Encoder-Decoder models

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

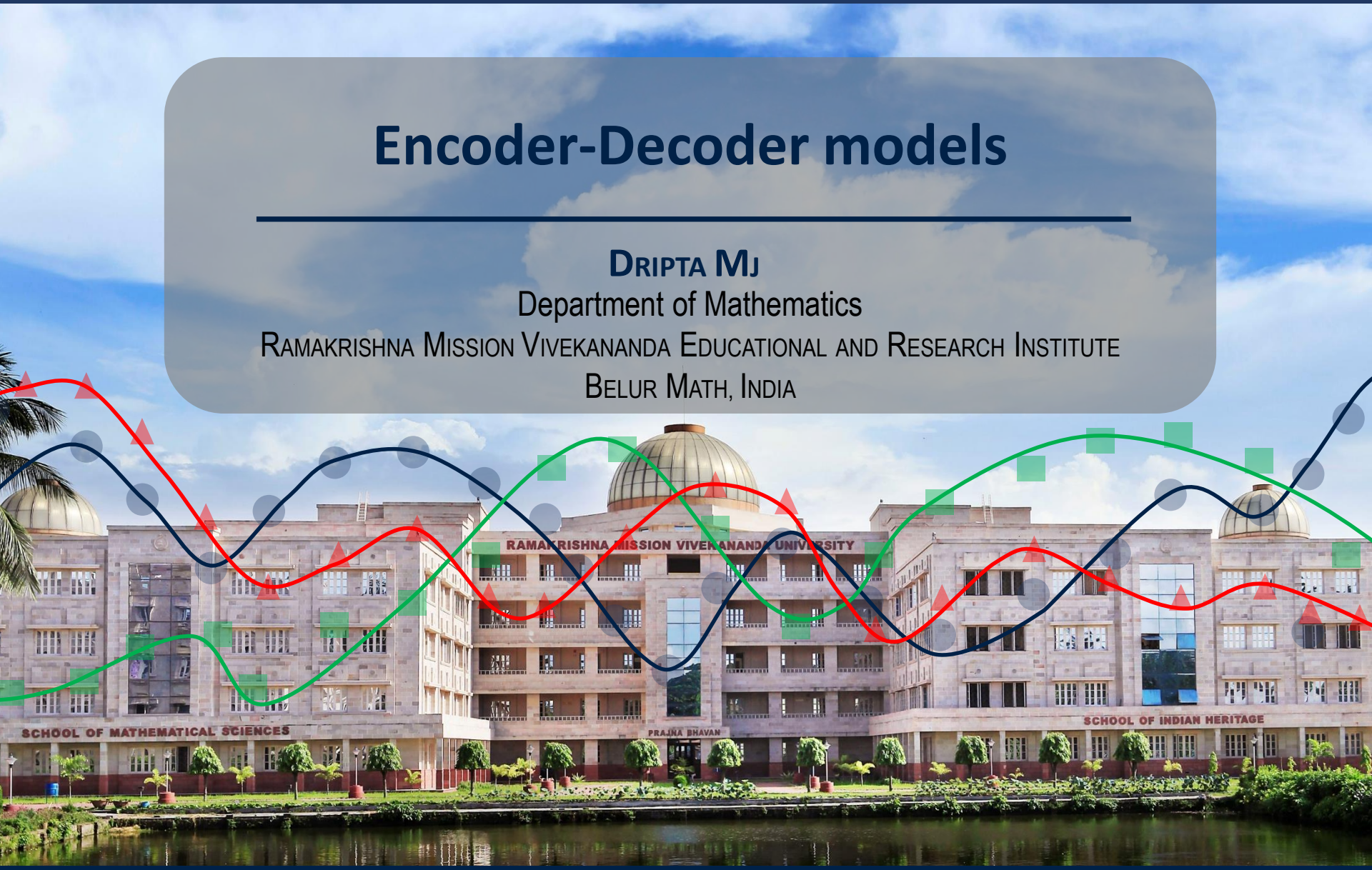
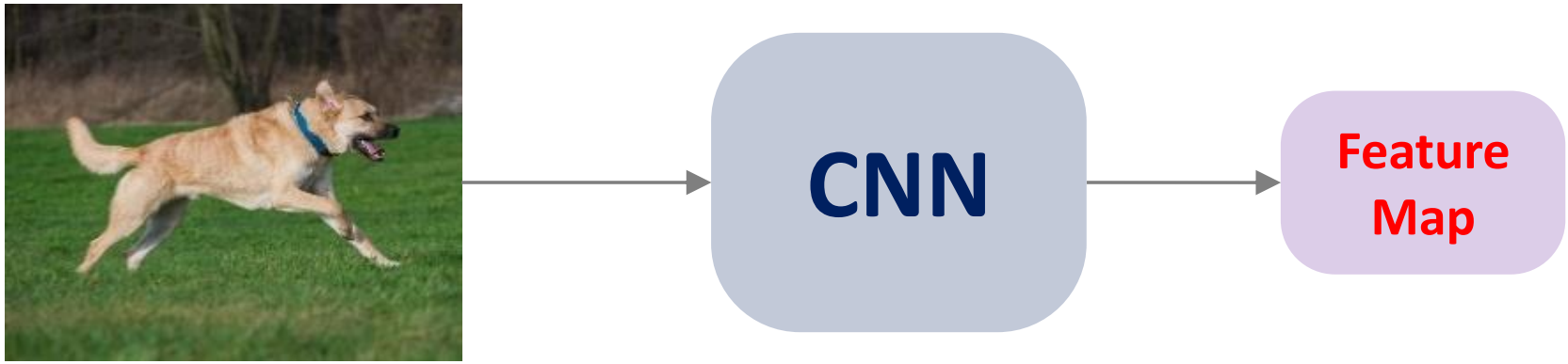


Image feature maps



- But how to handle text-data?

How to represent words?

- Consider vocabulary $\mathcal{V}$ with $|V|$ words
- Suppose vocabulary $\mathcal{V}$ is of the form

$$\mathcal{V} = \left\{ \begin{array}{c} \text{abandon} \\ \text{ability} \\ \cdot \\ \cdot \\ \text{hot} \\ \cdot \\ \text{is} \\ \cdot \\ \cdot \\ \text{today} \\ \cdot \\ \text{zoo} \end{array} \right\}$$

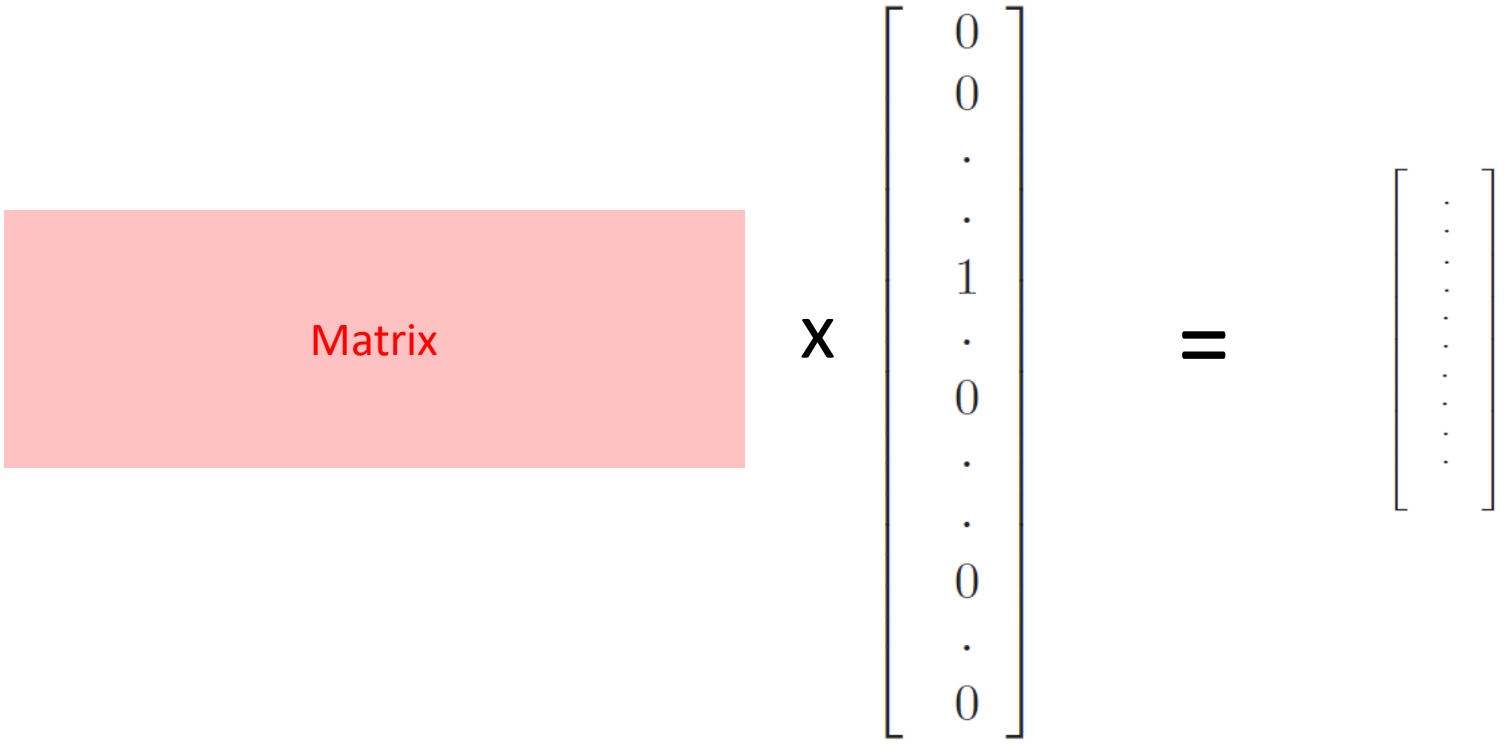
- $\mathbf{x}$ = "Today is hot"
- One-hot representation

$$\mathbf{x}_1 = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \\ \cdot \\ 0 \\ \cdot \\ \cdot \\ 1 \\ \cdot \\ 0 \end{bmatrix} \quad \mathbf{x}_2 = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \\ \cdot \\ 1 \\ \cdot \\ \cdot \\ 0 \\ \cdot \\ 0 \end{bmatrix} \quad \mathbf{x}_3 = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ 1 \\ \cdot \\ 0 \\ \cdot \\ \cdot \\ 0 \\ \cdot \\ 0 \end{bmatrix}$$

How to represent words?

- Words are represented as a completely independent entity.
- But all words in any language are not completely unrelated.
- Such representation does not give us any notion of similarity.
- Want to have a lower dimensional representation such that the subspace encodes relationship between words.

Low-dimensional word vector



How to represent words?

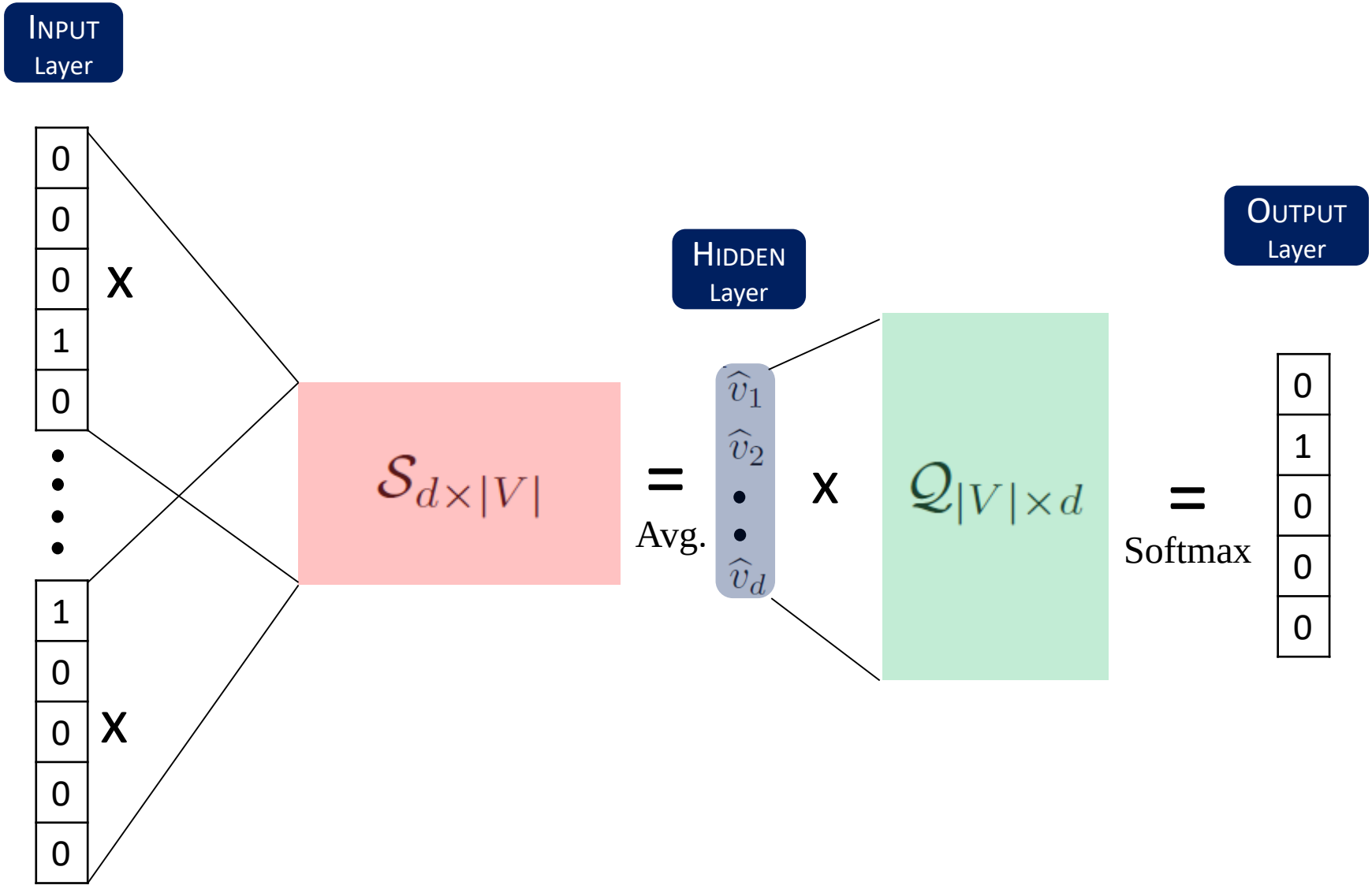
- Words are represented as a completely independent entity.
- But all words in any language are not completely unrelated.
- Such representation does not give us any notion of similarity.
- Want to have a lower dimensional representation such that the subspace encodes relationship between words.
- This lower dimensional representation of words is known as word embedding (word vector).
- Methods:
  - Singular Value Decomposition based methods
  - Iteration based methods
    - Continuous bag of words model
    - Skip-Gram model

Continuous bag of words (CBOW) model

- Given context predict the word.
- Example: { "The", "dog", "is", " ", "across", "the", "field" }.
  - Want the model to predict the middle word "located".
- Let the size of the embedding space be d.
- Let $\mathcal{S}$ be the input word matrix of size $d \times |V|$.
  - s_n: the nth column of $\mathcal{S}$ is the d dimensional vector representing the embedding for word w_n.
- Let $\mathcal{Q}$ be the output word matrix of size $|V| \times d$.
  - q_n: the nth row of $\mathcal{Q}$ is the d dimensional embedding vector associated with the nth word of the vocabulary.

Ref: Stanford NLP (http://web.stanford.edu/class/cs224n/)

Continuous bag of words (CBOW) model



CBOW: Procedure

- One hot vectors representing the input context:

$$\left\{\mathbf{x}_{(i-m)}, ..., \mathbf{x}_{(i-1)}, \mathbf{x}_{(i+1)}, \mathbf{x}_{(i+m)}\right\}$$

- Generate embedded word vectors for the input context

$$\left\{\mathbf{v}_{(i-m)}, ..., \mathbf{v}_{(i-1)}, \mathbf{v}_{(i+1)}, \mathbf{v}_{(i+m)}\right\}$$

  where $\mathbf{v}_{(i-j)} = \mathcal{S}\mathbf{x}_{(i-j)}$

- Compute the average of these vectors:

$$\widehat{\mathbf{v}} = \frac{\mathbf{v}_{(i-m)} + \mathbf{v}_{(i-m+1)} + ... + \mathbf{v}_{(i+m-1)} + \mathbf{v}_{(i+m)}}{2m}$$

- Score vector for the given context $\mathbf{u} = \mathbf{Q}\widehat{\mathbf{v}}$.

- Convert the scores into probabilities using the softmax function and make prediction

$$y^* = \mathcal{V}\left[\arg\max_j \left(\text{softmax}(\mathbf{u}_j)\right)\right]$$

CBOW: Loss

- Cross-entropy loss (for one example):

$$\mathcal{L} = -\log P(y = w_c | \mathbf{x}_{(i-m)}, ..., \mathbf{x}_{(i-1)}, \mathbf{x}_{(i+1)}, \mathbf{x}_{(i+m)})$$

$$= -\log \frac{\exp\left(\mathbf{q}_c \widehat{\mathbf{v}}\right)}{\sum_{j=1}^{|V|} \exp\left(\mathbf{q}_j \widehat{\mathbf{v}}\right)}$$

$$= -\mathbf{q}_c \widehat{\mathbf{v}} + \log \sum_{j=1}^{|V|} \exp\left(\mathbf{q}_j \widehat{\mathbf{v}}\right)$$

Encoder-Decoder

- Sometimes referred to as sequence-to-sequence architecture.
- How to map a variable-length sequence to another variable-length sequence?
- An encoder RNN processes the input sequence.
  - It outputs the context.
  - The context can be the final hidden state (or some function of it) of the encoder RNN
- A decoder RNN is conditioned on the context vector.
  - It generates the output sequence.

Image captioning

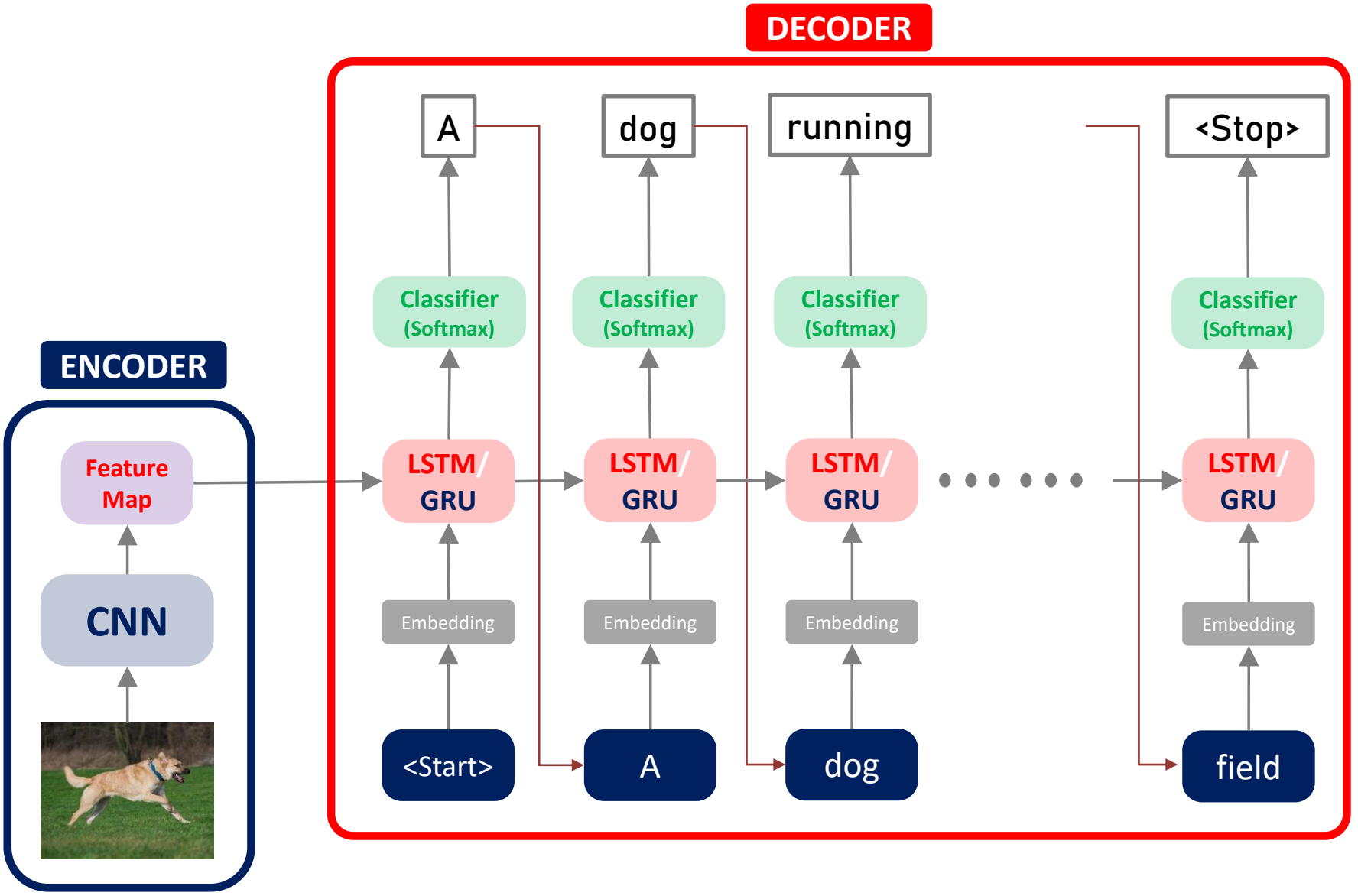
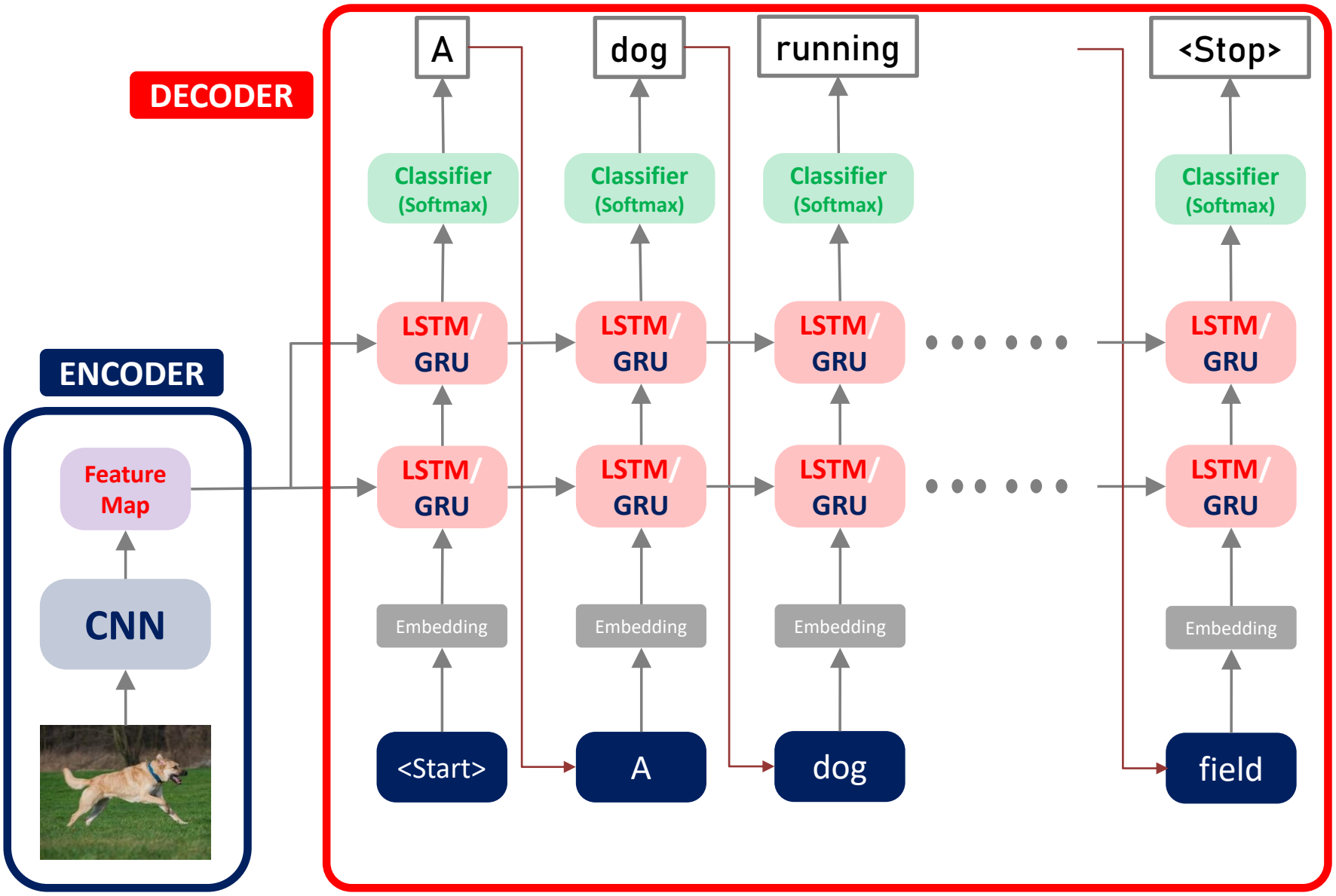
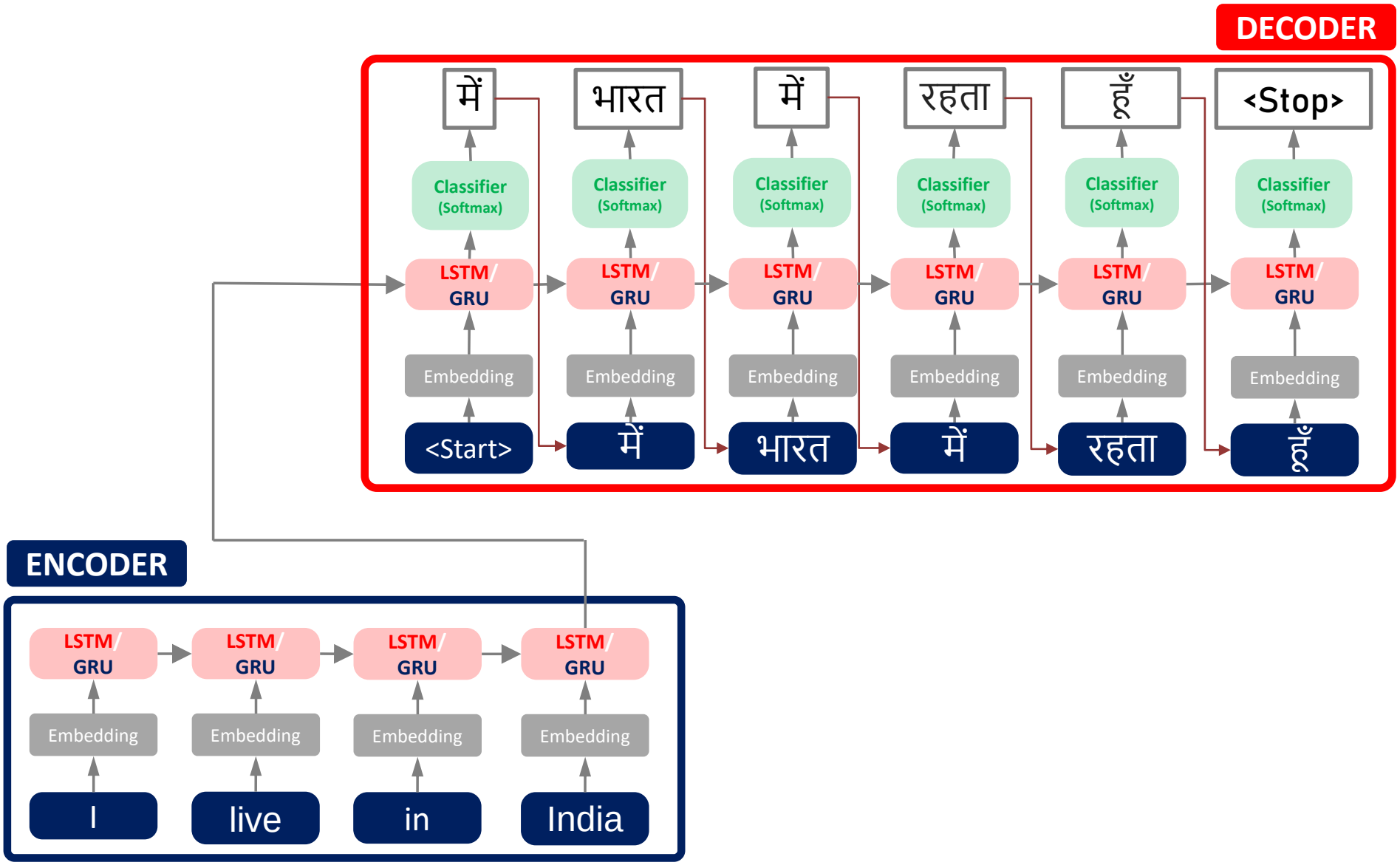


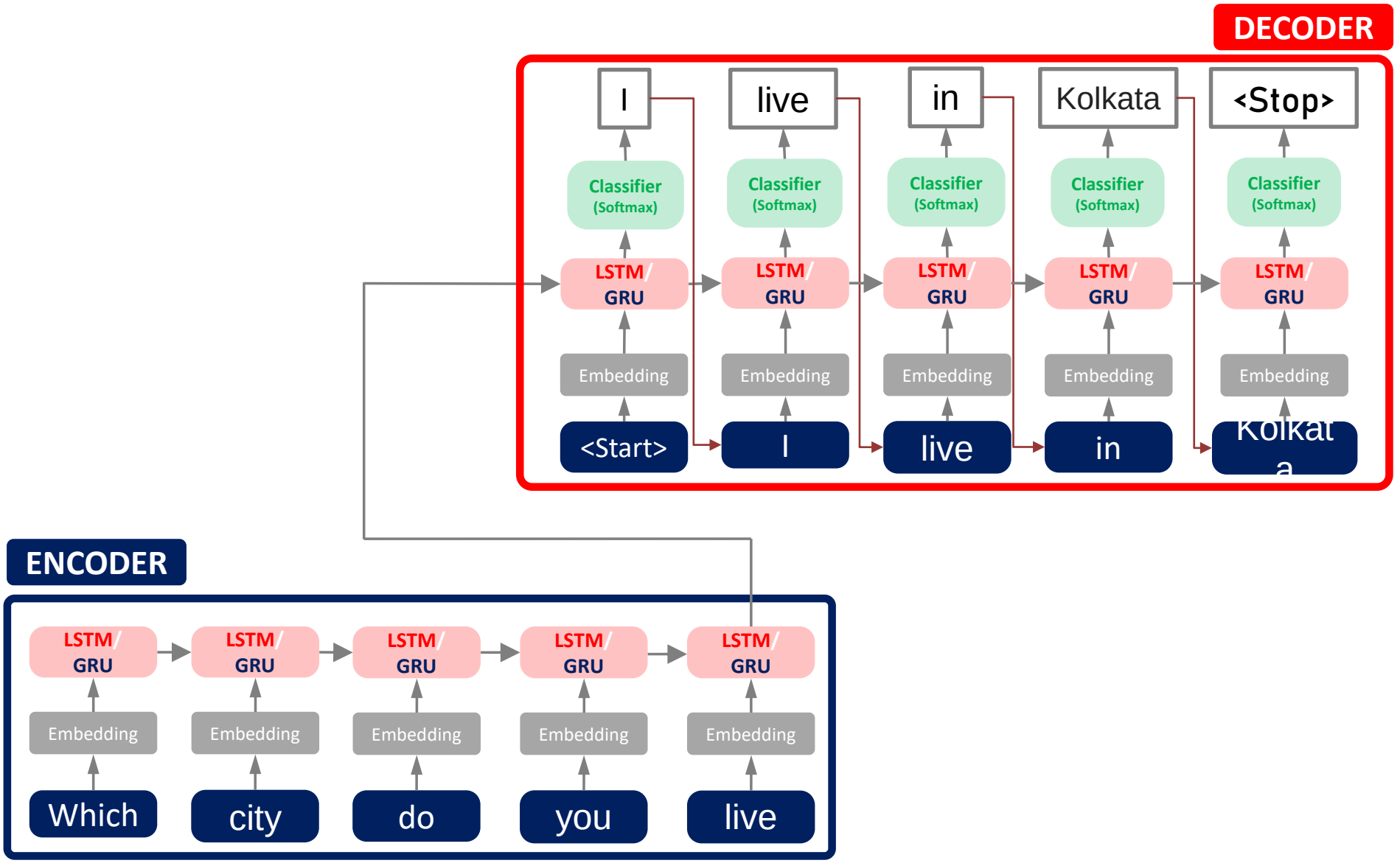
Image captioning



Translation



Conversation



Video captioning