HC-search for Incremental Parsing

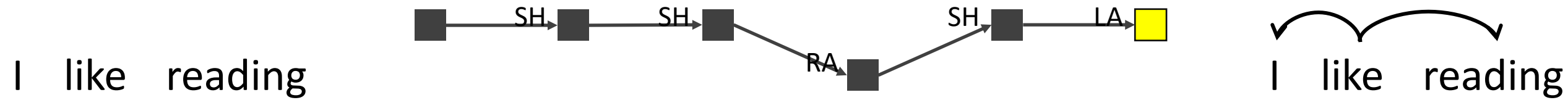
Yijia Liu, Wanxiang Che, Bing Qin, and Ting Liu

Research Center for Social Computing and Information Retrieval

Harbin Institute of Technology

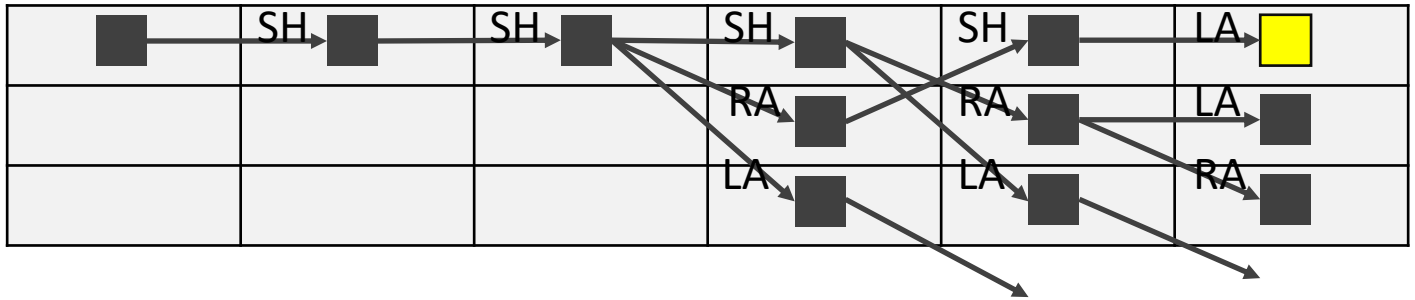
Standard Incremental Parsing

Standard Incremental Parsing



Standard Incremental Parsing

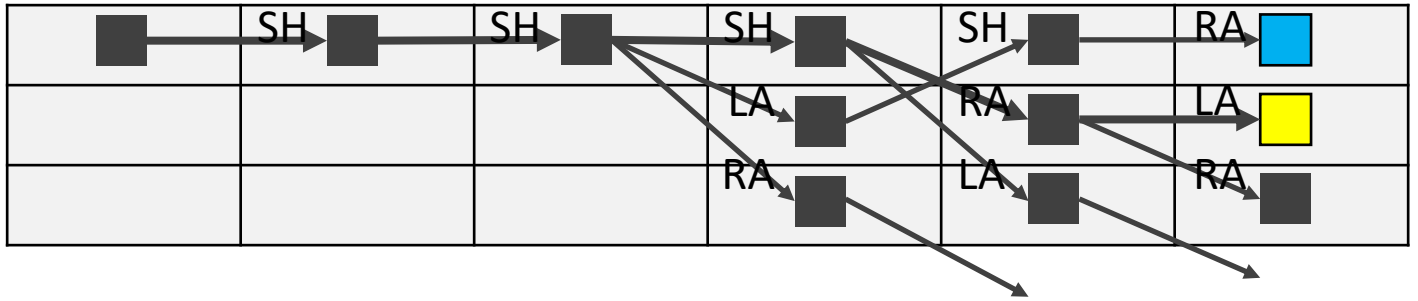
I like reading



I like reading

Learning in Standard Incremental Parsing

I like reading

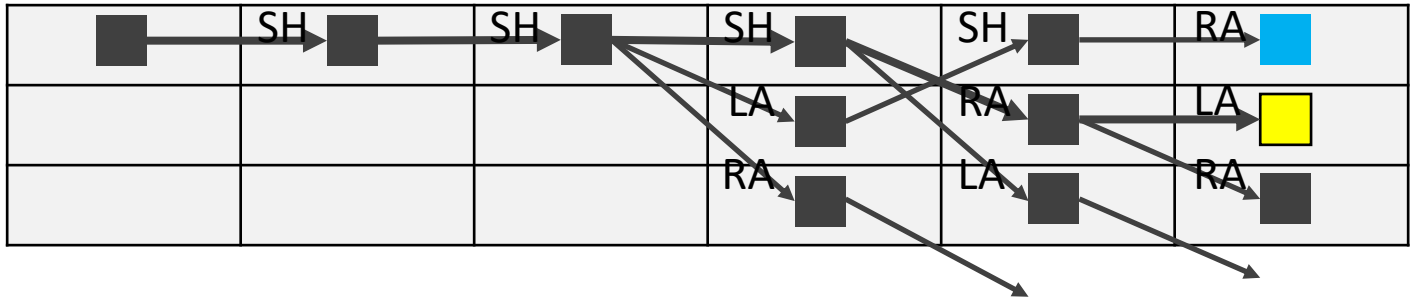


I like reading

- Goal: learning a function ***S*** to give ***oracle sequence*** highest score
- Typical 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

Two types of Errors in Learning Standard Incremental Parsing

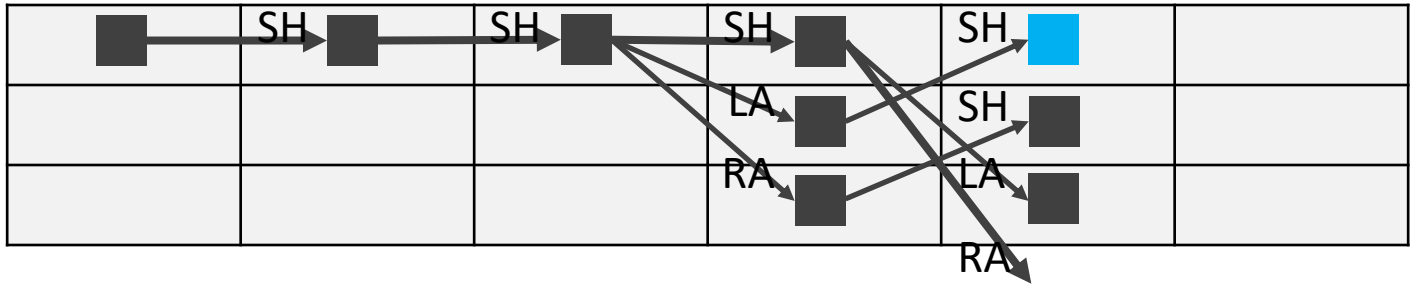
I like reading



I like reading

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

I like reading



I like reading

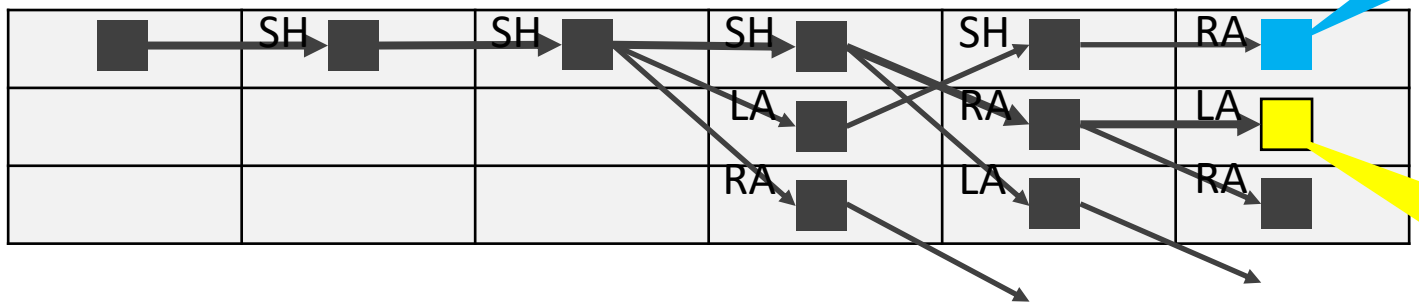
oracle sequence falls out beam

Two types of Errors in Learning Standard Incremental Parsing

- a function ***S*** services a dual-role:
  - scoring *gold tree* highest (reduce first type errors)
  - keeping *gold tree* in beam (reduce second type errors)

Ambiguous in Transition System

I like reading

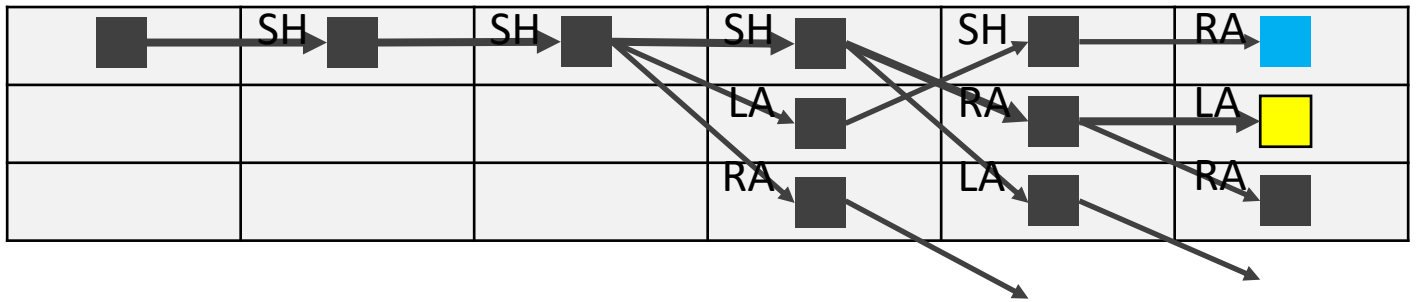


I like reading

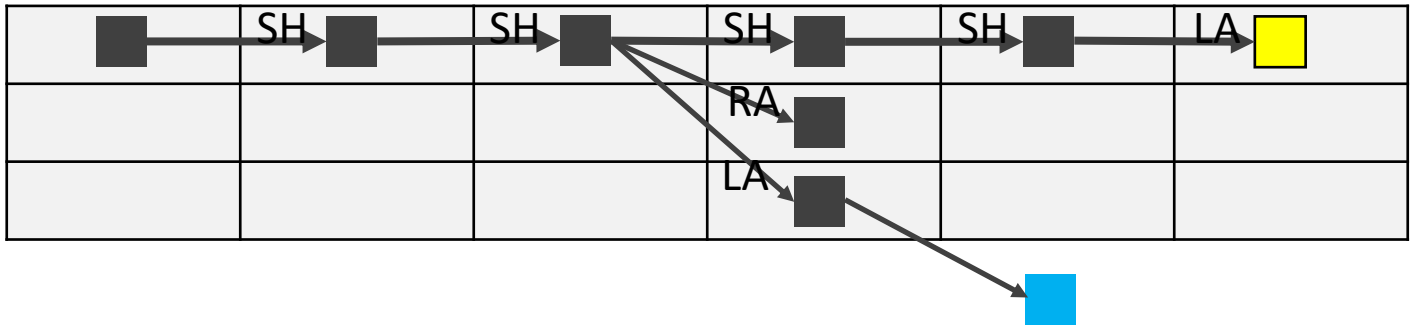
I like reading

- decreasing weight for ***highest sequence*** results in wrong punishment
- single function ***S*** serves the dual-role can be problematic
  - due to two roles conflict sometimes

Ambiguous in Transition System



after updates



$\mathcal{HC}$-search Framework

- 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

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

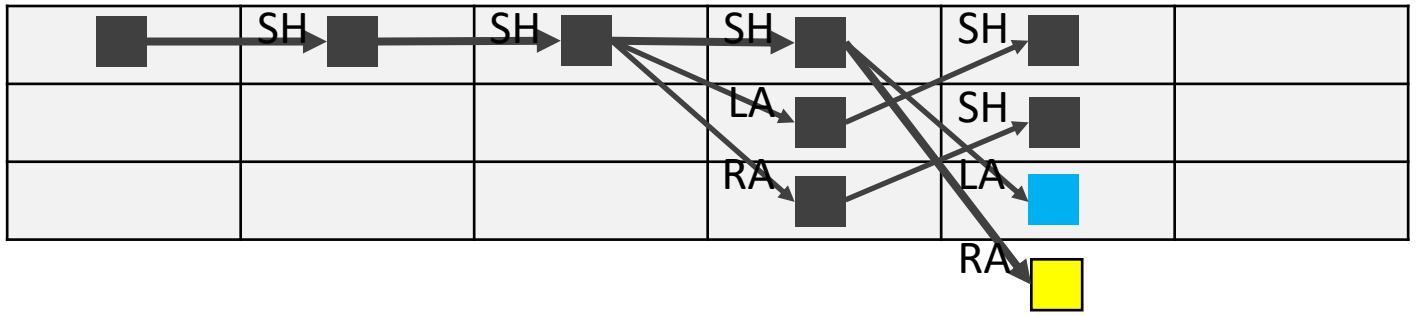
Our method

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

Our method

- $\mathcal{H}$-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

I like reading



I like reading

Our method

- $\mathcal{C}$-step learning scheme
  - a typical ranking problem
    - employing a evaluation function on the set of outputs
  - *COARSE* grain ranking:
    - rank the best outputs higher than the rest
  - *FINE* grain ranking:
    - rank the better outputs higher

Experiment

- On PTB and CTB5, beam = 64

Parser	PTB			CTB5		
	Dev	Test	SPD	Dev	Test	SPD
BASELINE	92.95	92.48	1x	86.76	86.44	1x
BASELINE+FINE	93.06	92.53 (+0.05)		87.07	86.70 (+0.26)	

Experiment

- On PTB and CTB5, beam = 64

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)	

Error Decomposition Analysis

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

- Relaxed $\mathcal{H}$-step learning objective
  - recall more high-quality output
  - increase difficulty of ranking

Conclusion

- We decompose incremental parsing loss based on the HC-search.
- We propose the relaxed H-step learning objective that recalls more high-quality outputs
- We found fine-grained ranking is more suitable for ranking in parsing
- Code can be found at: https://github.com/ExpResults/hc-incremental-parsing

Thanks and Questions!