Seq2Seq

Sequence to Sequence Learning with Neural Networks

< 2024 IDEA LLM 세미나 >

2024년 08월 06일

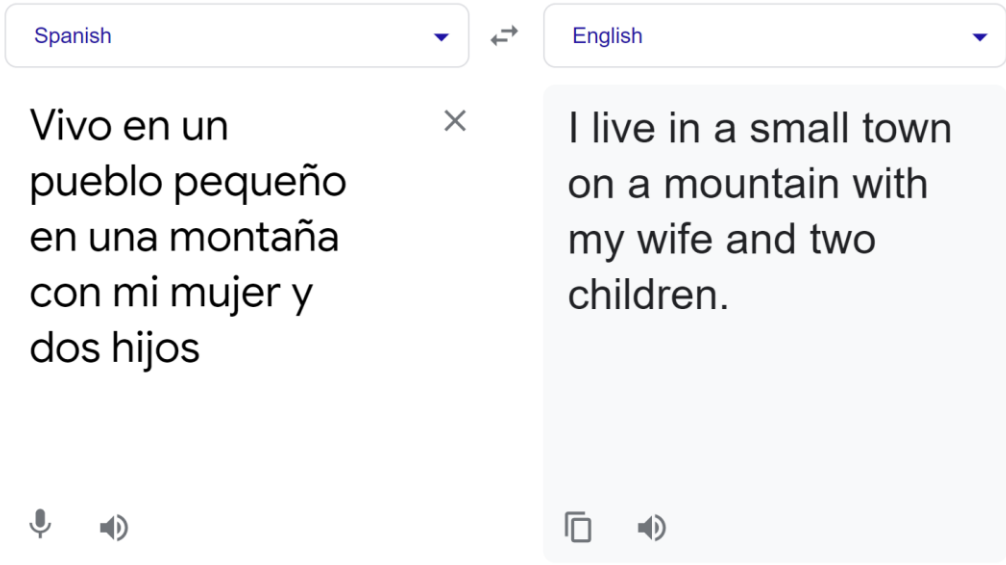
이 해 영

Contents

- Reviews : Many-to-Many RNN
- Sequence-to-Sequence (seq2seq) Models
  - Encoder
  - Decoder

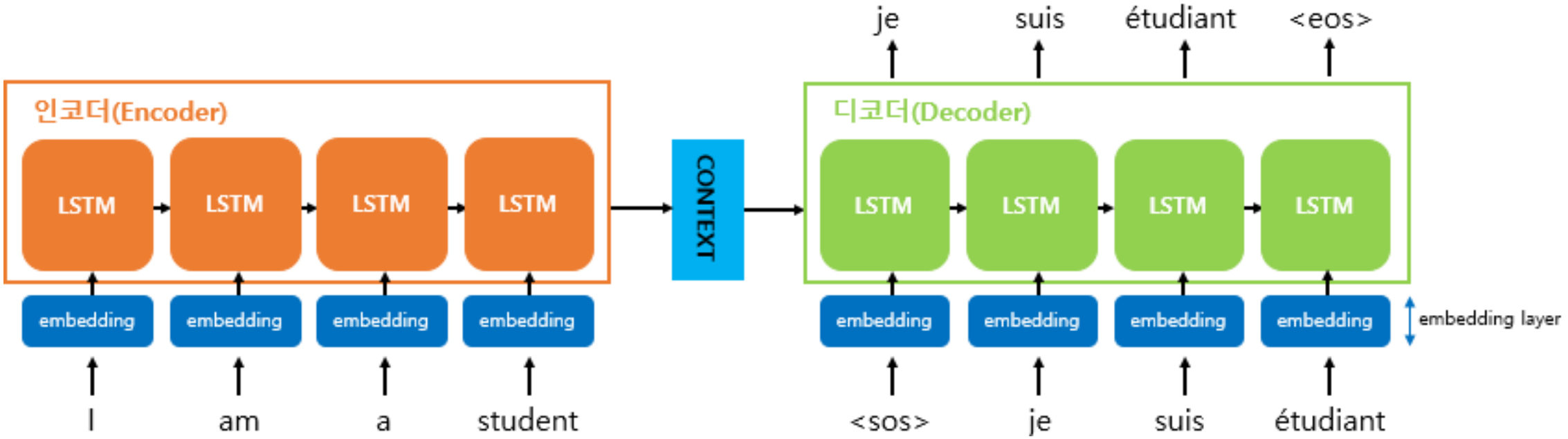
Sequence-to-Sequence (seq2seq) Models

- Machine Translation Problem
  - Given a sentence in one language, the task is generating a sentence of same meaning in another l anguage.



Sequence-to-Sequence (seq2seq) Models

- Encoder-Decoder

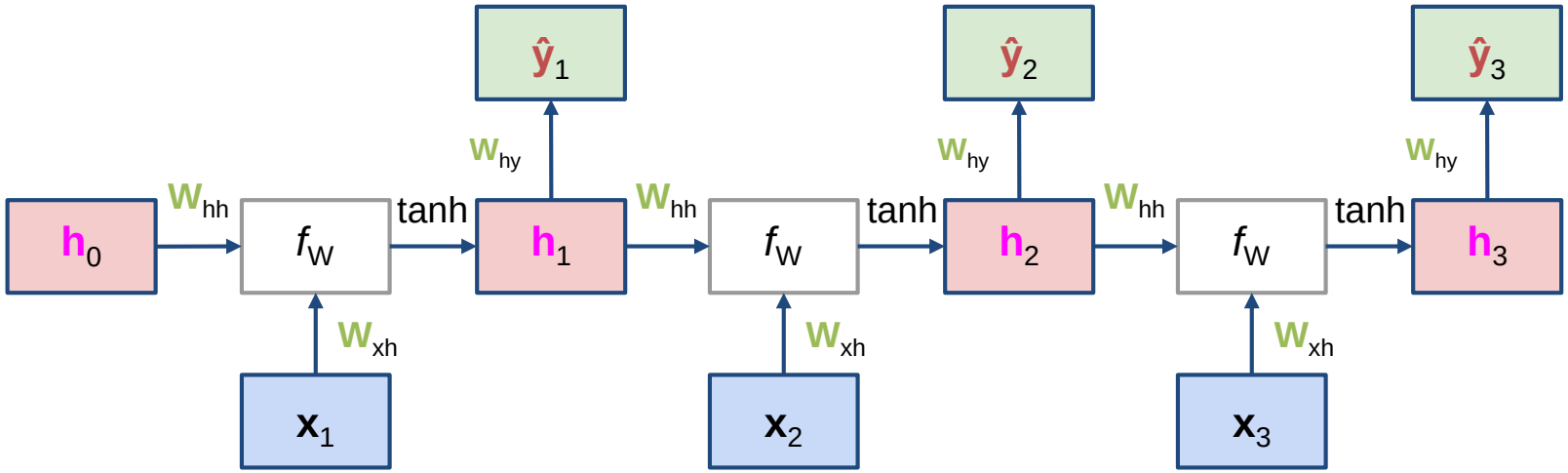


Review : RNN Trade-offs

- RNN is not perfect yet :
  - Issue 1) RNNs suffer from exploding/vanishing gradient problem, and thus it's hard to model long-range dependency. → **LSTM/GRU**
  - Issue 2) many-to-many RNN is not flexible enough to deal with input/output sequences of different length. → **Seq2seq model**

Sequence-to-Sequence Models

- Many-to-Many RNN
  - Input : $(x_1, \dots, x_T)$
  - Output : $(\hat{y}_1, \dots, \hat{y}_T)$



For binary classification:

$$\hat{y}_t = \sigma(W_{hy} h_t)$$

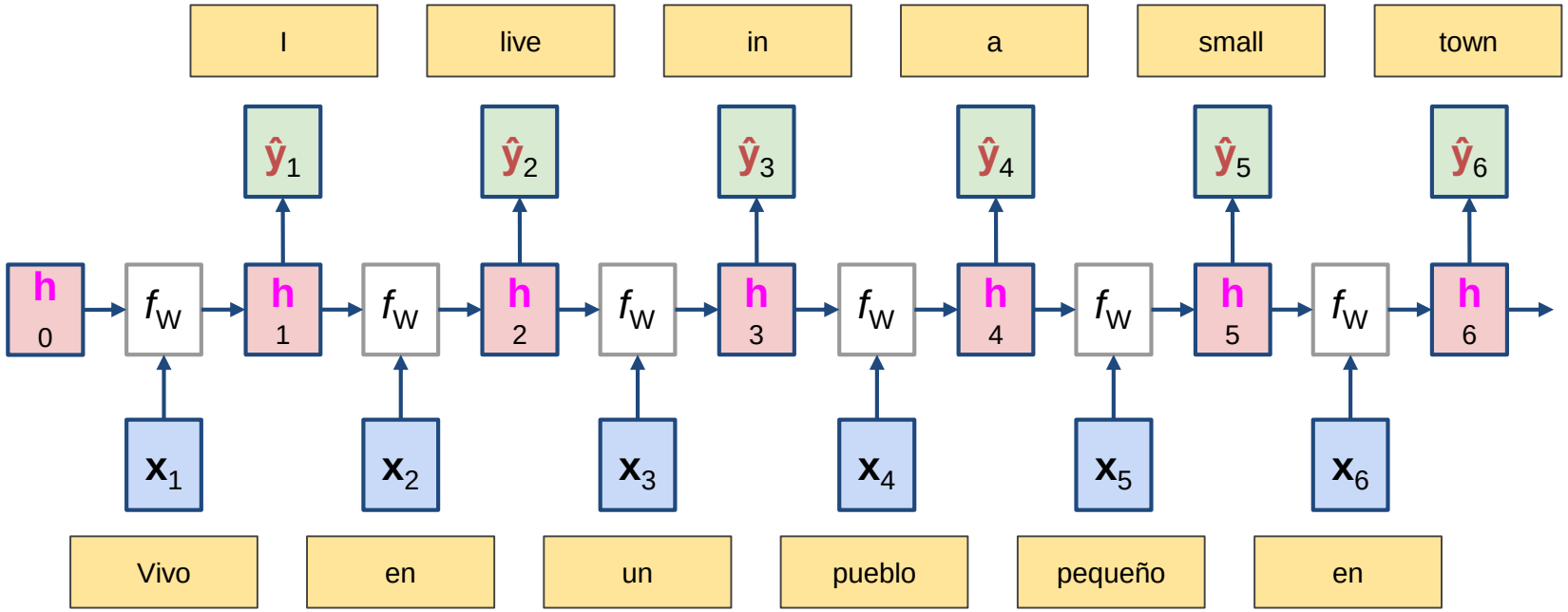
For regression:

$$\hat{y}_t = W_{hy} h_t$$

Sequence-to-Sequence Models

- Machine Translation Problem

Do you see any problem?



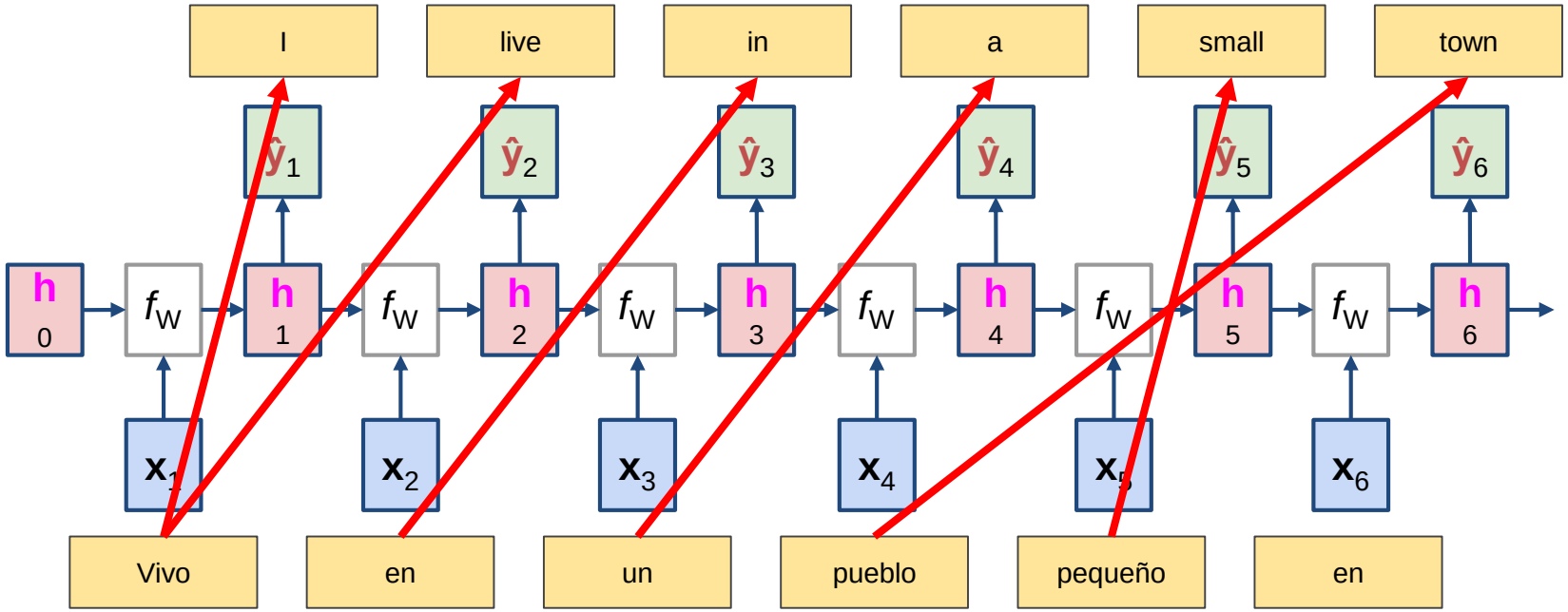
Sequence-to-Sequence Models

- Machine Translation Problem
  - Our RNN assumes 1:1 relationship.
    - Input length = output length
    - Semantics of input[k] = output[k].

For machine translation,

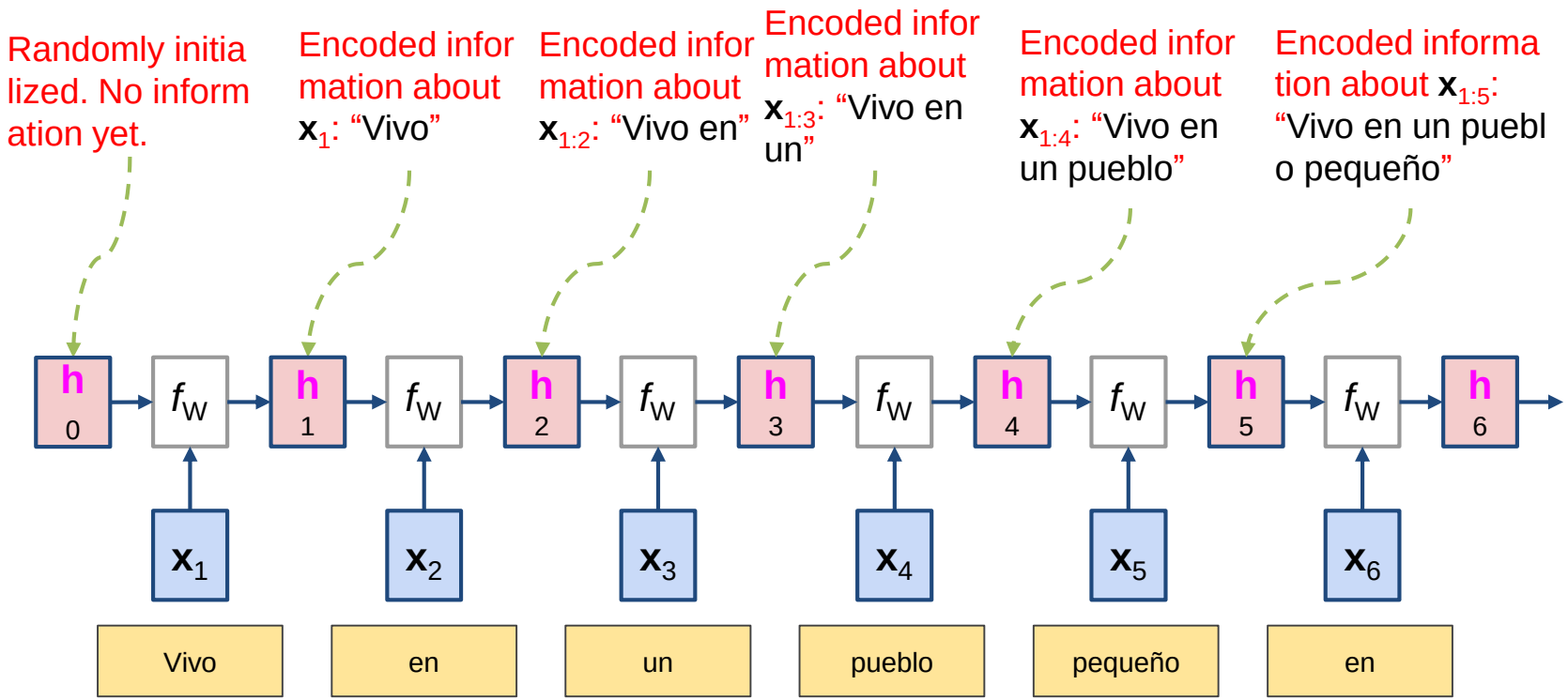
Sentence length varies by language.

Word may appear in different order!



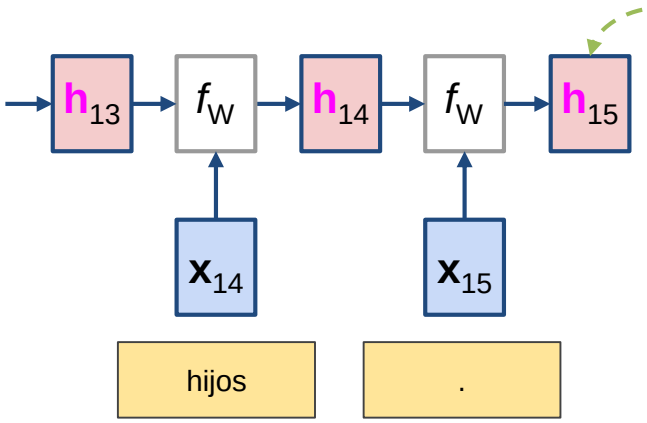
Sequence-to-Sequence Models

- Encoder-Decoder Structure
  - Let's step back to the original encoder structure, **without outputting** at each step:



Sequence-to-Sequence Models

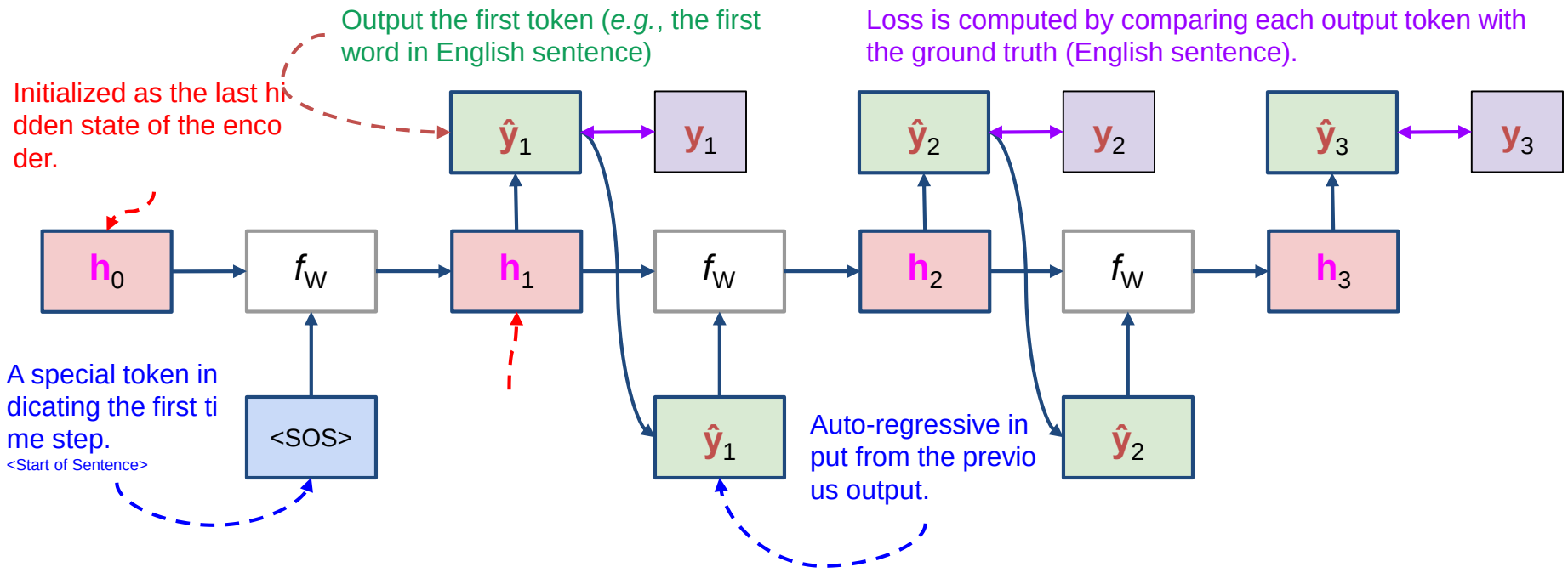
- Encoder-Decoder Structure



At the end of the sequence, h_{15} encodes the entire sequence $x_{1:15}$:
"Vivo en un pueblo pequeño en una montaña con mi mujer y dos hijos."

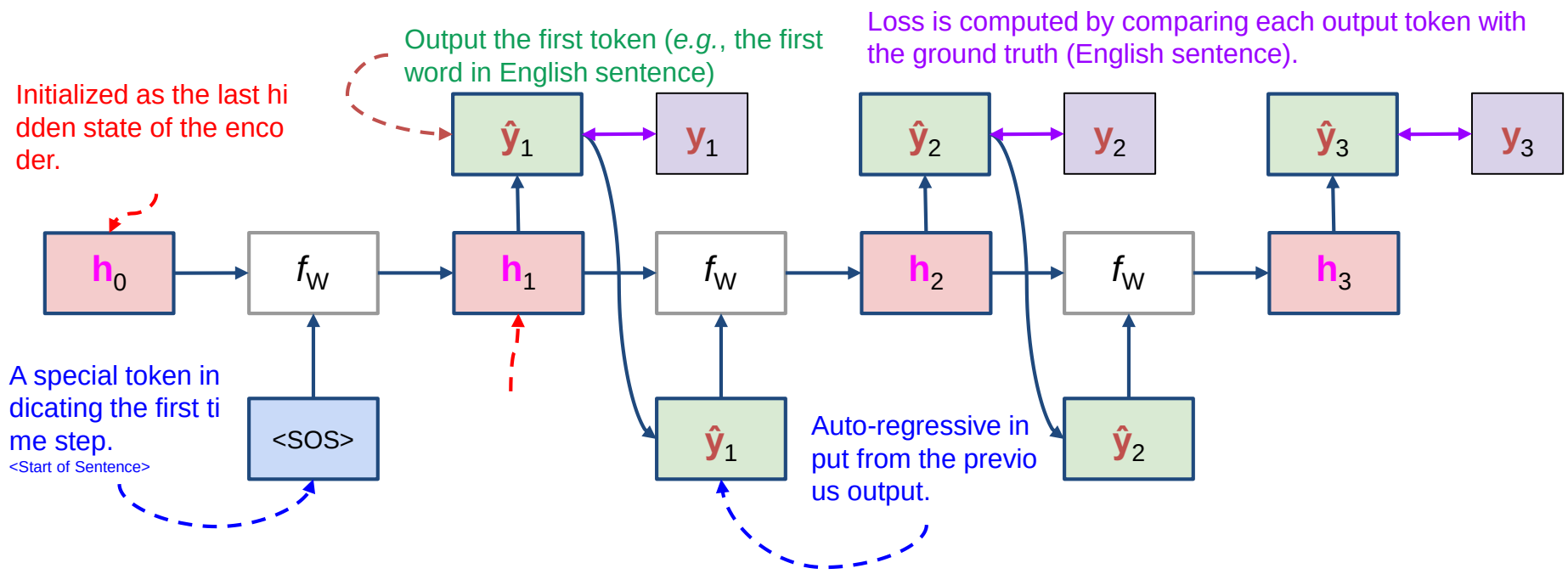
Sequence-to-Sequence Models

- Decoder : Auto-Regressive Generation
  - At each step, given a hidden state (expected to carry information about input sequence; **context**) and the last output (indicating where we are), it decides the next output token.



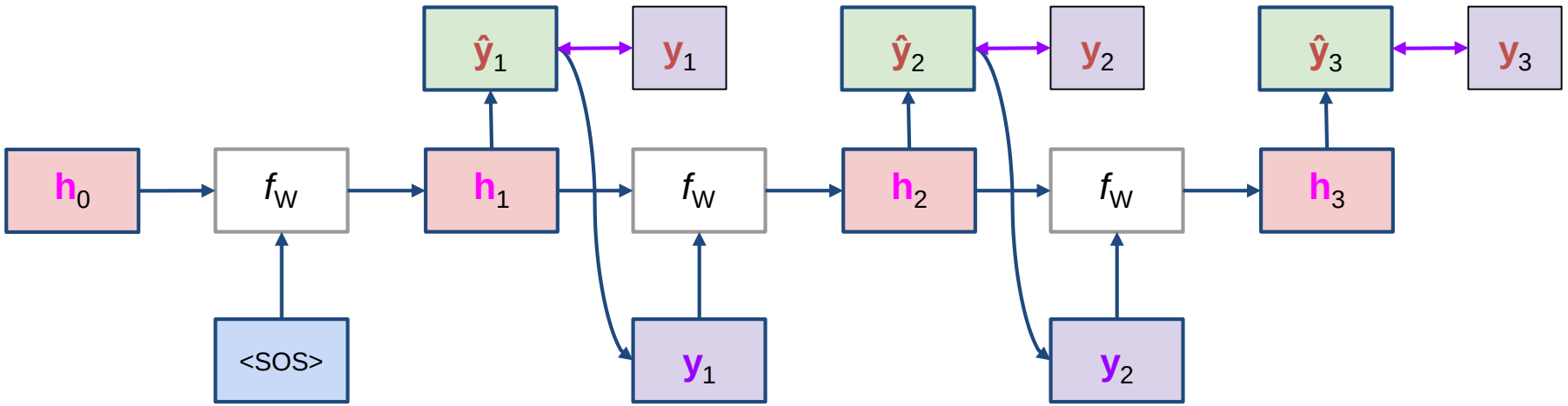
Sequence-to-Sequence Models

- Decoder : Auto-Regressive Generation
  - **Auto-regressive** input: the lagged (auto-regressive) values of the time series are used as inputs.



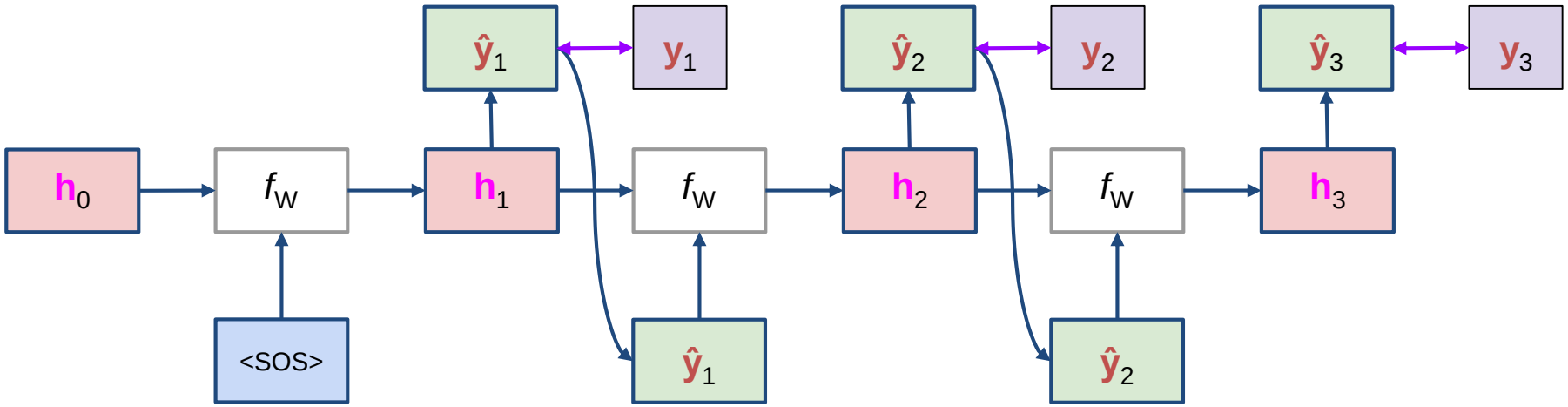
Sequence-to-Sequence Models

- Decoder : Teacher Forcing
  - At training, we use the ground truth y_{t-1} as input, because the model needs to learn what to output from the correct inputs.
    - Otherwise, the model may not train anything at the beginning!



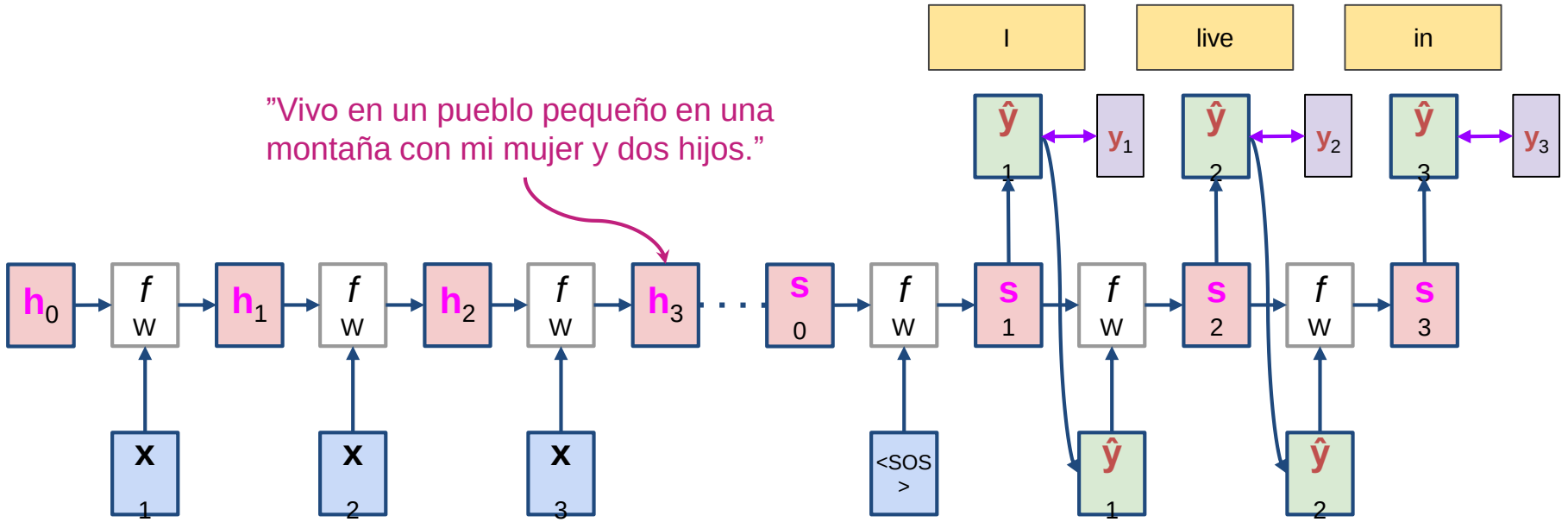
Sequence-to-Sequence Models

- Decoder : Teacher Forcing
  - At inference, we do not have access to the ground truth y_{t-1},
    so we actually feed the previous output $\hat{y}_{t-1}$ auto-regressively.



Sequence-to-Sequence Models

- Overall Sequence-to-Sequence (seq2seq) Model
  - Many-to-one as **encoder**, then one-to-many as **decoder**.
  - The input sequence is encoded as a single vector at the end of the encoder.
  - From this single vector, the decoder generates output sequence.



Comments / Q&A