 (+0.25)	1.33x	**87.27**	**87.15** (+0.71)	1.22x
WORST+COARSE	92.89	92.47 (-0.01)	1.30x	86.95	86.82 (+0.38)	1.20x
BASELINE+FINE	93.06	92.53 (+0.05)		87.07	86.70 (+0.26)	

Results on PTB, CTB5 with beam=64, HC-decomposition improves performance

Parser	PTB			CTB5		
	$\epsilon_{\mathcal{H}}$	$\epsilon_{\mathcal{C}\|\mathcal{H}}$	$\mathcal{E}_{\mathcal{HC}}$	$\epsilon_{\mathcal{H}}$	$\epsilon_{\mathcal{C}\|\mathcal{H}}$	$\mathcal{E}_{\mathcal{HC}}$
BEST+FINE	3.69	**3.90**	**6.87**	8.77	**5.72**	12.75
BEST+COARSE		4.14	7.06		6.93	13.39
WORST+FINE	**3.05**	4.62	6.88	**7.75**	7.33	**12.73**
WORST+COARSE		5.09	7.11		7.58	13.05
BASELINE+FINE	3.70	4.10	6.94	8.81	6.27	12.93

Error Decomposition Analysis: Relaxed H-step learning objective
☑recall more high-quality output ☑increase difficulty of ranking

Parser	*non-mixture*	*mixture*
BASELINE	92.48	
BASELINE+FINE	92.53	92.94
BEST+FINE	92.76	93.02
WORST+FINE	92.73	93.05

Improvement can be further achieved by mixing H- and C- step scores.