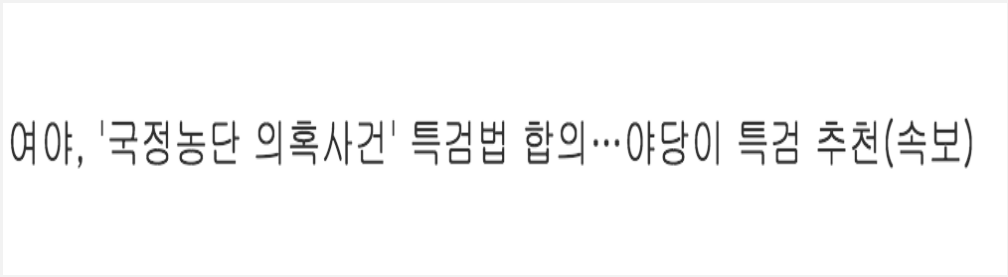
Abstractive Sentence Summarization with Attentive Recurrent Neural Networks

고려대학교 데이터시각화분석연구실
강경필

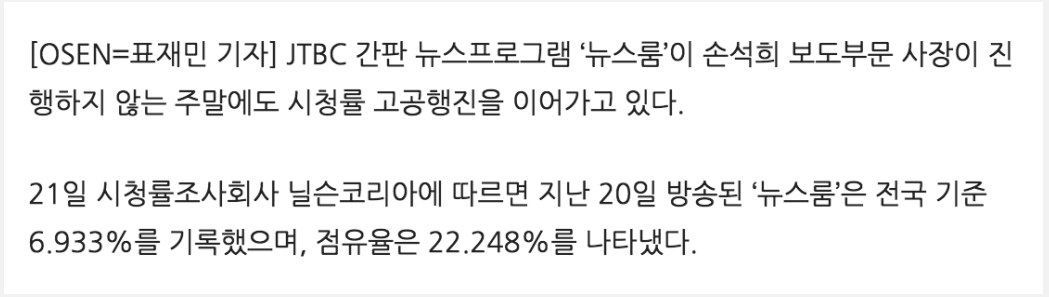
Contents

Related Work & Background

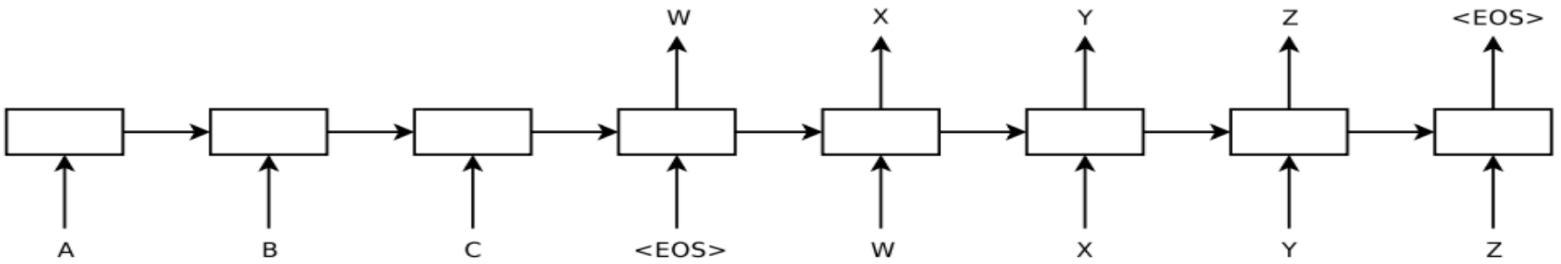
Extractive model : 본문에 나온 중요한 문장/어구를 가지고 조합



Abstractive model : 본문의 문맥을 통해 본문의 사용되지 않은 단어들도 사용



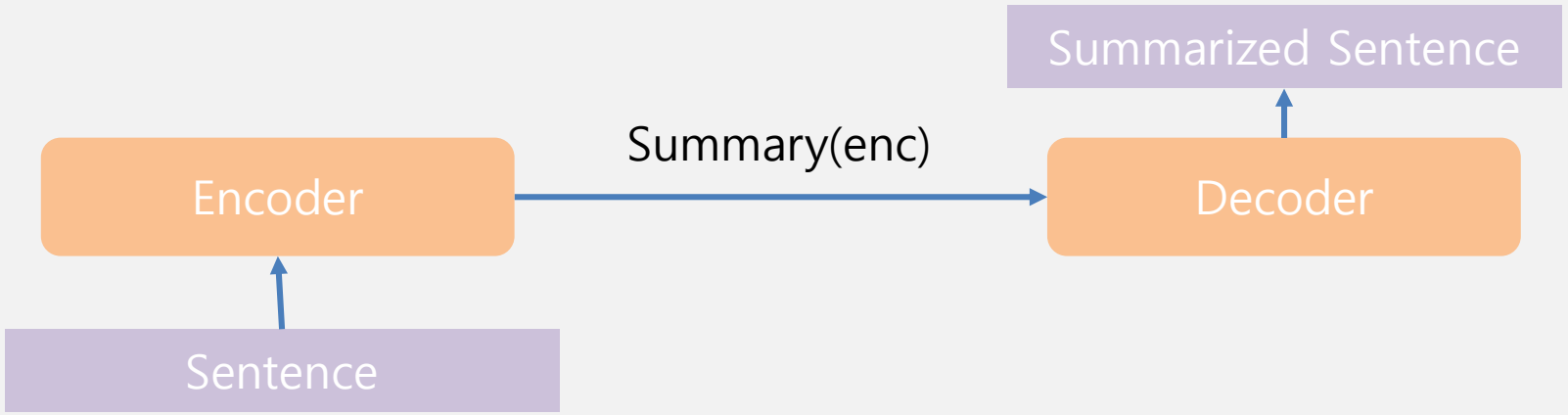
Google Text Summarization using Sequence to Sequence learning



Input: Article 1st sentence	Model-written headline
metro-goldwyn-mayer reported a third-quarter net loss of dlrs 16 million due mainly to the effect of accounting rules adopted this year	mgm reports 16 million net loss on higher revenue
starting from july 1, the island province of hainan in southern china will implement strict market access control on all incoming livestock and animal products to prevent the possible spread of epidemic diseases	hainan to curb spread of diseases
australian wine exports hit a record 52.1 million liters worth 260 million dollars (143 million us) in september, the government statistics office reported on monday	australian wine exports hit record high in september

뒤에 Luong-NMT(Stacked LSTM + Attention Mechanism)로 나옴

A Neural Attention Model for Abstractive Sentence Summarization



enc(x, y_c)는 일종의 summary

$$
\begin{aligned}
p(\mathbf{y}_{i+1}|\mathbf{y}_c, \mathbf{x}; \theta) &\propto \exp(\mathbf{V}\mathbf{h} + \mathbf{W}\mathrm{enc}(\mathbf{x}, \mathbf{y}_c)), \\
\tilde{\mathbf{y}}_c &= [\mathbf{E}\mathbf{y}_{i-C+1}, \ldots, \mathbf{E}\mathbf{y}_i], \\
\mathbf{h} &= \tanh(\mathbf{U}\tilde{\mathbf{y}}_c).
\end{aligned}
$$

(주의) $\tilde{y}_c$ 는 concatenated vector

$E \in \mathbb{R}^{D\times V}$ $y \in \mathbb{R}^{V}$ => $\tilde{y}_c \in \mathbb{R}^{CD}$

Language Model (Decoder)

A Neural Attention Model for Abstractive Sentence Summarization

Encoder는 3가지를 가지고 비교

Bag-of-Words Encoder Our most basic model simply uses the bag-of-words of the input sentence embedded down to size H, while ignoring properties of the original order or relationships between neighboring words. We write this model as:

$$\begin{aligned} \text{enc}_1(\mathbf{x}, \mathbf{y}_c) &= \mathbf{p}^\top \tilde{\mathbf{x}}, \\ \mathbf{p} &= [1/M, \ldots, 1/M], \\ \tilde{\mathbf{x}} &= [\mathbf{F}\mathbf{x}_1, \ldots, \mathbf{F}\mathbf{x}_M]. \end{aligned}$$

Where the input-side embedding matrix $\mathbf{F} \in \mathbb{R}^{H \times V}$ is the only new parameter of the encoder and $\mathbf{p} \in [0, 1]^M$ is a uniform distribution over the input words.

단어들의 Embedding Vector를 합한 것을 summary로 사용

A Neural Attention Model for Abstractive Sentence Summarization

L개의 Convolutional Layer

Activation, Max Pooling

$$\forall j, \quad \mathrm{enc}_2(\mathbf{x}, \mathbf{y}_c)_j = \max_i \tilde{\mathbf{x}}^L_{i,j}, \tag{5}$$

$$\forall i, l \in \{1, \ldots L\}, \quad \tilde{\mathbf{x}}^l_j = \tanh(\max\{\bar{\mathbf{x}}^l_{2i-1}, \bar{\mathbf{x}}^l_{2i}\}), \tag{6}$$

$$\forall i, l \in \{1, \ldots L\}, \quad \bar{\mathbf{x}}^l_i = \mathbf{Q}^l \tilde{\mathbf{x}}^{l-1}_{[i-Q,\ldots,i+Q]}, \tag{7}$$

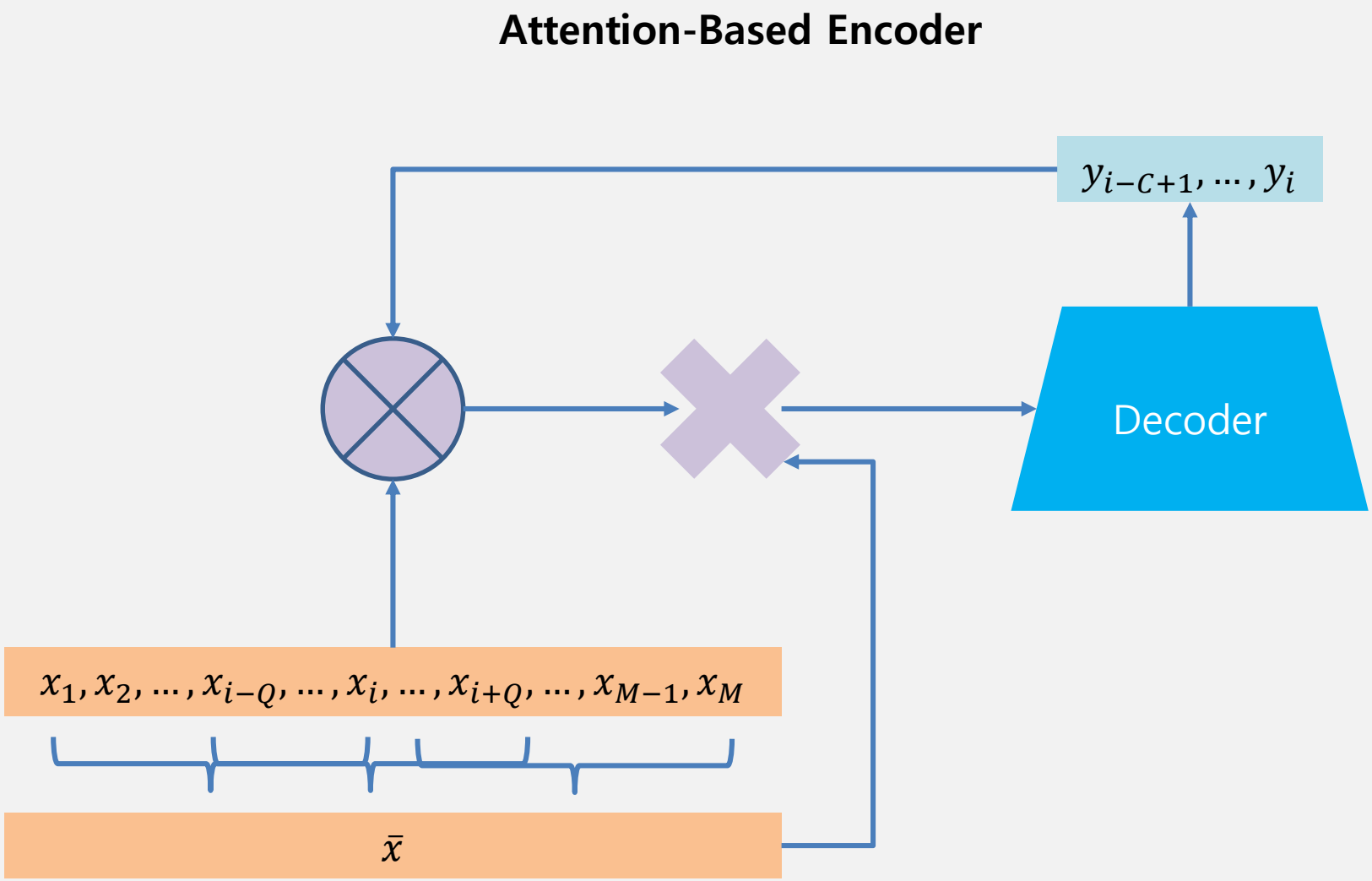
$$\tilde{\mathbf{x}}^0 = [\mathbf{F}\mathbf{x}_1, \ldots, \mathbf{F}\mathbf{x}_M]. \tag{8}$$

Filter $Q \in R^{L \times H \times (2Q+1)}$

$M > 2^L$

Convolutional Encoder

A Neural Attention Model for Abstractive Sentence Summarization



A Neural Attention Model for Abstractive Sentence Summarization

Attention-Based Encoder

$$
\begin{aligned}
\text{⑤}\ \mathrm{enc}_3(\mathbf{x}, \mathbf{y}_c) &= \mathbf{p}^\top \bar{\mathbf{x}}, \\
\text{④}\ \mathbf{p} &\propto \exp(\tilde{\mathbf{x}} \mathbf{P} \tilde{\mathbf{y}}'_c), \\
\text{①}\ \tilde{\mathbf{x}} &= [\mathbf{F}\mathbf{x}_1, \ldots, \mathbf{F}\mathbf{x}_M], \\
\text{②}\ \tilde{\mathbf{y}}'_c &= [\underbrace{\mathbf{G}\mathbf{y}_{i-C+1}, \ldots, \mathbf{G}\mathbf{y}_i}_{i+Q}], \\
\text{③}\ \forall i \quad \bar{\mathbf{x}}_i &= \sum_{q=i-Q} \tilde{\mathbf{x}}_i / Q.
\end{aligned}
$$

(주의)
$\tilde{y}_c$ 는 concatenated vector
$\tilde{x}$ 는 concatenated vector 아님!

$$x, y \in \mathbb{R}^V$$

$$\mathrm{F} \in \mathbb{R}^{H \times V}, \mathrm{G} \in \mathbb{R}^{D \times V}, \mathrm{P} \in \mathbb{R}^{H \times (CD)}$$

$$\bar{x}, \tilde{x} \in \mathbb{R}^{M \times H}$$

$$\tilde{y_c} \in \mathbb{R}^{CD}$$

$$p \in \mathbb{R}^M$$

$$\mathrm{enc} \in \mathbb{R}^H$$

이전 C개의 generated 단어들의 문맥

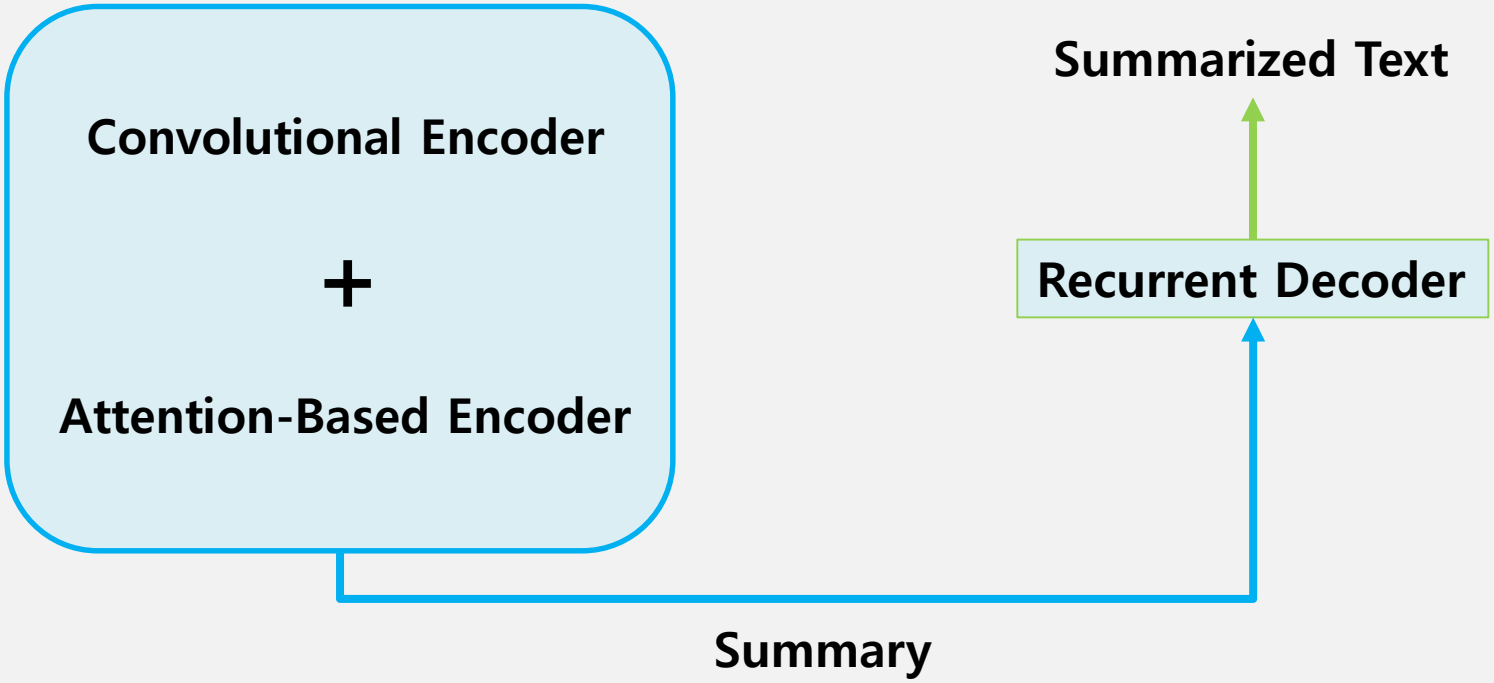
기존의 Attention과 같이 입력 단어를 attention 한다는 것은 아님!
생성된 C개의 단어를 입력 단어들과 다시 조합해서 그 때의 문맥(enc)을 생성

A Neural Attention Model for Abstractive Sentence Summarization

$$2^{H(p)} = 2^{-\sum_x p(x) \log_2 p(x)}$$

Model	Encoder	Perplexity
KN-Smoothed 5-Gram	none	183.2
Feed-Forward NNLM	none	145.9
Bag-of-Word	enc_1	43.6
Convolutional (TDNN)	enc_2	35.9
Attention-Based (ABS)	enc_3	27.1

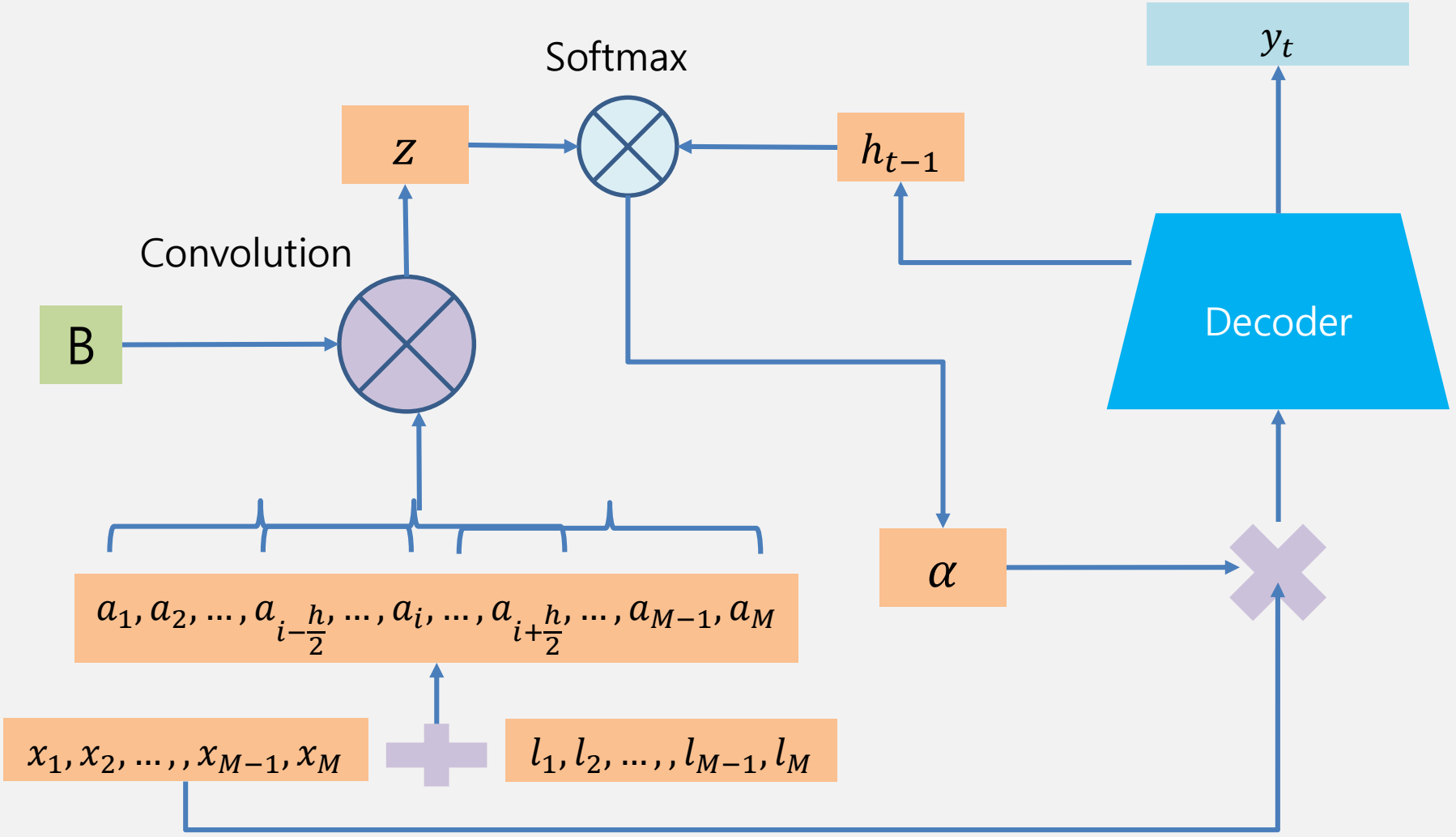
Abstractive Sentence Summarization with Attentive Recurrent Neural Networks



RAS(Recurrent Attentive Summarizer)

A Neural Attention Model for Abstractive Sentence Summarization

Attention-Based Encoder



Encoder

$x_i \in \mathbb{R}^d$: 단어의 learnable embedding vector

$l_i \in \mathbb{R}^d$: i번째 위치정보 learnable embedding vector

$a_i = x_i + l_i$: 단어의 full embedding vector

$B^k \in \mathbb{R}^{q \times d}$: convolution의 Filter matrix, d개 있음

Feature map: $z_{ik} = \sum_{h=-q/2}^{q/2} a_{i+h} \cdot b^k_{\frac{q}{2}+h}$, $z_i \in \mathbb{R}^d$

i번째 단어 앞뒤 각 q/2 단어를 convolution함

$$\alpha_{j,t-1} = \frac{\exp(z_j \cdot h_{t-1})}{\sum_{i=1}^{M} \exp(z_i \cdot h_{t-1})}$$

$h_t \in \mathbb{R}^d$: decoder에서의 t번째 state

$$c_t = \sum_{j=1}^{M} \alpha_{j,t-1} x_j$$

Decoder

Our Elman RNN takes the following form (Elman, 1990):

$$
\begin{aligned}
h_t &= \sigma(W_1 y_{t-1} + W_2 h_{t-1} + W_3 c_t) \\
P_t &= \rho(W_4 h_t + W_5 c_t),
\end{aligned}
$$

Elman RNN(기본 RNN)

The LSTM decoder is defined as (Hochreiter and Schmidhuber, 1997):

$$
\begin{aligned}
i_t &= \sigma(W_1 y_{t-1} + W_2 h_{t-1} + W_3 c_t) \\
i'_t &= \tanh(W_4 y_{t-1} + W_5 h_{t-1} + W_6 c_t) \\
f_t &= \sigma(W_7 y_{t-1} + W_8 h_{t-1} + W_9 c_t) \\
o_t &= \sigma(W_{10} y_{t-1} + W_{11} h_{t-1} + W_{12} c_t) \\
m_t &= m_{t-1} \odot f_t + i_t \odot i'_t \\
h_t &= m_t \odot o_t \\
P_t &= \rho(W_{13} h_t + W_{14} c_t).
\end{aligned}
$$

LSTM RNN

Loss, Training

$$\mathcal{L} = -\sum_{i=1}^{S}\sum_{t=1}^{N} \log P(y_t^i | \{y_1^i, \ldots, y_{t-1}^i\}, \mathbf{x}^i; \theta),$$

Negative Conditional log likelihood
(실제 정답 단어에 대한 log probability의 합)

S: training corpus 갯수
N : 문장 최대 길이

evaluation measure : perplexity

Loss, Training

```
<DOC id="AFE19940512.0003" type="story" >
<HEADLINE>
 Tributes pour in for late British Labour Party leader
</HEADLINE>
<DATELINE>
UNDATED, May 12 (AFP)
</DATELINE>
<TEXT>
<P>
Tributes poured in from around the world Thursday
to the late Labour Party leader John Smith, who died earlier from a massive
heart attack aged 55.
</P>
```

Training dataset : Gigaword corpus
Evaluation & Testset : Gigaword(held-out) 2000, DUC-2004

Loss, Training

Library : Torch (Facebook)
stochastic gradient descent(mini-batchs : 32)

grid search로 hyper-parameter들 찾음

RAS-Elman
H = 512, learning_rate = 0.5, learning rate annealing : 2
gradient clipping threashold : 10

RAS-LSTM
H = 512, learning_rate = 0.1, learning rate annealing : 2
gradient clipping threashold : 10

Results

Model	Perplexity
Bag-of-Words	43.6
Convolutional (TDNN)	35.9
Attention-based (ABS)	27.1
RAS-Elman	**18.9**
RAS-LSTM	20.3

Table 1: Perplexity on the Gigaword validation set. Bag-of-words, Convolutional (TDNN) and ABS are the different encoders of Rush et. al., 2015.

Results

	RG-1	RG-2	RG-L
ABS	29.55	11.32	26.42
ABS+	29.76	11.88	26.96
RAS-Elman ($k = 1$)	33.10	14.45	30.25
RAS-Elman ($k = 10$)	**33.78**	**15.97**	**31.15**
RAS-LSTM ($k = 1$)	31.71	13.63	29.31
RAS-LSTM ($k = 10$)	32.55	14.70	30.03
Luong-NMT	33.10	14.45	30.71

Table 2: F1 ROUGE scores on the Gigaword test set. ABS and ABS+ are the systems of Rush et al. 2015. k refers to the size of the beam for generation; $k = 1$ implies greedy generation. RG refers to ROUGE. Rush et al. (2015) previously reported ROUGE recall, while as we use the more balanced F-measure.

- RAS-Elman이 RAS-LSTM보다 성능이 좋음

- beam search를 한 경우 성능 향상

- ABS+는 부가정보를 사용했음에도 RAS 성능이 더 높음

- ABS+는 문장의 단어를 92% copy하지만 RSA은 74%만 copy

- Luong-NMT(Stacked LSTM + Attention Mechanism)의 경우에 비해 RAS는 훨씬 단순함에도 성능 차이는 거의 없음

Results

	RG-1	RG-2	RG-L
ABS (k = 50)	26.55	7.06	22.05
ABS+ (k = 50)	28.18	8.49	23.81
RAS-Elman ($k = 1$)	29.13	7.62	23.92
RAS-Elman ($k = 10$)	**28.97**	**8.26**	**24.06**
RAS-LSTM ($k = 1$)	26.90	6.57	22.12
RAS-LSTM ($k = 10$)	27.41	7.69	23.06
Luong-NMT	28.55	8.79	24.43

Table 3: ROUGE results (recall-only) on the DUC-2004 test sets. ABS and ABS+ are the systems of Rush et al. 2015. k refers to the size of the beam for generation; $k = 1$ implies greedy generation. RG refers to ROUGE.

- Gigaword에 비해 DUC dataset에 대해서는 성능향상이 그리 크지 않음
  - dataset tokenization이 살짝 다름
  - Gigaword의 headline이 더 짧음

Results

I(1): brazilian defender pepe is out for the rest of the season with a knee injury , his porto coach jesualdo ferreira said saturday .
G: football : pepe out for season
A+: ferreira out for rest of season with knee injury
R: brazilian defender pepe out for rest of season with knee injury

I(2): economic growth in toronto will suffer this year because of sars , a think tank said friday as health authorities insisted the illness was under control in canada 's largest city .
G: sars toll on toronto economy estimated at c$ # billion
A+: think tank under control in canada 's largest city
R: think tank says economic growth in toronto will suffer this year

I(3): colin l. powell said nothing – a silence that spoke volumes to many in the white house on thursday morning .
G: in meeting with former officials bush defends iraq policy
A+: colin powell speaks volumes about silence in white house
R: powell speaks volumes on the white house

실수

I(4): an international terror suspect who had been under a controversial loose form of house arrest is on the run , british home secretary john reid said tuesday .
G: international terror suspect slips net in britain
A+: reid under house arrest terror suspect on the run
R: international terror suspect under house arrest

Conclusion

- Convolutional attention encoder + RNN decoder
- 단순한 모델이지만 Luong-NMT만큼의 성능이 나옴
- 후속 연구로 앞서 말한 실수를 해결하는 모델 개발

Thank you