Neural Machine Translation by Jointly Learning to Align and Translate

JUHYEONG LEE, KNUAIR

Agenda

I. Introduction

II. Background

III. Model Architecture : Learning to Align and Translate

IV. Experiment & Results

Introduction

Dzmitry Bahdanau

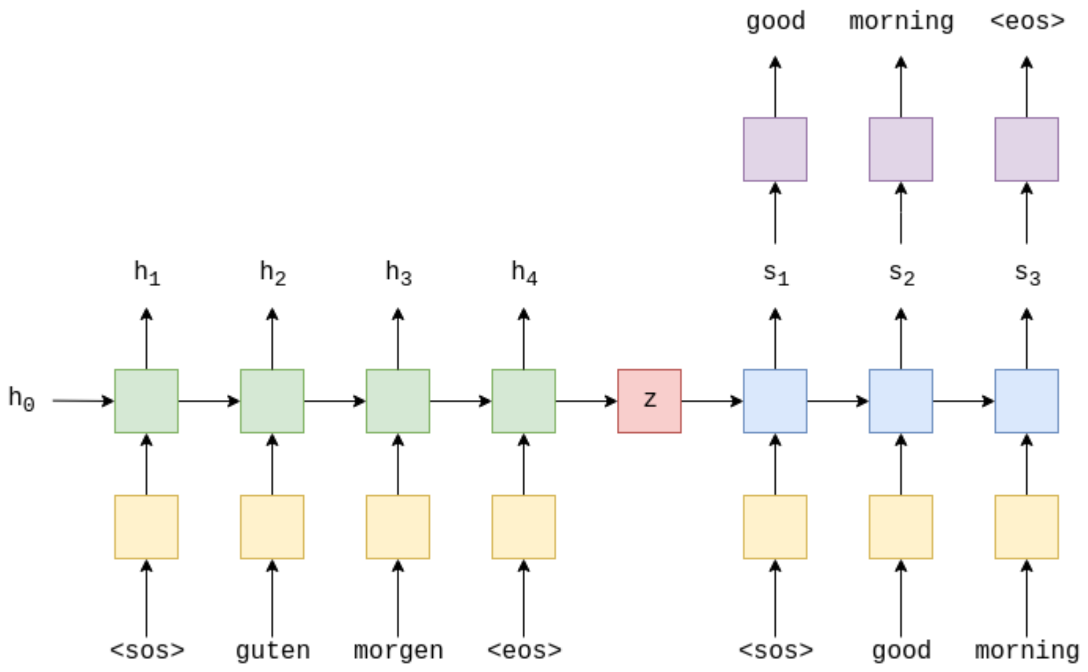
Kyunghyun Cho

Yoshua Bengio

- *Bahdanau, D., Cho, K. H., & Bengio, Y. (2015, January). Neural machine translation by jointly learning to align and translate. In 3rd International Conference on Learning Representations, ICLR 2015.*

Introduction

- Previous Presentation : Seq-to-Seq



Source : https://github.com/bentrevett/pytorch-seq2seq

Introduction

- A Family of Encoder–Decoders
  - Encode a source sentence into a *fixed-length vector*
  - needs to be able to compress all the necessary information of a source sentence into a fixed-length vector

Introduction

- A Family of Encoder–Decoders
  - **Encode a source sentence into a** *fixed-length vector*
  - needs to be able to **compress all the necessary information** of a source sentence into a fixed-length vector

- Problem
  - the use of a fixed-length vector is a bottleneck
  - difficult for the neural network to cope with long sentences

Introduction

- Proposal Method
  - Extend this by allowing a model to automatically (soft-)search for parts of a source sentence that are *relevant to predicting a target word*, without having to form these parts as a hard segment explicitly.

Attention Mechanism

- Searches for a set of positions in a source sentence where the most relevant information is concentrated

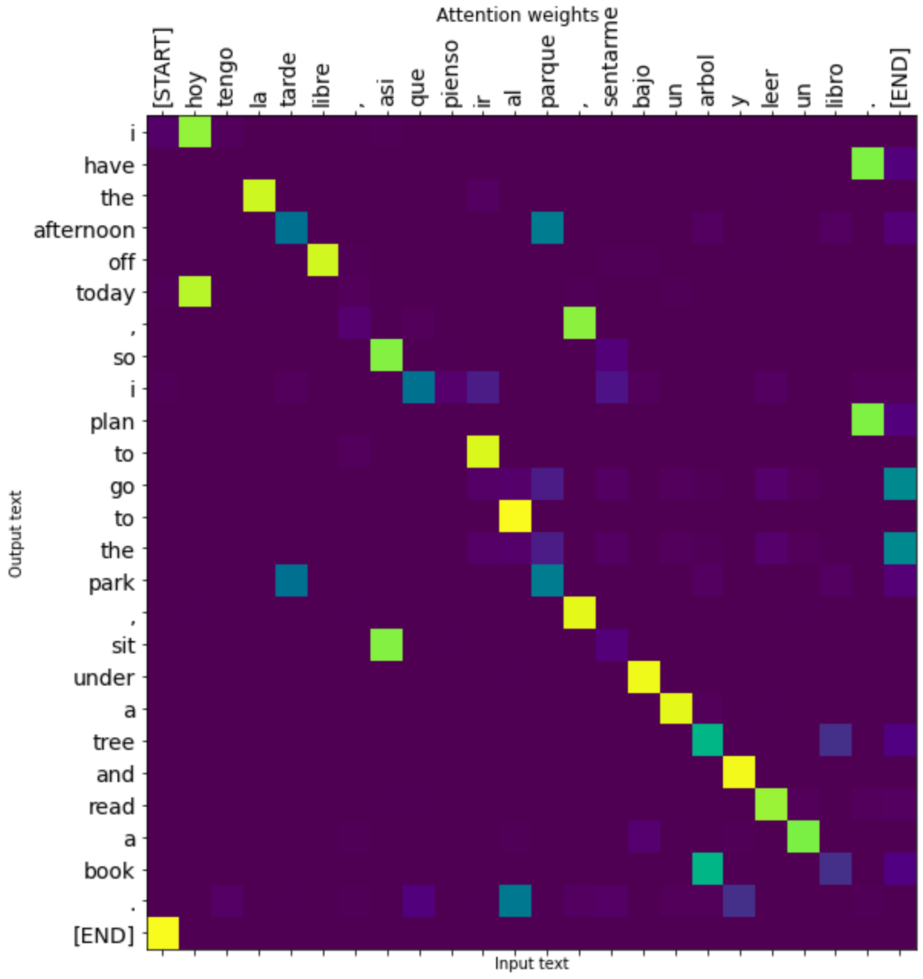


Image Source : https://towardsdatascience.com/end-to-end-attention-based-machine-translation-model-with-minimum-tensorflow-code-ae2f08cc8218

Attention Mechanism

- Searches for a set of positions in a source sentence **where the most relevant information is concentrated**
- Predicts a target word based on **the context vectors associated with these source positions** and all the previous generated target words

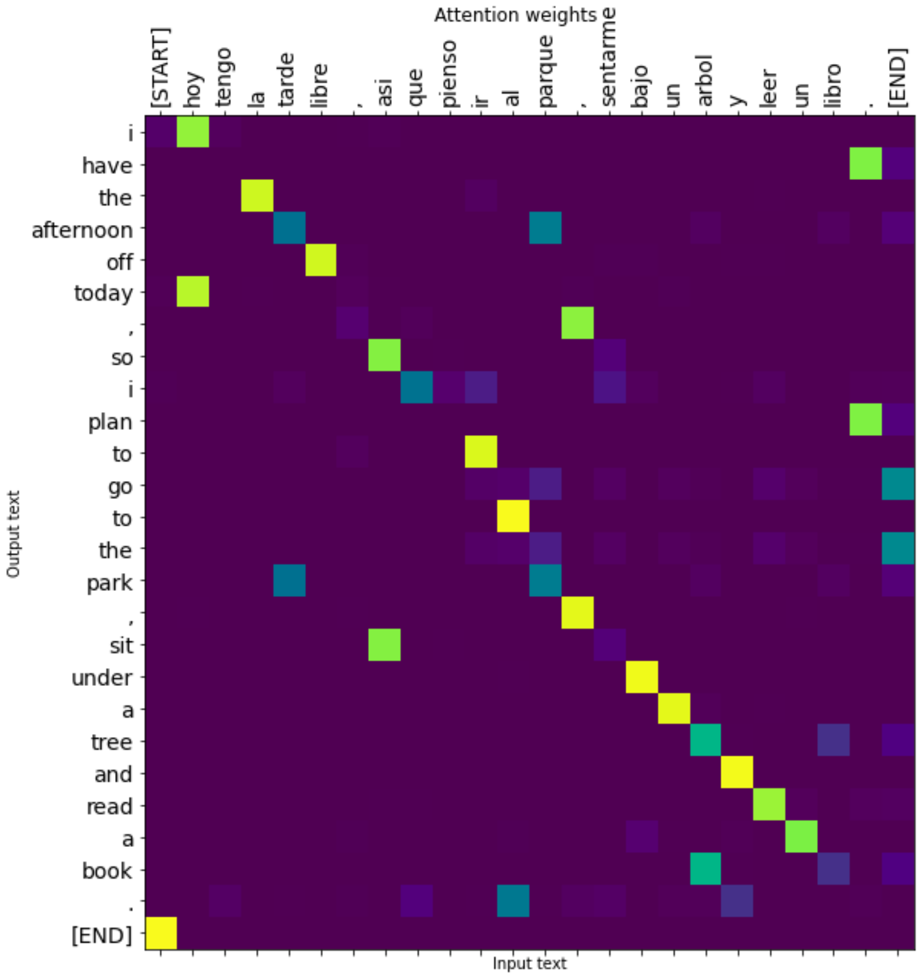


Image Source : https://towardsdatascience.com/end-to-end-attention-based-machine-translation-model-with-minimum-tensorflow-code-ae2f08cc8218

Attention Mechanism

- Distinguishing Feature
  - Does not attempt to encode a whole input sentence into a single fixed-length vector

Attention Mechanism

- Distinguishing Feature
  - Does not attempt to encode a whole input sentence into a single fixed-length vector
  - Encodes the input sentence into a sequence of vectors and chooses a subset of these vectors adaptively while decoding the translation

- The improvement is more apparent with longer sentences

Background : Neural Machine Translation

- NMT : Probabilistic Perspective
  - Finding a target sentence $\boldsymbol{y}$ that maximizes the cond. prob. of $\boldsymbol{y}$, given a source sentence x
  - $\mathrm{argmax}_y \, p(y|x)$

Background : Neural Machine Translation

- NMT : Probabilistic Perspective
  - Finding a target sentence $\boldsymbol{y}$ that maximizes the cond. prob. of $\boldsymbol{y}$, given a source sentence x
  - $\mathrm{argmax}_y\, p(y|x)$
  - Given a source sentence, a corresponding translation can be generated by searching for the sentence that maximizes the conditional probability

Background : RNN Enc-Dec

- Encoder : reads the input sentence, a seq. of vectors $\mathbb{x} = (x_1, \dots, x_{T_x})$, into a vector c.

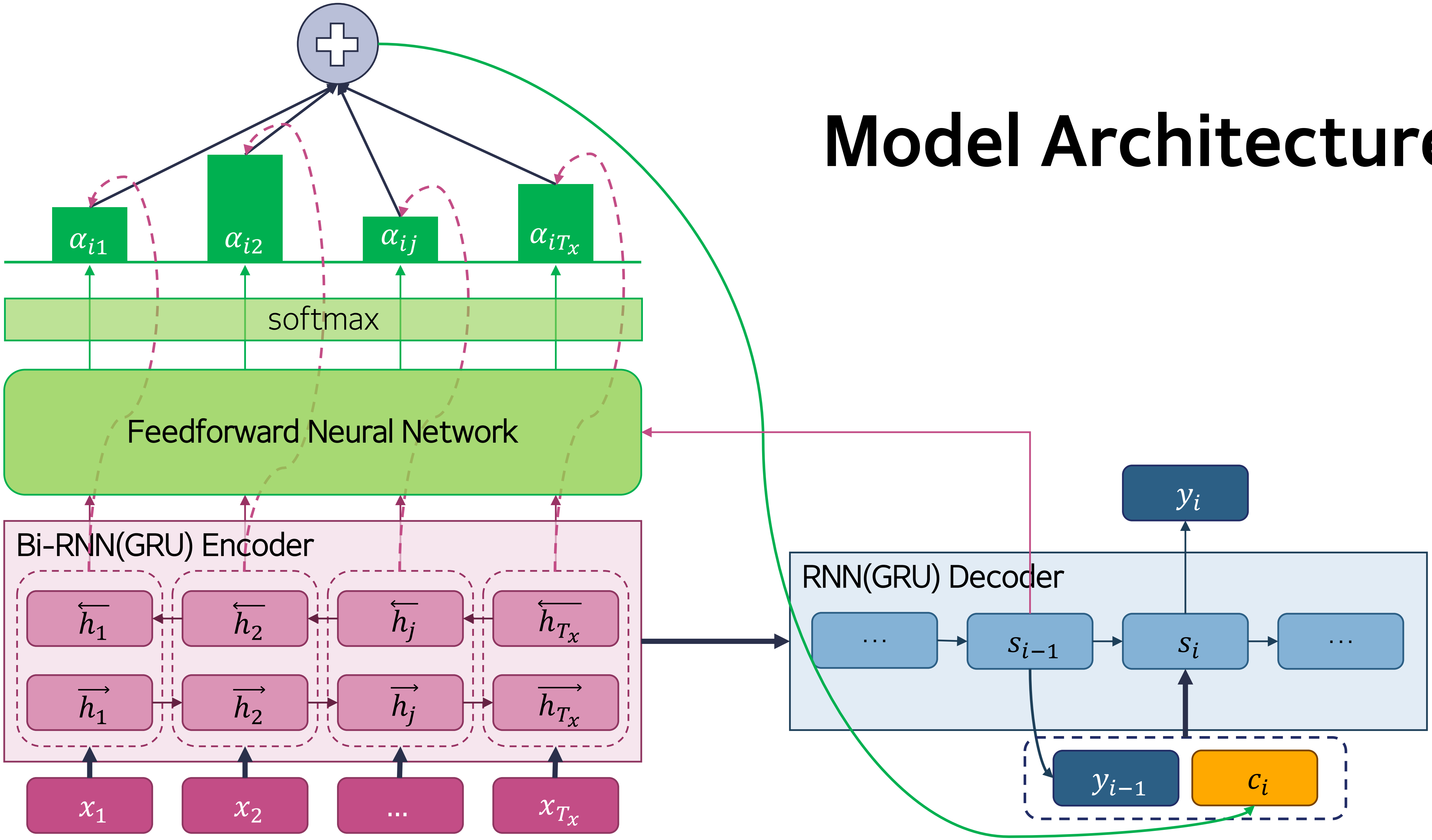
Background : RNN Enc-Dec

- Encoder : reads the input sentence, a seq. of vectors $\mathbb{x} = (x_1, \ldots, x_{T_x})$, into a vector c.
- Decoder : trained to predict the next word $y_{t'}$ given the context vector c, and all the previously predicted words $\{y_1, \ldots, y_{t'-1}\}$.

Background : RNN Enc-Dec

- Encoder : reads the input sentence, a seq. of vectors $\mathbb{x} = (x_1, \dots, x_{T_x})$, into a vector c.
- Decoder : trained to predict the next word $y_{t'}$ given the context vector c, and all the previously predicted words $\{y_1, \dots, y_{t'-1}\}$.
- $p(\mathbb{y}) = \prod_{t=1}^{N} p(y_t | \{y_1, \dots, y_{t'-1}\}, c)$, where
  $p(y_t | \{y_1, \dots, y_{t'-1}\}, c) = g(y_{t-1}, s_t, c)$,
  s_t be the hidden state of the RNN

Model Architecture

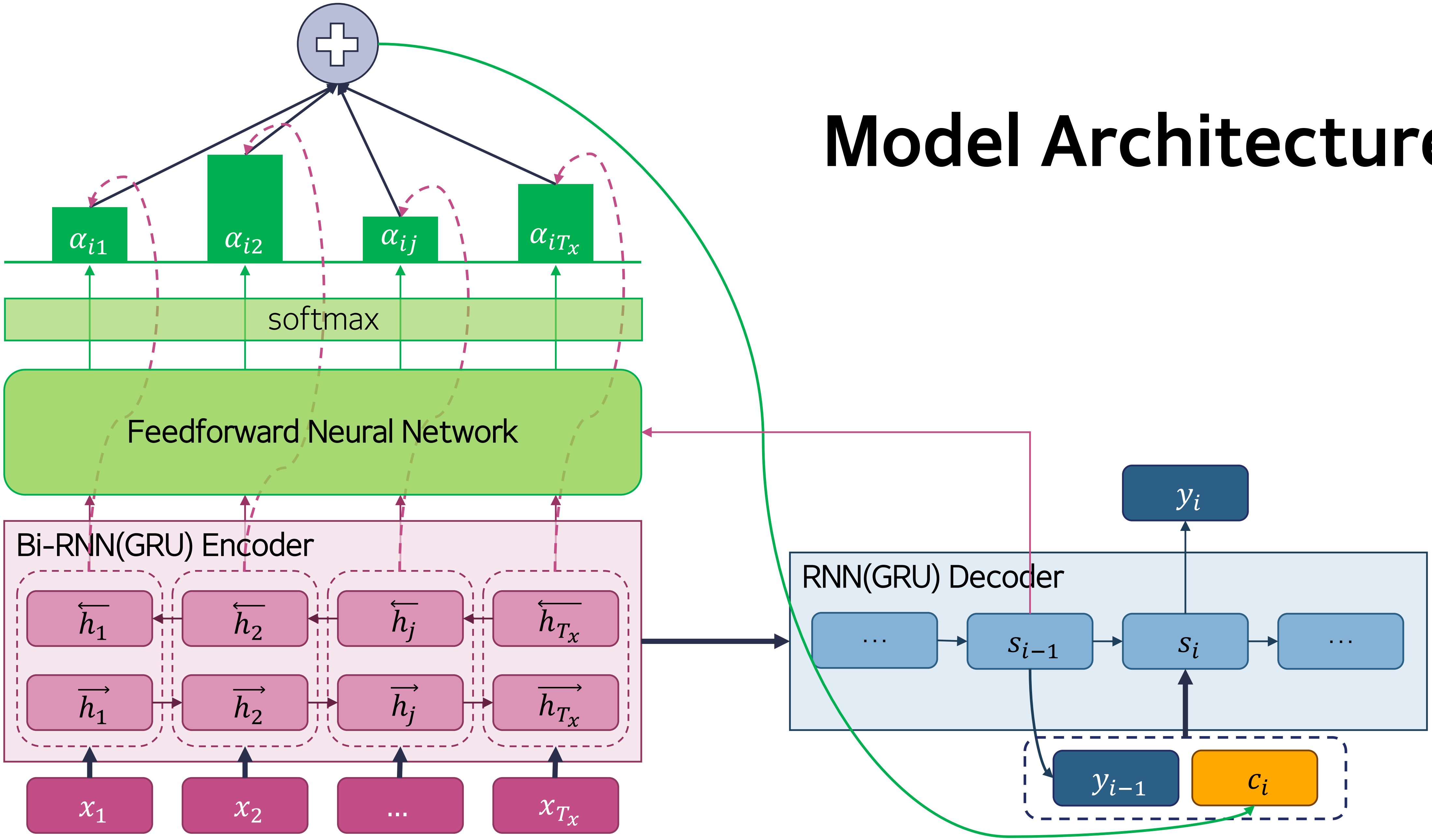


Decoder with Attention

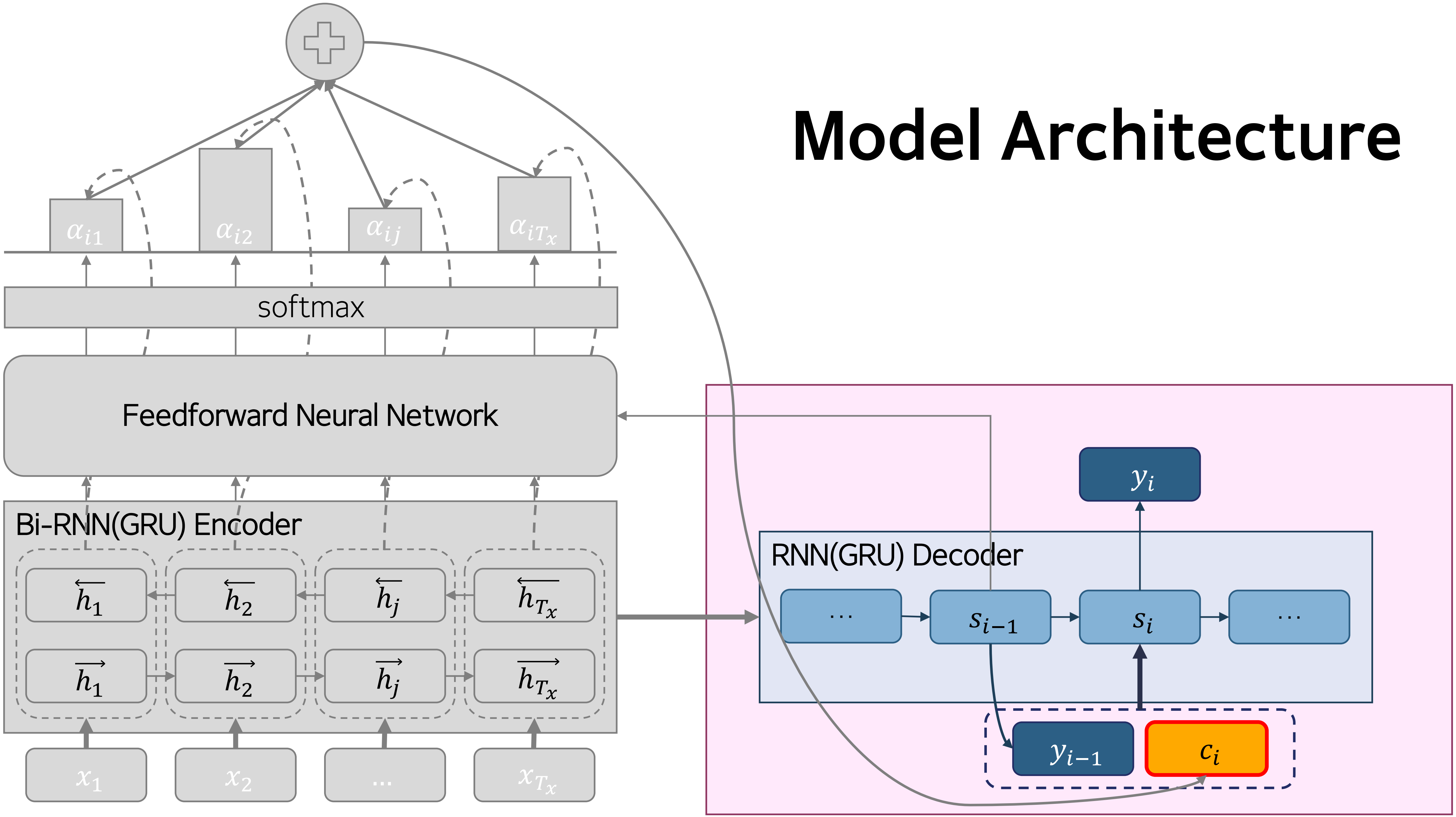
- New cond. prob. In new model:
  $p(y_i|\{y_1, \dots, y_{i-1}\}, \mathbb{x}) = g(y_{i-1}, s_i, \boldsymbol{c_i})$, where $s_i = f(s_{i-1}, y_{i-1}, c_i)$

- The probability is conditioned on a distinct context vector $\boldsymbol{c_i}$ for each target word y_i.

Model Architecture



Model Architecture



Constructing context vector c_i

- Encoder maps the input sentence $\mathbb{x}$ to a sequence of *annotations*

$$\left(h_1, \dots, h_{T_x}\right) = \text{hidden states of RNN Encoder}$$

  and context vector c_i depends on the sequence.

Constructing context vector $\boldsymbol{c_i}$

- Encoder maps the input sentence $\mathbb{x}$ to a sequence of *annotations*
$$(h_1, \dots, h_{T_x}) = \textbf{hidden states of RNN Encoder}$$
and **context vector** $\boldsymbol{c_i}$ depends on the sequence.

- Each annotation $\boldsymbol{h_i}$ contains information about the whole input sequence with a strong focus on the parts surrounding the $\boldsymbol{i}$-th word of the input sequence.

Constructing context vector c_i

- The *context vector* c_i is then computed as a weighted sum of these annotations h_i:

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

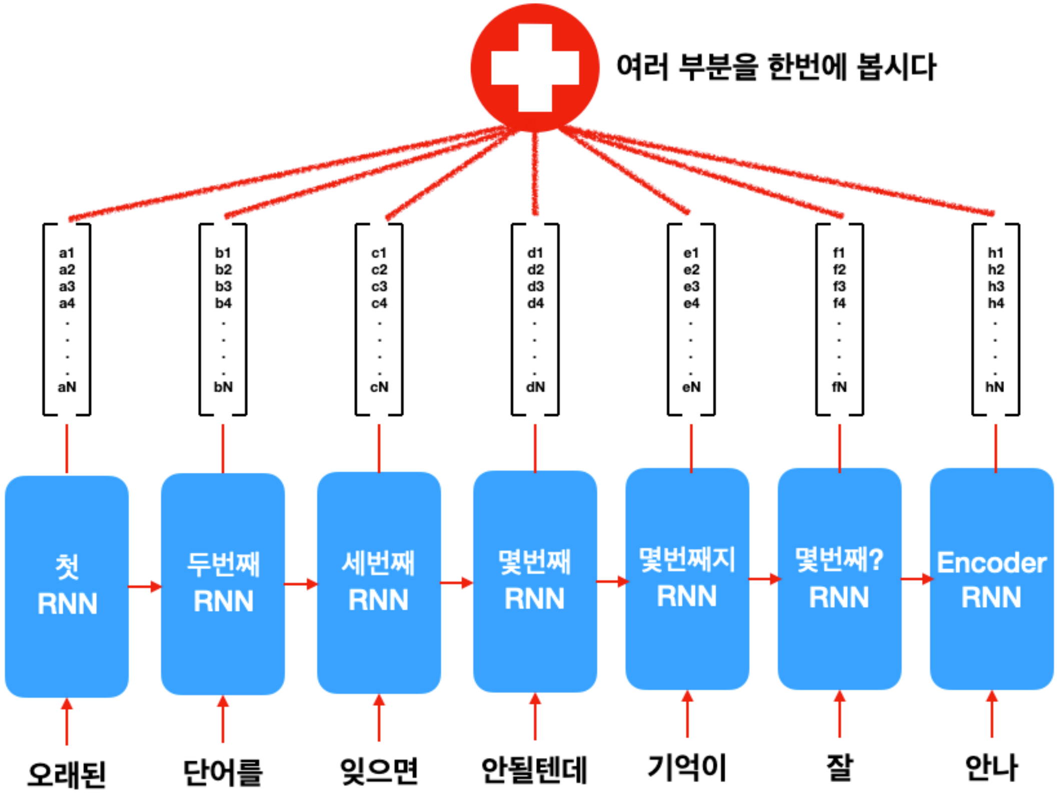
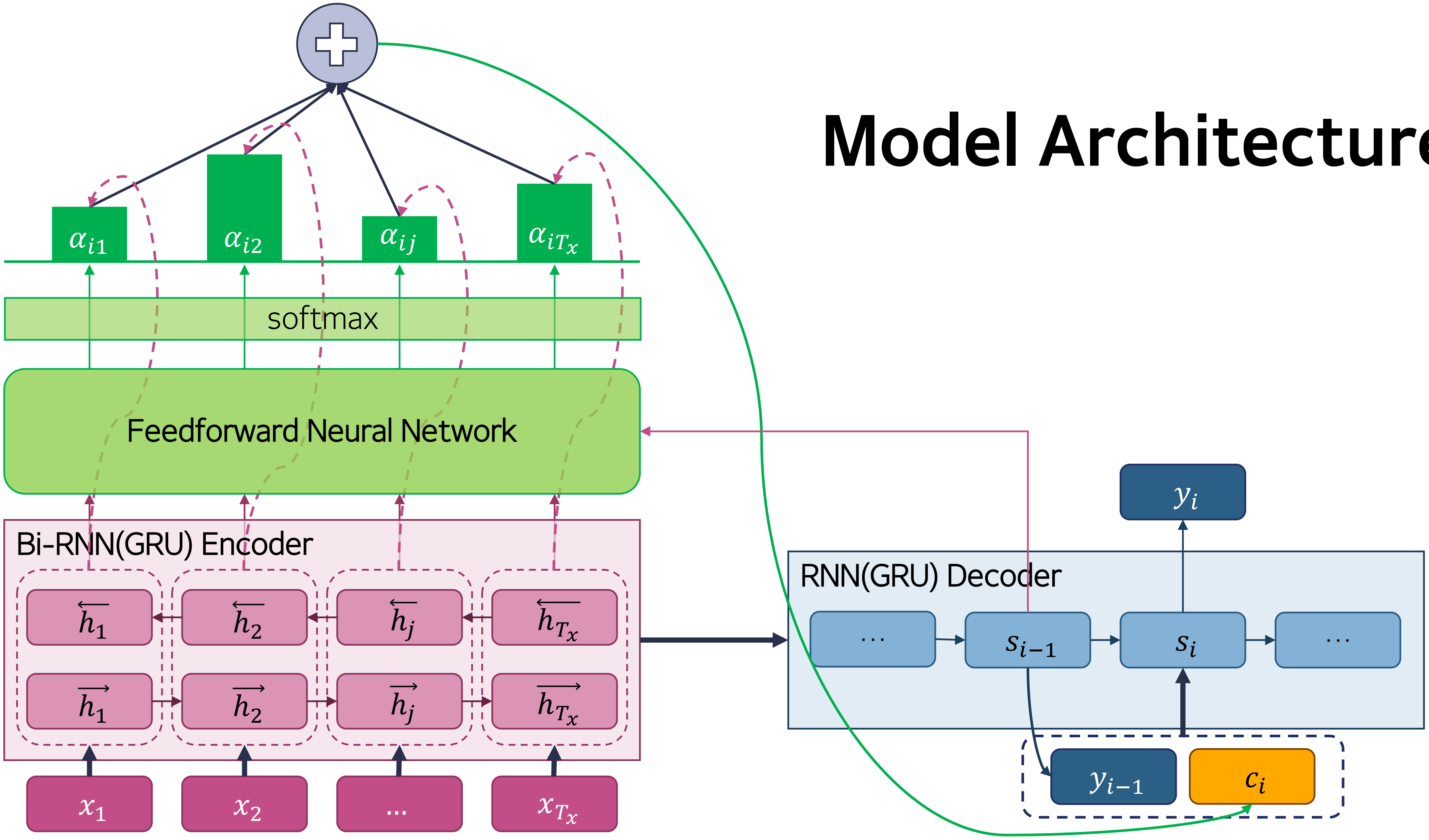
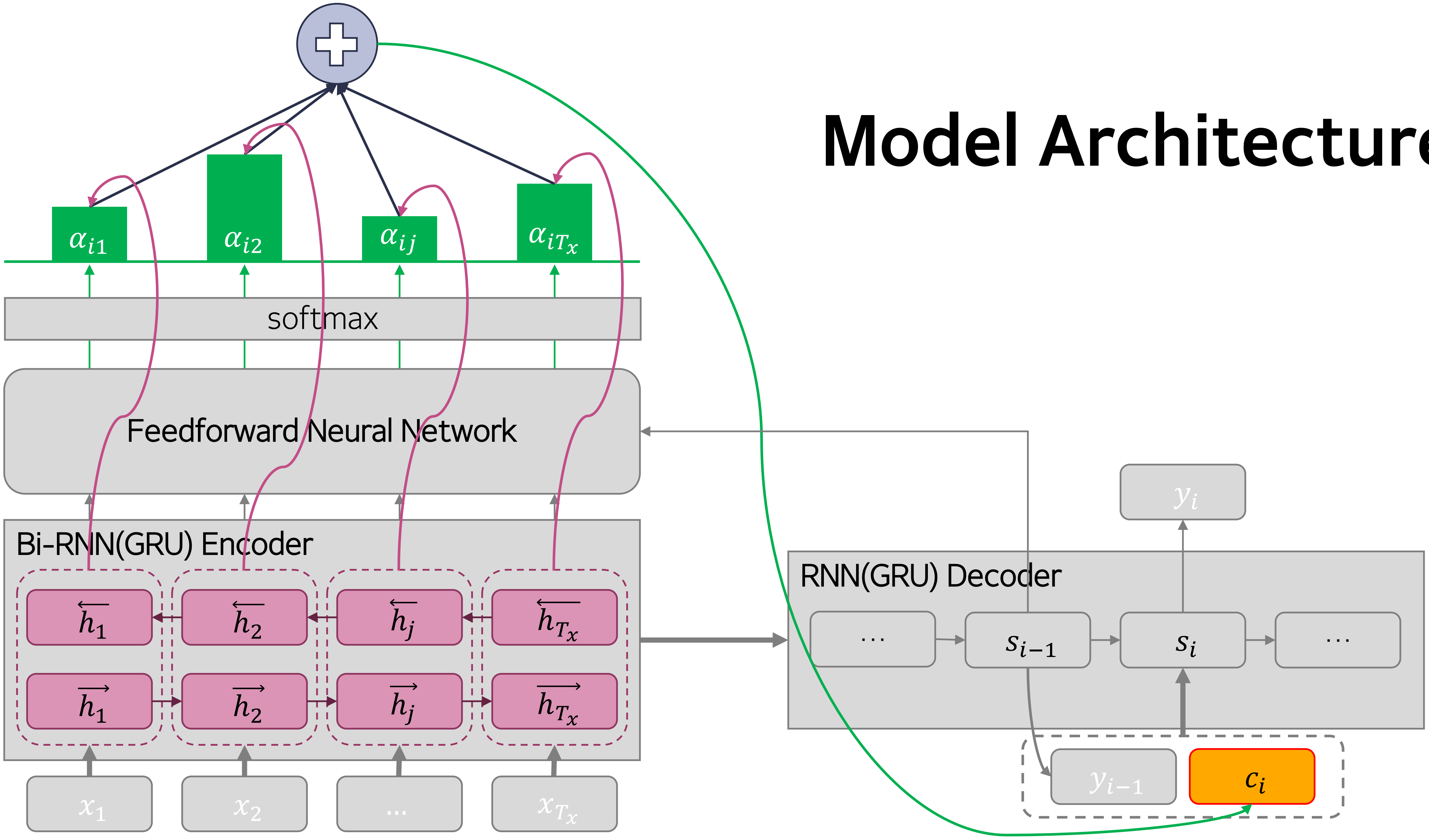


Image Source : https://jiho-ml.com/weekly-nlp-23/

Model Architecture



Model Architecture



Constructing context vector c_i

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

- $\alpha_{ij} = {\exp(e_{ij})} \Big/ {\sum_{k=1}^{T_x} \exp(e_{ik})}$, where

  $e_{ij} = a(s_{i-1}, h_j)$

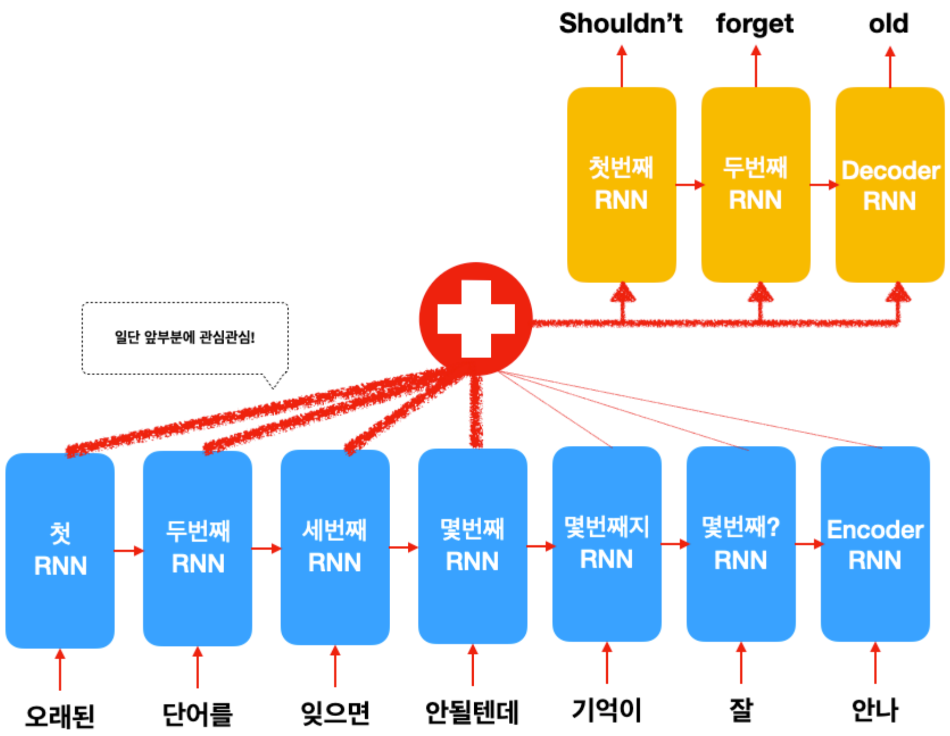


Image Source : https://jiho-ml.com/weekly-nlp-23/

Constructing context vector c_i

- $c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$
- $\alpha_{ij} = \exp(e_{ij}) \big/ \sum_{k=1}^{T_x} \exp(e_{ik})$, where
  $e_{ij} = a(s_{i-1}, h_j)$
- An energy e_{ij} :
  - scores how well
    the inputs around position $j\,(h_j)$ and
    the output at position $i(s_{i-1})$ match.

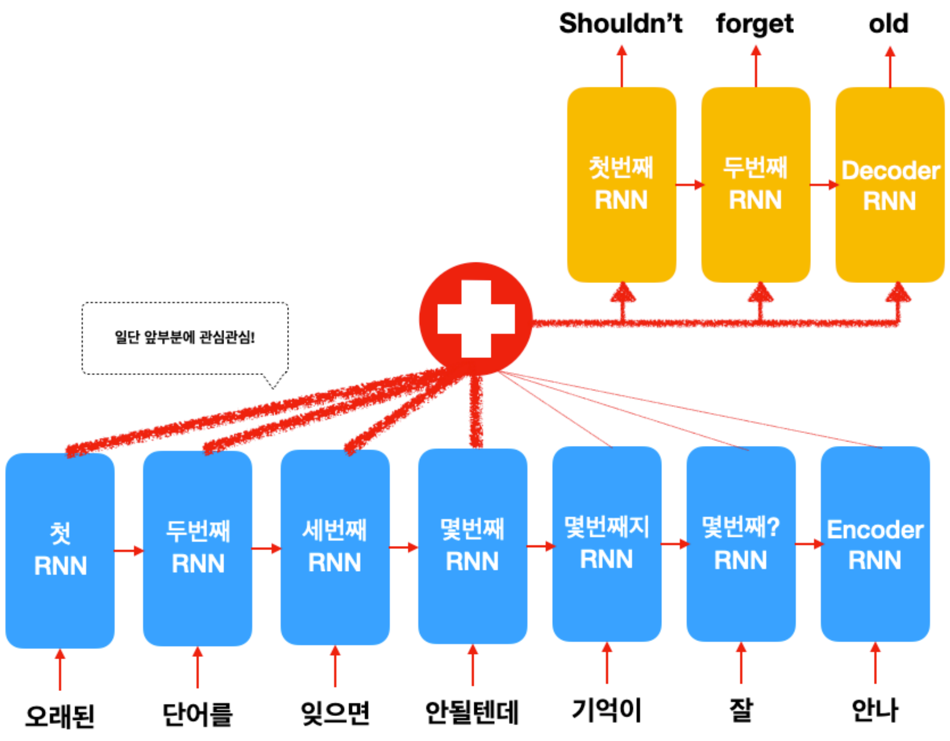
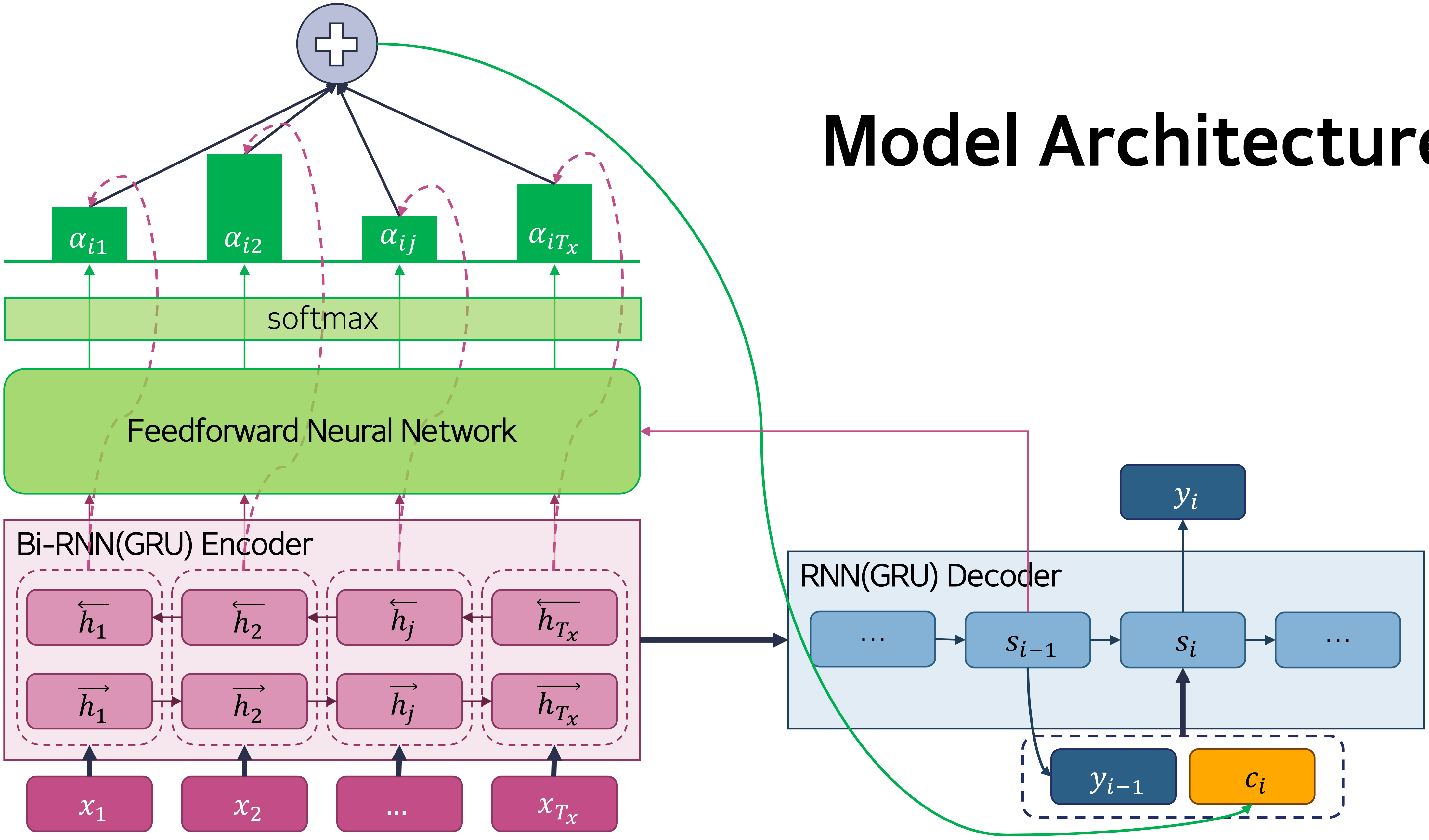
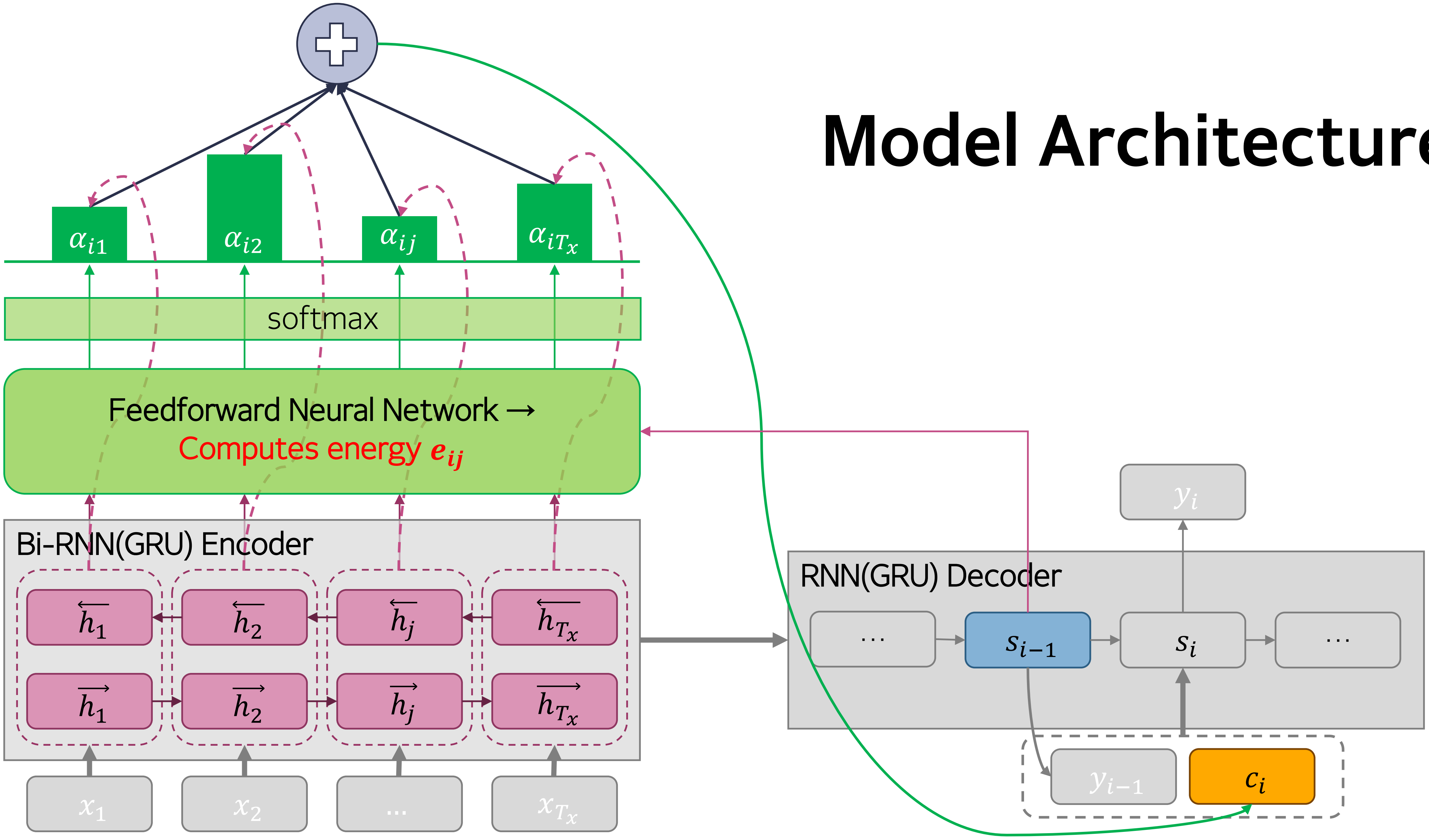


Image Source : https://jiho-ml.com/weekly-nlp-23/

Model Architecture



Model Architecture



Decoder with Attention

- Summary
  - context vector c_i : expected annotation

Decoder with Attention

- Summary
  - context vector $\boldsymbol{c_i}$: expected annotation
  - Weight $\boldsymbol{\alpha_{ij}}$: prob. that the target word y_i is aligned to, or translated from, a source word x_j

Decoder with Attention

- Summary
  - context vector $\boldsymbol{c_i}$: expected annotation
  - Weight $\boldsymbol{\alpha_{ij}}$: prob. that the target word y_i is aligned to, or translated from, a source word x_j
  - α_{ij} reflects the importance of the annotation $\boldsymbol{h_j}$ w.r.t. the previous hidden state $\boldsymbol{s_{i-1}}$ in deciding the next state $\boldsymbol{s_i}$ and generating $\boldsymbol{y_i}$

Decoder with Attention

- Summary
  - This implements a *mechanism of attention of decoder*
  - Decoder decides parts of the source sentence to pay attention to

Decoder with Attention

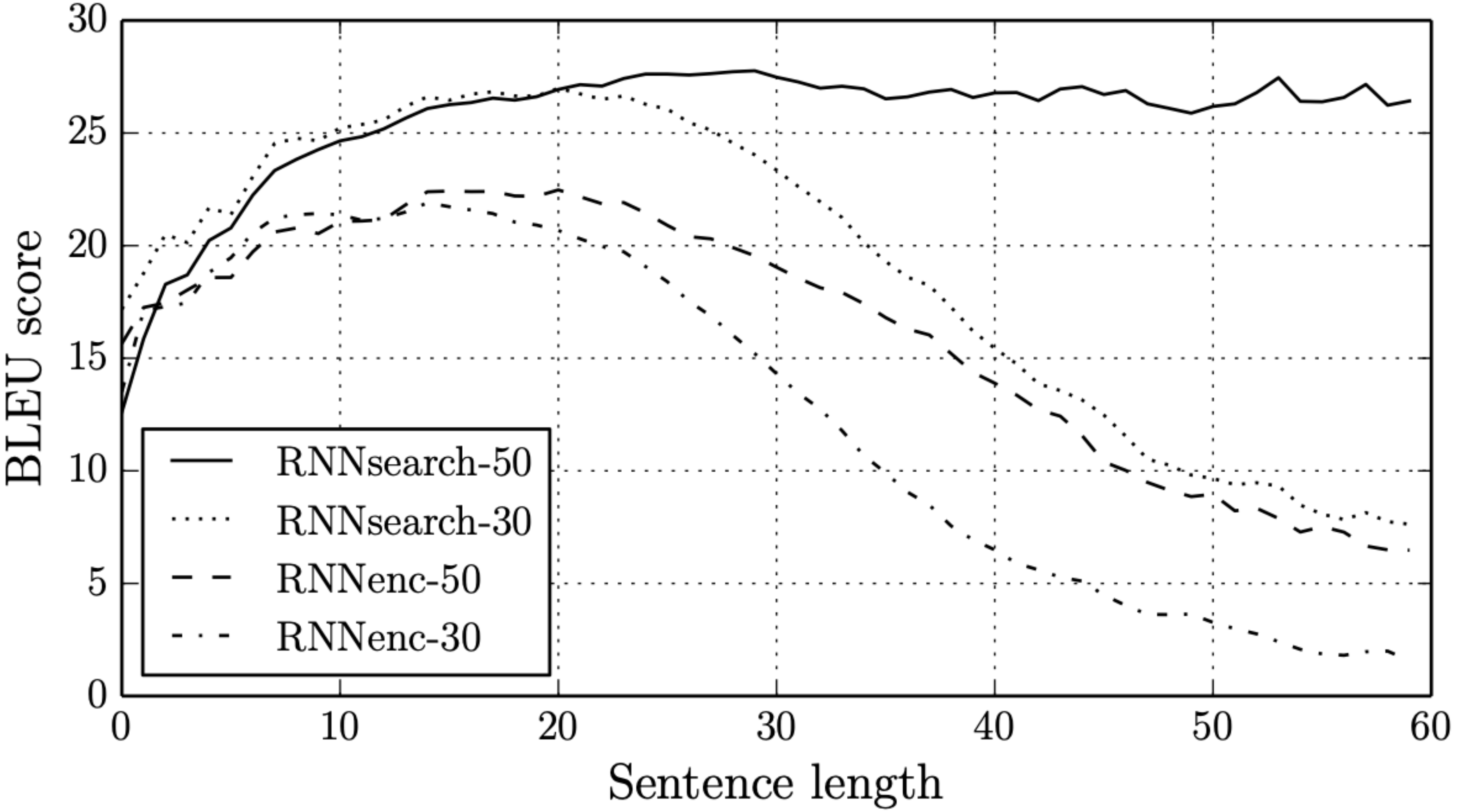
- Summary
  - This implements a *mechanism of attention of decoder*
  - **Decoder decides parts of the source sentence** to pay attention to

- The information can be spread throughout the sequence of annotations, which can be selectively retrieved by the decoder accordingly

Encoder : Bi-RNN for Annotating

- Obtain an annotation for each word $\boldsymbol{x_j}$
  by concatenating $\overrightarrow{\boldsymbol{h_j}}$, $\overleftarrow{\boldsymbol{h_j}}$, i.e., $\boldsymbol{h_j} = \left[\overrightarrow{\boldsymbol{h_j}}^{\boldsymbol{T}};\ \overleftarrow{\boldsymbol{h_j}}^{\boldsymbol{T}}\right]$
- The annotation h_j
  - contains the summaries of
    both the preceding words and the following words
  - will be focused on the words around $\boldsymbol{x_j}$

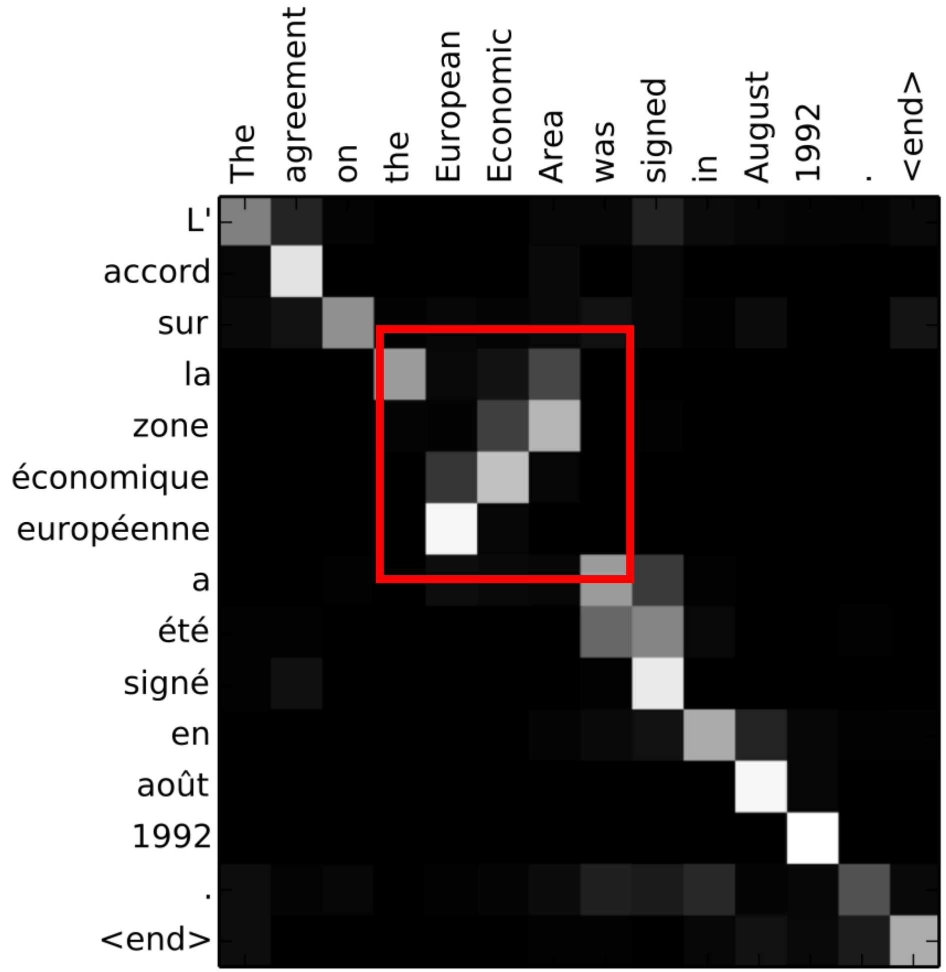
Experiments & Results



Experiments & Results

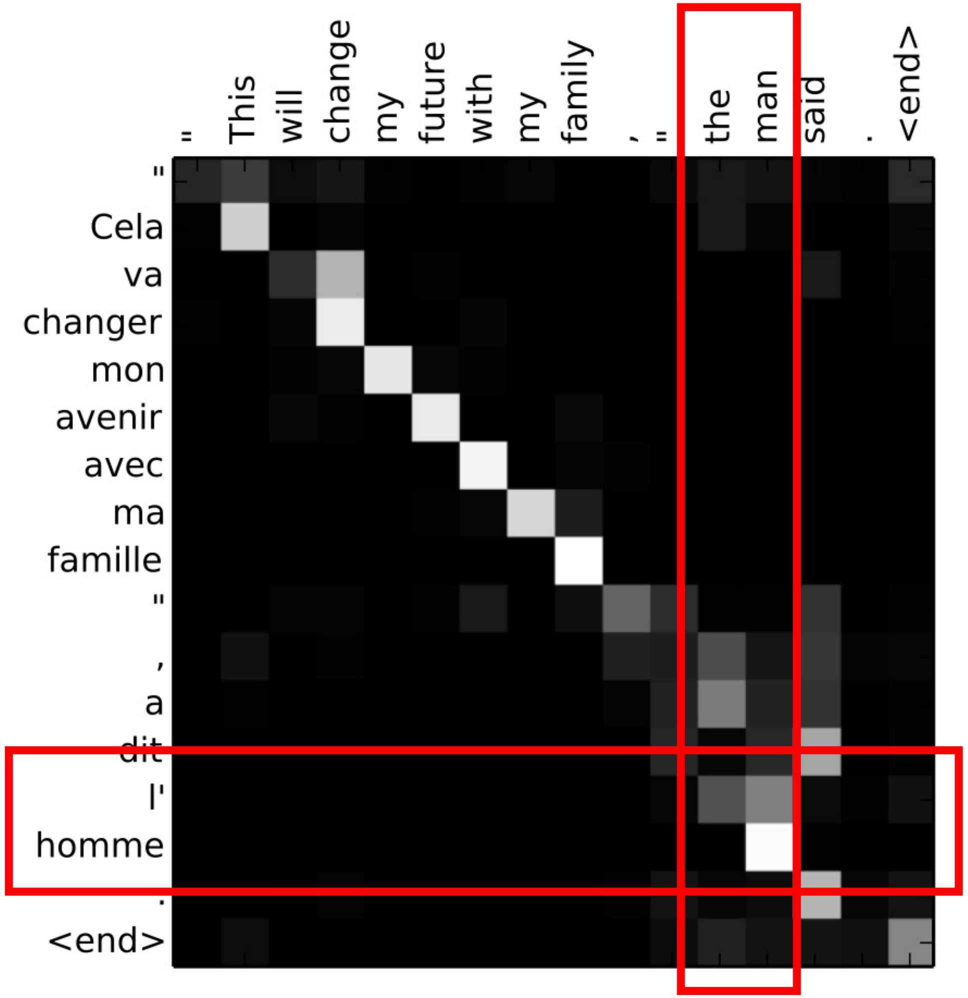
- English-French Translation
- “European economic Area”
- “Zone économique européenne”

- Soft-Alignment:
  Able to correctly align words



Experiments & Results

- English *"the"*, in French, correspo nds to one of *"le", "la", "les", or "l'"*
- Hard-alignment : hard to choose which is correct
- Soft-alignment : solves this issue by look both "the" and "man"
- Able to find a linguistically plausible alignment



Q&A