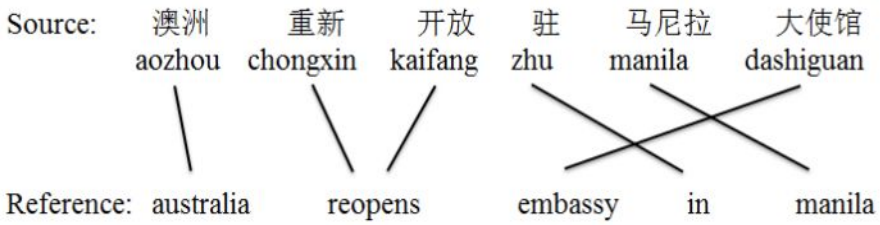
Improved Neural Machine Translation with a Syntax-Aware Encoder and Decoder (2017)

Huadong Chen, Shujian Huang, David Chiang, Jiajun Chen

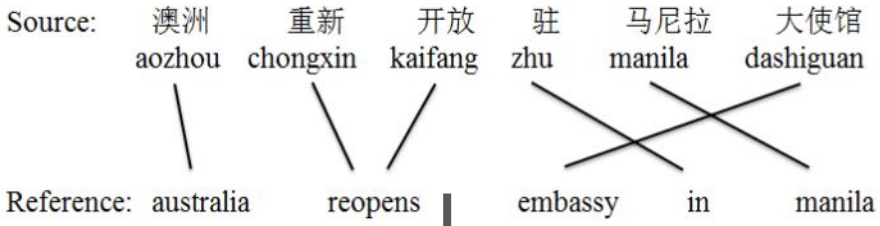
Presenter: Ziqiao Ding

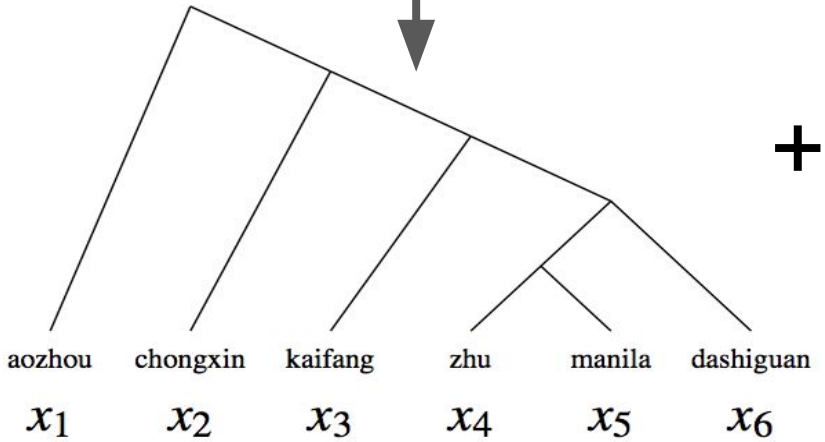
Motivation [Shi et al. (2016)]



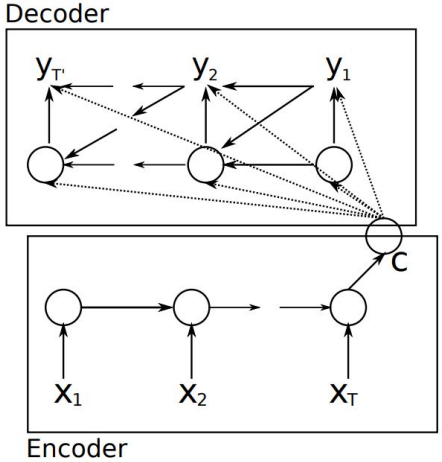
The current encoder-decoder model relies solely on the implicit structure it learns during training and can suffer from syntactic errors.

Proposal



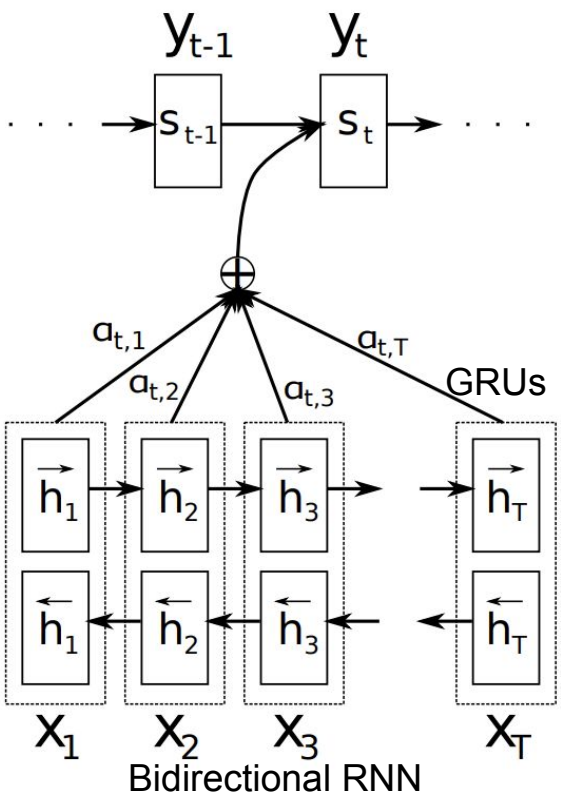


+



Incorporate source-side syntactic (binarized) tree with both encoder and decoder

What we have (Before Spring Break) [Bahdanau et al. (2015)]



A Forward GRU **Decoder** (With Attention)

$$P(y_j \mid \mathbf{y}_{<\mathbf{j}}, \mathbf{x}; \theta) = \mathrm{softmax}(t_{j-1}, d_j, c_j) \qquad \alpha_{j,i} = \frac{\exp(e_{j,i})}{\sum_{i'=1}^{I} \exp(e_{j,i'})}$$

$$d_j = \mathrm{GRU}(d_{j-1}, t_{j-1}, c_j)$$

$$c_j = \sum_{i=1}^{I} \alpha_{j,i} \overleftrightarrow{h_i} \qquad e_{j,i} = v_a^{\mathrm{T}} \tanh(W_a d_{j-1} + U_a \overleftrightarrow{h_i})$$

A Bidirectional Sequential **Encoder**

$$\overrightarrow{h_i} = \mathrm{GRU}(\overrightarrow{h_{i-1}}, s_i) \qquad \overleftrightarrow{h_i} = \begin{bmatrix} \overrightarrow{h_i} \\ \overleftarrow{h_i} \end{bmatrix}$$

$$\overleftarrow{h_i} = \mathrm{GRU}(\overleftarrow{h_{i-1}}, s_i)$$

GRU (quick reminder)

2 Main parts!

Reset Gate (How much info to forget):

$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r)$$

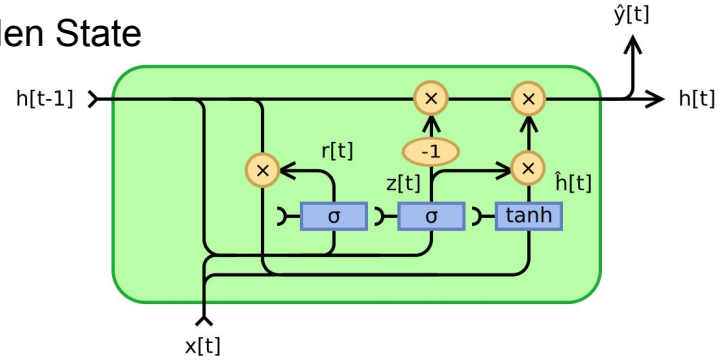
Update Gate (How much info to pass on):

$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z)$$

Updated Hidden State:

$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \sigma_h(W_h x_t + U_h(r_t \circ h_{t-1}) + b_h)$$

Hidden State



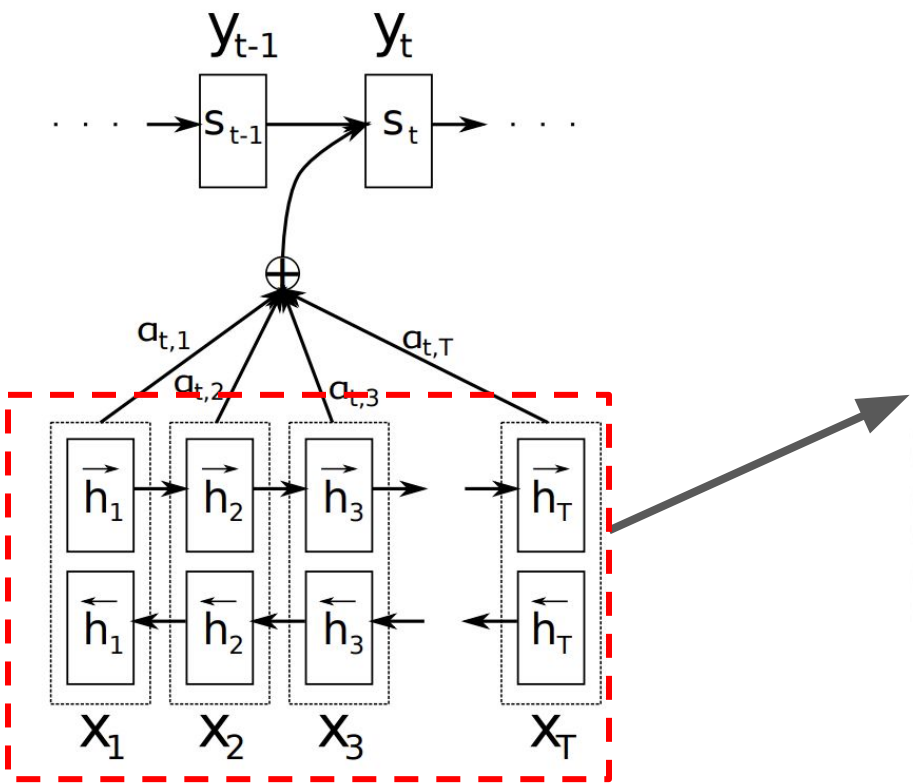
(Auxiliary) Inputs

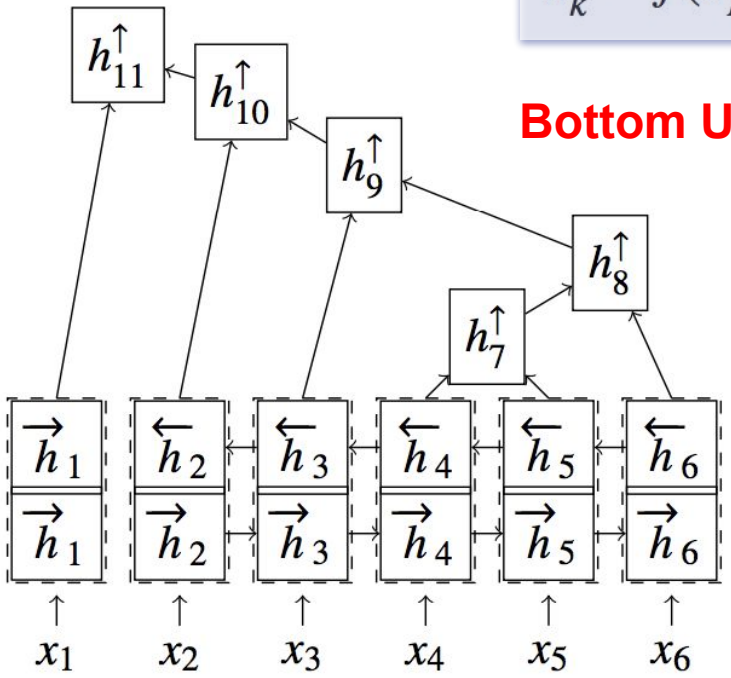
Tree-GRU Encoder [Eriguchi et al. (2016)]

Leaf Node: $h_k^{\uparrow} = \overleftrightarrow{h_k}$

Internal Node: $h_k^{\uparrow} = f(h_{L(k)}^{\uparrow}, h_{R(k)}^{\uparrow})$

Bottom Up Tree!





(a) Tree-GRU Encoder

Tree-GRU Encoder [Eriguchi et al. (2016)]

$$h_k^{\uparrow} = f(h_{L(k)}^{\uparrow}, h_{R(k)}^{\uparrow})$$

Children Nodes:

$$r_L = \sigma(U_L^{(rL)} h_{L(k)}^{\uparrow} + U_R^{(rL)} h_{R(k)}^{\uparrow} + b^{(rL)})$$

$$r_R = \sigma(U_L^{(rR)} h_{L(k)}^{\uparrow} + U_R^{(rR)} h_{R(k)}^{\uparrow} + b^{(rR)})$$

$$z_L = \sigma(U_L^{(zL)} h_{L(k)}^{\uparrow} + U_R^{(zL)} h_{R(k)}^{\uparrow} + b^{(zL)})$$

$$z_R = \sigma(U_L^{(zR)} h_{L(k)}^{\uparrow} + U_R^{(zR)} h_{R(k)}^{\uparrow} + b^{(zR)})$$

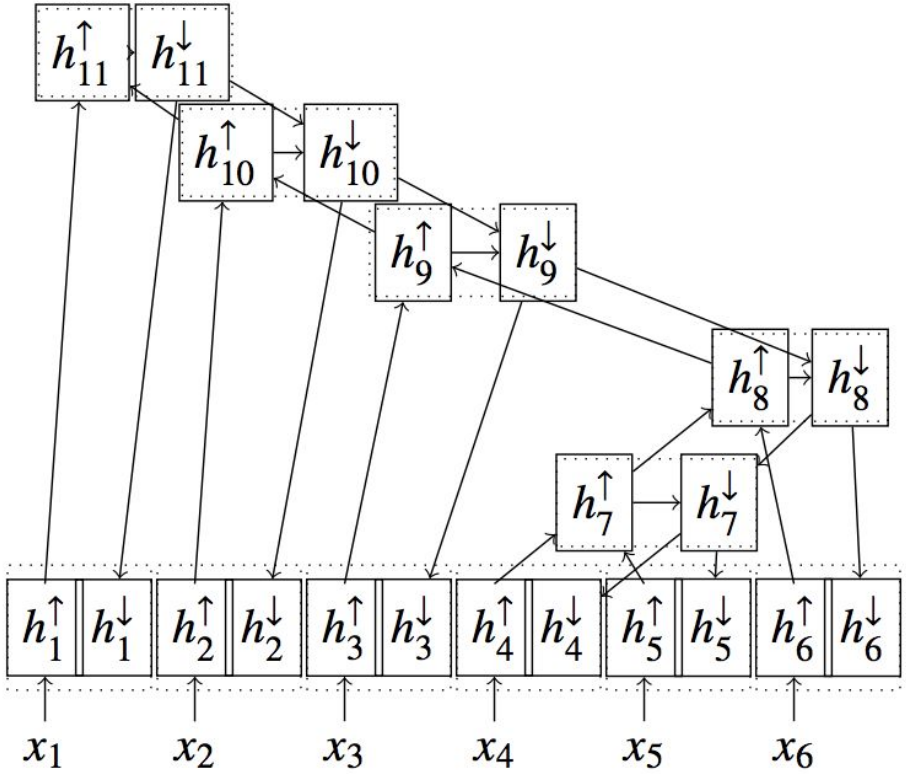
Current Node:

$$z = \sigma(U_L^{(z)} h_{L(k)}^{\uparrow} + U_R^{(z)} h_{R(k)}^{\uparrow} + b^{(z)})$$

$$\tilde{h}_k^{\uparrow} = \tanh\left(U_L(r_L \odot h_{L(k)}^{\uparrow}) + U_R(r_R \odot h_{R(k)}^{\uparrow})\right)$$

$$h_k^{\uparrow} = z_L \odot h_{L(k)}^{\uparrow} + z_R \odot h_{R(k)}^{\uparrow} + z \odot \tilde{h}_k^{\uparrow}$$

Bidirectional Tree-GRU Encoder (This Paper)



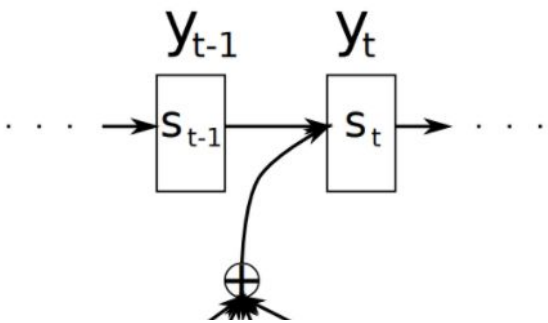
Root: $$h_{\rho}^{\downarrow} = \tanh\left(W h_{\rho}^{\uparrow} + b\right)$$

Non-root: $$h_{k}^{\downarrow} = \mathrm{GRU}(h_{p(k)}^{\downarrow}, h_{k}^{\uparrow})$$

Add Top Down Layer!

$$h_{k}^{\updownarrow} = \begin{bmatrix} h_{k}^{\uparrow} \\ h_{k}^{\downarrow} \end{bmatrix}$$

Decoder (Coverage Model) [Tu et al. (2016)]



A Forward GRU **Decoder** (With Attention)

$$P(y_j \mid \mathbf{y}_{<\mathbf{j}}, \mathbf{x}; \theta) = \text{softmax}(t_{j-1}, d_j, c_j)$$

$$d_j = \text{GRU}(d_{j-1}, t_{j-1}, c_j)$$

$$c_j = \sum_{i=1}^{I} \alpha_{j,i} \overleftrightarrow{h_i}$$

$$\alpha_{j,i} = \frac{\exp(e_{j,i})}{\sum_{i'=1}^{I} \exp(e_{j,i'})}$$

$$e_{j,i} = v_a^{\mathrm{T}} \tanh(W_a d_{j-1} + U_a \overleftrightarrow{h_i})$$

+

Coverage Vector: $$C_{j,i} = \text{GRU}(C_{j-1,i}, \alpha_{j,i}, d_{j-1}, h_i)$$

$$e_{j,i} = v_a^{\mathrm{T}} \tanh(W_a d_{j-1} + U_a h_i + V_a C_{j-1,i})$$

Coverage? What?

More intuitive in SMT:
(Which word is translated already?)

$$\mathbf{x} = \{x_1, x_2, x_3, x_4\}$$

$$\mathcal{C} = \{0,0,0,0\} \rightarrow \mathcal{C} = \{0,1,1,0\} \rightarrow \mathcal{C} = \{1,1,1,1\}$$

Initial

Goal!

But harder to model single word in NN, so we model the attention:

$$C_{j,i} = \mathrm{GRU}(C_{j-1,i}, \alpha_{j,i}, d_{j-1}, h_i)$$

Hidden State! Attention Weight, decoder state, encoder state

Attention adjusted according to coverage:

$$e_{j,i} = v_a^{\mathrm{T}} \tanh\left(W_a d_{j-1} + U_a h_i + V_a C_{j-1,i}\right)$$

Tree Coverage Decoder (This paper)

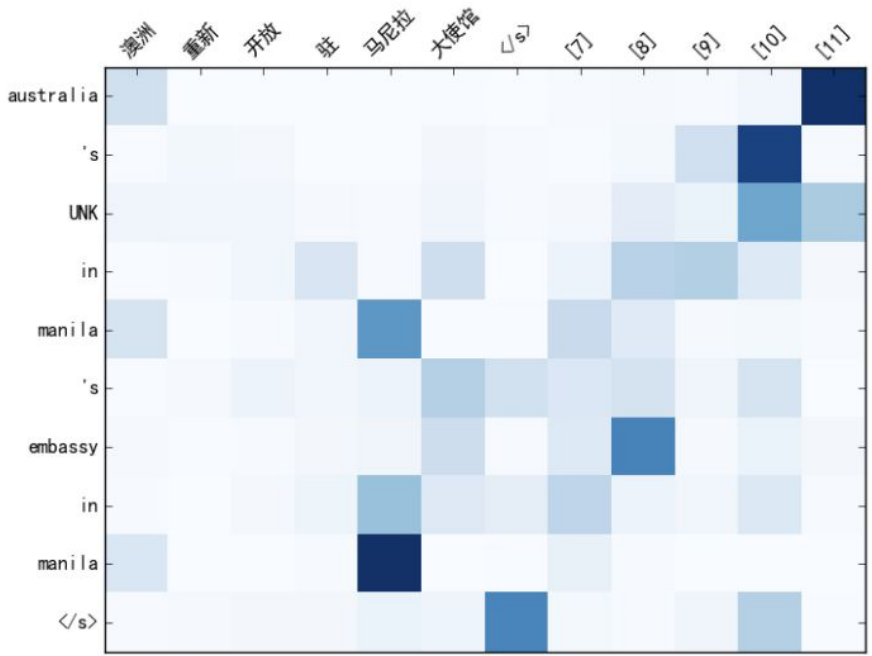
$$C_{j,i} = \mathrm{GRU}(C_{j-1,i}, \alpha_{j,i}, d_{j-1}, h_i)$$

Tree Coverage Vector:

$$C_{j,i} = \mathrm{GRU}(C_{j-1,i}, \alpha_{j,i}, d_{j-1}, h_i, C_{j-1,L(i)}, \alpha_{j,L(i)}, C_{j-1,R(i)}, \alpha_{j,R(i)}).$$

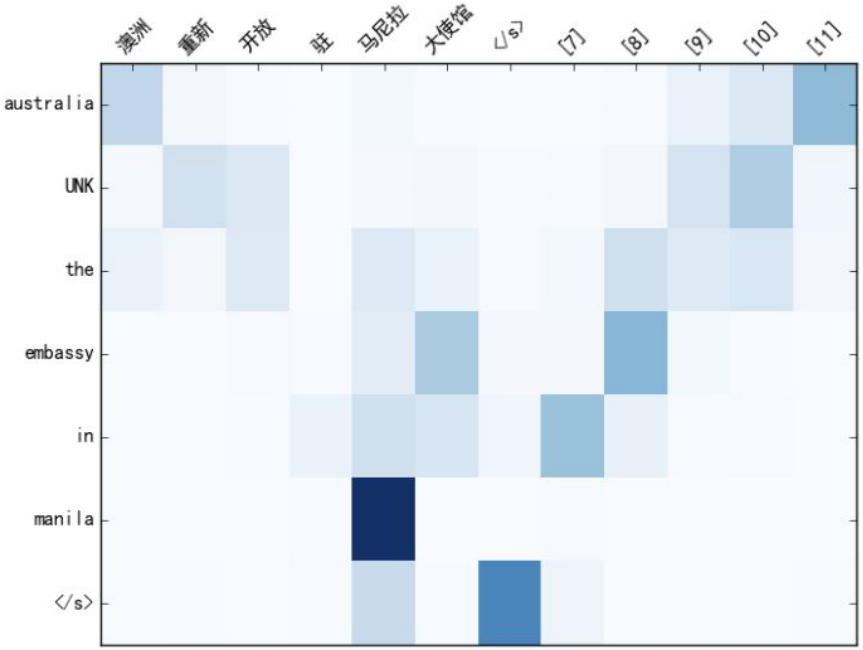
$$e_{j,i} = v_a^{\mathrm{T}} \tanh\left(W_a d_{j-1} + U_a h_i + V_a C_{j-1,i}\right)$$

And there’s a point in using Coverage Model? Yes...



(a) Tree-GRU Encoder

Reduced Concentrated Area



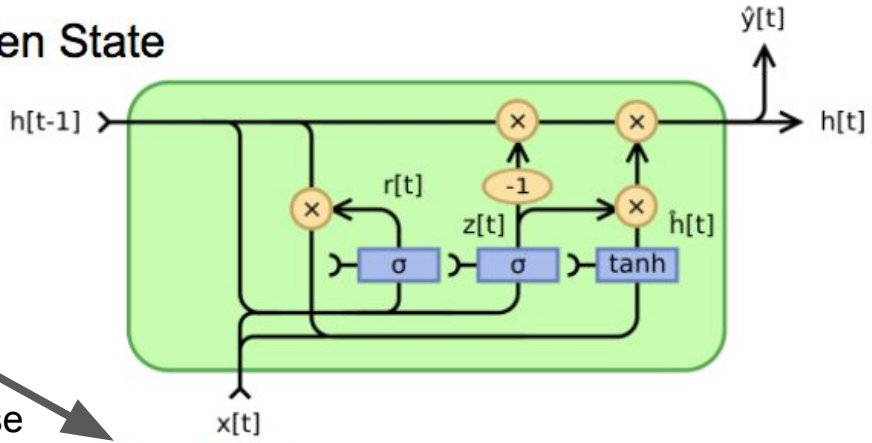
(b) + Tree-Coverage Model

Note: Extending GRU parameters

Coverage for h_i at time j-1

$$C_{j,i} = \mathrm{GRU}(C_{j-1,i}, \alpha_{j,i}, d_{j-1}, h_i, C_{j-1,L(i)}, \alpha_{j,L(i)}, C_{j-1,R(i)}, \alpha_{j,R(i)}).$$

Hidden State



Concatenate everything else

(Auxiliary) Inputs

Performance (Encoder Only)

#	Encoder	Coverage	MT02	MT03	MT04	MT05	MT06	Average
1	Sequential	no	33.76	31.88	33.15	30.55	27.47	30.76
2	Tree-LSTM	no	33.83	33.15	33.81	31.22	27.86	31.51(+0.75)
3	Tree-GRU	no	35.39	33.62	35.1	32.55	28.26	32.38(+1.62)
4	Bidirectional	no	35.52	33.91	35.51	33.34	29.91	33.17(+2.41)

BLEU scores against:

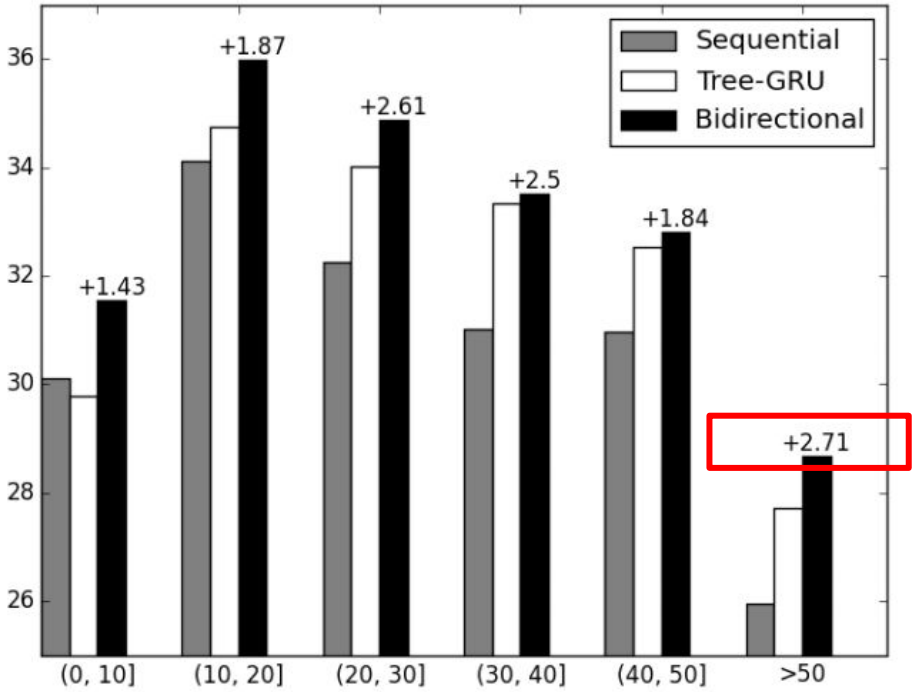
- **NMT**: the standard attentional NMT model (Bahdanau et al., 2015).
- **Tree-LSTM**: the attentional NMT model extended with the Tree-LSTM encoder (Eriguchi et al., 2016).

Performance (w. Word / Tree Coverage)

#	Encoder	Coverage	MT02	MT03	MT04	MT05	MT06	Average
5	Sequential	word	34.21	32.73	34.17	31.64	28.29	31.71(+0.95)
6	Tree-LSTM	word	35.81	33.62	34.84	32.6	28.52	32.40(+1.64)
7	Tree-GRU	word	35.91	33.71	35.46	33.02	29.14	32.84(+2.08)
8	Bidirectional	word	36.14	35.00	36.07	33.74	30.40	33.80(+3.04)
9	Tree-LSTM	tree	34.97	33.91	35.21	33.08	29.38	32.90(+2.14)
10	Tree-GRU	tree	35.67	34.25	35.72	33.47	29.95	33.35(+2.59)
11	Bidirectional	tree	**36.57**	**35.64**	**36.63**	**34.35**	**30.57**	**34.30(+3.54)**

- **Coverage**: the attentional NMT model extended with word coverage (Tu et al., 2016).

Performance against sentence length



Potential Further Improvements

- More training complexity analysis

- Make use of node labels from the tree

- Using syntactic information on the target side