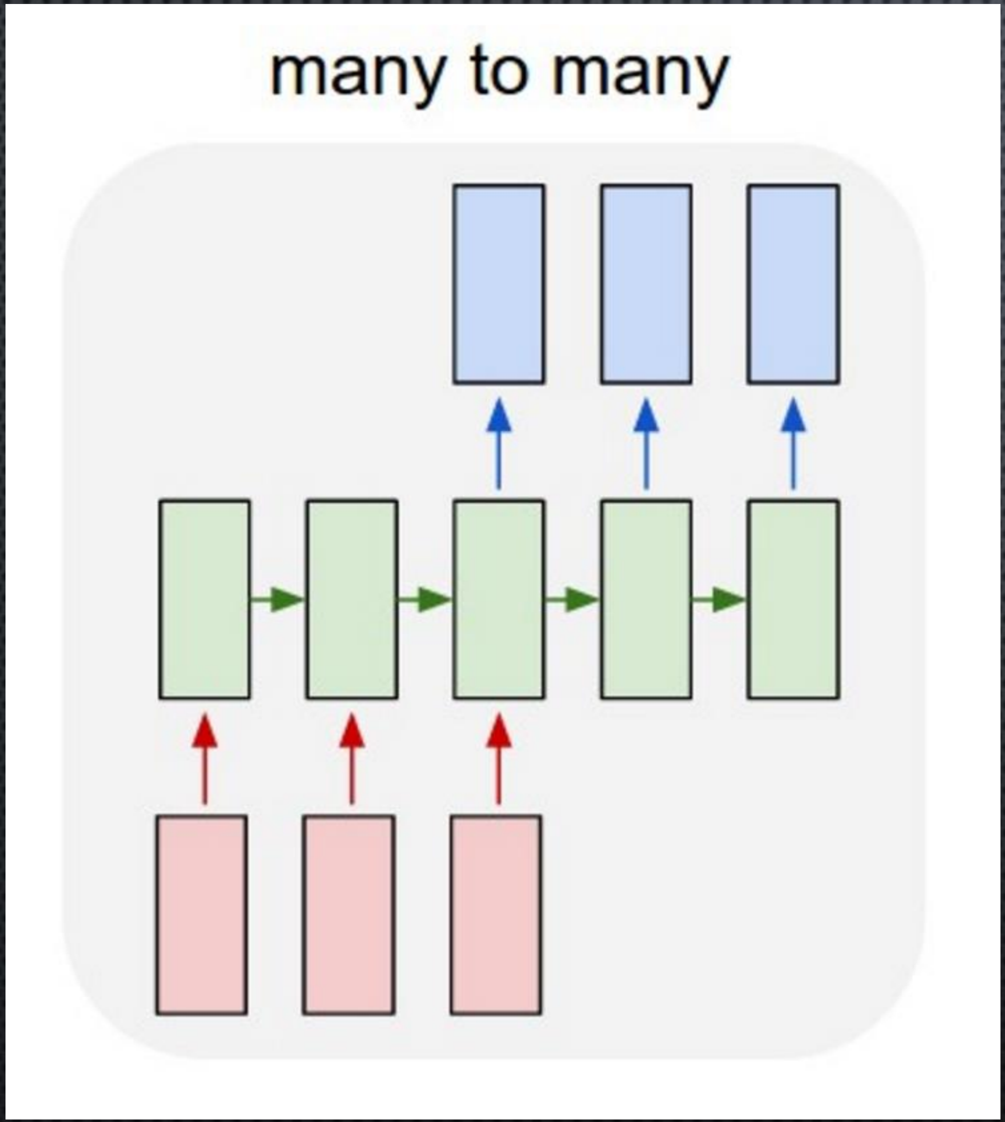
TEXT SUMMARIZATION

고려대학교 데이터마이닝 시각화 연구실
강경필



Text summarization ~ Machine Translation

Sequence-to-Sequence
with Attention Model for Text Summarization
(2016, June)

Effective Approaches to Attention-based
Neural Machine Translation
(2015)

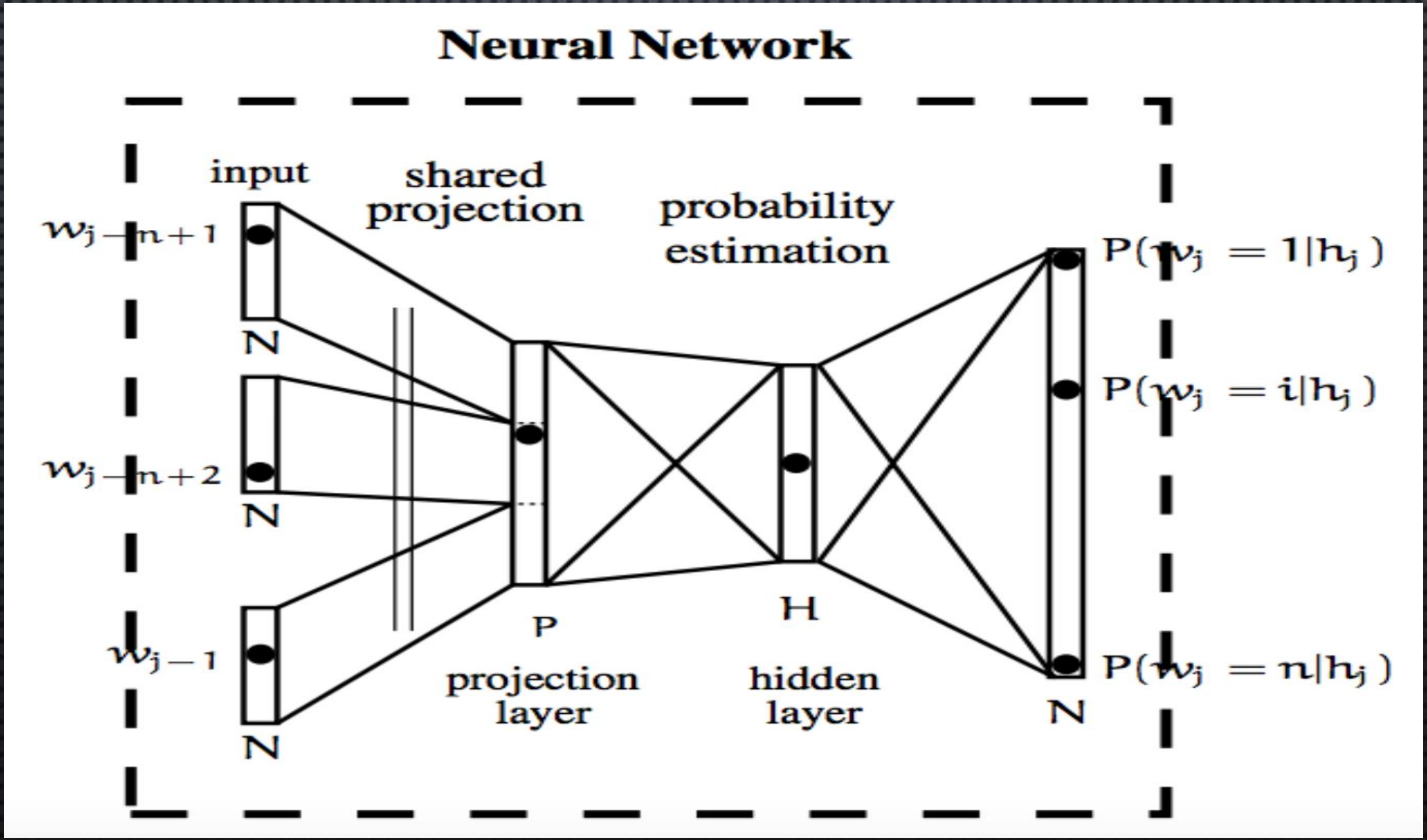
Sequence to Sequence Learning
with Neural Networks
(2014)

Neural Machine Translation
by Jointly Learning to Align and Translate
(2014)

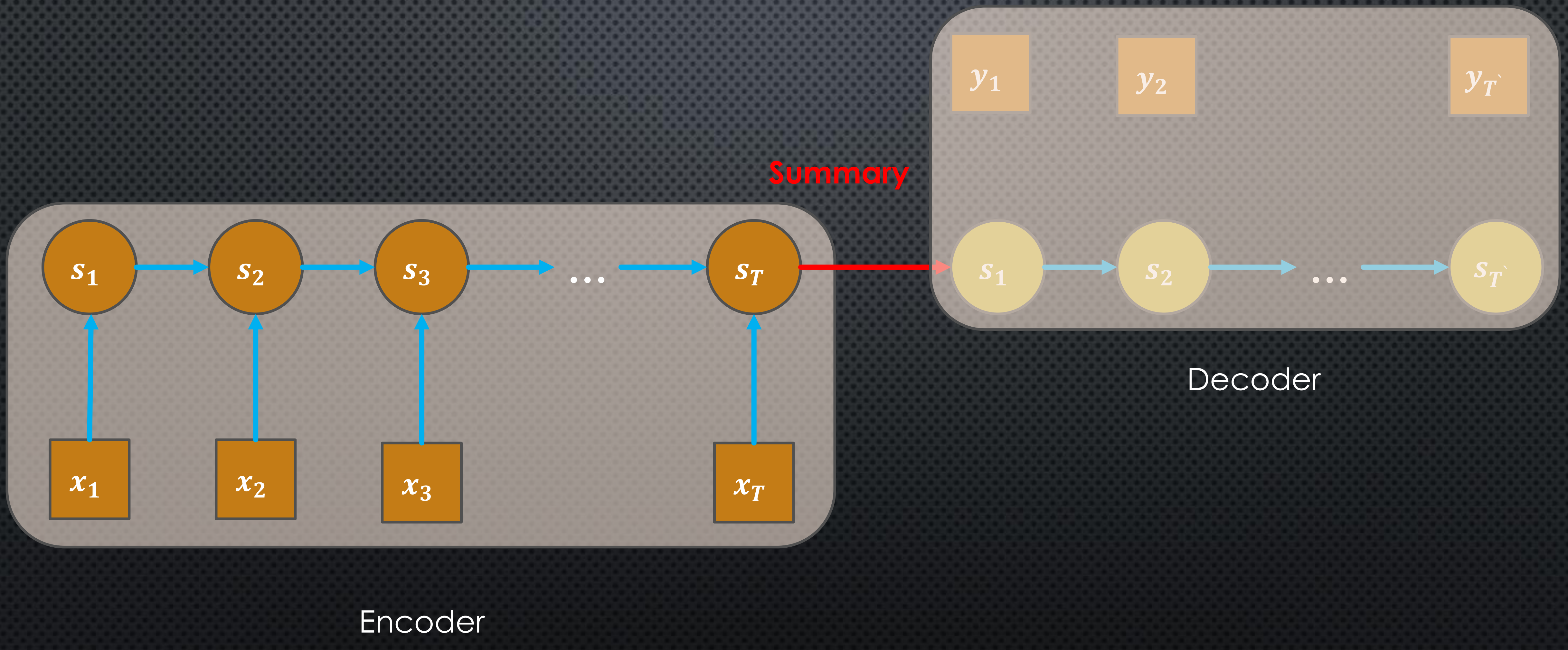
Learning Phrase Representations using
RNN Encoder–Decoder for Statistical Machine Translation
(2014)

Continuous-Space Language Models
for Statistical Machine Translation
(2014)

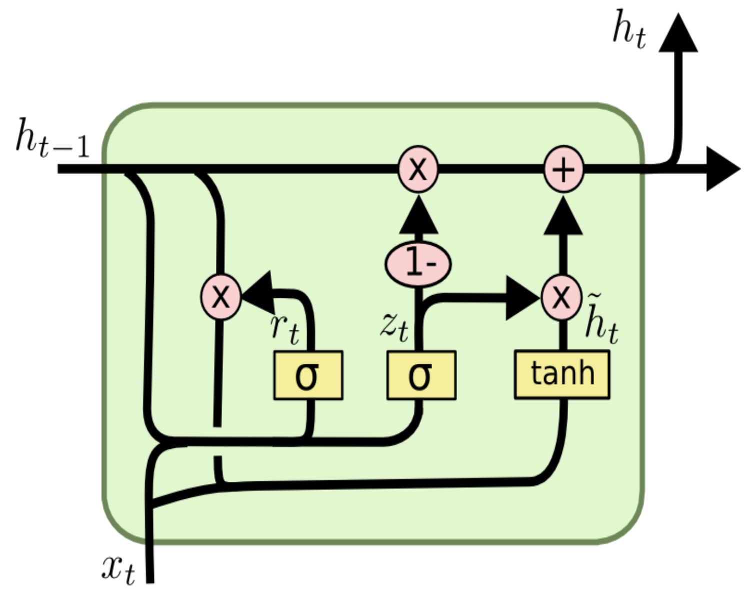
Continuous-Space Language Models (CSLM) for Statistical Machine Translation (2014)



Learning Phrase Representations using
RNN Encoder–Decoder for Statistical Machine Translation (2014)



Learning Phrase Representations using
RNN Encoder–Decoder for Statistical Machine Translation (2014)



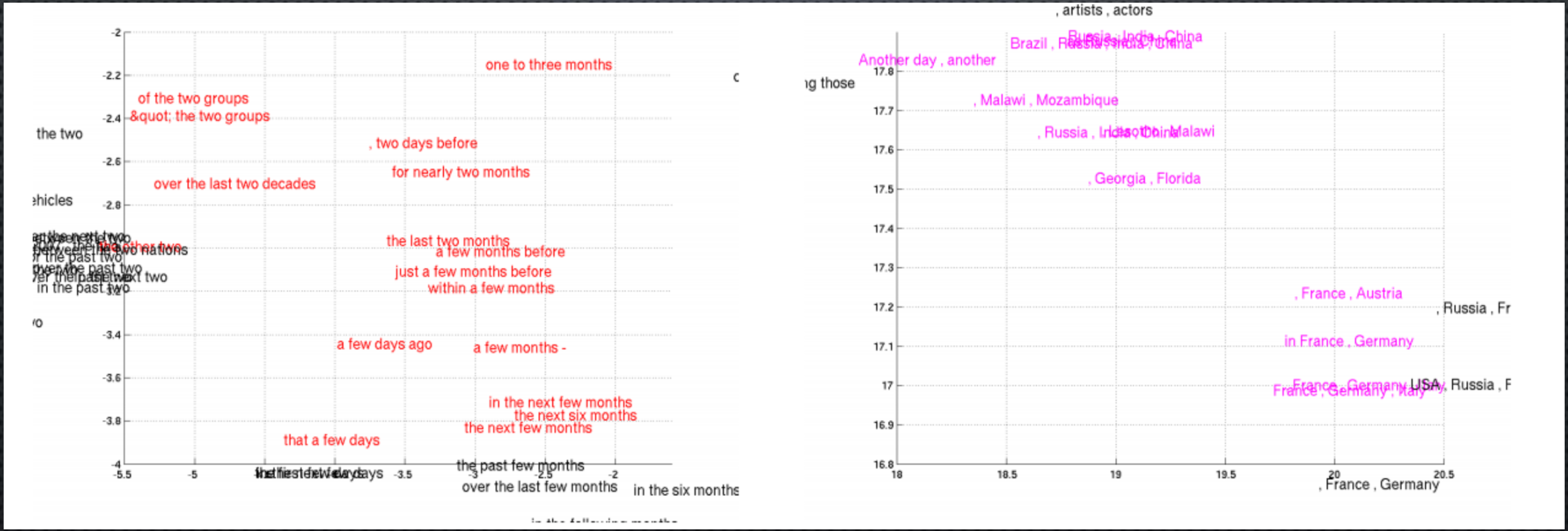
$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$ Update gate

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$ Reset gate

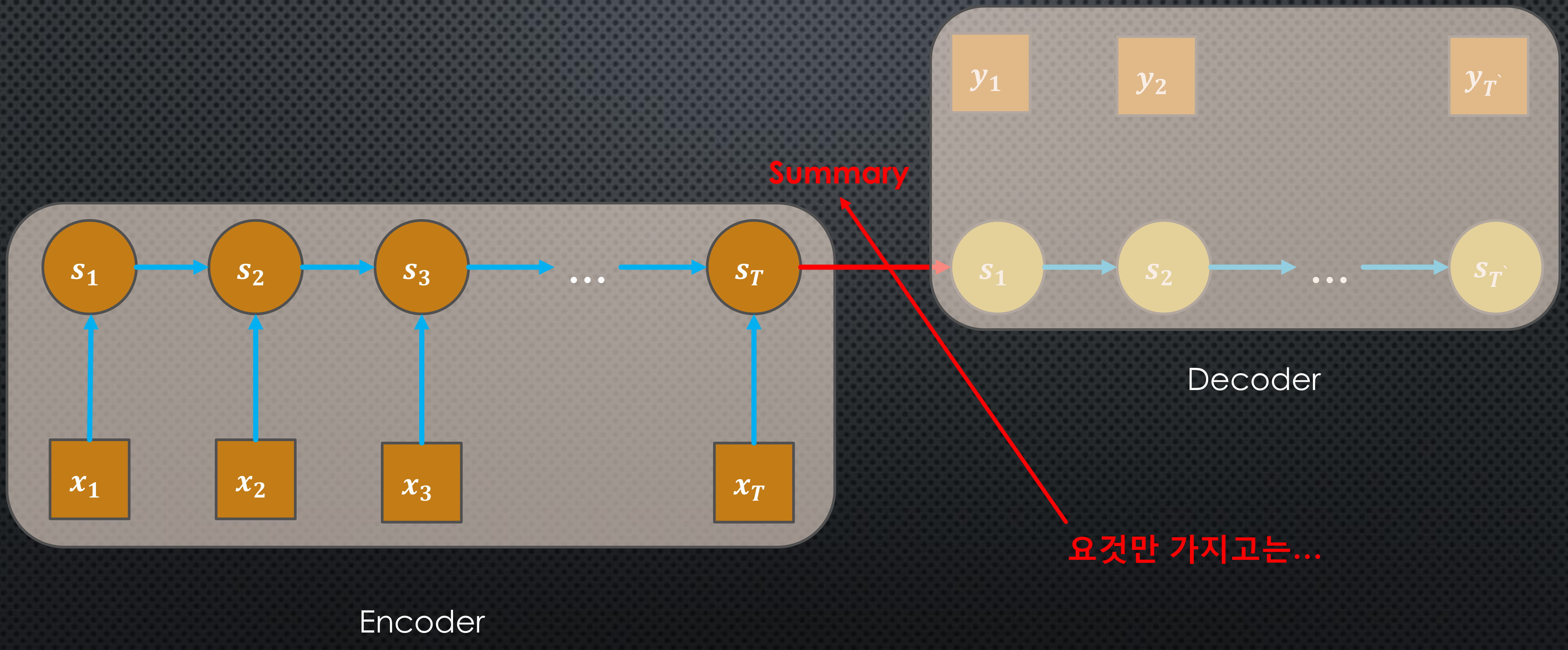
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$ New hidden state

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$ Final hidden state

Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation (2014)



Neural Machine Translation
by Jointly Learning to Align and Translate(2014)



Neural Machine Translation
by Jointly Learning to Align and Translate(2014)

Encoder

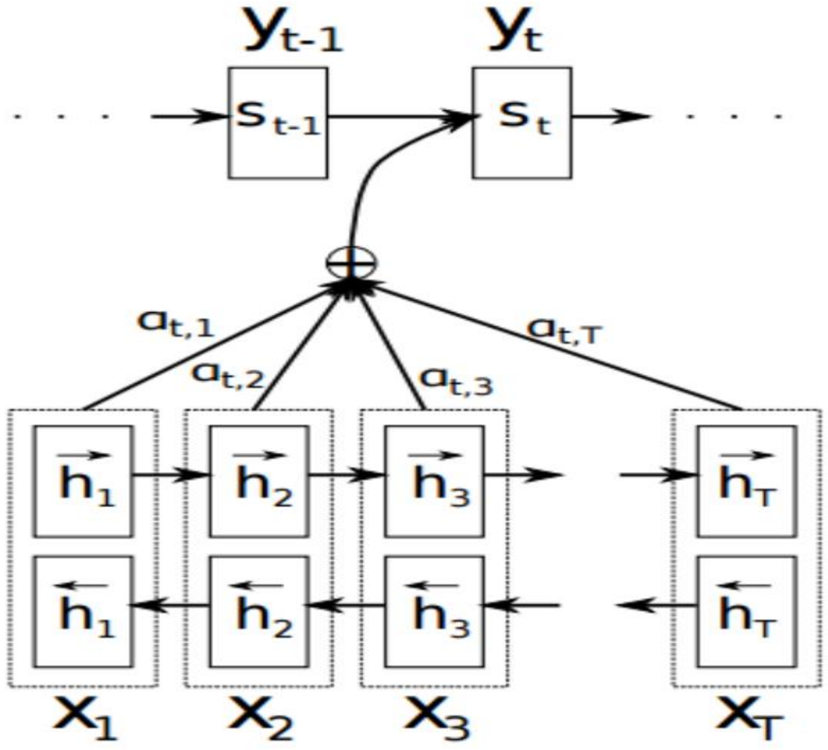


Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

Bidirectional RNN
(GRU)

$$h_j = \left[\overrightarrow{h}_j^\top ; \overleftarrow{h}_j^\top\right]^\top$$

$$\begin{bmatrix} \overrightarrow{h}_j^T \\ \overleftarrow{h}_j^T \end{bmatrix} \quad \mathbb{R}^{2n\times 1}$$

Neural Machine Translation by Jointly Learning to Align and Translate(2014)

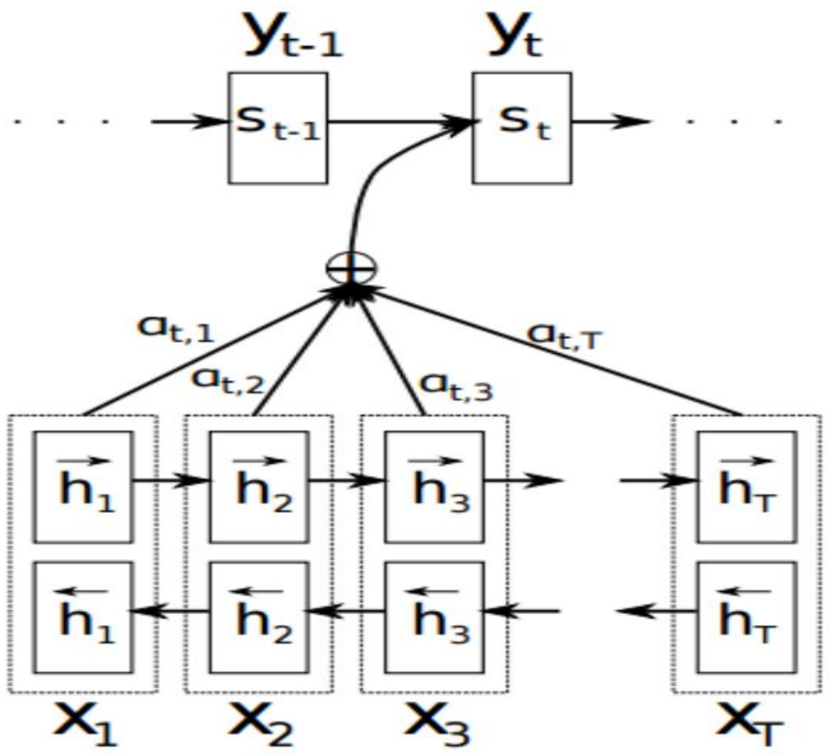


Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

$$\overrightarrow{h}_i = \begin{cases} (1-\overrightarrow{z}_i)\circ \overrightarrow{h}_{i-1} + \overrightarrow{z}_i \circ \underline{\overrightarrow{h}}_i & ,\text{if } i>0 \\ 0 & ,\text{if } i=0 \end{cases}$$

$$\underline{\overrightarrow{h}}_i = \tanh\left(\overrightarrow{W}\overline{E}x_i + \overrightarrow{U}\left[\overrightarrow{r}_i \circ \overrightarrow{h}_{i-1}\right]\right)$$

$$\overrightarrow{z}_i = \sigma\left(\overrightarrow{W}_z\overline{E}x_i + \overrightarrow{U}_z\overrightarrow{h}_{i-1}\right)$$

$$\overrightarrow{r}_i = \sigma\left(\overrightarrow{W}_r\overline{E}x_i + \overrightarrow{U}_r\overrightarrow{h}_{i-1}\right).$$

$\overleftarrow{h}_j^T$ 도 마찬가지로 계산,
$\bar{E}\ matrix$는 embedding matrix

Neural Machine Translation
by Jointly Learning to Align and Translate(2014)

Decoder

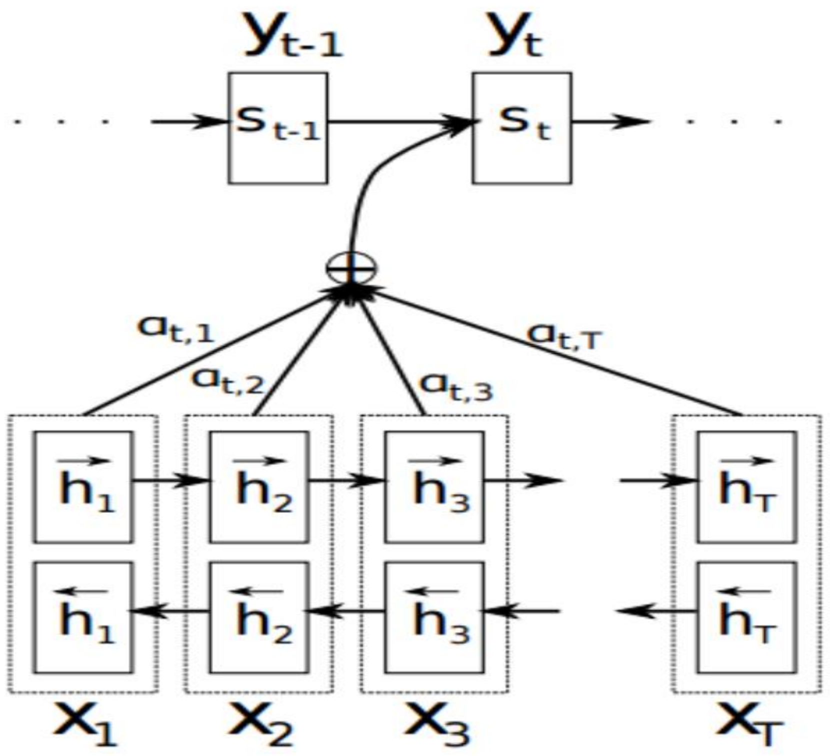


Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}$$

$$e_{ij} = v_a^\top \tanh(W_a s_{i-1} + U_a h_j),$$

and h_j is the j-th annotation in the source sentence (see Eq. (7)). $v_a \in \mathbb{R}^{n'}, W_a \in \mathbb{R}^{n' \times n}$ and $U_a \in \mathbb{R}^{n' \times 2n}$ are weight matrices. Note that the model becomes RNN Encoder–Decoder (Cho *et al.*, 2014a), if we fix c_i to $\overrightarrow{h}_{T_x}$.

i번째 단어를 생성할 때
source sentence의 j번째 단어를
얼마만큼 attention 할지!

Neural Machine Translation
by Jointly Learning to Align and Translate(2014)

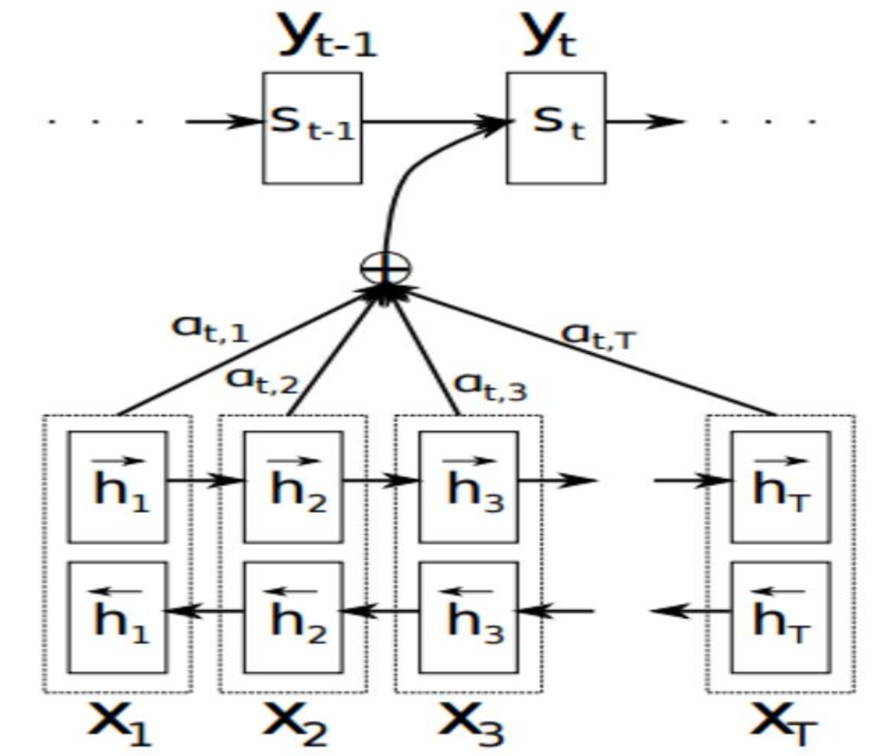


Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j,$$

i번째 단어를 만들때의 Context

Neural Machine Translation
by Jointly Learning to Align and Translate(2014)

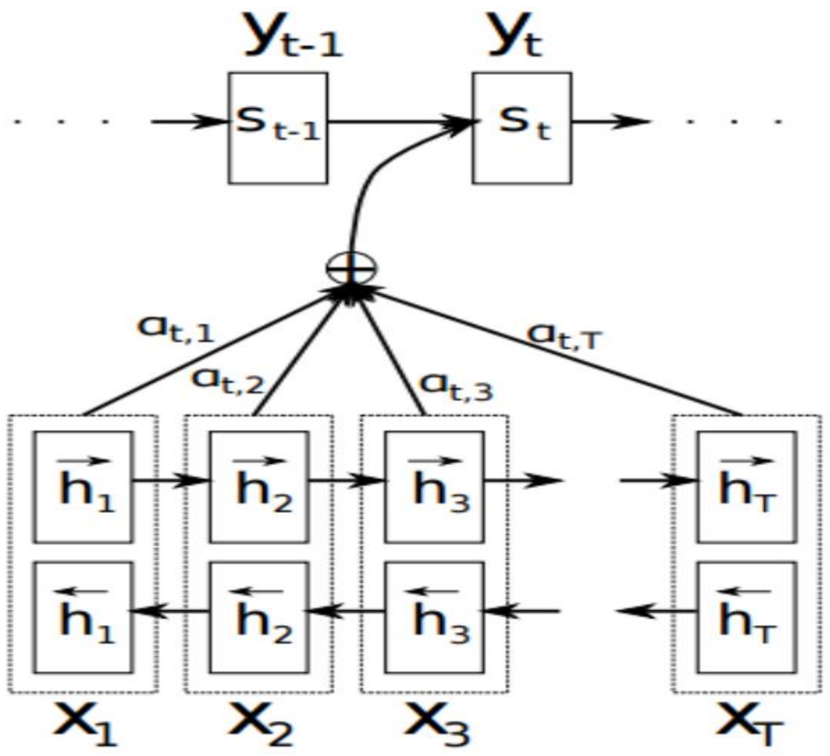


Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

$$s_i = (1 - z_i) \circ s_{i-1} + z_i \circ \tilde{s}_i,$$

$$\tilde{s}_i = \tanh\left(W E y_{i-1} + U\left[r_i \circ s_{i-1}\right] + C c_i\right)$$
$$z_i = \sigma\left(W_z E y_{i-1} + U_z s_{i-1} + C_z c_i\right)$$
$$r_i = \sigma\left(W_r E y_{i-1} + U_r s_{i-1} + C_r c_i\right)$$

$$s_0 = \tanh(W_s \overleftarrow{h_1})$$

Neural Machine Translation by Jointly Learning to Align and Translate(2014)

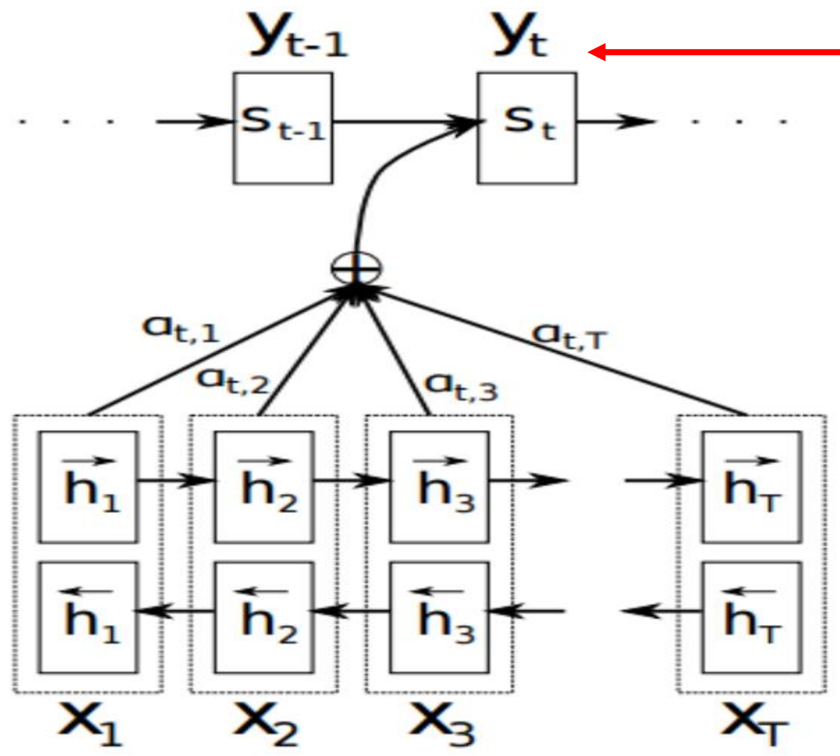


Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

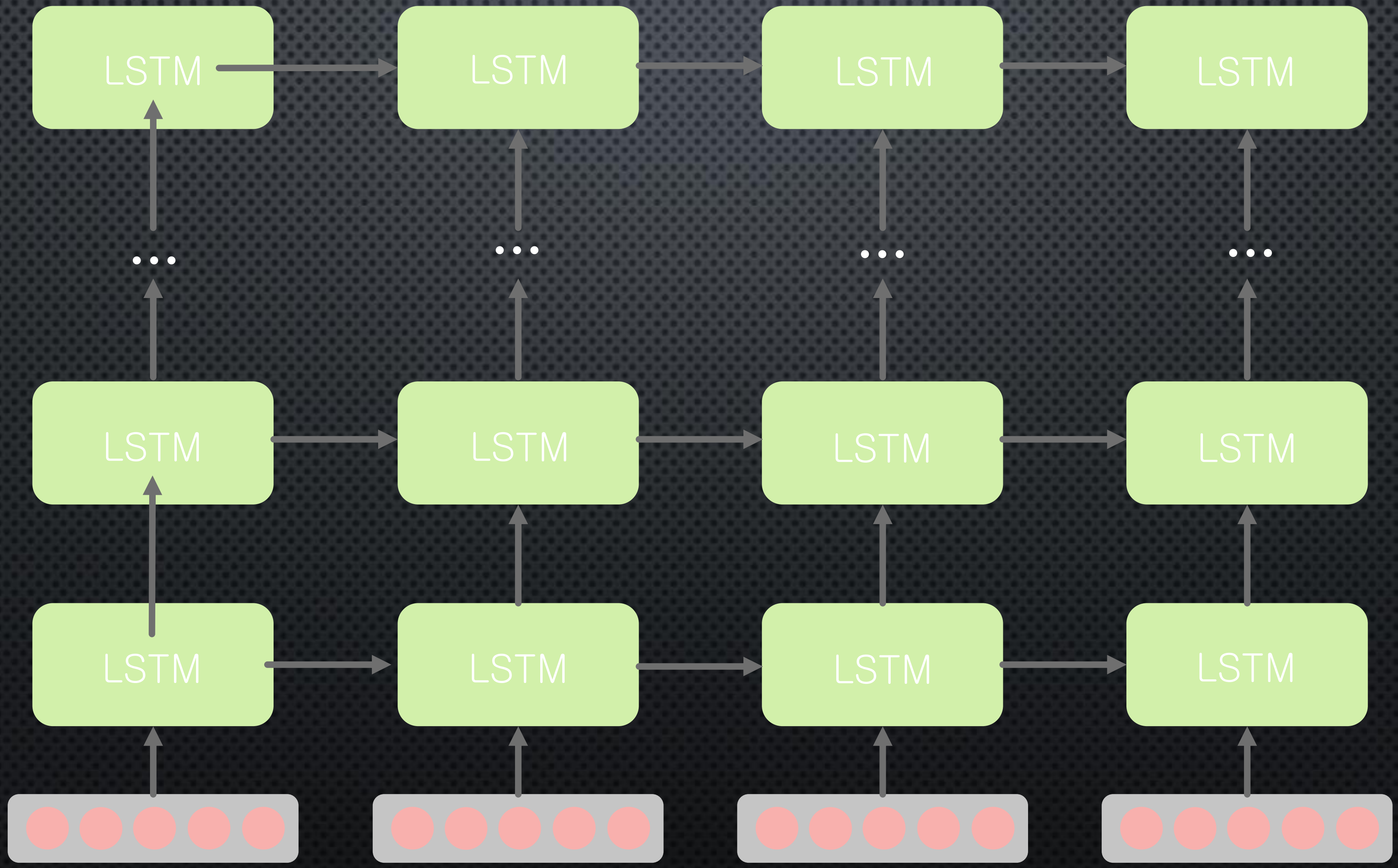
y_t를 생성할 때는 *maxout network* 사용(size : 500)

$$\tilde{t}_i = U_o s_{i-1} + V_o E y_{i-1} + C_o c_i.$$

$$t_i = \left[\max\left\{\tilde{t}_{i,2j-1}, \tilde{t}_{i,2j}\right\}\right]^\top_{j=1,\ldots,l}$$

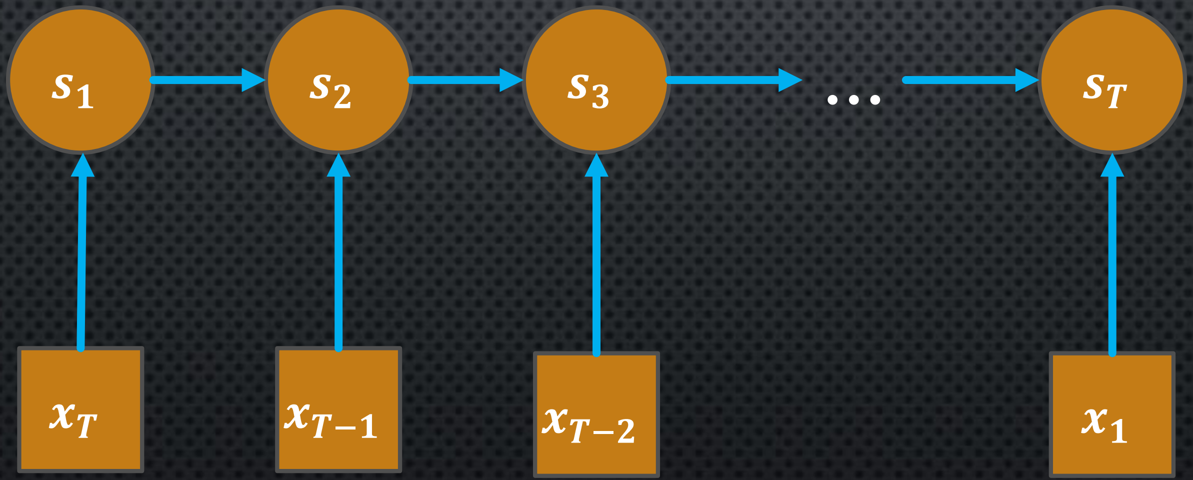
$$p(y_i | s_i, y_{i-1}, c_i) \propto \exp\left(y_i^\top W_o t_i\right),$$

Sequence to Sequence Learning
with Neural Networks (2014)

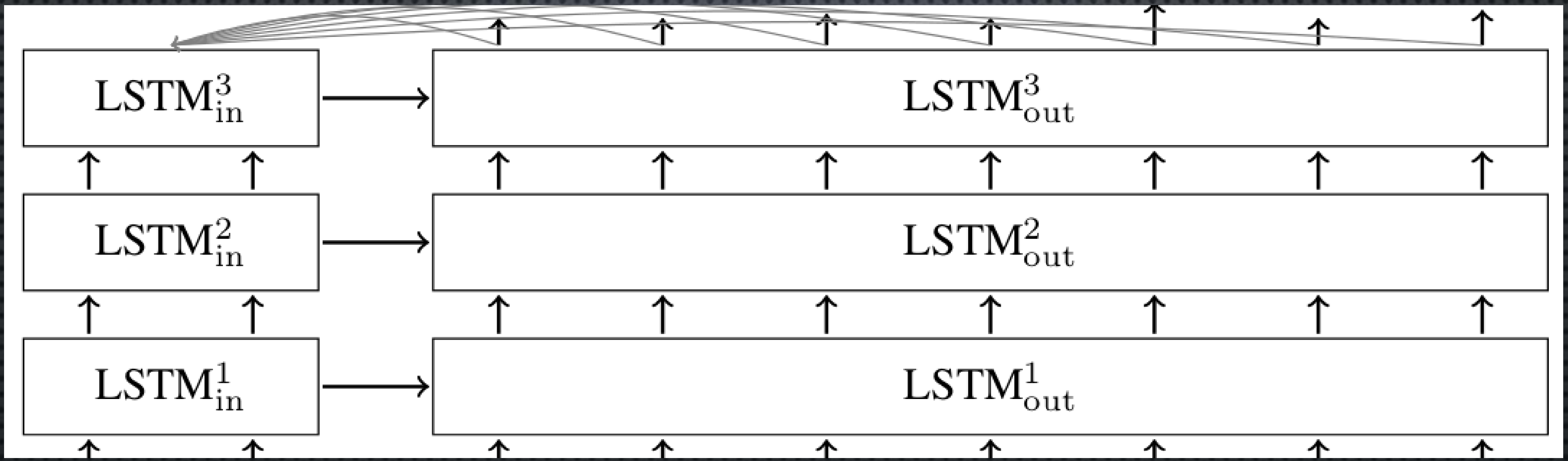


Sequence to Sequence Learning with Neural Networks (2014)

Reversing



Effective Approaches to Attention-based Neural Machine Translation (2015)



Sequence to Sequence Learning with Neural Networks (2014)

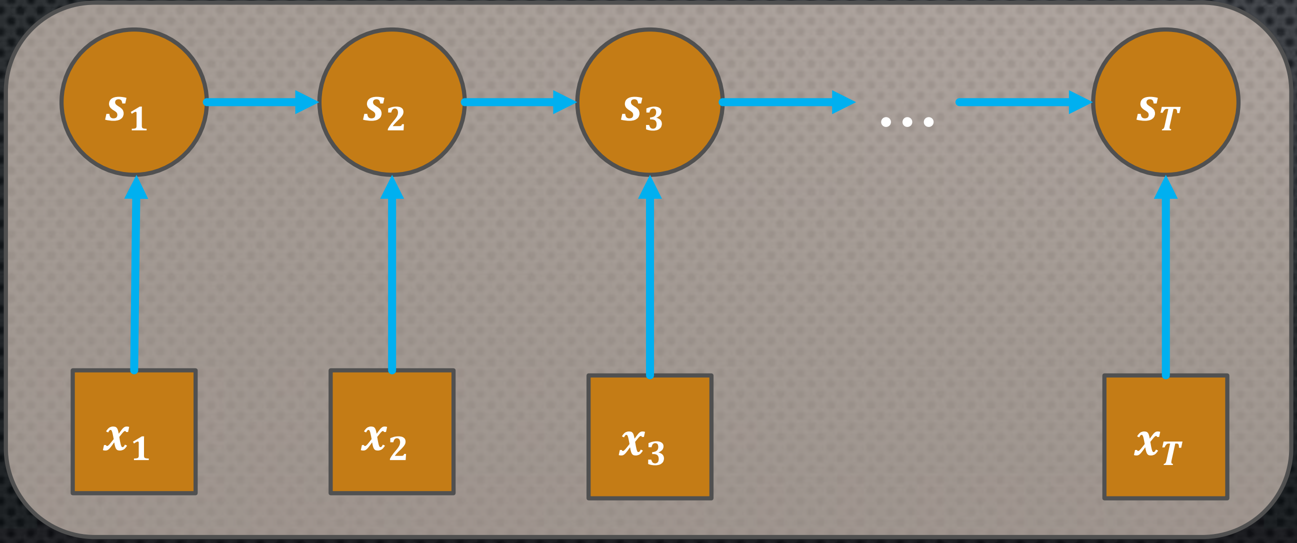
Deep RNN Encoder-Decoder

+

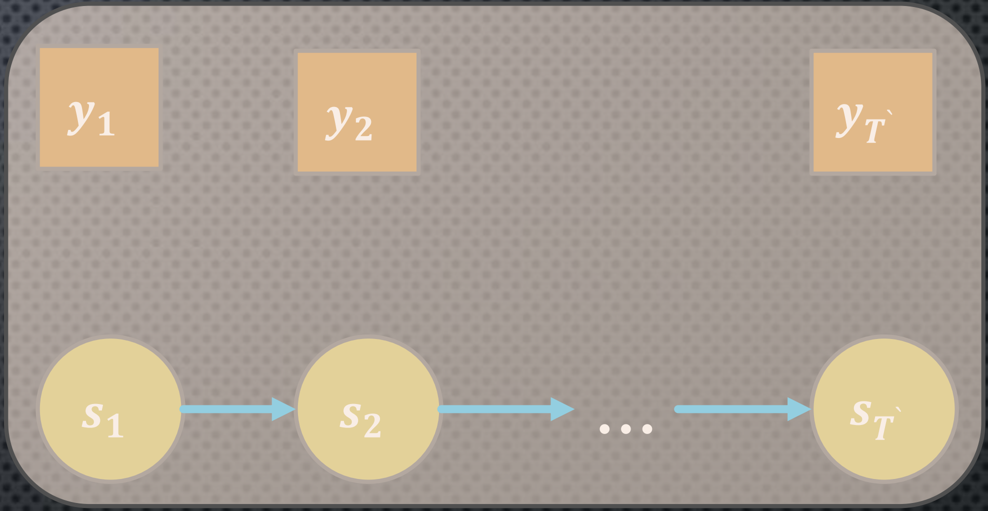
Neural Machine Translation by Jointly Learning to Align and Translate(2014)

Attention

Sequence-to-Sequence
with Attention Model for Text Summarization (2016, June)



Encoder



Decoder

Seq-to-seq with attention
+ GigaWord dataset

Sequence-to-Sequence with Attention Model for Text Summarization (2016, June)

Configuration:

Following is the configuration for the best trained model on Gigaword:

batch_size: 64

bidirectional encoding layer: 4

article length: first 2 sentences, total words within 120.

summary length: total words within 30.

word embedding size: 128

LSTM hidden units: 256

Sampled softmax: 4096

vocabulary size: Most frequent 200k words from dataset's article and summaries.

Sequence-to-Sequence with Attention Model for Text Summarization (2016, June)

Examples:

The following are some text summarization examples, including experiments using dataset other than Gigaword.

article: novell inc. chief executive officer eric schmidt has been named chairman of the internet search-engine company google .

human: novell ceo named google chairman

machine: novell chief executive named to head internet company

Sequence-to-Sequence
with Attention Model for Text Summarization (2016, June)

Experiment Result

8000 examples from testset are sampled to generate summaries and rouge score is calculated for the generated summaries. Here is the best rouge score on Gigaword dataset:

ROUGE-1 Average_R: 0.38272 (95%-conf.int. 0.37774 - 0.38755)

ROUGE-1 Average_P: 0.50154 (95%-conf.int. 0.49509 - 0.50780)

ROUGE-1 Average_F: 0.42568 (95%-conf.int. 0.42016 - 0.43099)

ROUGE-2 Average_R: 0.20576 (95%-conf.int. 0.20060 - 0.21112)

ROUGE-2 Average_P: 0.27565 (95%-conf.int. 0.26851 - 0.28257)

ROUGE-2 Average_F: 0.23126 (95%-conf.int. 0.22539 - 0.23708) https://www.tensorflow.org/versions/r0.10/tutorials/seq2seq/index.html

```
outputs, states = basic_rnn_seq2seq(encoder_inputs, decoder_inputs, cell)
```

- 참고

- https://www.tensorflow.org/versions/r0.10/tutorials/seq2seq/index.html
- Sequence-to-Sequence with Attention Model for Text Summarization
- Effective Approaches to Attention-based Neural Machine Translation
- Sequence to Sequence Learning with Neural Networks
- NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE
- Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation
- Continuous-Space Language Models for Statistical Machine Translation
- https://research.googleblog.com/2016/08/text-summarization-with-tensorflow.html
- https://github.com/tensorflow/models/tree/master/textsum

감사합니다