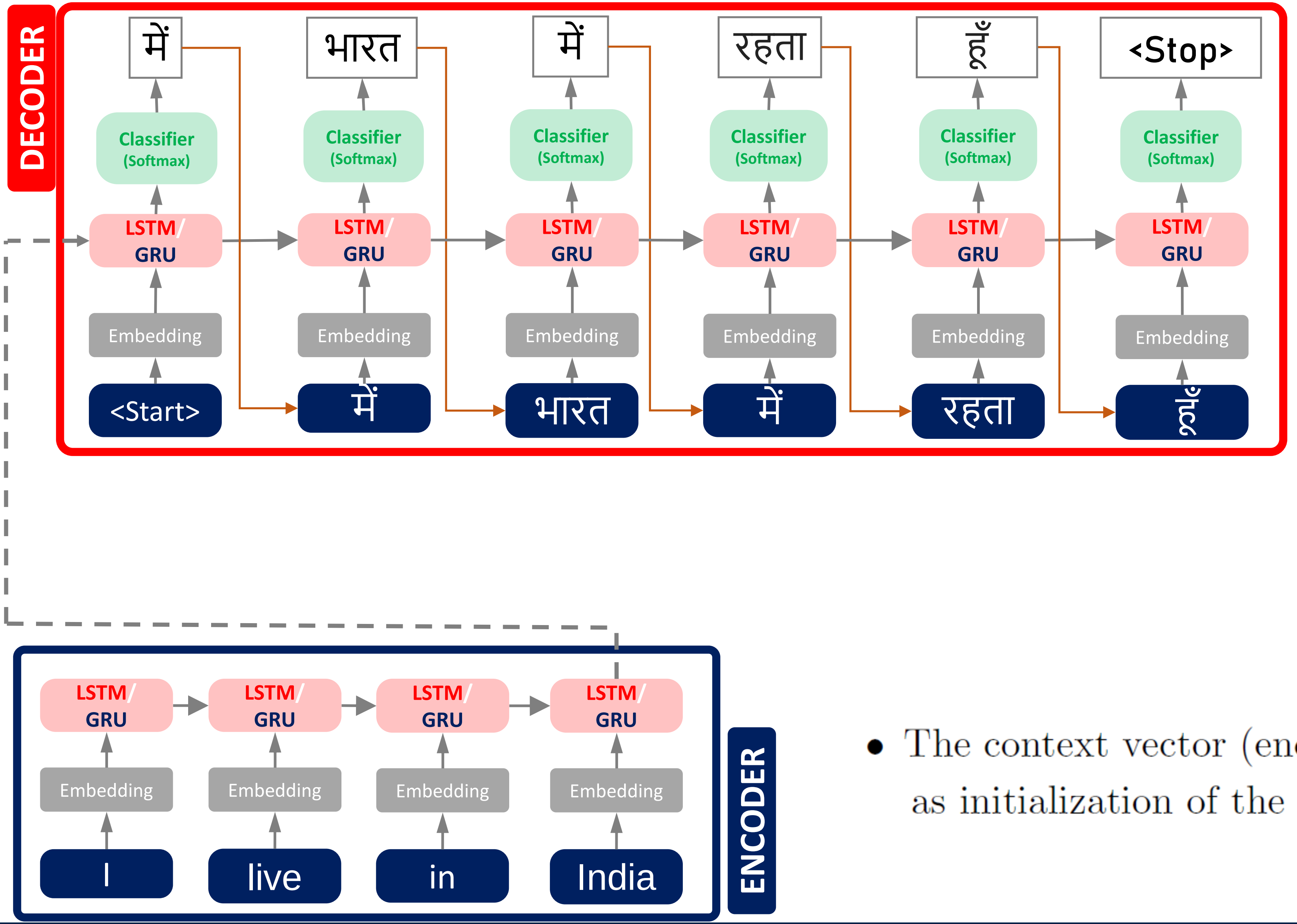
Attention Mechanism

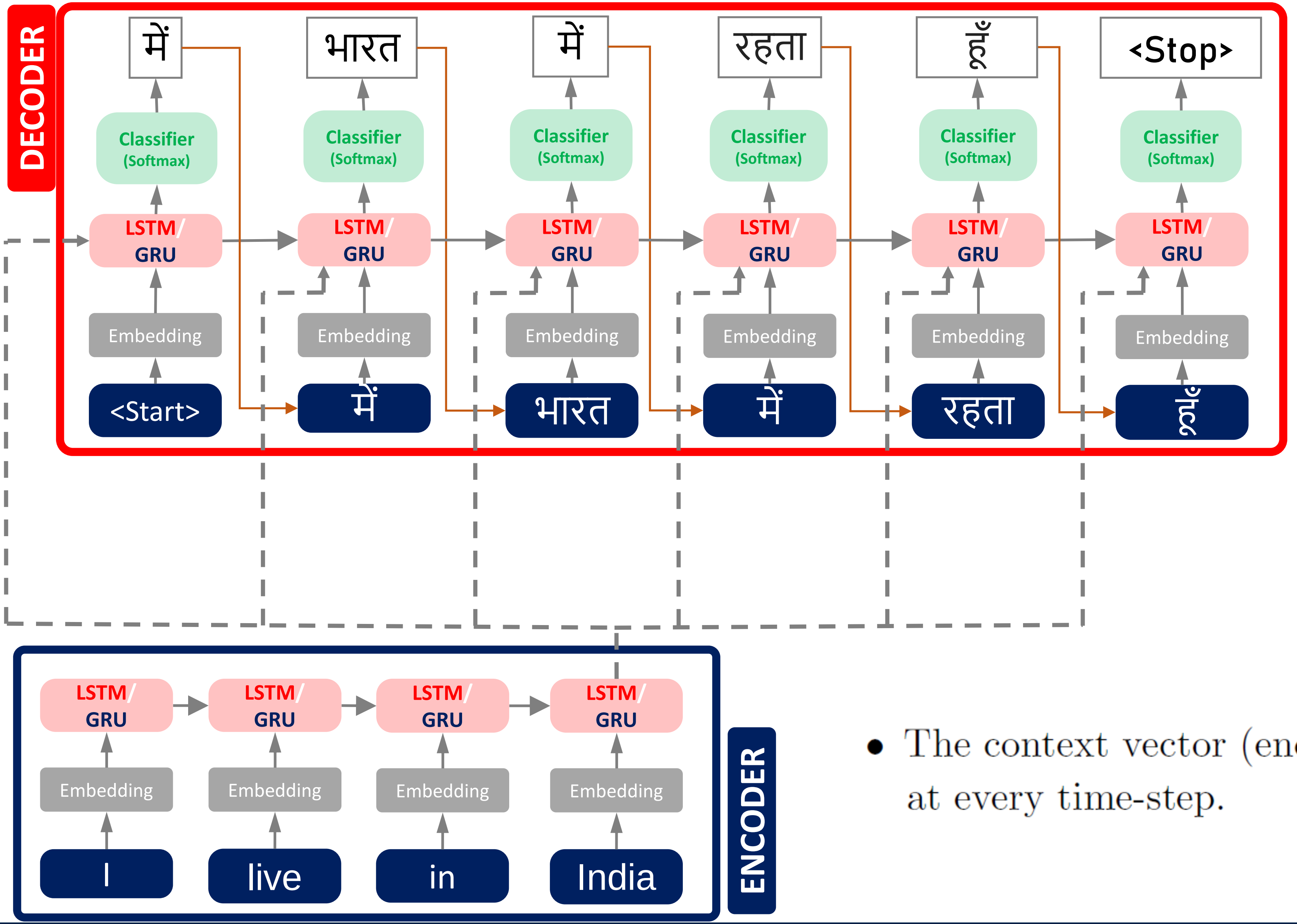
Dripta Mj
Department of Mathematics
Ramakrishna Mission Vivekananda Educational and Research Institute
Belur Math, India

Translation: Review



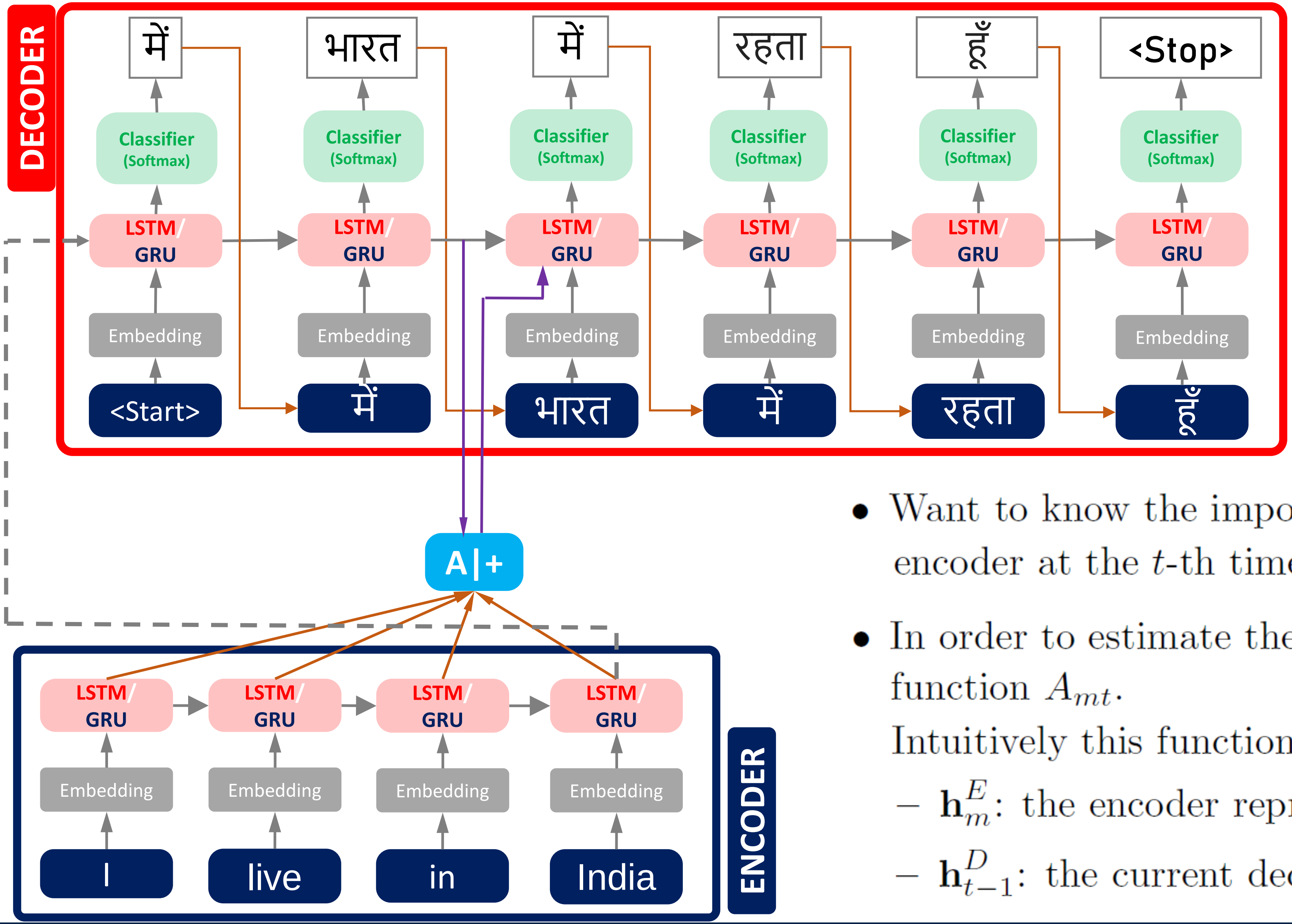
- The context vector (encoding) can be passed to the decoder as initialization of the hidden state (of decoder).

Translation: Review



- The context vector (encoding) can be passed to the decoder at every time-step.

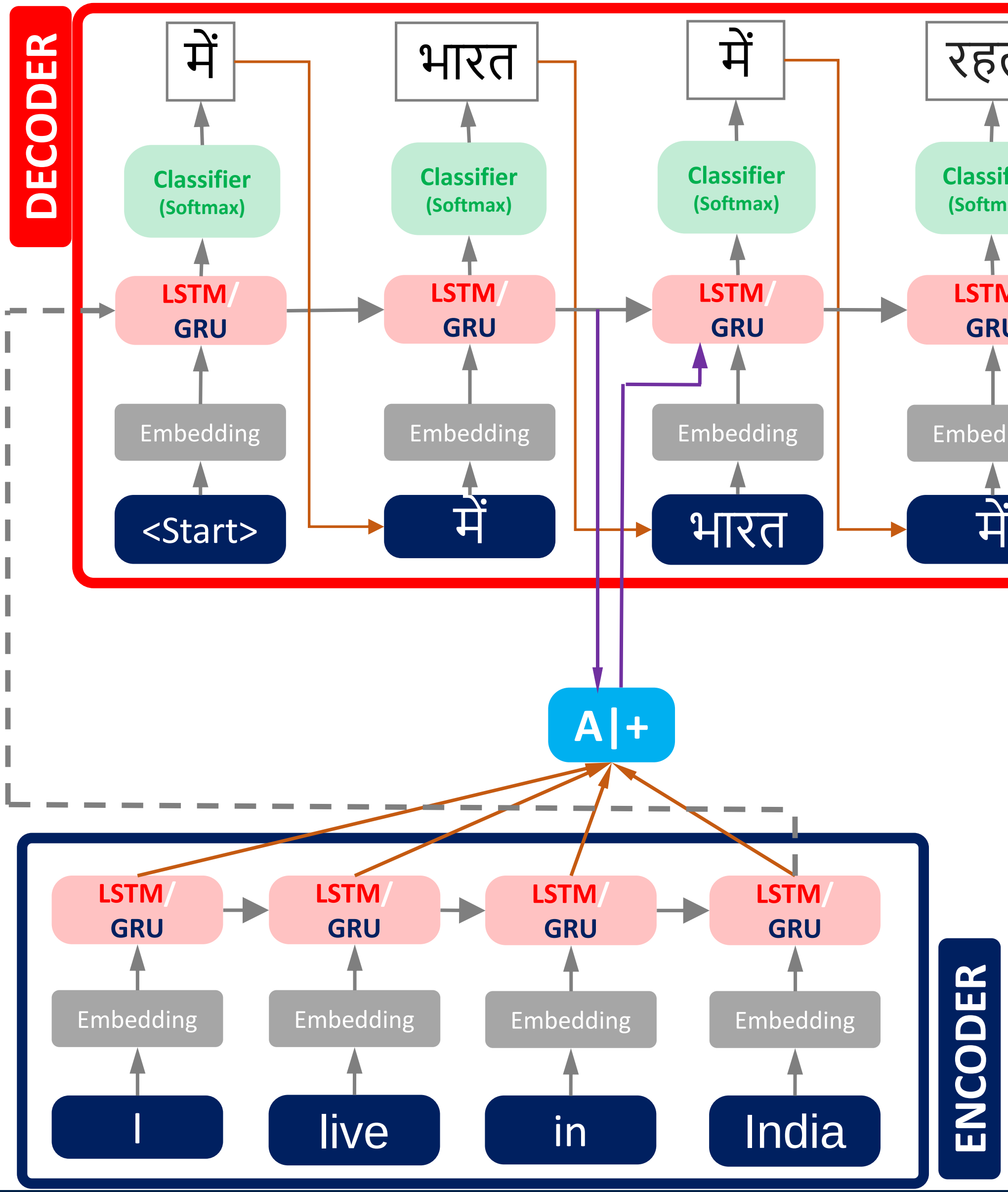
Translation with Attention



- Want to know the importance of the m-th word of the encoder at the t-th time-step of the decoder: β_{mt}.
- In order to estimate the importance, we define a score function A_{mt}.

  Intuitively this function can be taken to depend upon
  - $\mathbf{h}_m^E$: the encoder representation of the m-th word
  - $\mathbf{h}_{t-1}^D$: the current decoder state

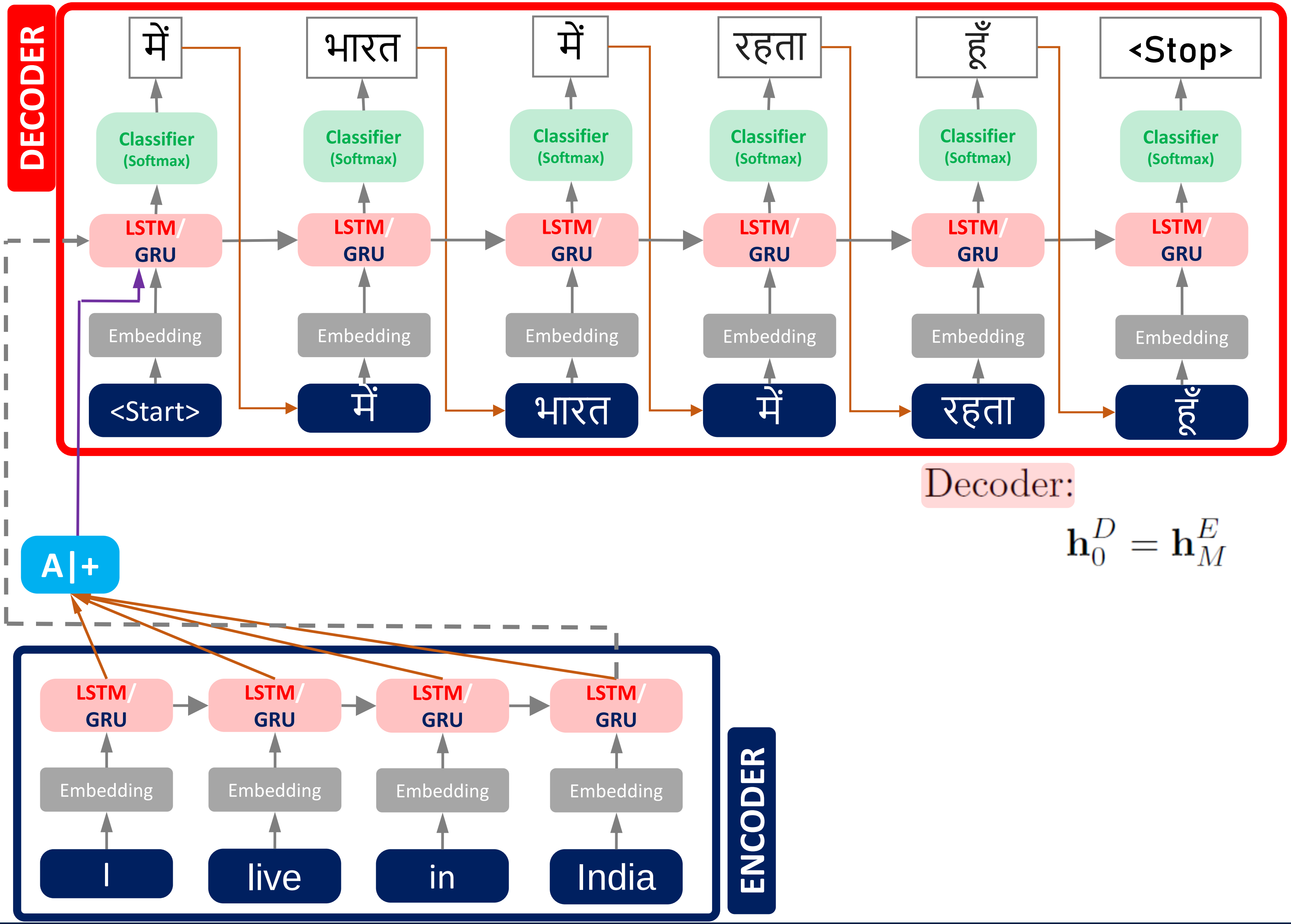
Translation with Attention



- Want to know the importance of the m-th word of the encoder at the t-th time-step of the decoder: β_{mt}.
- In order to estimate the importance, we define a score function A_{mt}.
  Mathematically

$$A_{mt} = f(\mathbf{h}_m^E, \mathbf{h}_{t-1}^D)$$

Translation with Attention



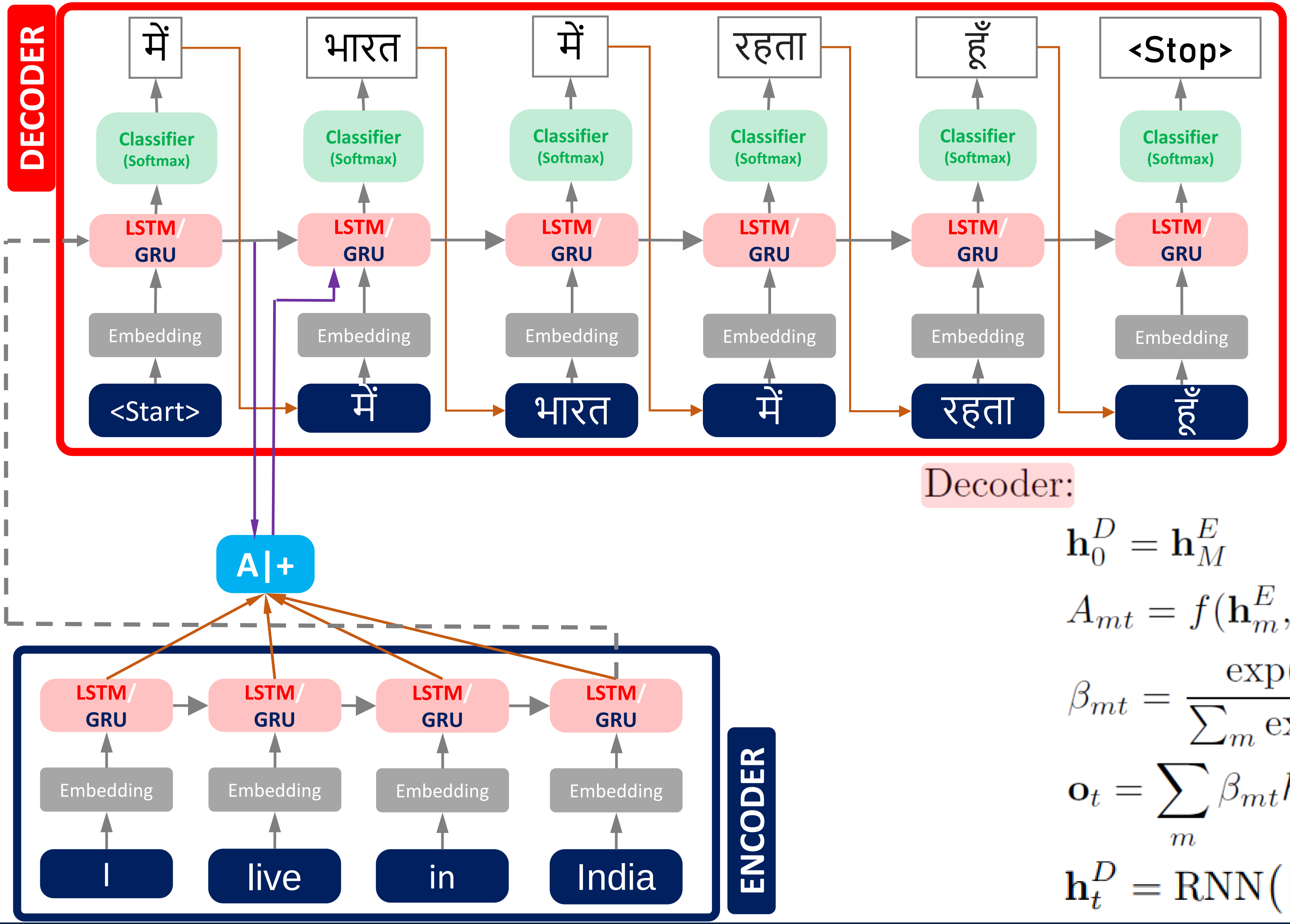
Encoder:

$$\mathbf{h}_m^E = \mathrm{RNN}\left(\mathrm{Embed}(\mathbf{x}_m), \mathbf{h}_{m-1}^E\right)$$

Decoder:

$$\mathbf{h}_0^D = \mathbf{h}_M^E$$

Translation with Attention



Encoder:

$$\mathbf{h}_m^E = \mathrm{RNN}\big(\mathrm{Embed}(\mathbf{x}_m), \mathbf{h}_{m-1}^E\big)$$

Decoder:

$$\mathbf{h}_0^D = \mathbf{h}_M^E$$

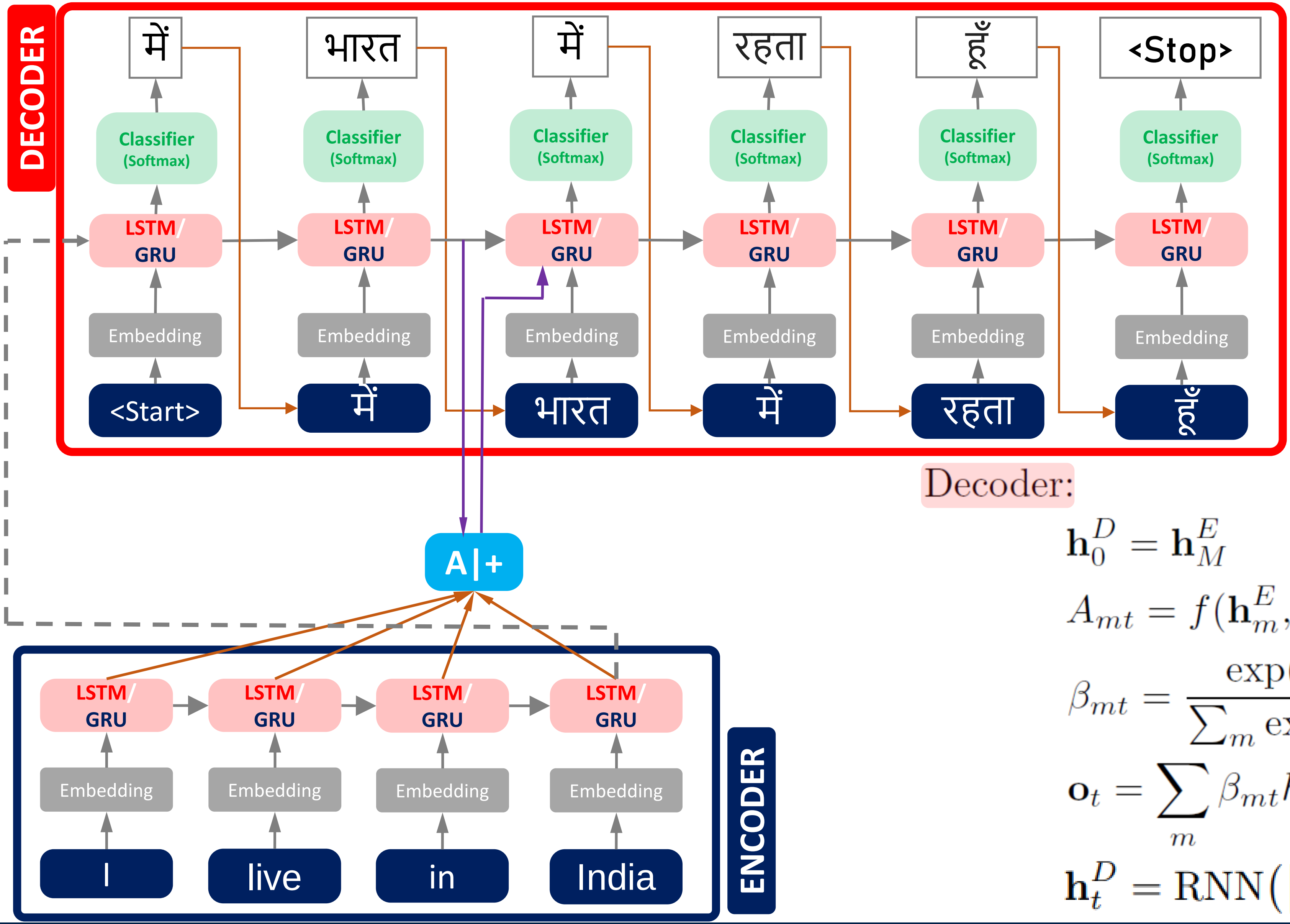
$$A_{mt} = f\big(\mathbf{h}_m^E, \mathbf{h}_{t-1}^D\big) \qquad \text{Score function}$$

$$\beta_{mt} = \frac{\exp(A_{mt})}{\sum_m \exp(A_{mt})} \qquad \text{Attention weights}$$

$$\mathbf{o}_t = \sum_m \beta_{mt} h_m^E \qquad \text{Attention layer outputs}$$

$$\mathbf{h}_t^D = \mathrm{RNN}\big(\big[\mathbf{o}_t, \mathrm{Embed}(y_{t-1}^*)\big], \mathbf{h}_{t-1}^D\big)$$

Translation with Attention



Encoder:

$$\mathbf{h}_m^E = \mathrm{RNN}\left(\mathrm{Embed}(\mathbf{x}_m), \mathbf{h}_{m-1}^E\right)$$

Decoder:

$$\mathbf{h}_0^D = \mathbf{h}_M^E$$

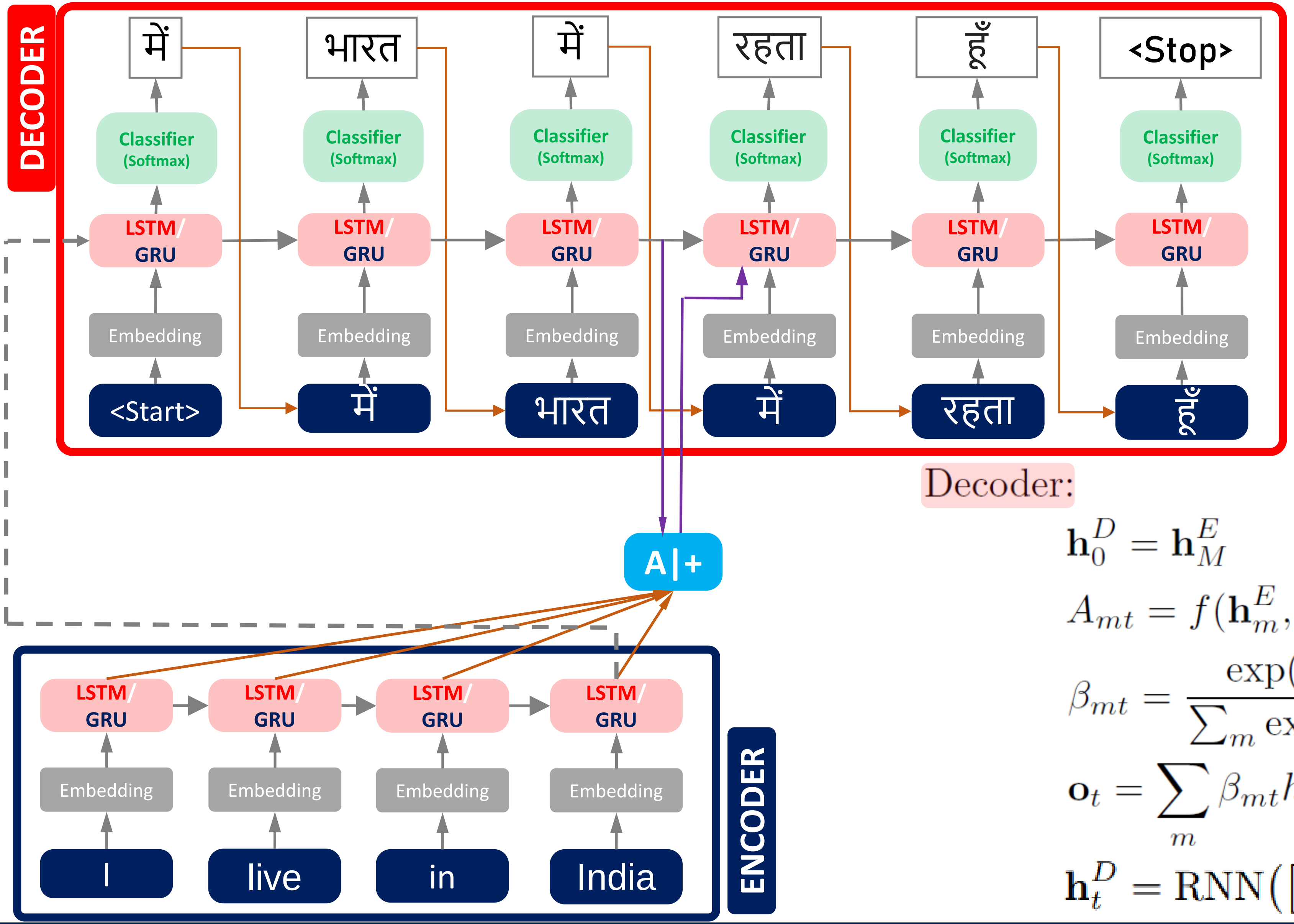
$$A_{mt} = f\left(\mathbf{h}_m^E, \mathbf{h}_{t-1}^D\right) \qquad \text{Score function}$$

$$\beta_{mt} = \frac{\exp(A_{mt})}{\sum_m \exp(A_{mt})} \qquad \text{Attention weights}$$

$$\mathbf{o}_t = \sum_m \beta_{mt} h_m^E \qquad \text{Attention layer outputs}$$

$$\mathbf{h}_t^D = \mathrm{RNN}\left(\left[\mathbf{o}_t, \mathrm{Embed}(y_{t-1}^*)\right], \mathbf{h}_{t-1}^D\right)$$

Translation with Attention



Encoder:

$$\mathbf{h}_m^E = \mathrm{RNN}\left(\mathrm{Embed}(\mathbf{x}_m), \mathbf{h}_{m-1}^E\right)$$

Decoder:

$$\mathbf{h}_0^D = \mathbf{h}_M^E$$

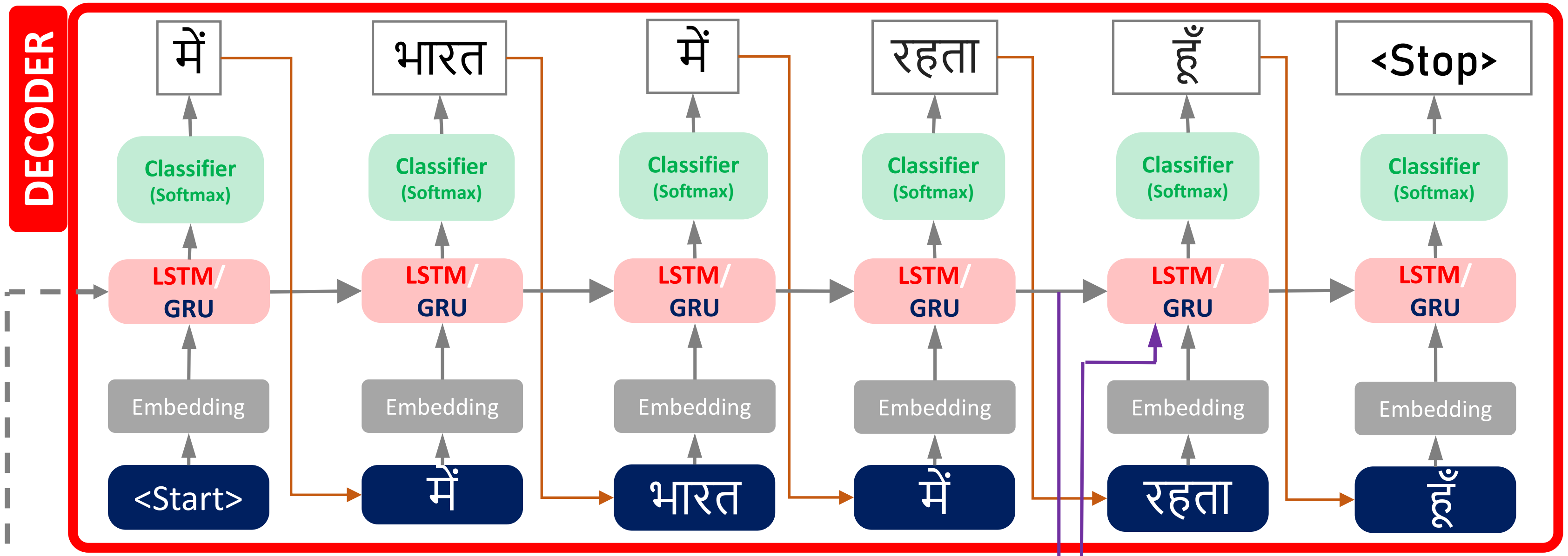
$$A_{mt} = f\left(\mathbf{h}_m^E, \mathbf{h}_{t-1}^D\right) \qquad \text{Score function}$$

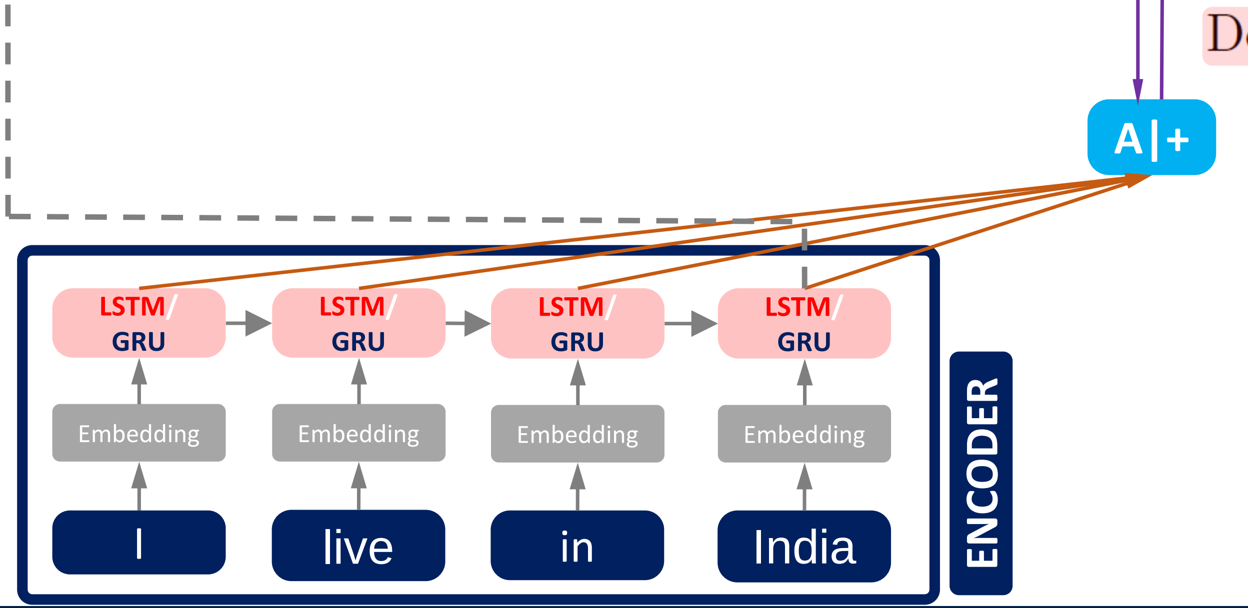
$$\beta_{mt} = \frac{\exp(A_{mt})}{\sum_m \exp(A_{mt})} \qquad \text{Attention weights}$$

$$\mathbf{o}_t = \sum_m \beta_{mt} h_m^E \qquad \text{Attention layer outputs}$$

$$\mathbf{h}_t^D = \mathrm{RNN}\left(\left[\mathbf{o}_t, \mathrm{Embed}(y_{t-1}^*)\right], \mathbf{h}_{t-1}^D\right)$$

Translation with Attention





Encoder:

$$\mathbf{h}_m^E = \mathrm{RNN}\big(\mathrm{Embed}(\mathbf{x}_m), \mathbf{h}_{m-1}^E\big)$$

Decoder:

$$\mathbf{h}_0^D = \mathbf{h}_M^E$$

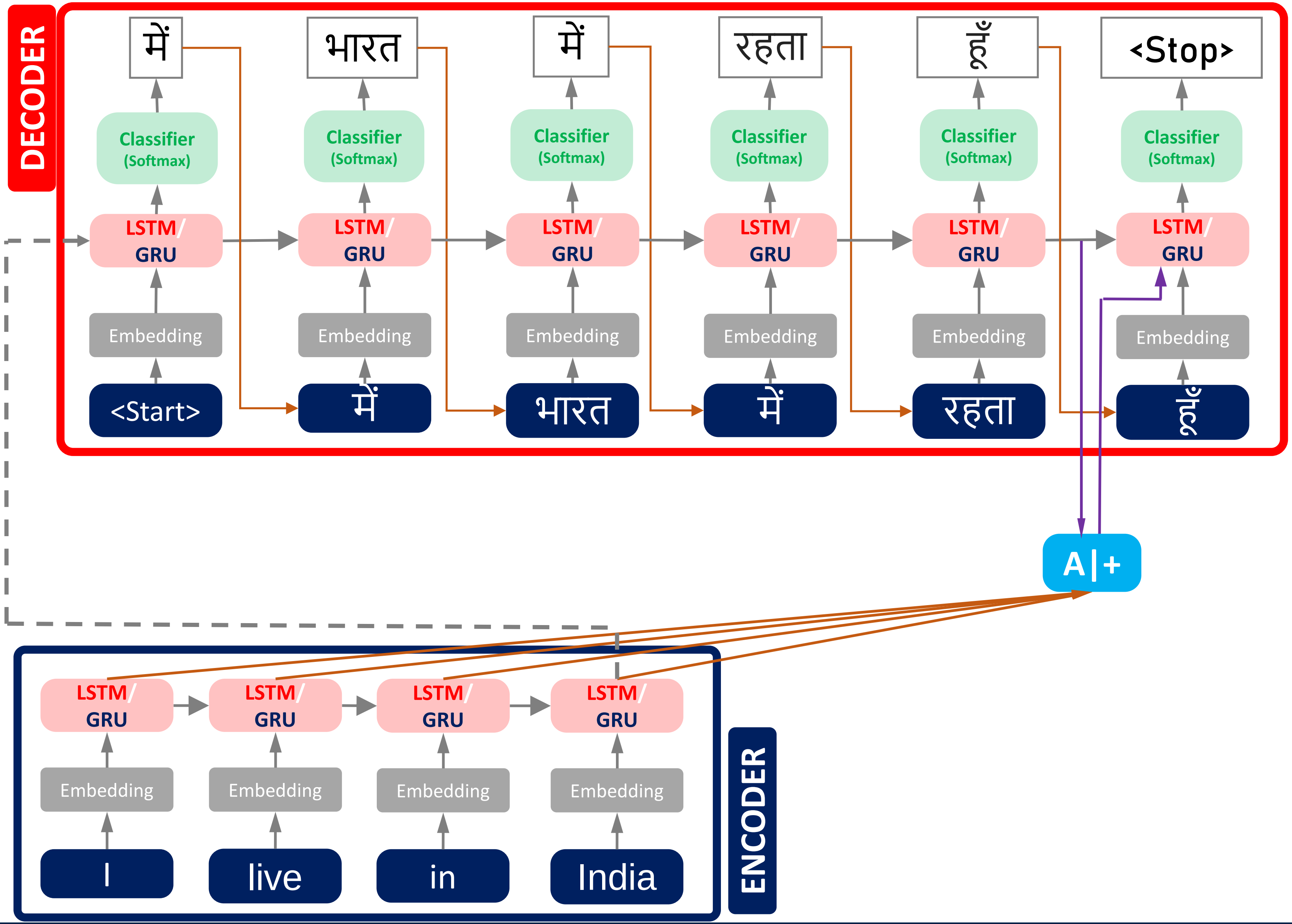
$$A_{mt} = f\big(\mathbf{h}_m^E, \mathbf{h}_{t-1}^D\big)$$ Score function

$$\beta_{mt} = \frac{\exp(A_{mt})}{\sum_m \exp(A_{mt})}$$ Attention weights

$$\mathbf{o}_t = \sum_m \beta_{mt} h_m^E$$ Attention layer outputs

$$\mathbf{h}_t^D = \mathrm{RNN}\Big(\big[\mathbf{o}_t, \mathrm{Embed}(y_{t-1}^*)\big], \mathbf{h}_{t-1}^D\Big)$$

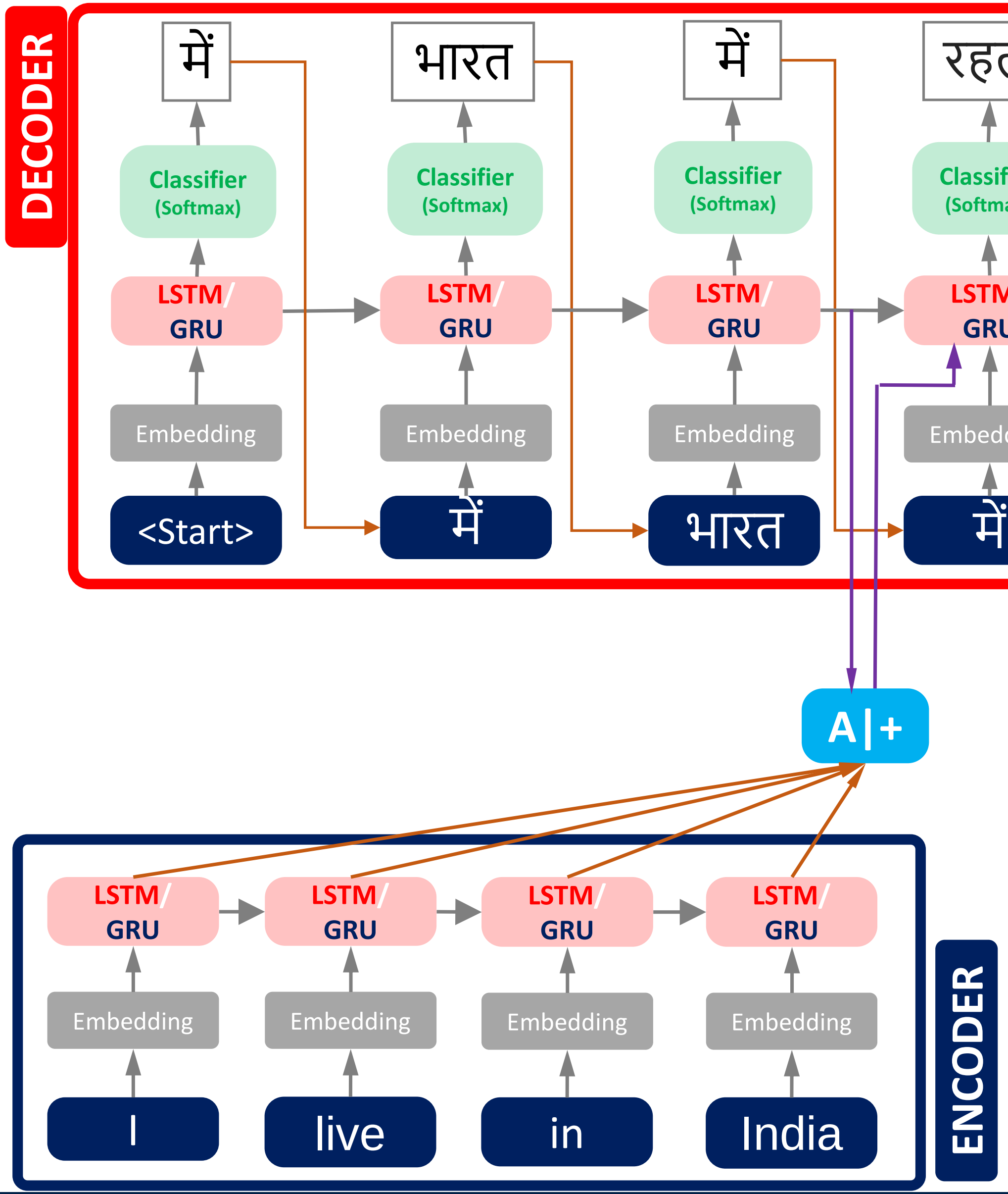
Translation with Attention



Masked softmax

- Often input sequences are padded.
- Masked softmax filters out those elements.
  - Need to specify the valid length of each sequence.
- Masked softmax assigns 0 weight to elements outside the valid length.
  - It takes those elements to be large negative, such that after softmax the weights become 0.

Bahdanau's attention

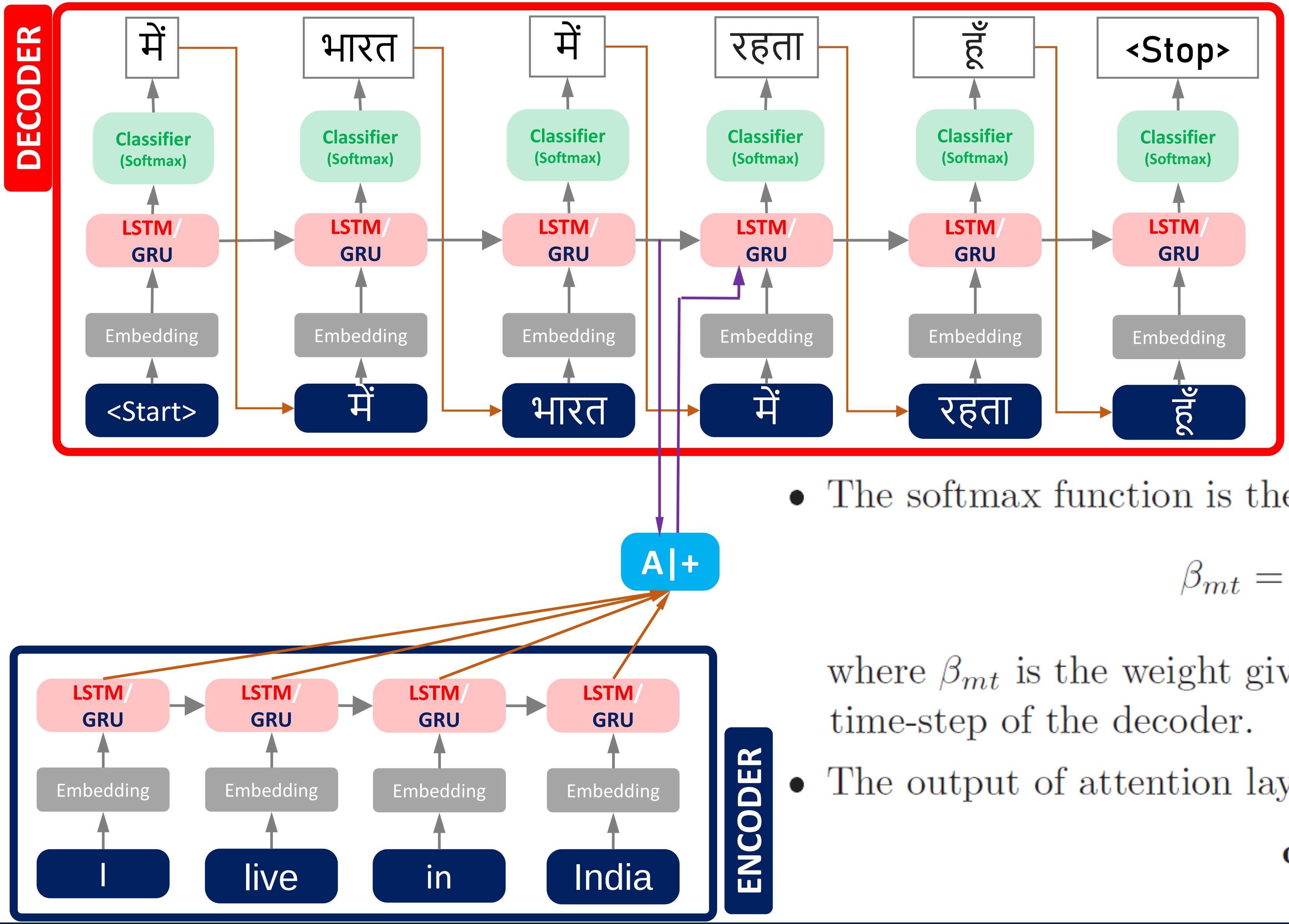


- The score function is defined as

$$A_{mt}(\mathbf{h}_m^E, \mathbf{h}_{t-1}^D) = \mathbf{v}^{\mathrm{T}} \tanh\left(\mathbf{W}_k \mathbf{h}_m^E + \mathbf{W}_q \mathbf{h}_{t-1}^D\right)$$

  - $\mathbf{h}_{t-1}^D \in \mathbb{R}^{H_D \times 1}$ is known as the query $\mathbf{q}$.
  - $\mathbf{h}_m^E \in \mathbb{R}^{H_E \times 1}$ is known as the key $\mathbf{k}$.
- Note, $A_{mt}(\mathbf{h}_m^E, \mathbf{h}_{t-1}^D)$ is scalar.
- $\mathbf{v}$, $\mathbf{W}_k$ and $\mathbf{W}_q$ are parameters that are learnt.

Bahdanau's attention



- The softmax function is then used to obtain the attention weights

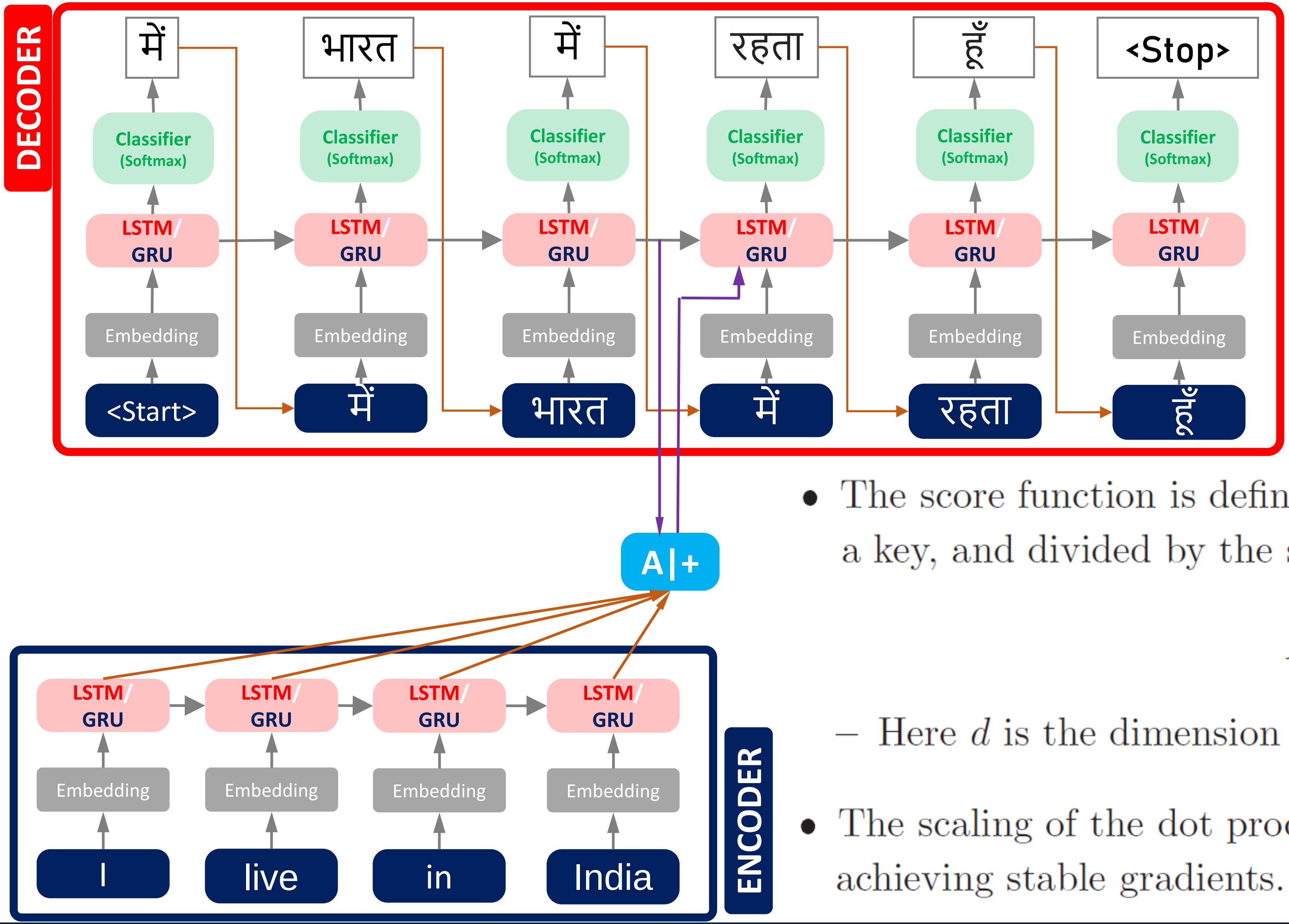
$$\beta_{mt} = \frac{\exp(A_{mt})}{\sum_m \exp(A_{mt})}$$

where β_{mt} is the weight given to the mth input word at the tth time-step of the decoder.

- The output of attention layer is the weighted sum of the values

$$\mathbf{o}_t = \sum_m \beta_{mt} h_m^E$$

Dot-product attention



- The score function is defined as the dot-product of the query and a key, and divided by the square-root of the dimension:

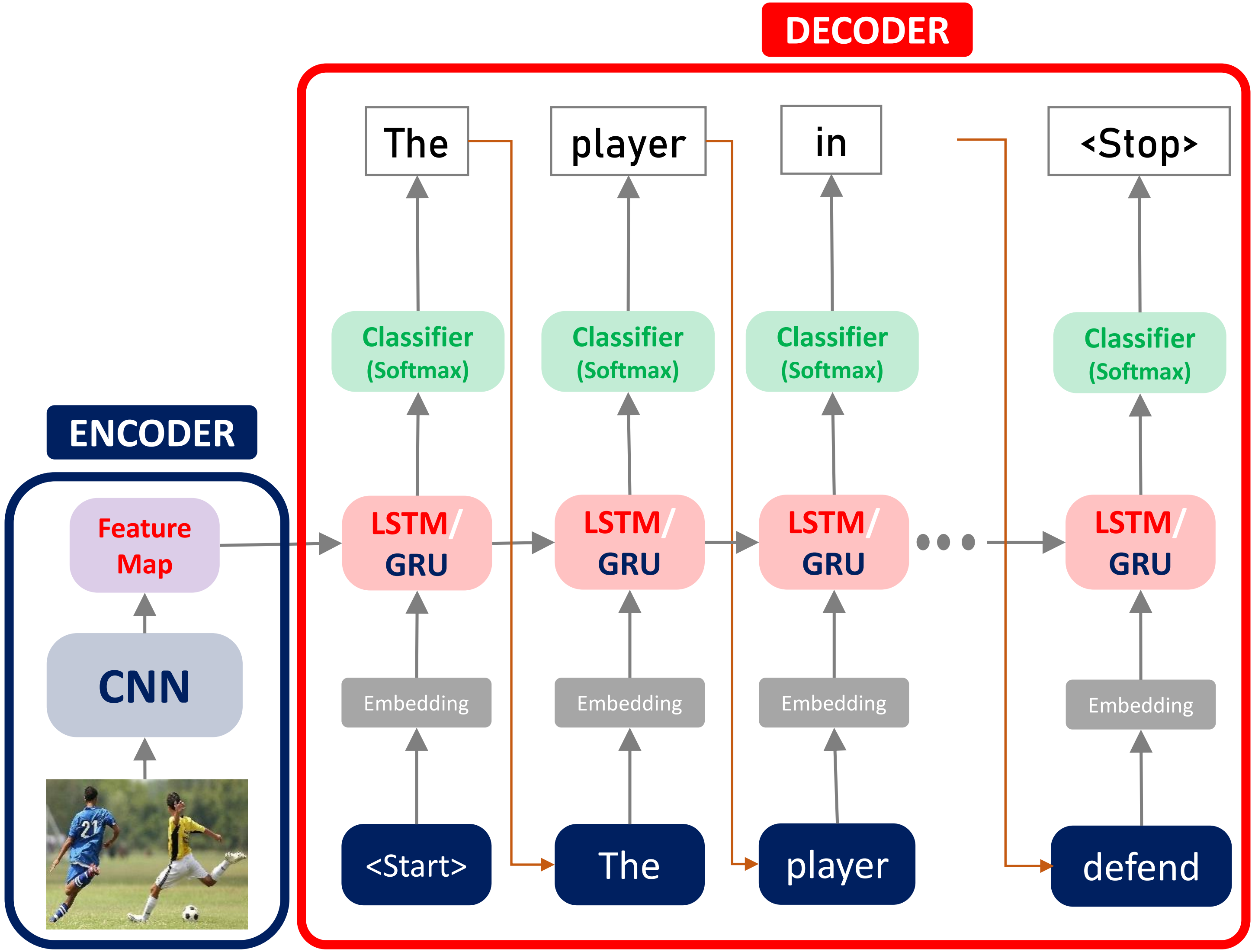
$$A(\mathbf{q}, \mathbf{k}) = \frac{< \mathbf{q}, \mathbf{k} >}{\sqrt{d}}$$

  – Here d is the dimension of the key vector.

- The scaling of the dot products by $1/\sqrt{d}$ is done to facilitate achieving stable gradients.

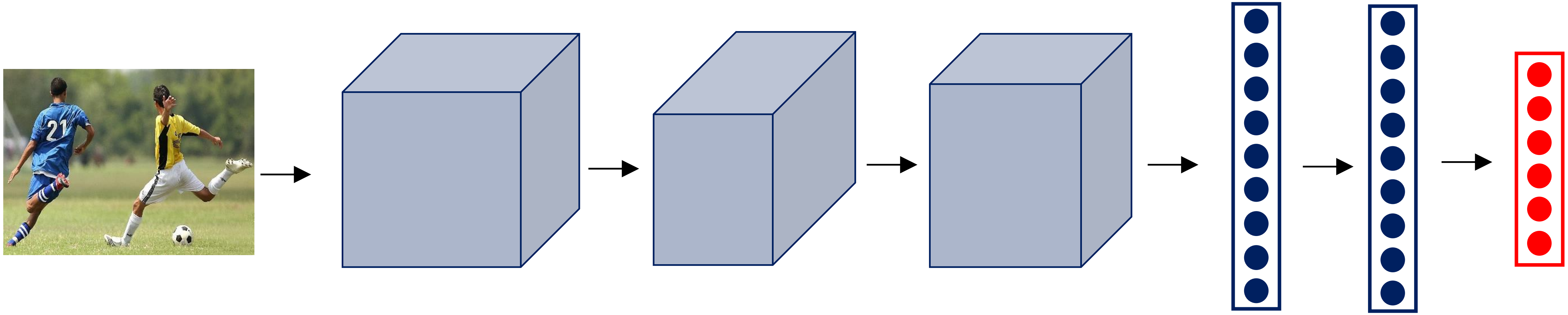
VISUAL
ATTENTION

Image captioning: standard model



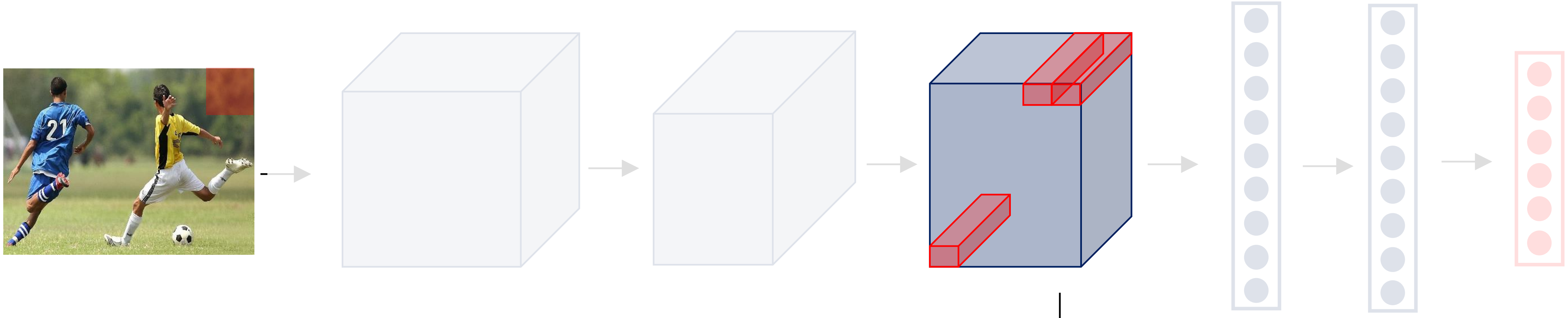
- A key aspect of the human visual system is attention.
- The decoder model presented here uses a static representation of the image.
- Attention mechanism enables salient features to play important roles when needed.

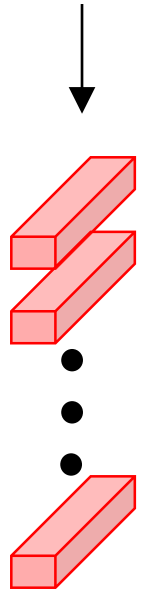
CNN feature map



- The feature map from a FC layer represent information in a very compact form.
  - This can lead to loss of useful information.
  - Also spatial information is not properly retained.
- Output is taken from one of the convolutional layers.
  - This is the feature map generated in one of the convolutional layers.

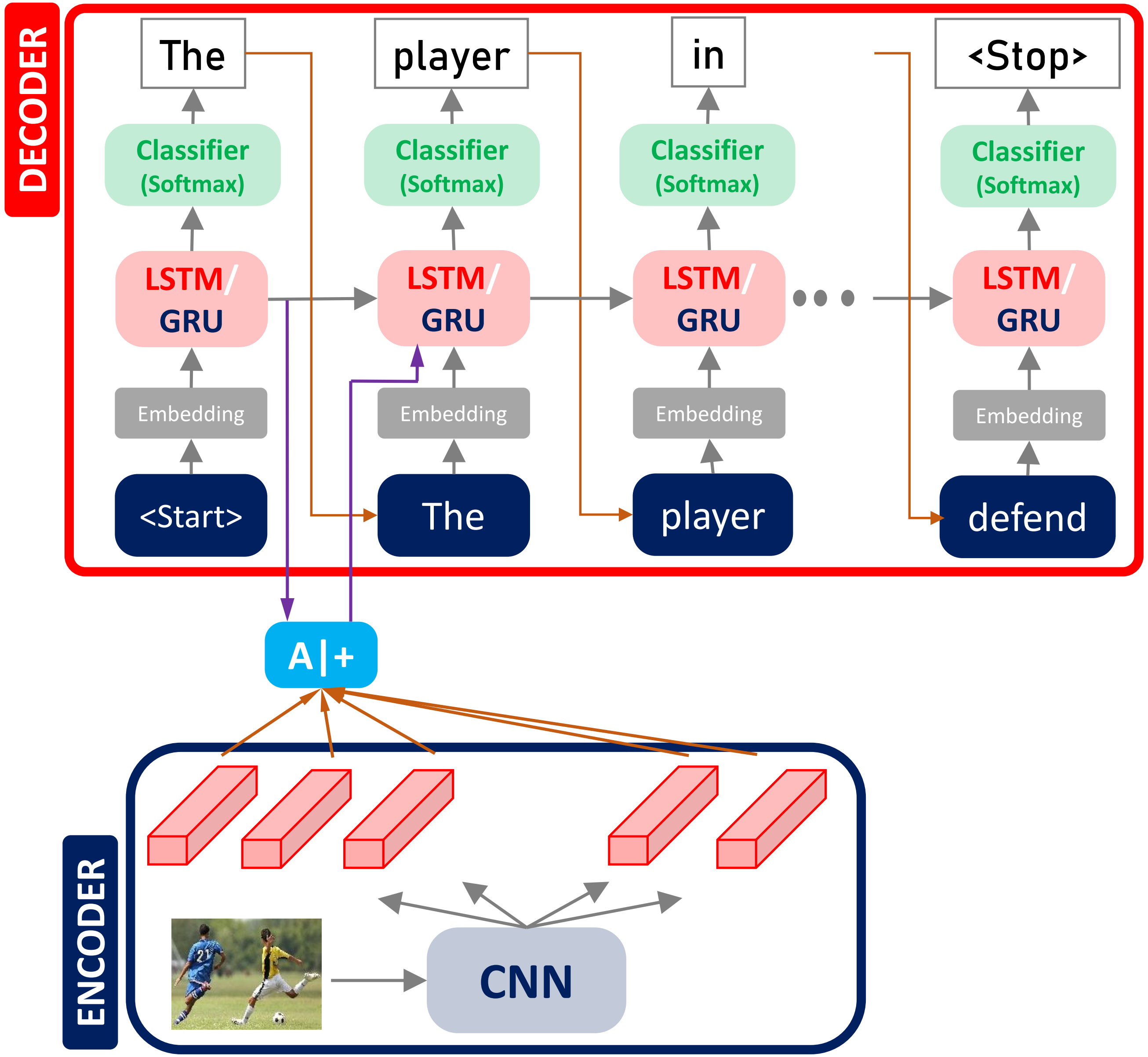
CNN feature map





- Points in a convolutional feature map defines a specially localized feature vectors.
- If the convolutional feature map is of size $8 \times 8 \times 50$, then there would be 64 vectors, each of size 50.

Visual attention model



- Suppose $\mathbf{f}_1$, $\mathbf{f}_2$,...., $\mathbf{f}_M$ are the feature vectors derived from a convolutional layer.
- Let $\mathbf{y}^{*(t)}$ be the output of the decoder at time-step t.
- The hidden state at time-step t of the RNN can be computed as

$$\mathbf{h}_t = \mathrm{RNN}\left(\mathbf{h}_{t-1}, [\mathbf{y}^{*(t-1)}, \mathbf{o}_t]\right)$$

where $\mathbf{o}_t$ is the weighted sum of the CNN feature vectors

$$\mathbf{o}_t = \sum_{m=1}^{M} \beta_{mt} \mathbf{f}_m$$