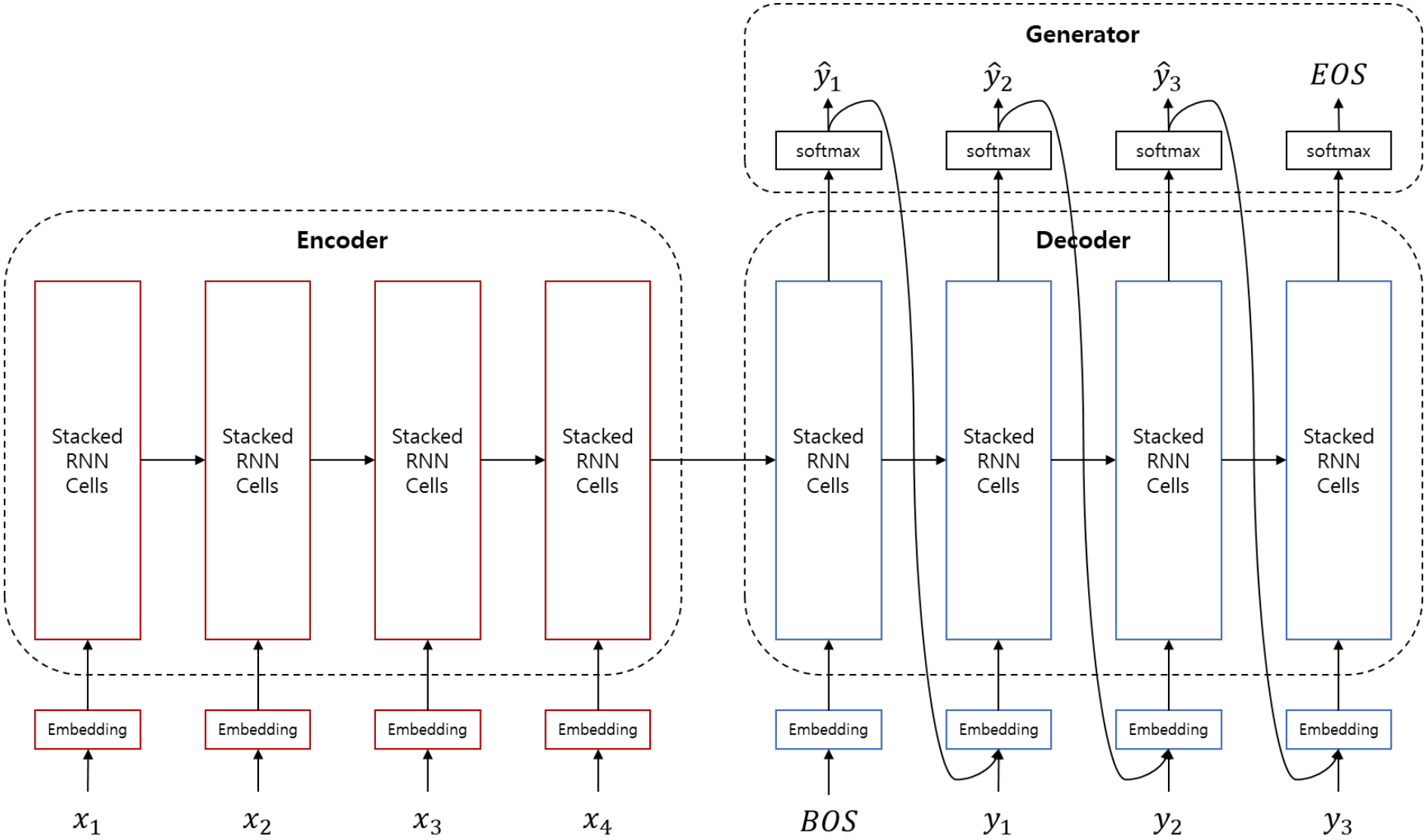
Attention

Ki Hyun Kim

nlp.with.deep.learning@gmail.com

Previous Work



What is Attention?

- Differentiable Key-Value Function

- 기존의 Key-Value 함수와 달리, Query와 Key의 유사도에 따라 Value를 반환

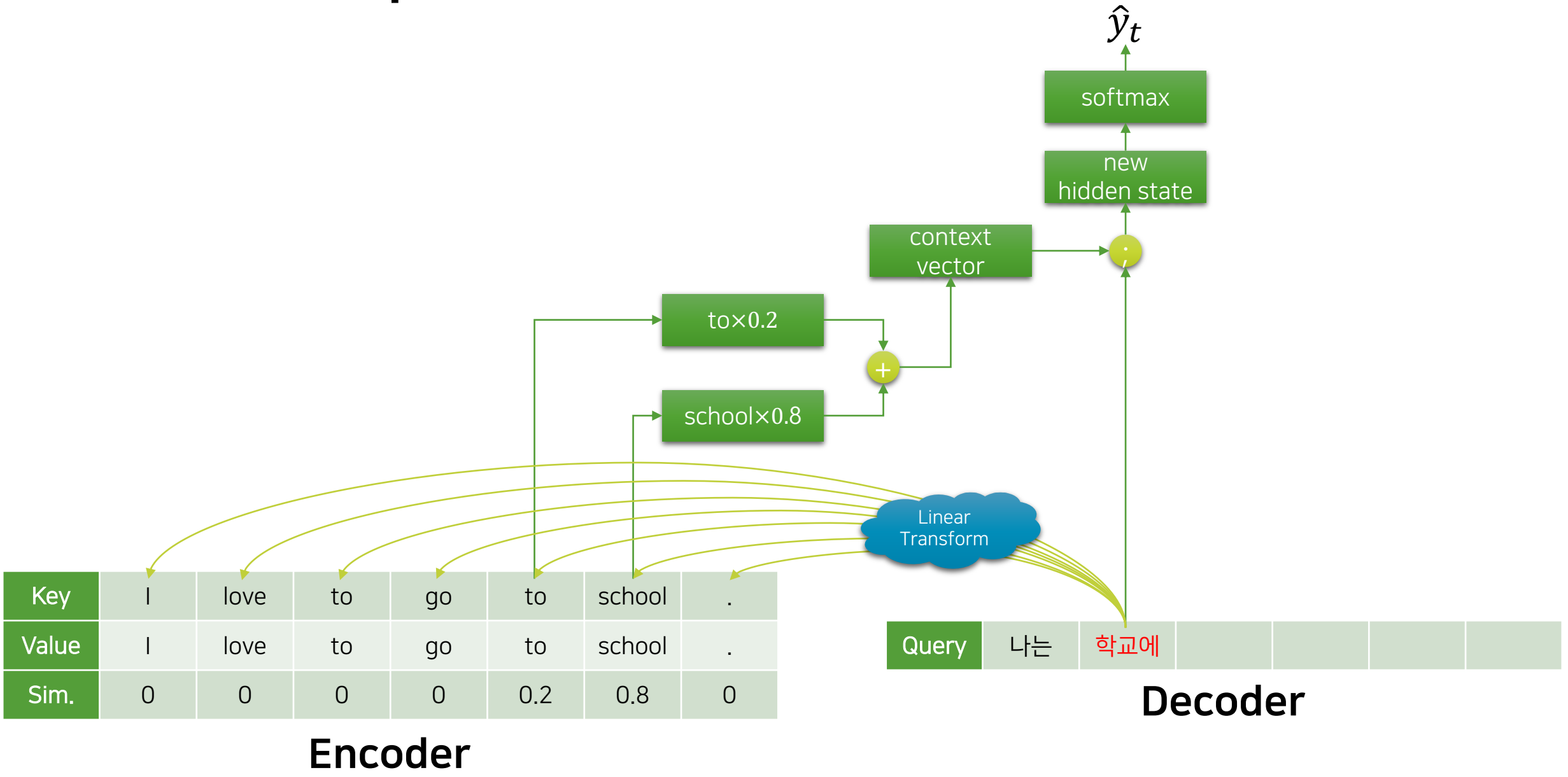
- Decoder RNN(LSTM)의 hidden state의 한계로 인해 부족한 정보를 직접 encoder에 조회하여 예측에 필요한 정보를 얻어오는 과정

- 정보를 잘 얻어오기 위해 Query를 잘 만들어내는 과정을 학습

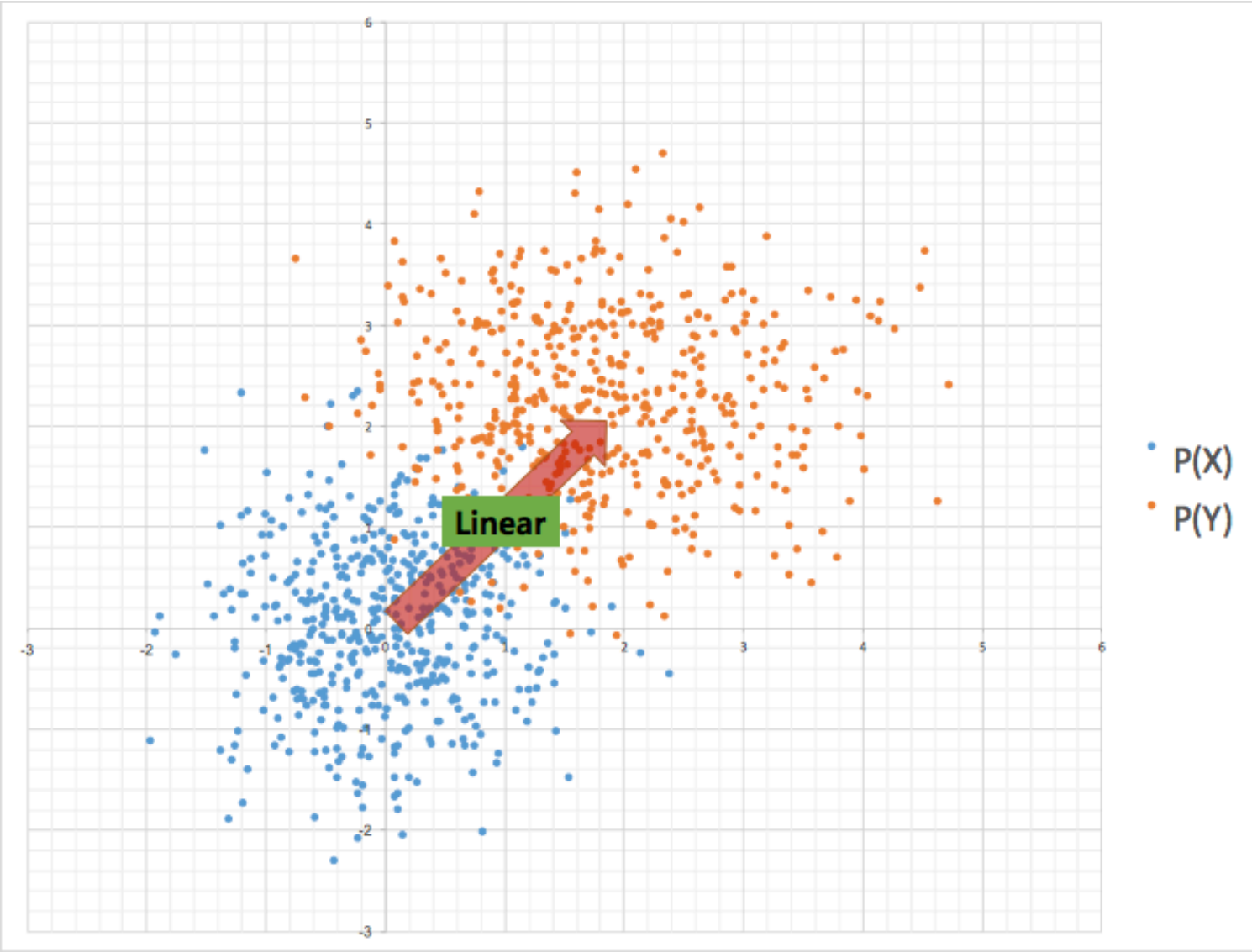
Attention in Seq2seq

- Query: 현재 time-step의 decoder output
- Keys: 각 time-step 별 encoder output
- Values: 각 time-step 별 encoder output

Intuitive Explanations

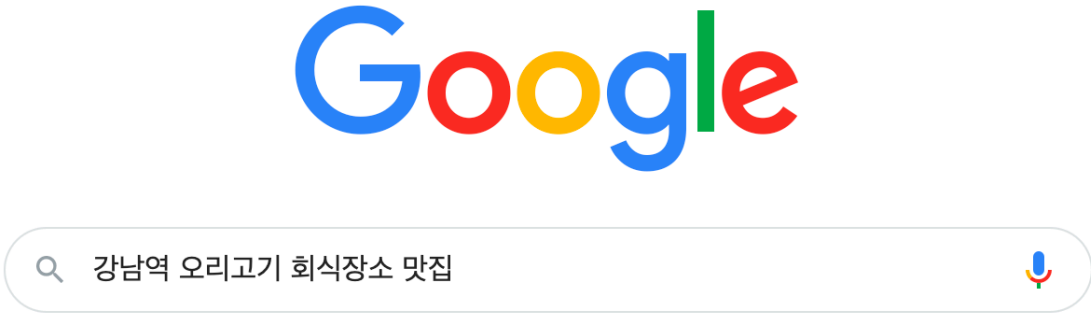


Linear Transformation



Linear Transformation

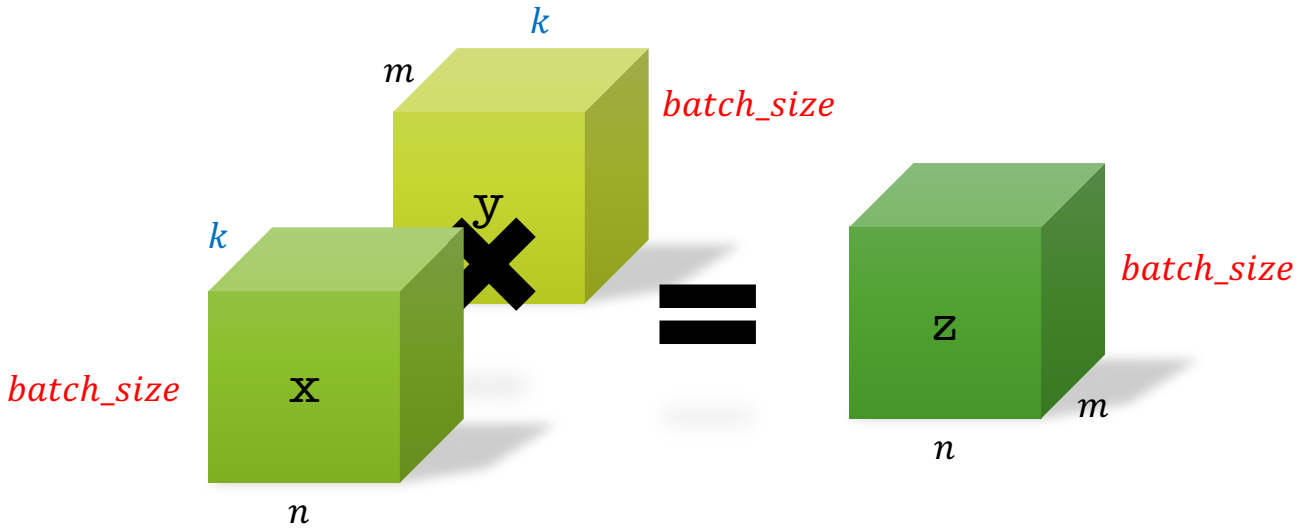
- Example:



마음의 상태(state)를 잘 반영하면서
좋은 검색 결과를 이끌어내는 쿼리를 얻기 위함

Before we start,

- Batch Matrix Multiplication (BMM)



z = torch.bmm(x, y)

$$(batch\_size, n, k) \times (batch\_size, k, m) = (batch\_size, n, m)$$

x y z

Equations

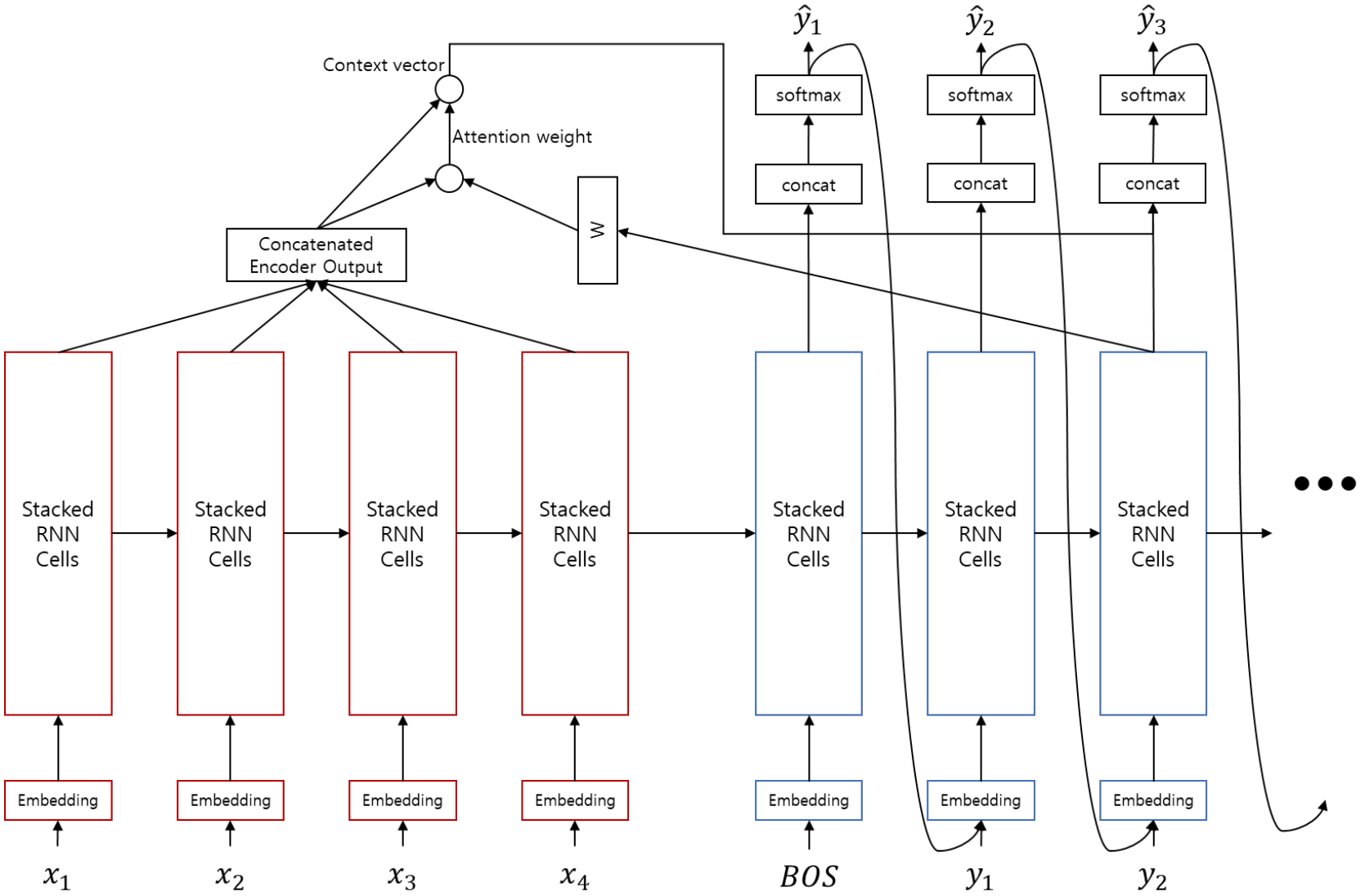
- With entire encoder's hidden states and current decoder's hidden state,

$$w = \text{softmax}(h_t^{\text{dec}} \cdot W_{\text{a}} \cdot h_{1:m}^{\text{enc}\,T})$$
$$c = w \cdot h_{1:m}^{\text{enc}},$$
$$\text{where } c \in \mathbb{R}^{\text{batch\_size} \times 1 \times \text{hidden\_size}} \text{ is a context vector, and } W_{\text{a}} \in \mathbb{R}^{\text{hidden\_size} \times \text{hidden\_size}}.$$

- Re-define decoder's hidden state, and feed into generator,

$$\tilde{h}_t^{\text{dec}} = \tanh([h_t^{\text{dec}}; c] \cdot W_{\text{concat}})$$
$$\hat{y}_t = \text{softmax}(\tilde{h}_t^{\text{dec}} \cdot W_{\text{gen}}),$$
$$\text{where } W_{\text{concat}} \in \mathbb{R}^{(2 \times \text{hidden\_size}) \times \text{hidden\_size}} \text{ and } W_{\text{gen}} \in \mathbb{R}^{\text{hidden\_size} \times |V|}.$$

Attention



Evaluation

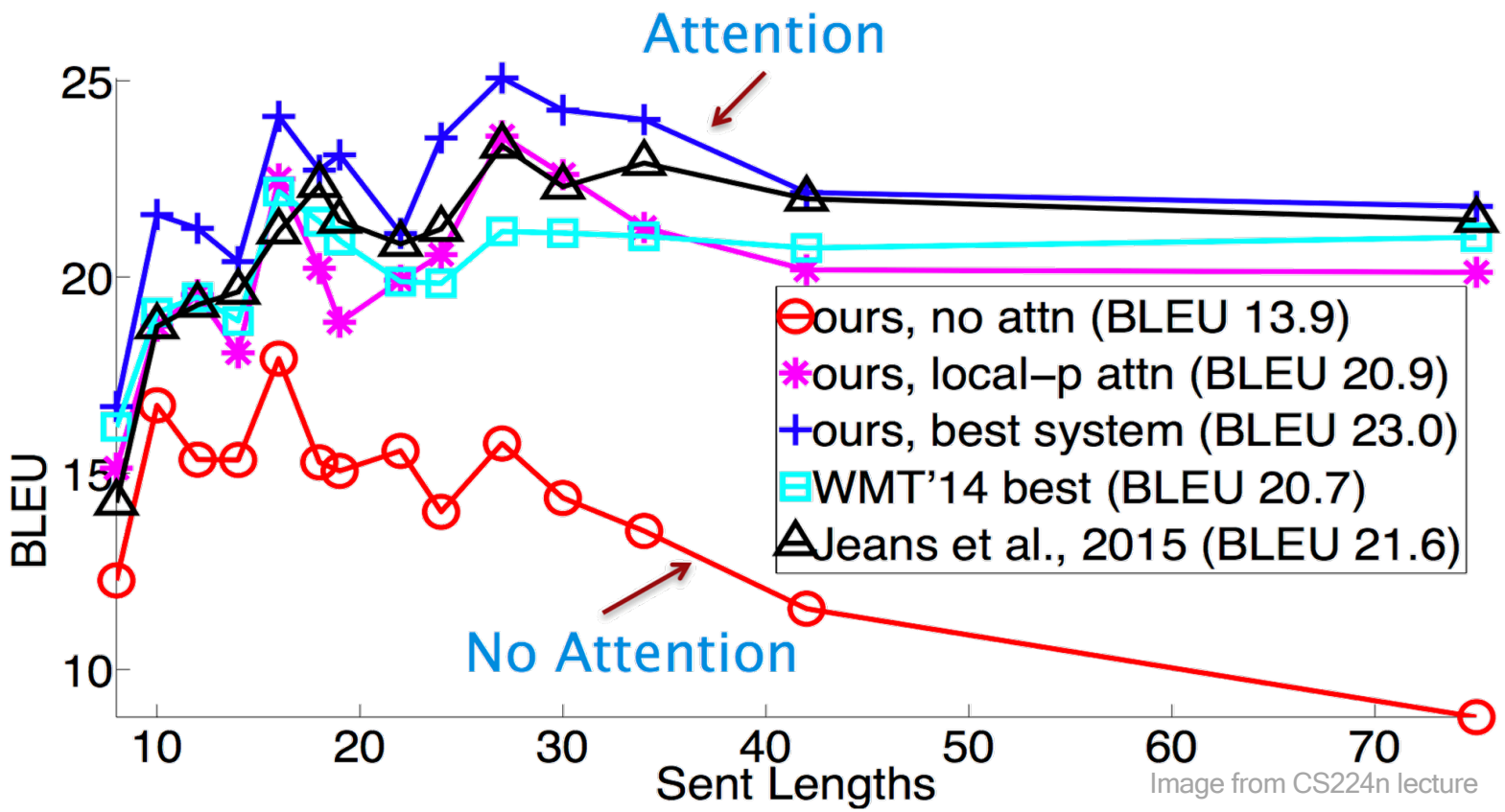


Image from CS224n lecture

Summary

- Attention은 미분 가능한 Key-Value Function이다.
  - Attention 함수의 입력은 Query, Key, Value.

- 정보를 잘 얻기 위한 Query를 변환하는 방법을 배우는 과정

- Attention을 통해 RNN의 hidden state의 한계를 극복 가능
  - LSTM을 쓰더라도 context vector에 모든 정보를 담기에는 한계가 있음
  - 더 긴 길이의 입력/출력에도 대처할 수 있게 됨