Expected F-Measure Training for Shift-Reduce Parsing with Recurrent Neural Networks

Wenduan Xu

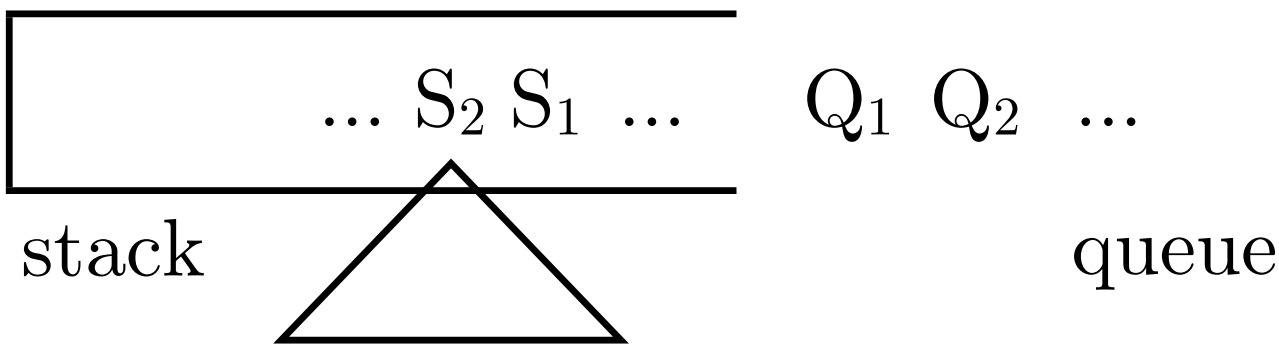
Cambridge University

12th June, 2016

Another Title...

Training a global shift-reduce model that is also optimized towards the final evaluation metric

Shift-Reduce Parsing



Shift-Reduce Parsing

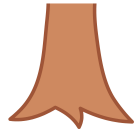
step	stack $(s_n, \ldots, s_1, s_0)$	queue $(q_0, q_1 \ldots, q_n)$	action
0		Ms. Haag plays Elianti	

Shift-Reduce Parsing

step	stack $(s_n, \ldots, s_1, s_0)$	queue $(q_0, q_1 \ldots, q_n)$	action
0		Ms. Haag plays Elianti	
1		Haag plays Elianti	SHIFT
2		plays Elianti	SHIFT

Shift-Reduce Parsing

step	stack $(s_n, \ldots, s_1, s_0)$	queue $(q_0, q_1 \ldots, q_n)$	action
0		Ms. Haag plays Elianti	
1		Haag plays Elianti	SHIFT
2		plays Elianti	SHIFT
3		plays Elianti	REDUCE

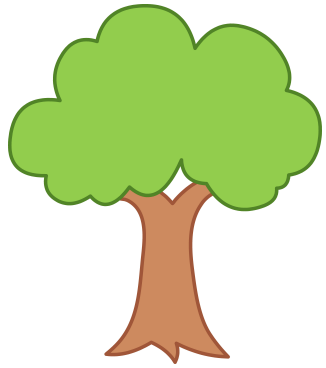


Shift-Reduce Parsing

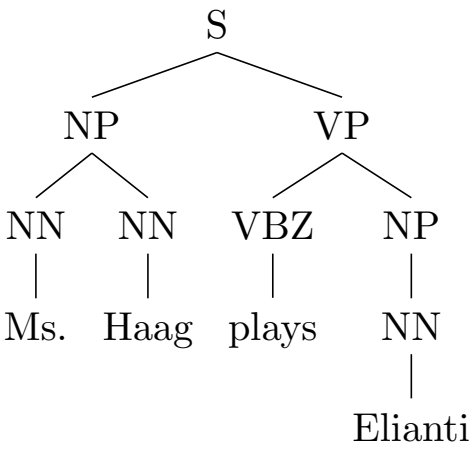
step	stack $(s_n, \ldots, s_1, s_0)$	queue $(q_0, q_1 \ldots, q_n)$	action
0		Ms. Haag plays Elianti	
1		Haag plays Elianti	SHIFT
2		plays Elianti	SHIFT
3		plays Elianti	REDUCE
4		plays Elianti	UNARY
5		Elianti	SHIFT
6			SHIFT
7			UNARY

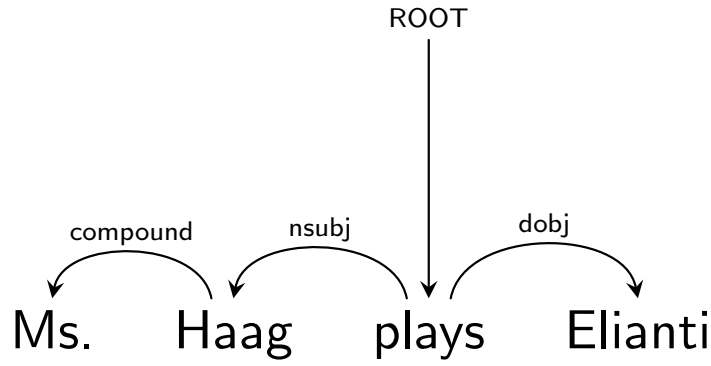
Shift-Reduce Parsing

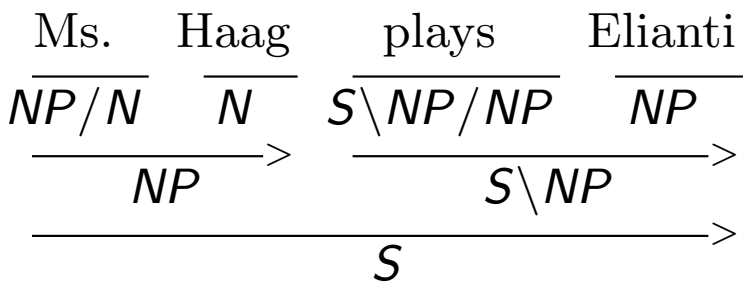
step	stack $(s_n, \ldots, s_1, s_0)$	queue $(q_0, q_1 \ldots, q_n)$	action
0		Ms. Haag plays Elianti	
1		Haag plays Elianti	SHIFT
2		plays Elianti	SHIFT
3		plays Elianti	REDUCE
4		plays Elianti	UNARY
5		Elianti	SHIFT
6			SHIFT
7			UNARY
8			REDUCE
9			REDUCE



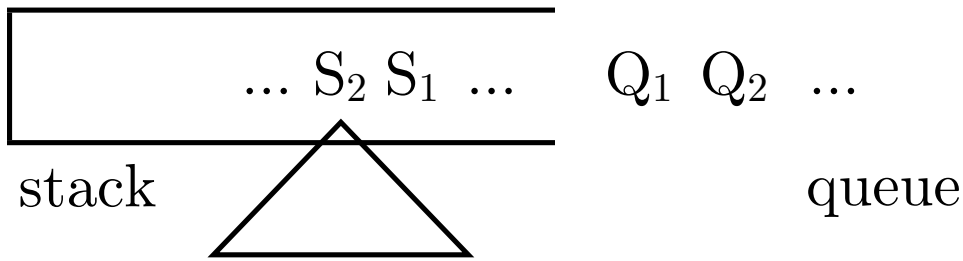
Shift-Reduce Parsing



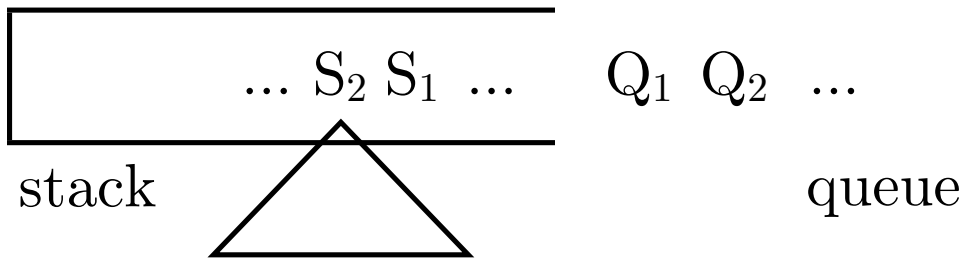




Shift-Reduce Parsing



Shift-Reduce Parsing



SHIFT SHIFT SHIFT REDUCE SHIFT SHIFT ...

Shift-Reduce Parsing

- Linear model (perceptron, SVM etc.)

$$score(t_i) = \mathbf{f}(\langle s, q \rangle, t_i) \cdot \mathbf{w}$$

Shift-Reduce Parsing

- Linear model (perceptron, SVM etc.)

$$score(t_i) = \mathbf{f}(\langle s, q \rangle, t_i) \cdot \mathbf{w}$$

$$score(t) = \sum_i score(t_i)$$

Shift-Reduce Parsing

- Linear model (perceptron, SVM etc.)

$$score(t_i) = \mathbf{f}(\langle s, q \rangle, t_i) \cdot \mathbf{w}$$

$$score(t) = \sum_i score(t_i)$$

$$t^* = \arg\max_t score(t)$$

Shift-Reduce Parsing

- Linear model (perceptron, SVM etc.)

$$score(t_i) = \mathbf{f}(\langle s, q \rangle, t_i) \cdot \mathbf{w}$$

$$score(t) = \sum_i score(t_i)$$

$$t^* = \arg\max_t score(t)$$

- Great flexibility in defining the feature functions

Shift-Reduce Parsing

- Linear model (perceptron, SVM etc.)

$$score(t_i) = \mathbf{f}(\langle s, q \rangle, t_i) \cdot \mathbf{w}$$

$$score(t) = \sum_i score(t_i)$$

$$t^* = \arg\max_t score(t)$$

- Great flexibility in defining the feature functions
  - results in millions of sparse indicator features

Shift-Reduce Parsing

- Linear model (perceptron, SVM etc.)

$$score(t_i) = \mathbf{f}(\langle s, q \rangle, t_i) \cdot \mathbf{w}$$

$$score(t) = \sum_i score(t_i)$$

$$t^* = \arg\max_t score(t)$$

- Great flexibility in defining the feature functions
  - results in millions of sparse indicator features

- Used in many parsers
  - (e.g., Yamada and Matsumoto, 2003; Huang and Sagae, 2010; Zhang and Clark, 2011; Zhang and Nivre, 2011; Goldberg and Nivre, 2012; Bohnet et al., 2013; Zhu et al., 2013;)

Shift-Reduce Parsing Models

	feature templates
1	S_0wp, S_0c, S_0pc, S_0wc, S_1wp, S_1c, S_1pc, S_1wc, S_2pc, S_2wc, S_3pc, S_3wc,
2	Q_0wp, Q_1wp, Q_2wp, Q_3wp,
3	S_0Lpc, S_0Lwc, S_0Rpc, S_0Rwc, S_0Upc, S_0Uwc, S_1Lpc, S_1Lwc, S_1Rpc, S_1Rwc, S_1Upc, S_1Uwc,
4	S_0wS_1wc, S_0cS_1w, S_0wS_1c, S_0cS_1c, S_0wQ_0wp, S_0cQ_0wp, S_0wQ_0p, S_0cQ_0p, S_1wQ_0wp, S_1cQ_0wp, S_1wQ_0p, S_1cQ_0p,
5	S_0wS_1cQ_0p, S_0cS_1wQ_0p, S_0cS_1cQ_0wp, S_0cS_1cQ_0p, S_0pS_1pQ_0p, S_0wQ_0pQ_1p, S_0cQ_0wQ_1p, S_0cQ_0pQ_1wp, S_0cQ_0pQ_1p, S_0pQ_0pQ_1p, S_0wS_1cS_2c, S_0cS_1wS_2c, S_0cS_1cS_2w, S_0cS_1cS_2c, S_0pS_1pS_2p,
6	S_0cS_0H cS_0Lc, S_0cS_0H cS_0Rc, S_1cS_1H cS_1Rc, S_0cS_0RcQ_0p, S_0cS_0RcQ_0w, S_0cS_0LcS_1c, S_0cS_0LcS_1w, S_0cS_1cS_1Rc, S_0wS_1cS_1Rc.

Table 1: Feature templates.

(Zhang and Clark, 2011)

Shift-Reduce Parsing with Neural Networks

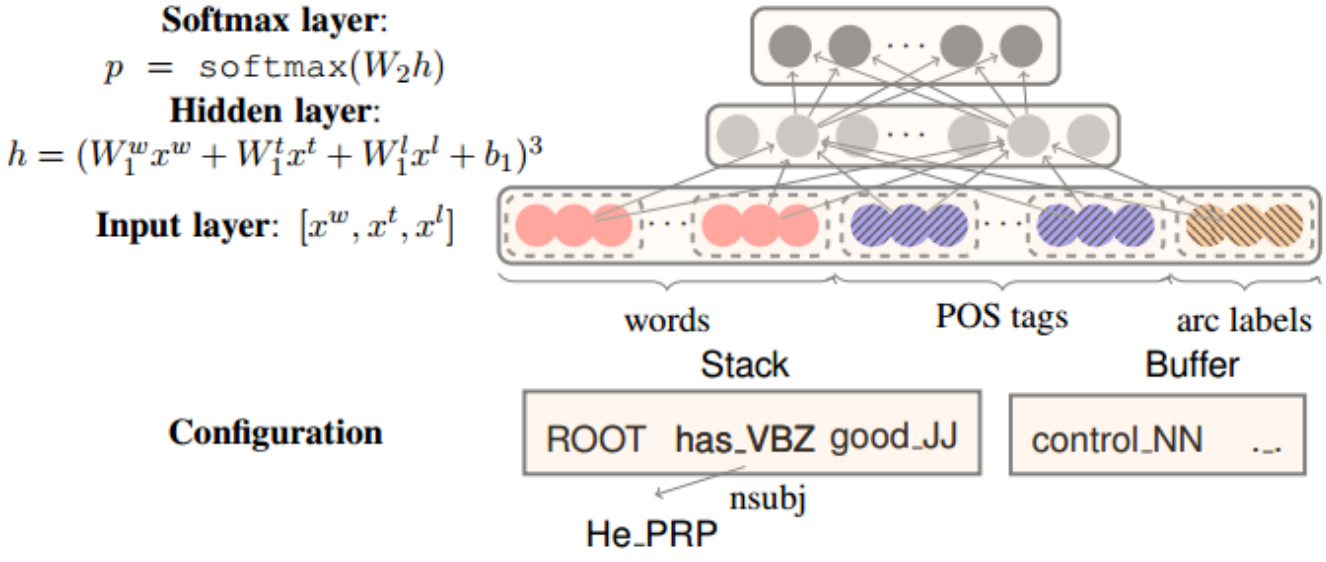


Figure 2: Our neural network architecture.

(Chen and Manning, 2014)

Shift-Reduce Parsing with Neural Networks

	NN	Beam (Train)	Beam (Test)	global
C&M, 2014	✓	✗	✓	✗

Shift-Reduce Parsing with Neural Networks

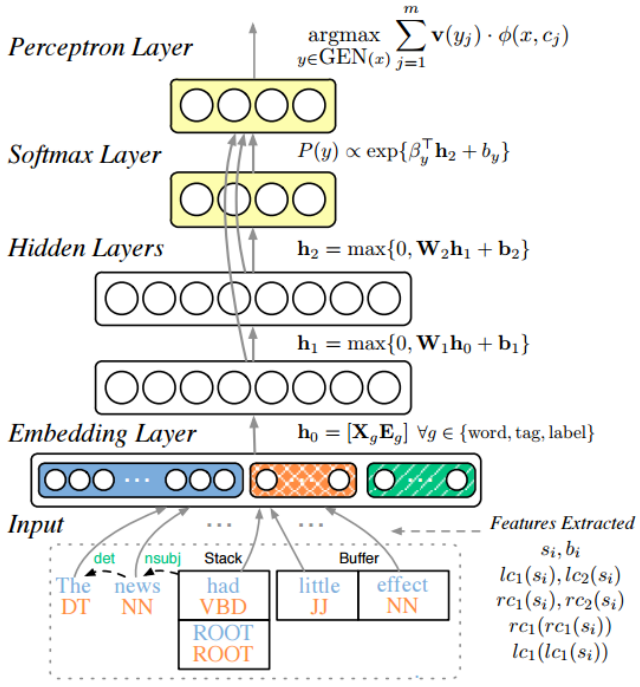
	NN	Beam (Train)	Beam (Test)	global
C&M, 2014	✓	✗	✓	✗
this work	✓	✓	✓	✓

Shift-Reduce Parsing with Neural Networks

	NN	Beam (Train)	Beam (Test)	global
C&M, 2014	✓	✗	✓	✗
this work	✓	✓	✓	✓

At the same time, the model is optimized towards the final evaluation metric ✓

Global Shift-Reduce NN Models



(Weiss et al., 2015)

Related Work

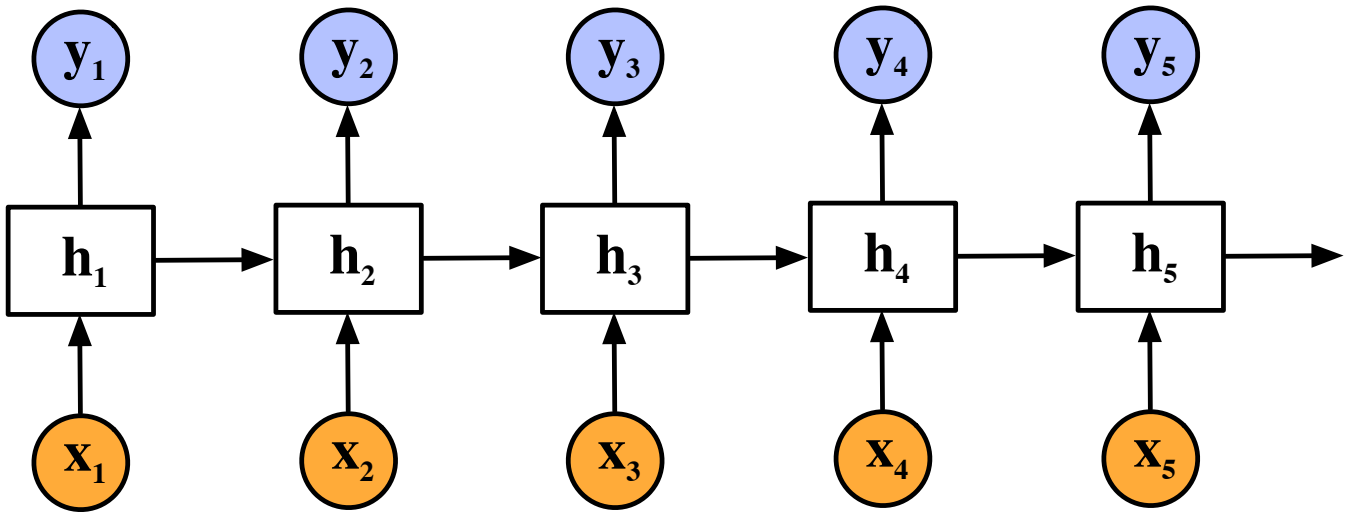
- Watanabe and Sumita, 2015
  - max-margin based objective
  - max-violation updates (Huang et al., 2012)
- Zhou et al., 2015
  - based on Chen and Manning, 2014
  - contrastive learning (Hinton, 2002; LeCun and Huang, 2005; Liang and Jordan, 2008)
- Andor et al., 2016
  - based on Chen and Manning, 2014 and Weiss et al., 2015
  - CRF (Bottou et al., 1997; Le Cun et al., 1998; Lafferty et al., 2001)

Related Work

- Watanabe and Sumita, 2015
  - max-margin based objective
  - max-violation updates (Huang et al., 2012)
- Zhou et al., 2015
  - based on Chen and Manning, 2014
  - contrastive learning (Hinton, 2002; LeCun and Huang, 2005; Liang and Jordan, 2008)
- Andor et al., 2016
  - based on Chen and Manning, 2014 and Weiss et al., 2015
  - CRF (Bottou et al., 1997; Le Cun et al., 1998; Lafferty et al., 2001)
- Optimizing task-specific metrics for parsing
  - (e.g., Goodman, 1996; Smith and Eisner, 2006; Auli and Lopez, 2011)

Expected F-Measure Training: Step 1

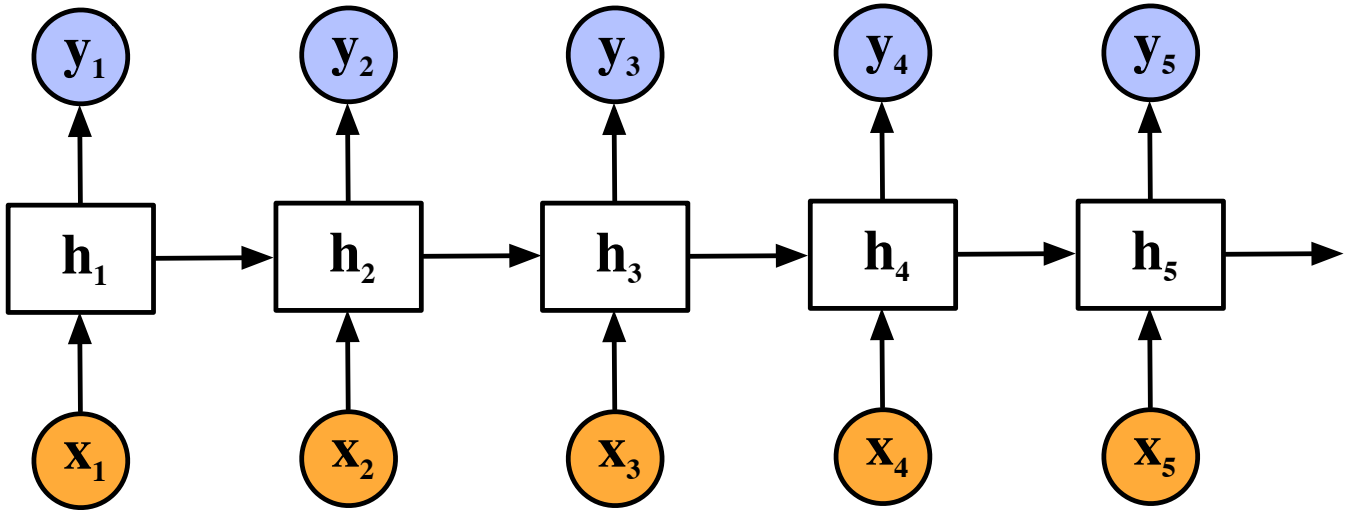
1. Train a baseline model using a cross-entropy loss (pretraining)



$$L(\theta) = -\sum_{k}^{T_i} p(t_k)$$

Expected F-Measure Training: Step 1

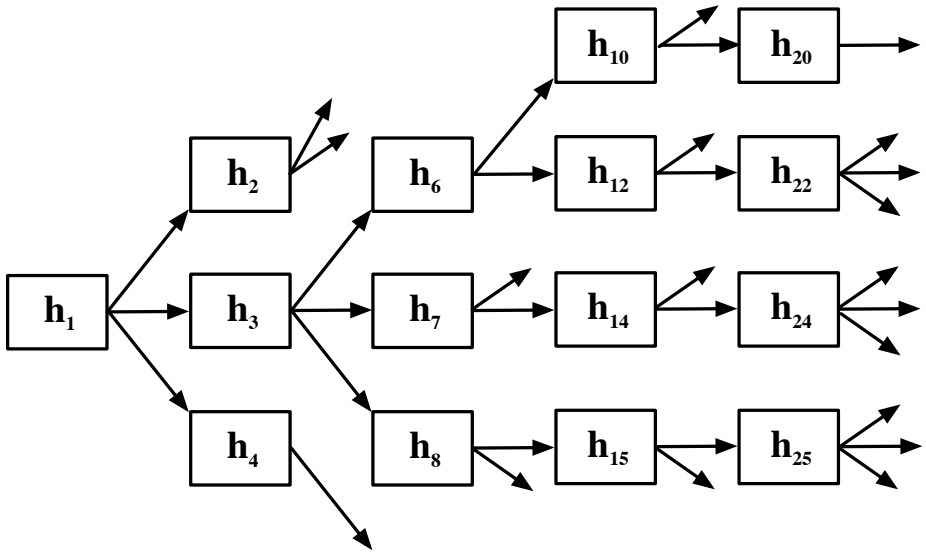
1. Train a baseline model using a cross-entropy loss (pretraining)



$$L(\theta) = -\sum_{k}^{T_i} p(t_k), \qquad \theta_B = \{\mathbf{U}, \mathbf{V}, \mathbf{W}\}$$

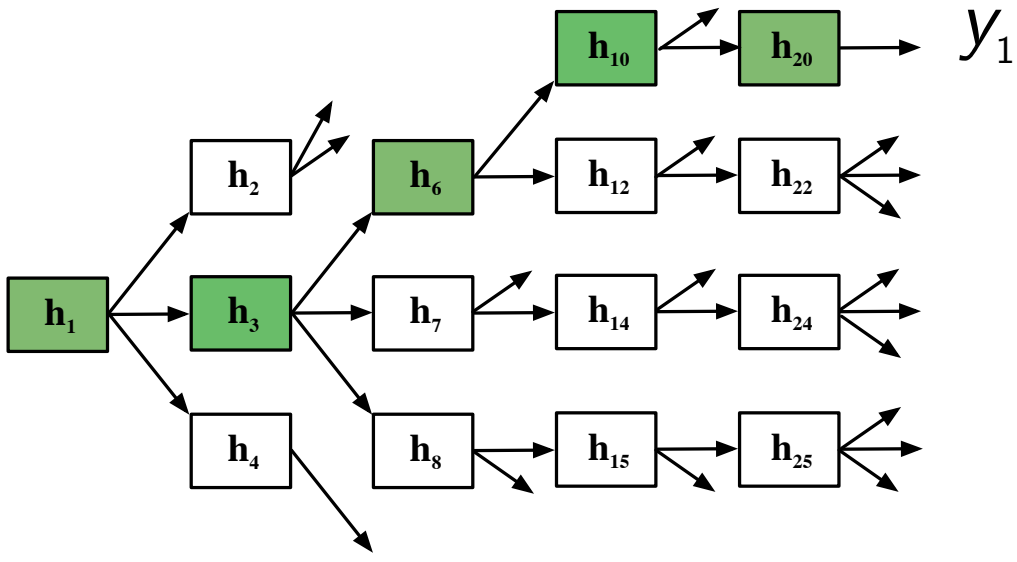
Expected F-Measure Training: Step 2

❷ Let $\theta = \theta_B$, and parse each sentence in the training data with beam search



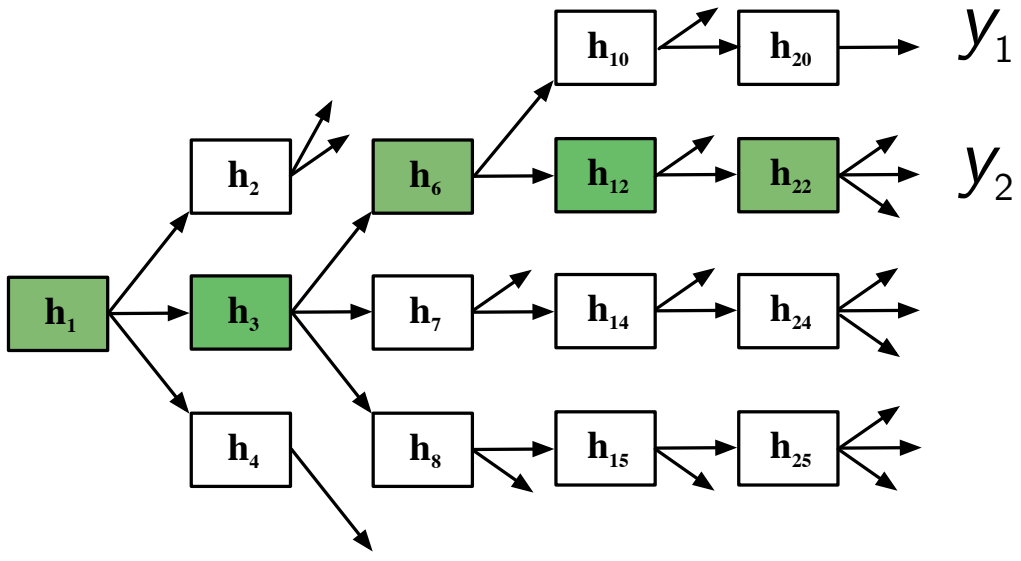
Expected F-Measure Training: Step 2

❷ Let $\theta = \theta_B$, and parse each sentence in the training data with beam search



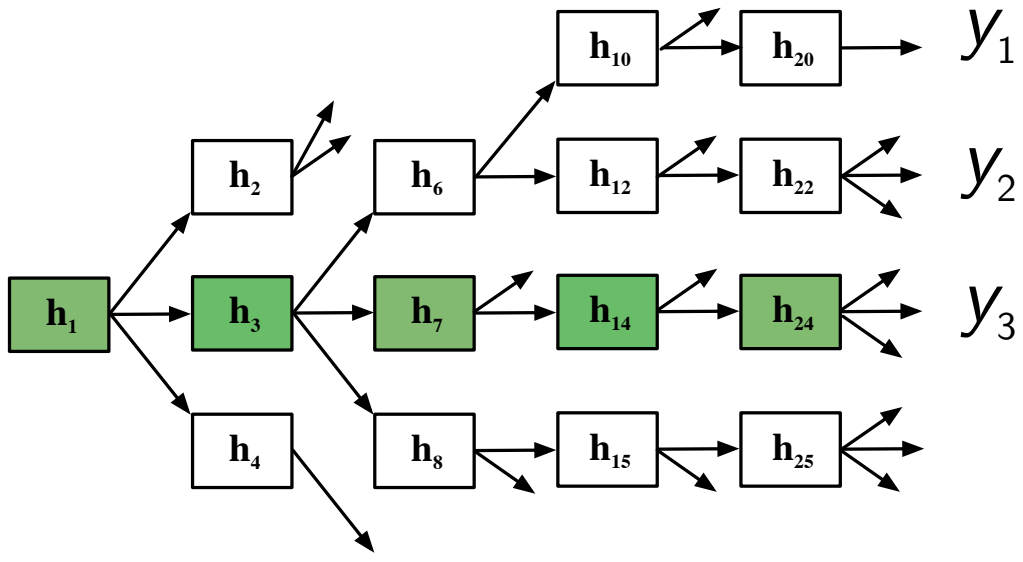
Expected F-Measure Training: Step 2

❷ Let $\theta = \theta_B$, and parse each sentence in the training data with beam search



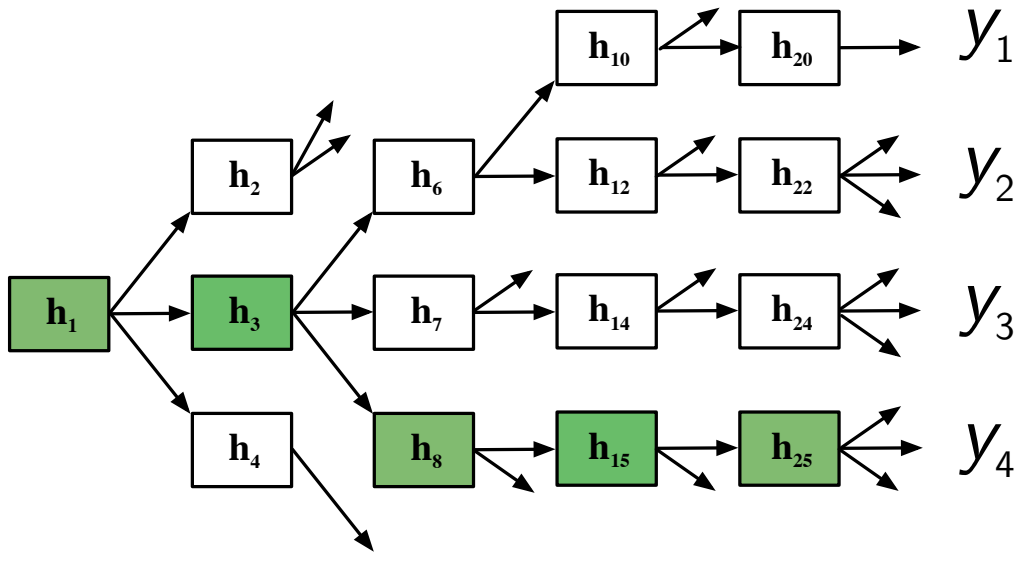
Expected F-Measure Training: Step 2

❷ Let $\theta = \theta_B$, and parse each sentence in the training data with beam search



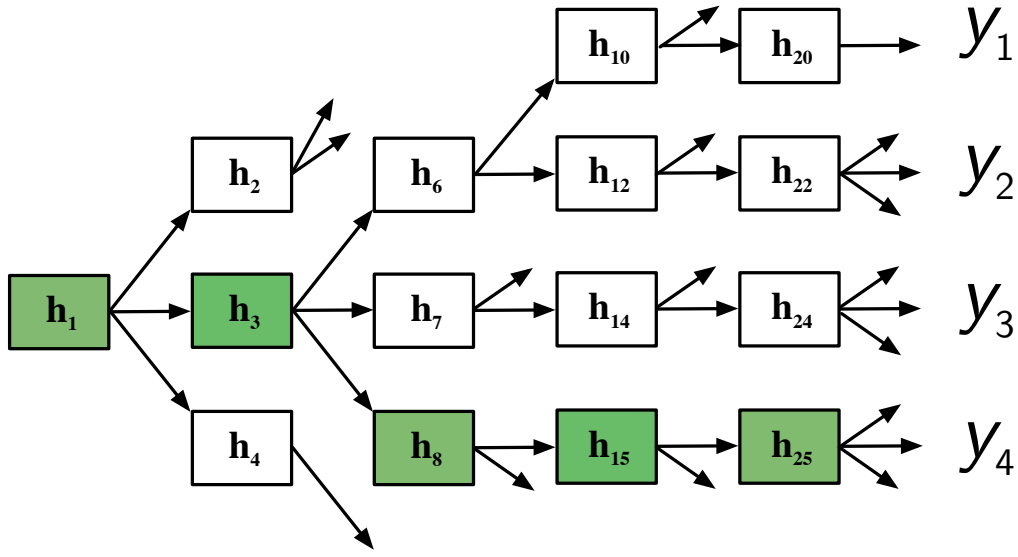
Expected F-Measure Training: Step 2

❷ Let $\theta = \theta_B$, and parse each sentence in the training data with beam search



Expected F-Measure Training: Step 2

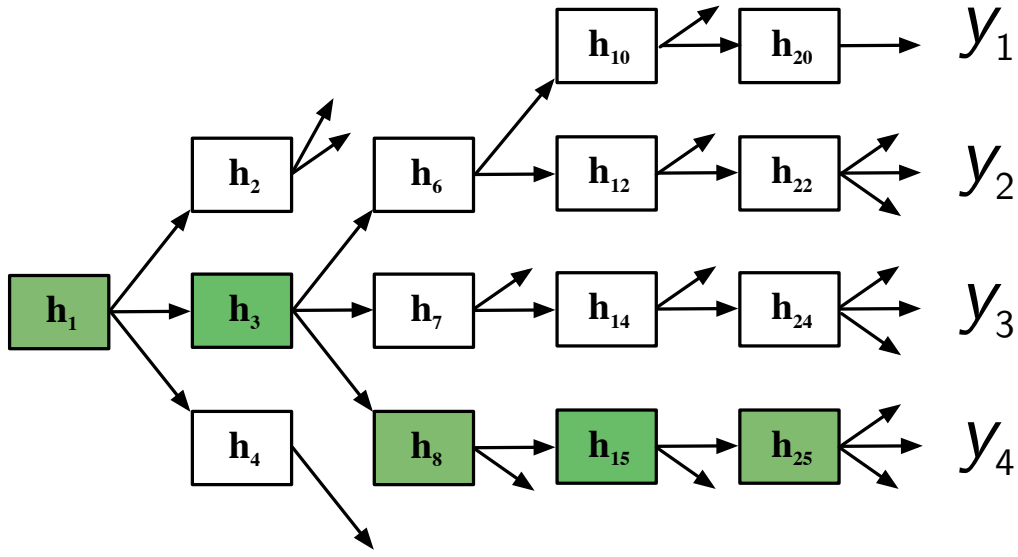
❷ Let $\theta = \theta_B$, and parse each sentence in the training data with beam search



$$\omega(y_i) = \sum_{j=1}^{|y_i|} \log s_\theta(y_{ij})$$

Expected F-Measure Training: Step 2

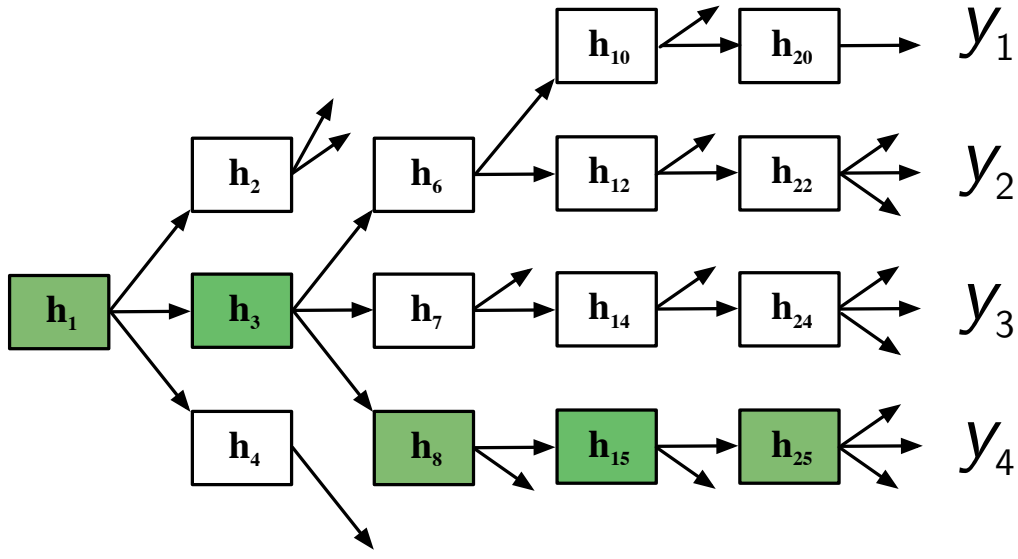
❷ Let $\theta = \theta_B$, and parse each sentence in the training data with beam search



$$\omega(y_i) = \sum_{j=1}^{|y_i|} \log s_\theta(y_{ij}), \qquad \text{F1}(\Delta_{y_i}, \Delta_{x_n}^G)$$

Expected F-Measure Training: Step 2

❷ Let $\theta = \theta_B$, and parse each sentence in the training data with beam search



$$\omega(y_i) = \sum_{j=1}^{|y_i|} \log s_\theta(y_{ij}), \qquad \text{F1}(\Delta_{y_i}, \Delta^G_{x_n})$$

Expected F-Measure Training: Step 3

❸ Compute the -xF1 loss for each sentence, do SGD update and iterate

$$J(\theta) = -\text{xF1}(\theta) = -\sum_{y_i \in \Lambda(x_n)} p(y_i|\theta)\text{F1}(\Delta_{y_i}, \Delta_{x_n}^{G}),$$

$$p(y_i|\theta) = \frac{\exp\{\omega(y_i)\}}{\sum_{y \in \Lambda(x_n)} \exp\{\omega(y)\}},$$

Expected F-Measure Training: Step 3

❸ Compute the -xF1 loss for each sentence, do SGD update and iterate

$$J(\theta) = -\text{xF1}(\theta) = -\sum_{y_i \in \Lambda(x_n)} p(y_i|\theta)\text{F1}(\Delta_{y_i}, \Delta^{G}_{x_n}),$$

$$p(y_i|\theta) = \frac{\exp\{\omega(y_i)\}}{\sum_{y \in \Lambda(x_n)} \exp\{\omega(y)\}},$$

Expected F-Measure Training: Step 3

❸ Compute the -xF1 loss for each sentence, do SGD update and iterate

$$J(\theta) = -\text{xF1}(\theta) = -\sum_{y_i \in \Lambda(x_n)} p(y_i|\theta)\text{F1}(\Delta_{y_i}, \Delta^{G}_{x_n}),$$

$$p(y_i|\theta) = \frac{\exp\{\omega(y_i)\}}{\sum_{y \in \Lambda(x_n)} \exp\{\omega(y)\}},$$

$$\frac{\partial J(\theta)}{\partial \theta} = -\sum_{y_i \in \Lambda(x_n)} \sum_{y_{ij} \in y_i} \frac{\partial J(\theta)}{\partial s_\theta(y_{ij})} \frac{\partial s_\theta(y_{ij})}{\partial \theta}$$

$$= -\sum_{y_i \in \Lambda(x_n)} \sum_{y_{ij} \in y_i} \delta_{y_{ij}} \frac{\partial s_\theta(y_{ij})}{\partial \theta}.$$

Expected F-Measure Training: Step 3

❸ Compute the -xF1 loss for each sentence, do SGD update and iterate

$$J(\theta) = -\text{xF1}(\theta) = -\sum_{y_i \in \Lambda(x_n)} p(y_i|\theta)\text{F1}(\Delta_{y_i}, \Delta^{G}_{x_n}),$$

$$p(y_i|\theta) = \frac{\exp\{\omega(y_i)\}}{\sum_{y \in \Lambda(x_n)} \exp\{\omega(y)\}},$$

$$\frac{\partial J(\theta)}{\partial \theta} = -\sum_{y_i \in \Lambda(x_n)} \sum_{y_{ij} \in y_i} \frac{\partial J(\theta)}{\partial s_\theta(y_{ij})} \frac{\partial s_\theta(y_{ij})}{\partial \theta}$$

$$= -\sum_{y_i \in \Lambda(x_n)} \sum_{y_{ij} \in y_i} \delta_{y_{ij}} \frac{\partial s_\theta(y_{ij})}{\partial \theta}.$$

Expected F-Measure Training

output	action sequence	$\omega(y_i)$	F1
y_1	$y_{11}\ y_{12} \ldots y_{1i}$	-0.60	0.67
y_2	$y_{21}\ y_{22} \ldots y_{2j}$	-1.5	0.81
y_3	$y_{31}\ y_{32} \ldots y_{3k}$	-4.96	0.90

Expected F-Measure Training

output	action sequence	$\omega(y_i)$	F1
y_1	$y_{11}\ y_{12} \ldots y_{1i}$	-0.60	0.67
y_2	$y_{21}\ y_{22} \ldots y_{2j}$	-1.5	0.81
y_3	$y_{31}\ y_{32} \ldots y_{3k}$	-4.96	0.90

$$J(\theta) = -\text{xF1}(\theta) = -\sum_{y_i \in \Lambda(x_n)} p(y_i|\theta)\text{F1}(\Delta_{y_i}, \Delta^G_{x_n}) = 71.00$$

Expected F-Measure Training

output	action sequence	$\omega(y_i)$	F1
y_1	$y_{11}\ y_{12} \ldots y_{1i}$	-0.60	0.67
y_2	$y_{21}\ y_{22} \ldots y_{2j}$	-1.5	0.81
y_3	$y_{31}\ y_{32} \ldots y_{3k}$	-4.96	0.90

$$J(\theta) = -\text{xF1}(\theta) = -\sum_{y_i \in \Lambda(x_n)} p(y_i|\theta)\text{F1}(\Delta_{y_i}, \Delta^G_{x_n}) = 71.00$$

output	action sequence	$\omega(y_i)$	F1
z_1	$z_{11}\ z_{12} \ldots z_{1i}$	-0.90	0.75
z_2	$z_{21}\ z_{22} \ldots z_{2j}$	-0.99	0.85
z_3	$z_{31}\ z_{32} \ldots z_{3k}$	-3.76	0.95

Expected F-Measure Training

output	action sequence	$\omega(y_i)$	F1
y_1	$y_{11}\ y_{12} \ldots y_{1i}$	-0.60	0.67
y_2	$y_{21}\ y_{22} \ldots y_{2j}$	-1.5	0.81
y_3	$y_{31}\ y_{32} \ldots y_{3k}$	-4.96	0.90

$$J(\theta) = -\text{xF1}(\theta) = -\sum_{y_i \in \Lambda(x_n)} p(y_i|\theta)\text{F1}(\Delta_{y_i}, \Delta^G_{x_n}) = 71.00$$

output	action sequence	$\omega(y_i)$	F1
z_1	$z_{11}\ z_{12} \ldots z_{1i}$	-0.90	0.75
z_2	$z_{21}\ z_{22} \ldots z_{2j}$	-0.99	0.85
z_3	$z_{31}\ z_{32} \ldots z_{3k}$	-3.76	0.95

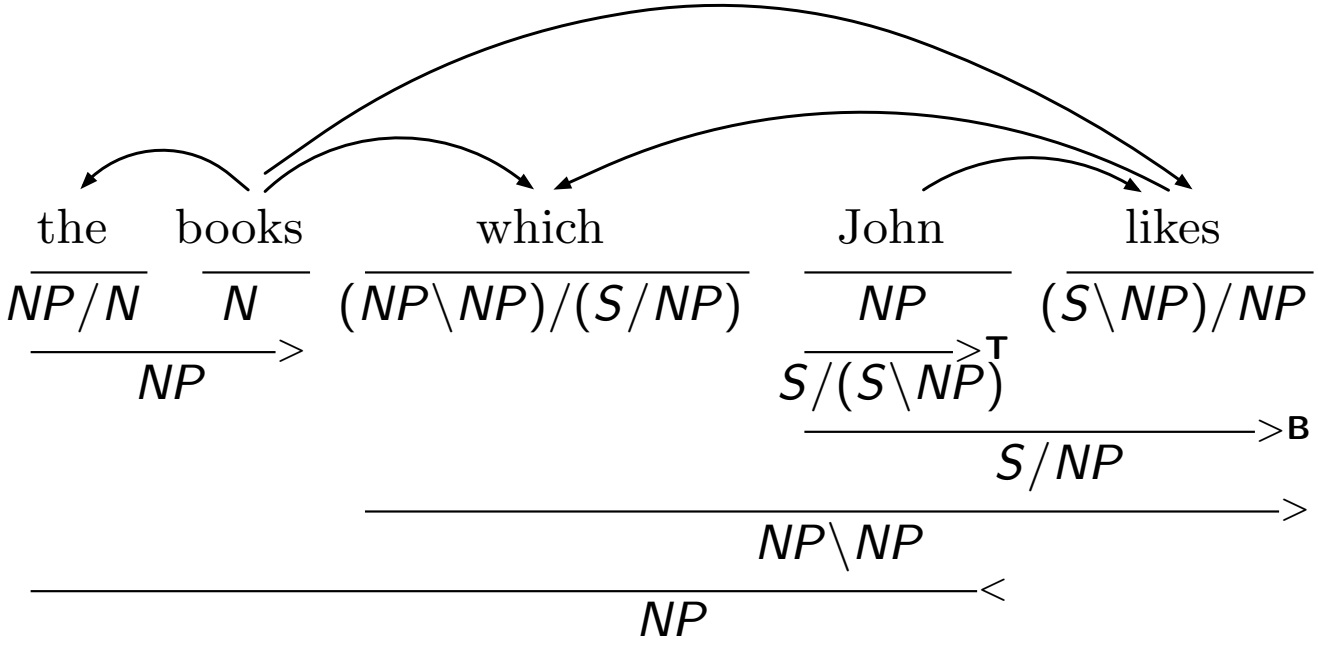
$$J(\theta) = -\text{xF1}(\theta) = -\sum_{z_i \in \Lambda(x_n)} p(z_i|\theta)\text{F1}(\Delta_{z_i}, \Delta^G_{x_n}) = 80.20$$

Experiments on CCGBank

Supertagger	Dev	Test
C&C	91.50	92.02
RNN	93.07	93.00
BRNN	**93.49**	**93.52**

- A bidirectional RNN supertagger extending the unidirectional one in Xu et al., 2015, using the same model and training parameters

Eval: F1 over Labeled, Directed CCG Deps



$\langle$*the*, NP/N_1, 1, *books*, $\rangle$

$\langle$*likes*, $(S\backslash NP_1)/NP_2$, 1, *John*$\rangle$

$\langle$*which*, $(NP/NP_1)/(S/NP)_2$, 2, *likes*$\rangle$

$\langle$*which*, $(NP/NP_1)/(S/NP)_2$, 1, *books*$\rangle$

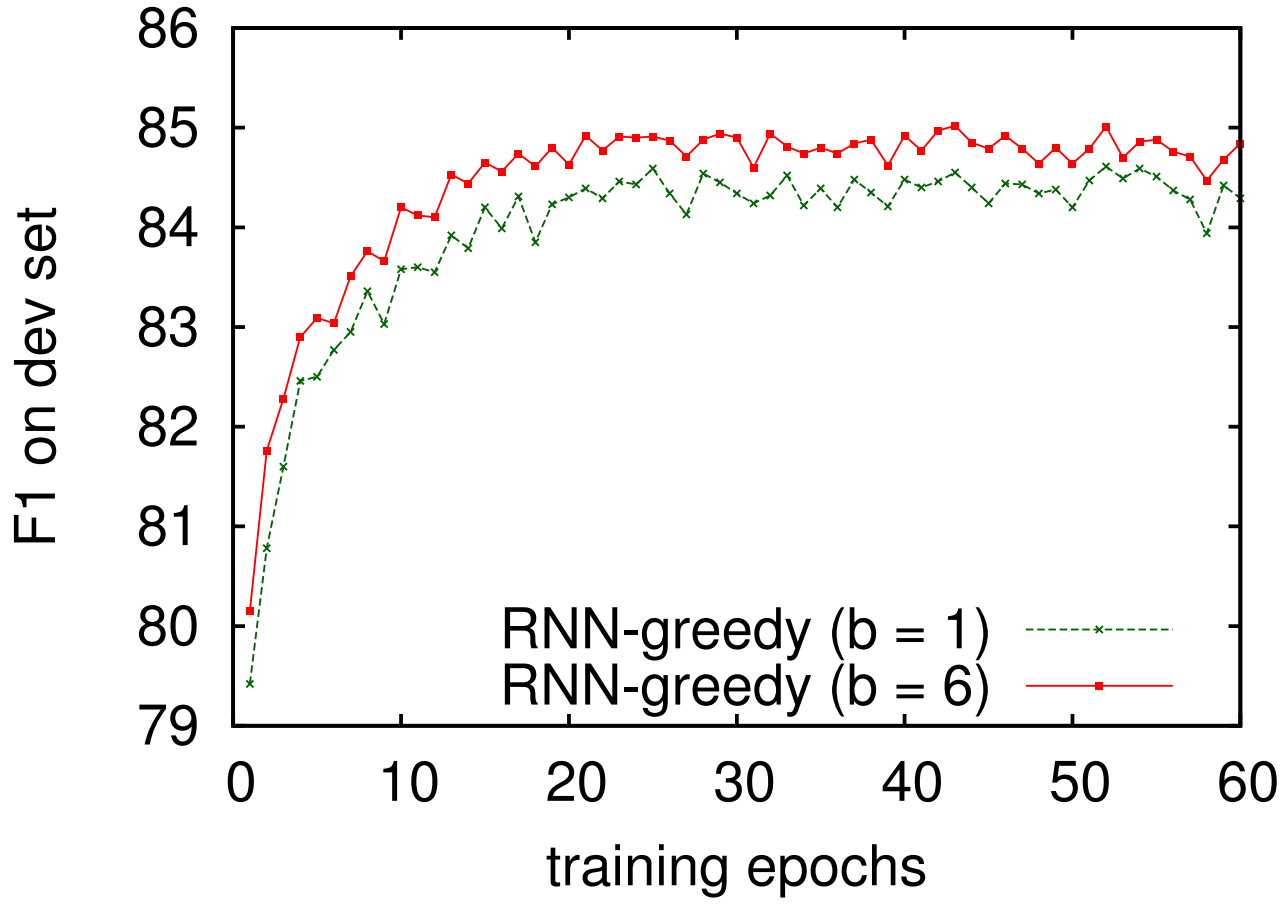
$\langle$*likes*, $(S\backslash NP_1)/NP_2$, 2, *books*$\rangle$

The Greedy Model and Beam Search (Dev)

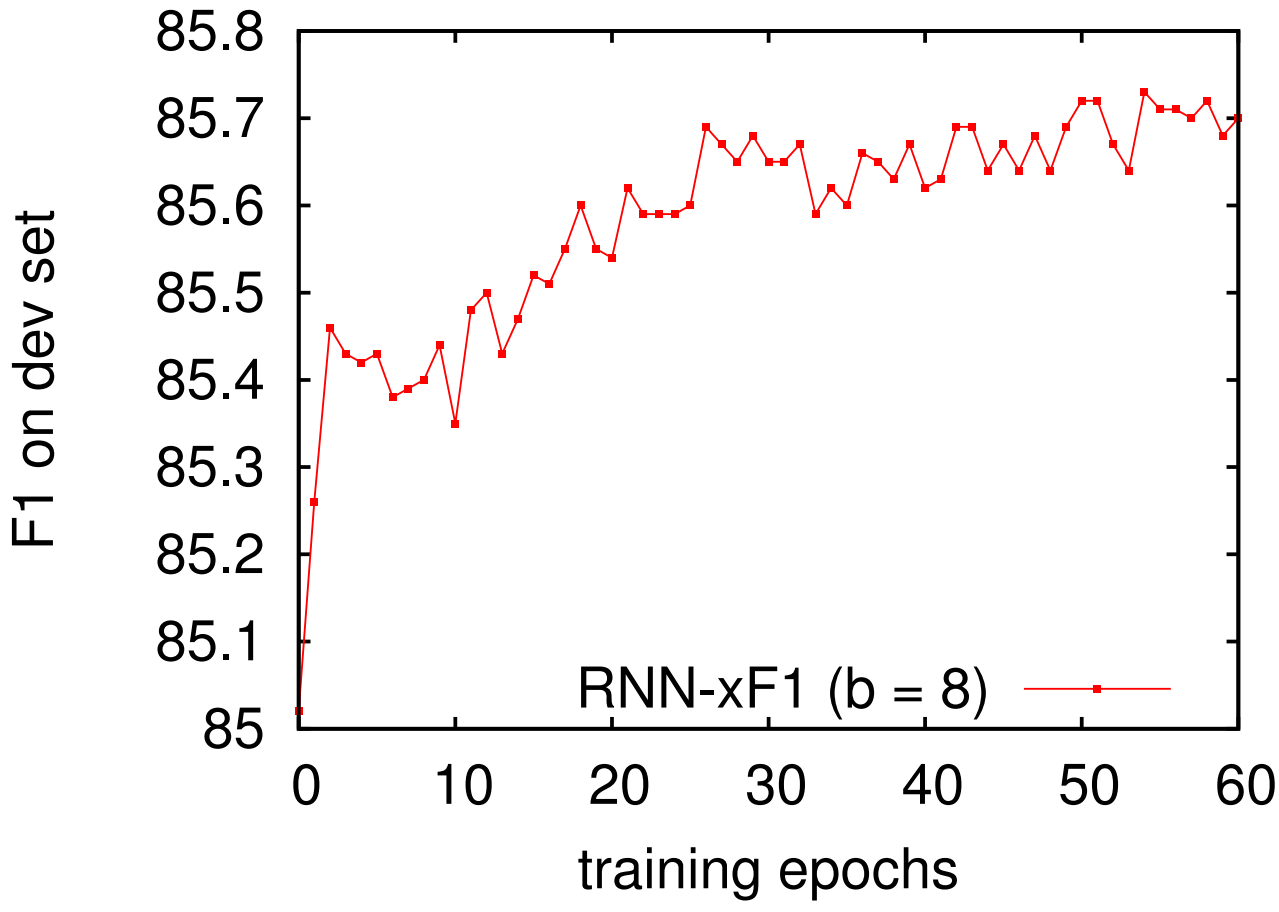
beam	F1
$b = 1$	84.61
$b = 2$	84.94
$b = 4$	85.01
$b = 6$	**85.02**
$b = 8$	**85.02**
$b = 16$	85.01

$b \in \{6, 8\}$ gives $+0.41\%$ F1 over $b = 1$

XEnt Model Dev F1 vs. Training Epochs



xF1 Model Dev F1 vs. Training Epochs



Test Set Parsing Results

Model	LP	LR	LF	CAT	Speed
C&C (normal)	85.45	83.97	84.70	92.83	97.90
C&C (hybrid)	86.24	84.17	85.19	93.00	95.25
Zhang11 ($b = 16$)	87.04	84.14	85.56	92.95	49.54
Xu14 ($b = 128$)	87.03	**85.08**	86.04	93.10	12.85
Am16 ($b = 1$)	-	-	83.27	91.89	350.00
Am16 ($b = 16$)	-	-	85.57	92.86	10.00
RNN-greedy ($b = 1$)	88.53	81.65	84.95	93.57	337.45
RNN-greedy ($b = 6$)	88.54	82.77	85.56	93.68	96.04
RNN-xF1 ($b = 8$)	**88.74**	84.22	**86.42**	**93.87**	67.65

- Zhang11 = Zhang and Clark, 2011*, Xu14 = Xu et al., 2014; AM16 = Ambati et al., 2016 (NN + Struct. Percep (Weiss et al., 2015))
- The xF1 model improves LR by 2.57% and LF by 1.47% over RNN-greedy ($b = 1$)

Test Set Parsing Results

Model	LP	LR	LF	CAT	Speed
C&C (normal)	85.45	83.97	84.70	92.83	97.90
C&C (hybrid)	86.24	84.17	85.19	93.00	95.25
Zhang11 ($b = 16$)	87.04	84.14	85.56	92.95	49.54
Xu14 ($b = 128$)	87.03	**85.08**	86.04	93.10	12.85
Am16 ($b = 1$)	-	-	83.27	91.89	350.00
Am16 ($b = 16$)	-	-	85.57	92.86	10.00
RNN-greedy ($b = 1$)	88.53	81.65	84.95	93.57	337.45
RNN-greedy ($b = 6$)	88.54	82.77	85.56	93.68	96.04
RNN-xF1 ($b = 8$)	**88.74**	84.22	**86.42**	**93.87**	67.65

- Zhang11 = Zhang and Clark, 2011*, Xu14 = Xu et al., 2014; AM16 = Ambati et al., 2016 (NN + Struct. Percep (Weiss et al., 2015))
- The xF1 model improves LR by 2.57% and LF by 1.47% over RNN-greedy ($b = 1$)
- Auli and Lopez, 2011 uses a softmax-margin objective (Gimpel and Smith, 2010) on the C&C parser

The End: Questions?

Model	LP	LR	LF	CAT	Speed
C&C (normal)	85.45	83.97	84.70	92.83	97.90
C&C (hybrid)	86.24	84.17	85.19	93.00	95.25
Zhang11 ($b = 16$)	87.04	84.14	85.56	92.95	49.54
Xu14 ($b = 128$)	87.03	**85.08**	86.04	93.10	12.85
Am16 ($b = 1$)	-	-	83.27	91.89	350.00
Am16 ($b = 16$)	-	-	85.57	92.86	10.00
RNN-greedy ($b = 1$)	88.53	81.65	84.95	93.57	337.45
RNN-greedy ($b = 6$)	88.54	82.77	85.56	93.68	96.04
RNN-xF1 ($b = 8$)	**88.74**	84.22	**86.42**	**93.87**	67.65

- Zhang11 = Zhang and Clark, 2011*, Xu14 = Xu et al., 2014, AM16 = Ambati et al., 2016 (NN + Struct. Percep (Weiss et al., 2015))
- The xF1 model improves LR by 2.57% and LF by 1.47% over RNN-greedy ($b = 1$)
- Auli and Lopez, 2011 uses a softmax-margin objective (Gimpel and Smith, 2010) on the C&C parser