Graph-Structured-Stack Beam Search for Shift-Reduce Constituency Parsing

Juneki Hong Liang Huang {juneki.hong, liang.huang.sh}@gmail.com

School of EECS, Oregon State University

Oregon State University

Motivations

Deep-Learning Constituency Parsers achieve state-of-the-art results but suffer from slow runtimes.

- previous deep-learning parsers have exhaustively encoded stacks, which cannot take advantage of dynamic programming.
- the latest state-of-the-art minimum-span parser can efficiently use LSTM span features, but still runs in $O(N^3)$ cubic time.

Here we propose:

- **Graph-Structured Stack**: a formalism that can represent multiple stacks at once. (Tomita 1991)
  - we can temporarily merge all equivalent looking states together, based on their top-most element.
  - this allows efficient beam search decoding in linear time.
- a fast left-to-right incremental **Shift-Reduce Constituency Parser**.
  - uses span features to efficiently query the LSTM.
  - modifying the width of the beam provides a speed-vs-accuracy tradeoff.

Graph Structured Stack

State Representation

- topmost span on top of the stack taken as the state
- stacks with the same span considered "equivalent" and merged together
- each state maintains left pointers to remember the next item in the stack when popped

State Merging

- *shift merge*: two states that shift into the same span
  - both states are temporarily merged together
  - reduce actions will "unpack" shift-merged states into multiple subsequent states.
- *reduce merge*: states that reduce into the same span
  - only the state with the higher model score is taken

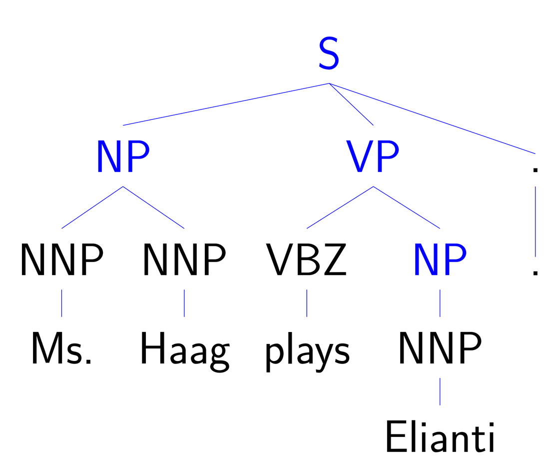

Figure: Example sentence and corresponding parse tree.

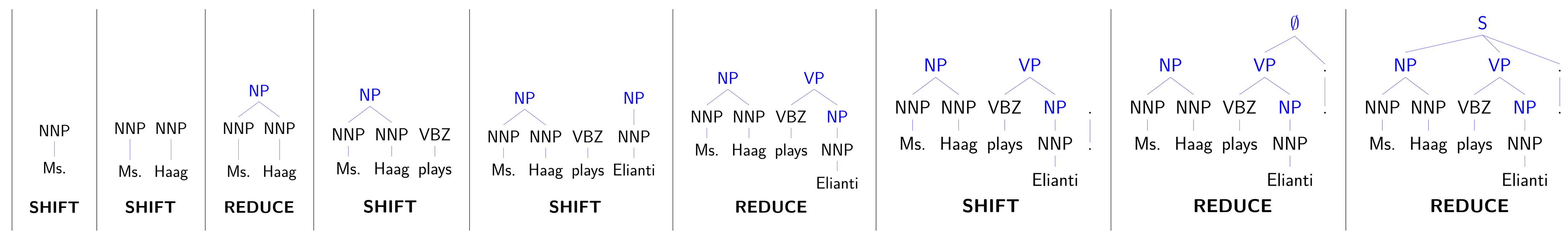

Figure: Derivation of a parse tree.

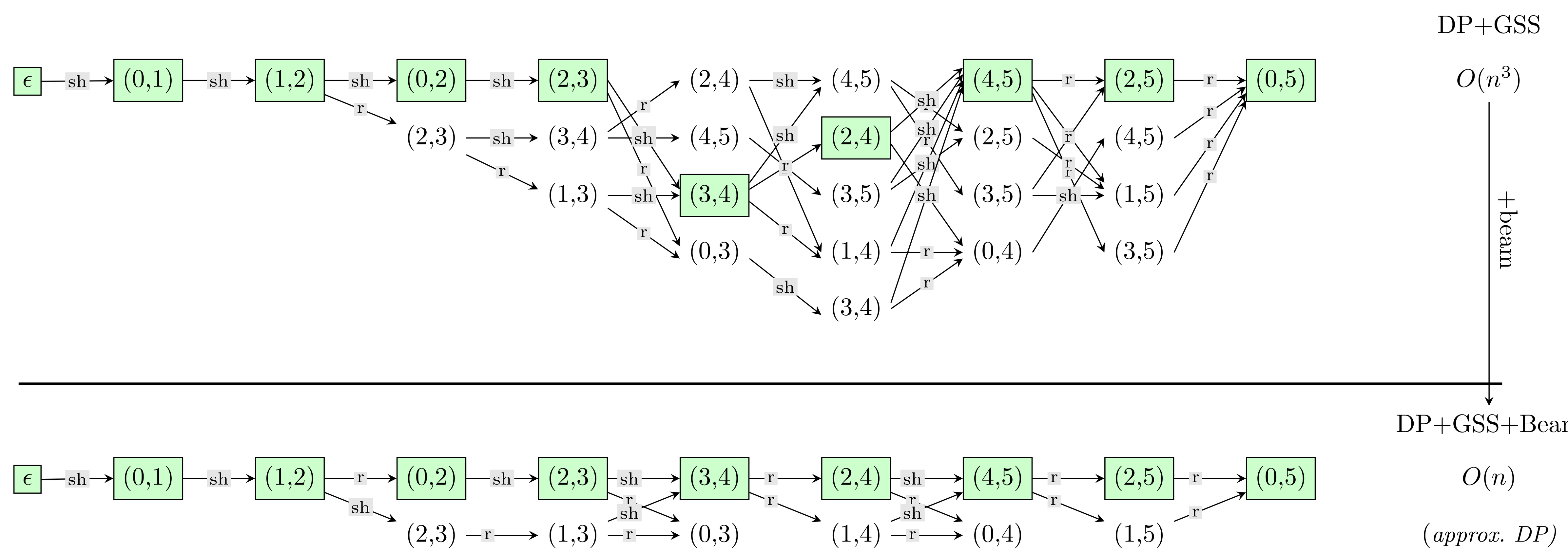

Figure: GSS dynamic programming during decoding; correct derivation colored green.

Syntactic Constituency Parsing

Linear-Time Parsing with beam search

- simulates greedy parsing at beam-size 1
- approximates exhaustive chart parsing at greater beam-sizes

Span-Based Parsing (Cross & Huang 2016)

- stack maintains spans instead of any subtrees
- no tree structure representations anywhere
- span features calculated as differences across Bi-LSTM output

Labeling Spans (label$_X$, nolabel)

- performed after each structural (shift, reduce) action
  - each action produces a subsequent span (i, j)
  - highest scoring label over the span is chosen
- nolabel makes binarization of the constituency tree unnecessary

Decoding With Pretrained Model

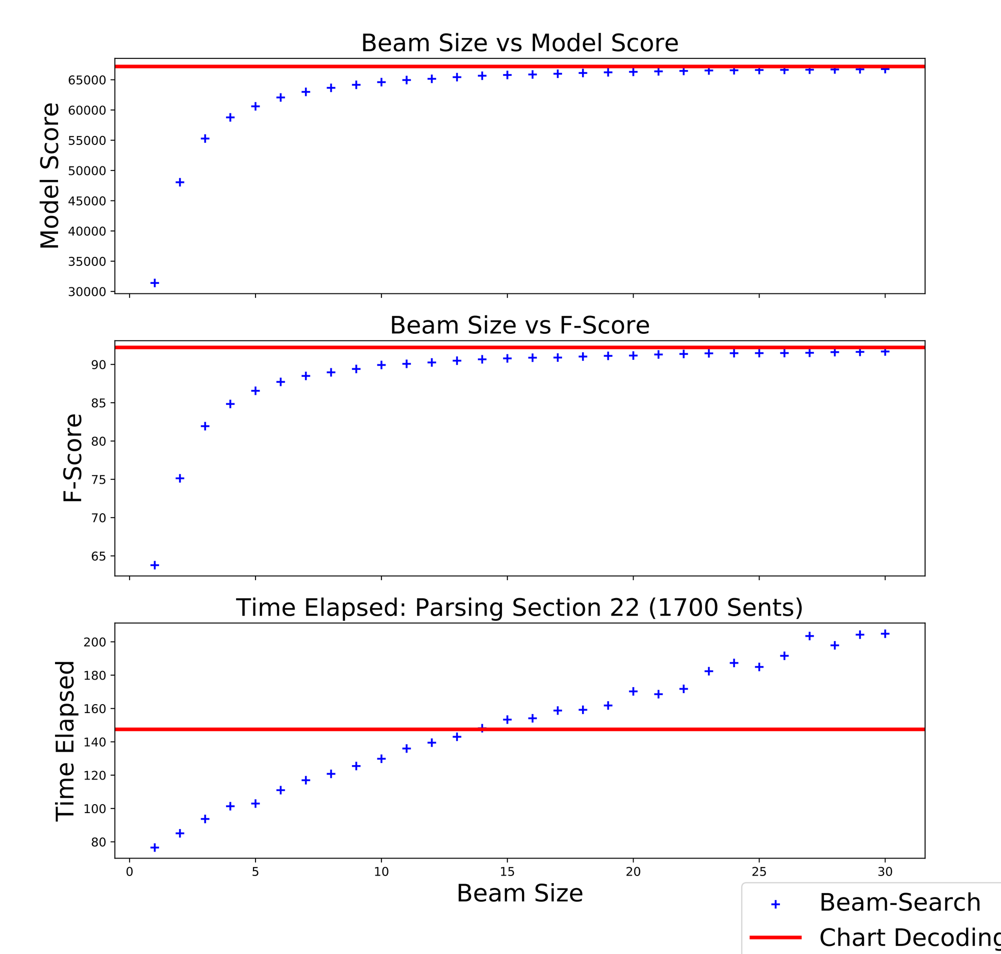

Figure: Speed-Accuracy tradeoff, decoding with different beam sizes.

Deductive System

input	$w_0 \dots w_{n-1}$
state form	$\ell : \langle i, j \rangle : (c, v, \pi)$
axiom(p_0)	$0 : \langle 0, 0 \rangle : (0, 0, \emptyset)$, goal 2n - 1 : $\langle 0, n \rangle : (c, c, \{p_0\})$

$$\textbf{shift} \quad \frac{\ell : \langle \_, j \rangle : (c, \_, \_)}{\ell + 1 : \langle j, j+1 \rangle : (c + \xi, 0, \{p\})} \; j < n$$

$$\textbf{reduce} \quad \frac{\_ : \langle k, i \rangle : (c', v', \pi') \quad \ell : \langle i, j \rangle : (\_, v, \pi)}{\ell + 1 : \langle k, j \rangle : (c' + v + \sigma, v' + v + \sigma, \pi')}$$

Deductive system, where ξ and δ are the shift and reduce costs. c and v are the prefix and inside scores of a state. π is a set of predictor states.

Recurrent Neural Models

- bi-directional LSTM in Cross & Huang (2016)
- no explicit discourse/syntactic tree structures represented in features
- span boundary LSTM representations are passed to FF network to calculate scores of actions/labels

Training & Emipirical Evaluation

Settings

- use training with max-violation updates & **dynamic oracle**
- set most hyperparams based on Stern et. al. 2017

Final Parsing Results (F1 scores)

	Development Set 22			Test Set 23		
	LR	LP	F1.	LR	LP	F1.
Durett and Klein(2015)	-	-	-	-	-	91.1
Cross and Huang (2016)	91.07	92.22	91.64	90.5	92.1	91.3
Liu and Zhang (2016)	-	-	-	91.3	92.1	91.7
Stern et. al. (2017)	92.12	92.09	91.95	90.63	92.98	**91.79**
GSS Beam 1	88.92	92.72	90.62	87.25	92.63	89.86
GSS Beam 10	91.13	92.90	92.01	90.39	92.85	91.60
GSS Beam 20	91.18	93.01	92.08	90.54	92.97	**91.74**

Speed Comparison (Sentences / Second)

	Complexity	Speed
Cross and Huang (2016)	$O(N)$	21.0
Stern et. al. (2017)	$O(N^3)$	12.2
GSS Beam 1	$O(N)$	22.2
GSS Beam 10	$O(N)$	14.2
GSS Beam 20	$O(N)$	10.7