Natural Language Processing (NLP)

Sequence to Sequence Applications

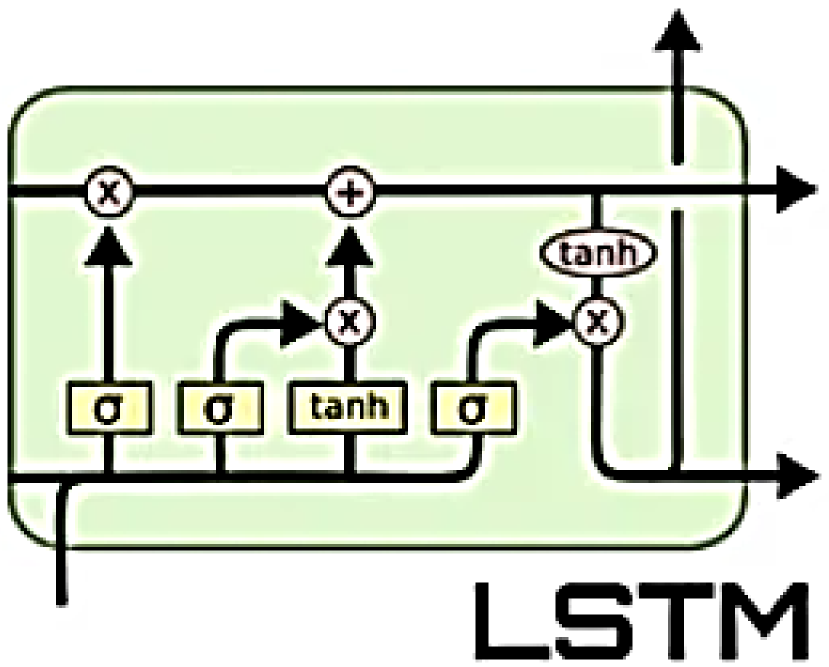
Next Word Prediction, Encoder & Decoder.

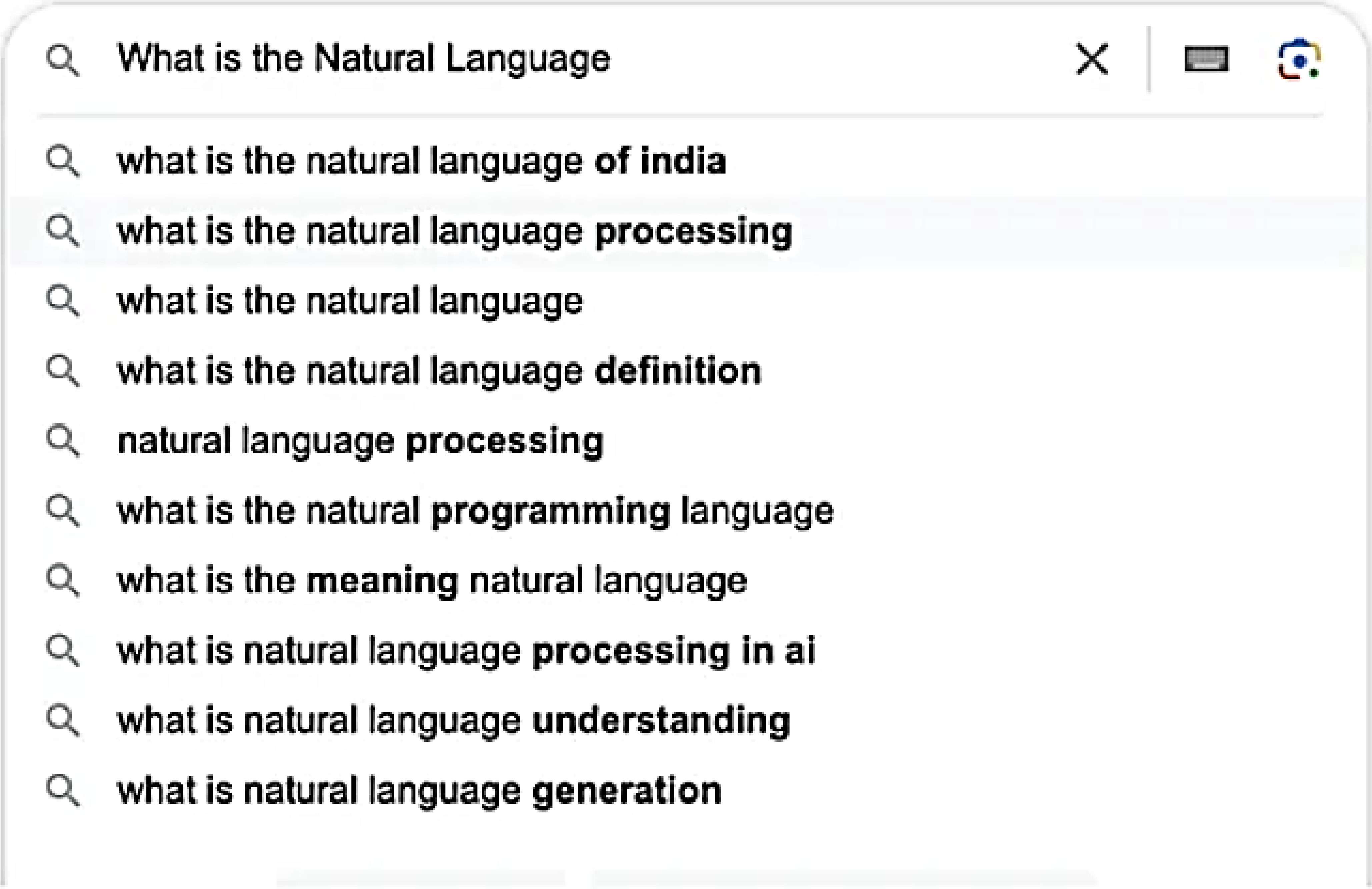
Language Translation

By:

Dr. Zohair Ahmed

Next Word Prediction





Complex Problem

- Natural language has always been complex.
- Especially when we see deleted words in some pictures, we try to guess that word.
- I wonder what way we should follow so that we can guess the word at the end of a sentence.
- Even though this seems really easy, we can drown in the pool of possibilities.

Let’s do an example

- Let’s do an example to show how complex it is.
- What is the weather …….. ?
- It can be; today, like, now, outside etc.
- As you can see, deciding what the next word is in a sentence can be quite complicated.
- Of course, what is important here may be issues such as the meaning of the previous sentence.

Code Example

- Let's jump into the Code

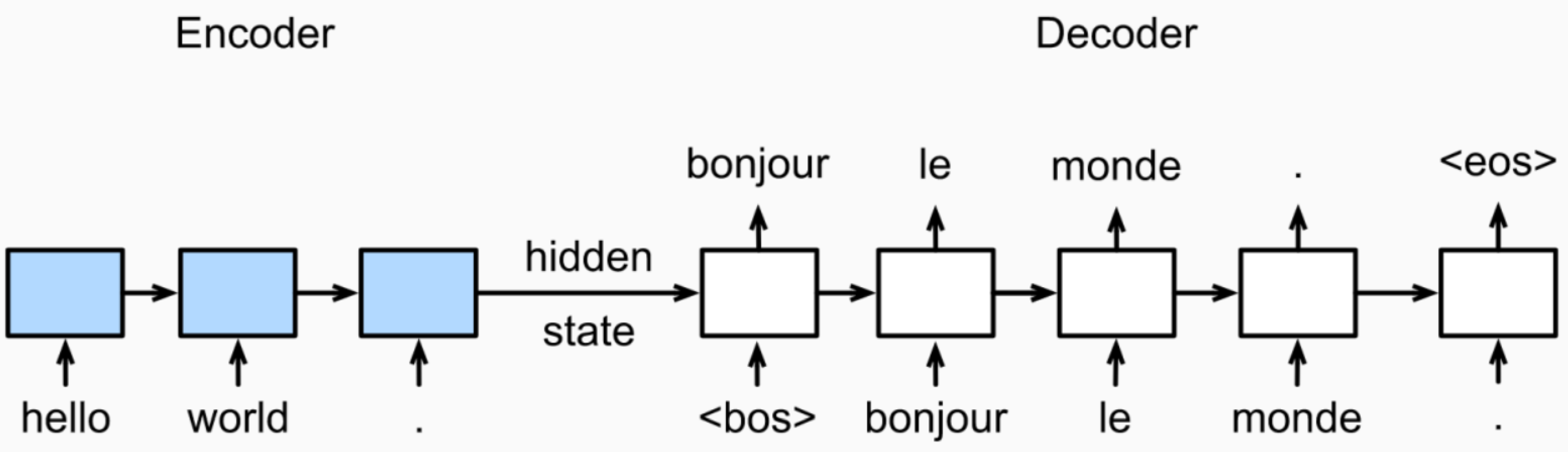
Seq2Seq Modeling - Language Translation

- A sequence-to-sequence (seq2seq) model is a type of neural network architecture widely used in various natural language processing (NLP) tasks, such as machine translation, text summarization, and dialogue systems.
- The key idea behind seq2seq models is to learn a mapping between input and output sequences of variable lengths.

Seq2Seq Modeling - Language Translation

- The sequence-to-sequence model has two main components: an **encoder** and a **decoder**.
- The encoder processes the input sequence and encodes it into a fixed-length vector representation, often called the context vector or the hidden state.
- The decoder then takes this context vector and generates the output sequence one element at a time, using the previous output elements to predict the next element.
- The encoder and decoder components are typically implemented using recurrent neural networks (RNNs), such as long short-term memory (LSTM) or gated recurrent units (GRU), which can handle sequential data.
- However, more recent architectures, like the Transformer model, have also been used for seq2seq tasks, achieving state-of-the-art performance in many applications.

Seq2Seq Modeling - Language Translation



The sequence to sequence model architecture.

Training Seq2Seq Model

- Training seq2seq models involves optimizing their parameters to minimize a loss function that measures the difference between the predicted target sequence and the actual target sequence.
- **1. Data Preparation**
- The training data consists of paired examples: **source language sentences** and their corresponding **target language translations**.
- Both **source** and **target** sentences are typically preprocessed, tokenized (broken down into individual words or units), and potentially padded to ensure consistent lengths.

Training Seq2Seq Model

- Training seq2seq models involves optimizing their parameters to minimize a loss function that measures the difference between the predicted target sequence and the actual target sequence.
- **1. Data Preparation**
- The training data consists of paired examples: **source language sentences** and their corresponding **target language translations**.
- Both **source** and **target** sentences are typically preprocessed, tokenized (broken down into individual words or units), and potentially padded to ensure consistent lengths.

2. Forward Pass

- During training, an input source language sentence is fed into the encoder's RNN (often an LSTM).
- The encoder processes the sentence word by word, capturing the meaning and generating the context vector.
- The decoder receives the context vector and starts generating the target language sentence one word at a time, again using an RNN (often an LSTM).
- At each step, the decoder predicts the next most likely word in the target sequence.

3. Loss Calculation and Backpropagation

- The predicted target word is compared to the actual word from the target sequence using a loss function (e.g., cross-entropy).
- This loss is calculated for each word in the target sequence.
- The total loss represents the overall discrepancy between the predicted and actual target sentence.
- Backpropagation is then used to propagate the error back through the network, adjusting the weights and biases of the RNNs in both the encoder and decoder to minimize the loss.

4. Teacher Forcing

- Teacher forcing is a technique commonly used during seq2seq model training to address the exposure problem.
- The exposure problem arises because the decoder might generate inaccurate words early in the target sequence during training. These inaccurate words then become the decoder's input for subsequent steps, potentially leading the model down the wrong path.
- Teacher forcing mitigates this by feeding the decoder with the **ground truth** (actual target word) during training for some initial steps instead of the decoder's prediction.
- This helps the model learn the correct sequence and improve its ability to generate accurate words later.
- As training progresses, teacher forcing is gradually reduced, allowing the decoder to rely more on its own predictions.

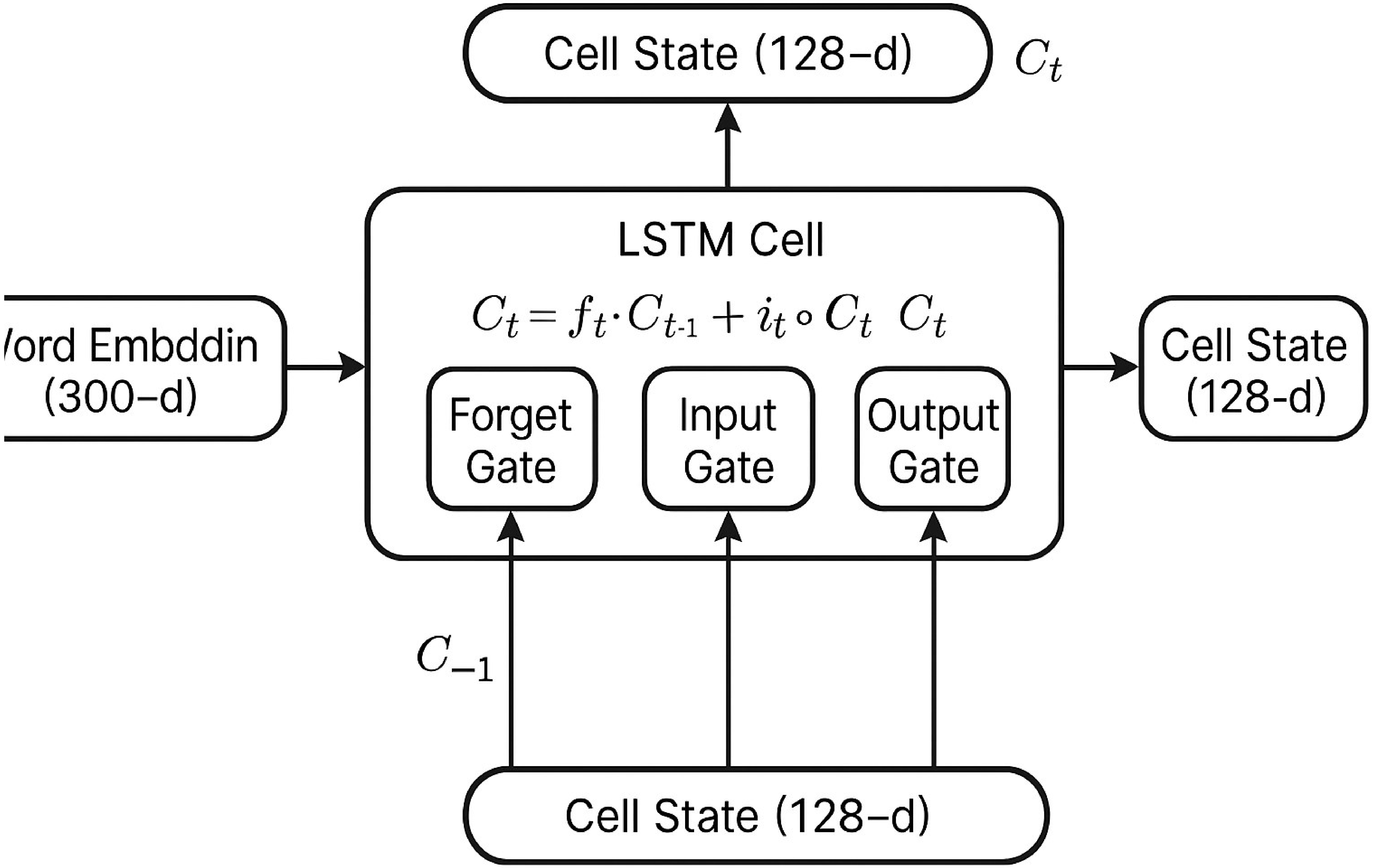
5. Iteration and Optimization

- The entire forward pass, loss calculation, backpropagation, and (potentially) teacher forcing process is repeated for multiple epochs (iterations) over the training data.
- Each iteration adjusts the model's parameters to minimize the overall loss, leading it to learn better representations and improve its translation accuracy.

5. Iteration and Optimization

- The entire forward pass, loss calculation, backpropagation, and (potentially) teacher forcing process is repeated for multiple epochs (iterations) over the training data.
- Each iteration adjusts the model's parameters to minimize the overall loss, leading it to learn better representations and improve its translation accuracy.

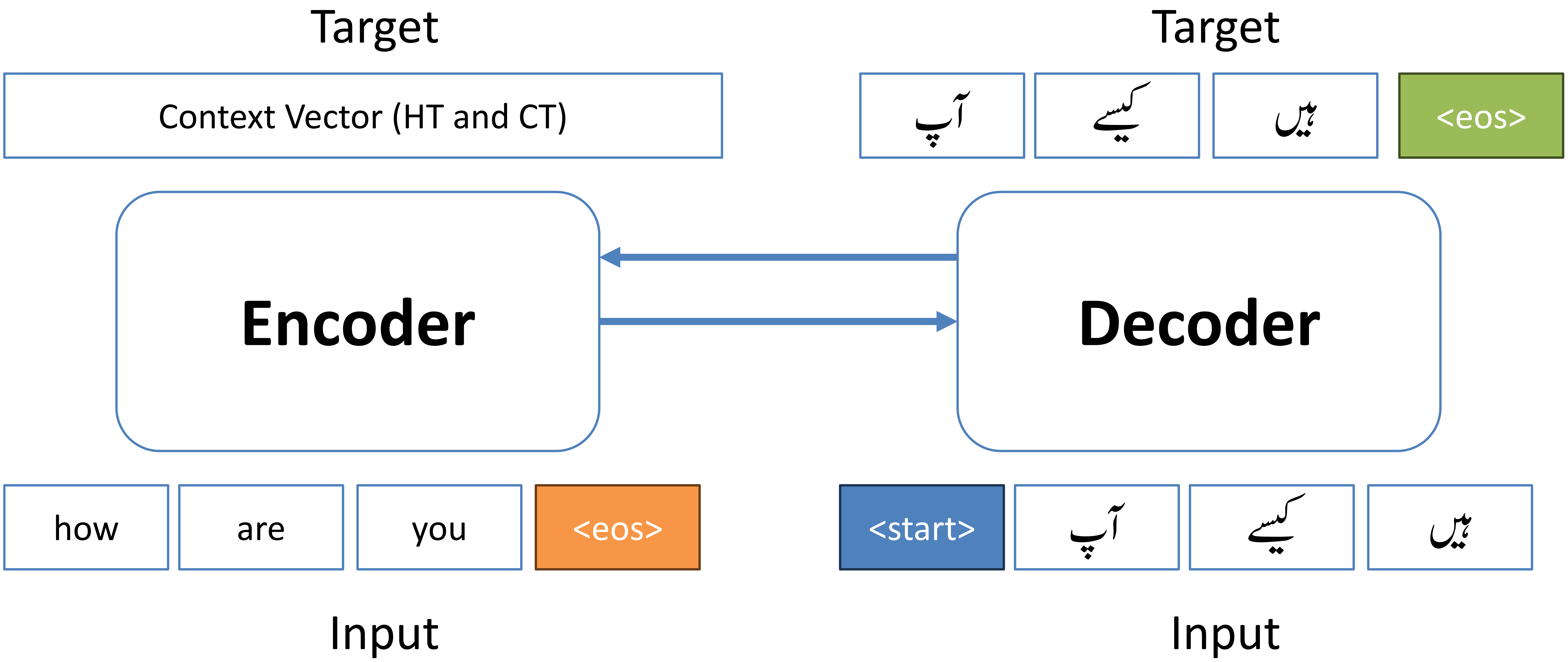
Things to Remember



Things to Remember

- **Backpropagation Flow in Seq2Seq**
- Here’s how it works:
- **Forward Pass**:
  - Encoder processes the input sentence and produces a context vector (final hidden and cell states).
  - Decoder uses this context to generate the output sentence word by word.
- **Loss Calculation**:
  - The decoder’s predictions are compared to the actual target sentence.
  - Loss is computed (usually using cross-entropy).
- **Backpropagation**:
  - The loss is backpropagated through the decoder’s layers.
  - Since the decoder’s initial hidden state came from the encoder, the gradients flow **back into the encoder**.
  - This updates the encoder’s weights, including the weights of its LSTM gates (input, forget, output).

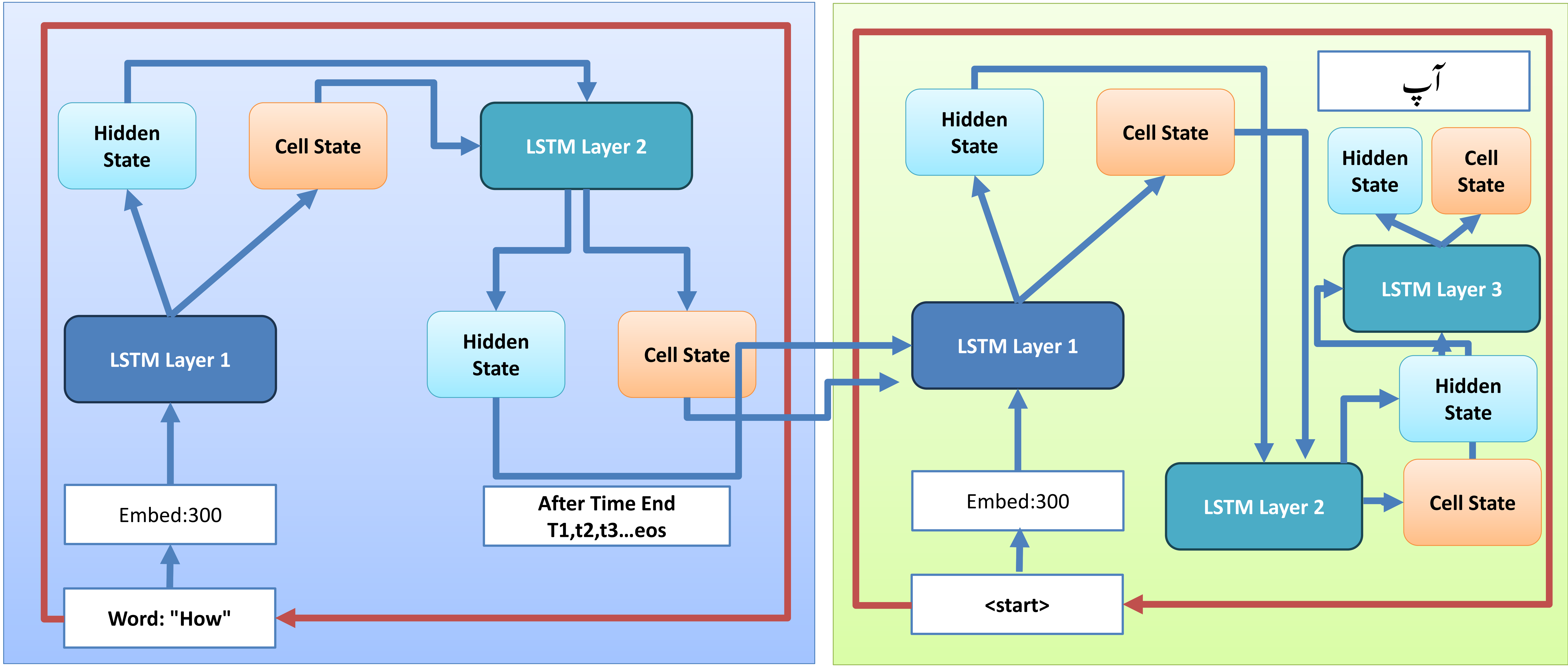
Things to Remember



Stacked LSTMS

- **2 LSTM** layers in the Encoder
- **3 LSTM** layers in the Decoder
- Input sentence: "How are you <eos>"
- Target sentence: "<start> آپ کیسے ہیں"

2 LSTM – Encoder 3 LSTM -Decoder



Code Example: Training Seq2Seq Model

- Let's jump into the Code