CST8507: NATURAL LANGUAGE PROCESSING

WEEK#6

SEQUENCE TO SEQUENCE MODELS (SEQ2SEQ)

DEVELOPED BY

HALA OWN, PH.D.

Lesson Agenda

- Midterm(week 7)
- Bidirectional Long Short Term Memory (Bi-LSTM)
- The encoder-decoder framework
- Attention mechanisms

Midterm

Midterm is on Tuesday, Feb. 23, at 2:00 pm.

- The exam will consist of **30 questions**, including multiple-choice and true/false questions, with no essay questions.
- Material includes from week 1 – week 6
- **You will have 60 minutes to complete the exam.**
- The exam is closed book. However, you may bring **one cheat sheet**: a single **letter-size page (8.5 × 11 inches)** that may be **used on both sides**.
- **Ensure that you leave a 5 cm by 5 cm space in the top-left corner of each side of your cheat sheet for the proctor's signature. If this specific area is missing, you will not be allowed to use any cheat sheet during the exam.**
- **Try to arrive early to allow sufficient time for setup.**

Midterm

- **Read the instruction before starting your exam.**
- Write your **name and ID number** on the spaces provided on the questionnaire and Answer sheet.
- **Make sure to have your ID.**
- Read carefully the ICT exam conduct outline
- Please do not forget to bring your **HB pencils and eraser**.
- Scantron answer sheets will be provided to you before the start of the exam together with the questionnaire.
- Submit **both** the questionnaire and the Scantron answer sheet

How to Prepare

- Lecture summary slides are a good place to start:

    they don't have all the details, but make sure you understand the details underlying the main points mentioned.

- Do the labs! Make sure you understand the answers you get.
- Code-Examples demonstrated during the lecture (check lecture materials folder on Brightspace).
- Hybrid work

ALGONQUIN COLLEGE

Questions

Recap :N-garm

N-grams

$$P(w_2|w_1) = \frac{\text{count}(w_1, w_2)}{\text{count}(w_1)} \longrightarrow \text{Bigrams}$$

$$P(w_3|w_1, w_2) = \frac{\text{count}(w_1, w_2, w_3)}{\text{count}(w_1, w_2)} \longrightarrow \text{Trigrams}$$

$$\boxed{P(w_1, w_2, w_3) = P(w_1) \times P(w_2|w_1) \times P(w_3|w_2)}$$

- Large N-grams to capture dependencies between distant words
- Need a lot of space and RAM

Bigram Probability

I have a dog whose name is Lucy.
I have two cats.
they like playing with Lucy.

$$P(A \mid B) = \frac{P(A,B)}{P(B)}$$

$$P(\text{have} \mid I) = \frac{P(\text{I have})}{P(I)} = \frac{2}{2} = 1$$

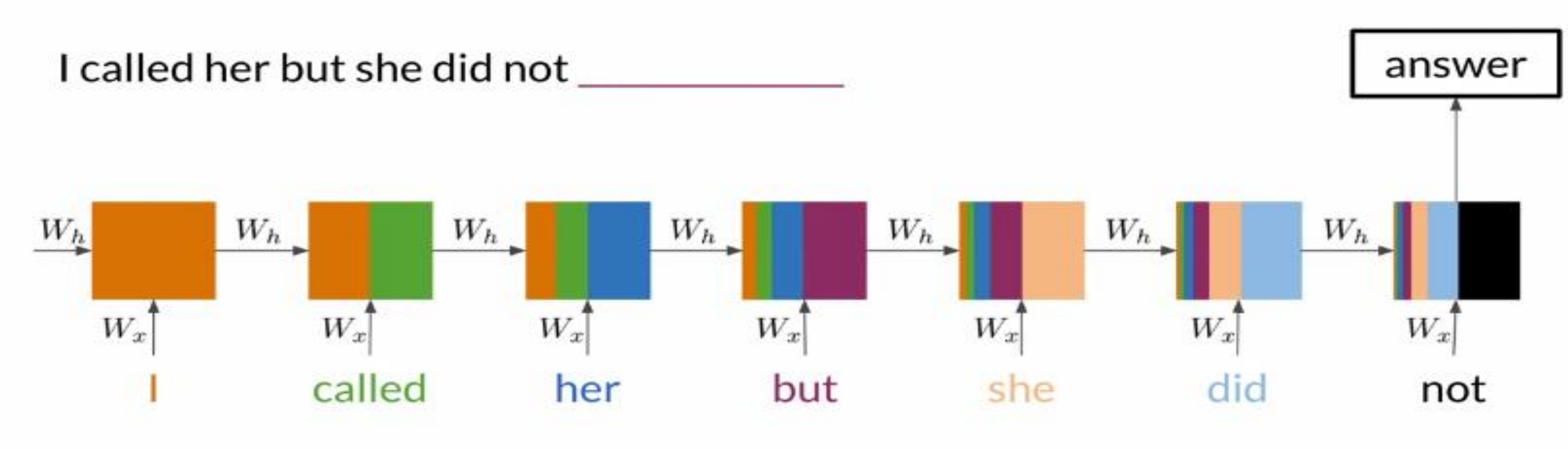
$$P(\text{two} \mid \text{have}) = \frac{P(\text{have two})}{P(\text{have})} = \frac{1}{2} = 0.5$$

$$P(\text{eating} \mid \text{have}) = \frac{P(\text{have eating})}{P(\text{have})} = \frac{0}{2} = 0$$

$$P(w_2|w_1) = \frac{C(w_1,w_2)}{\sum_w C(w_1,w)} = \frac{C(w_1,w_2)}{C(w_1)}$$

Recap :RNN

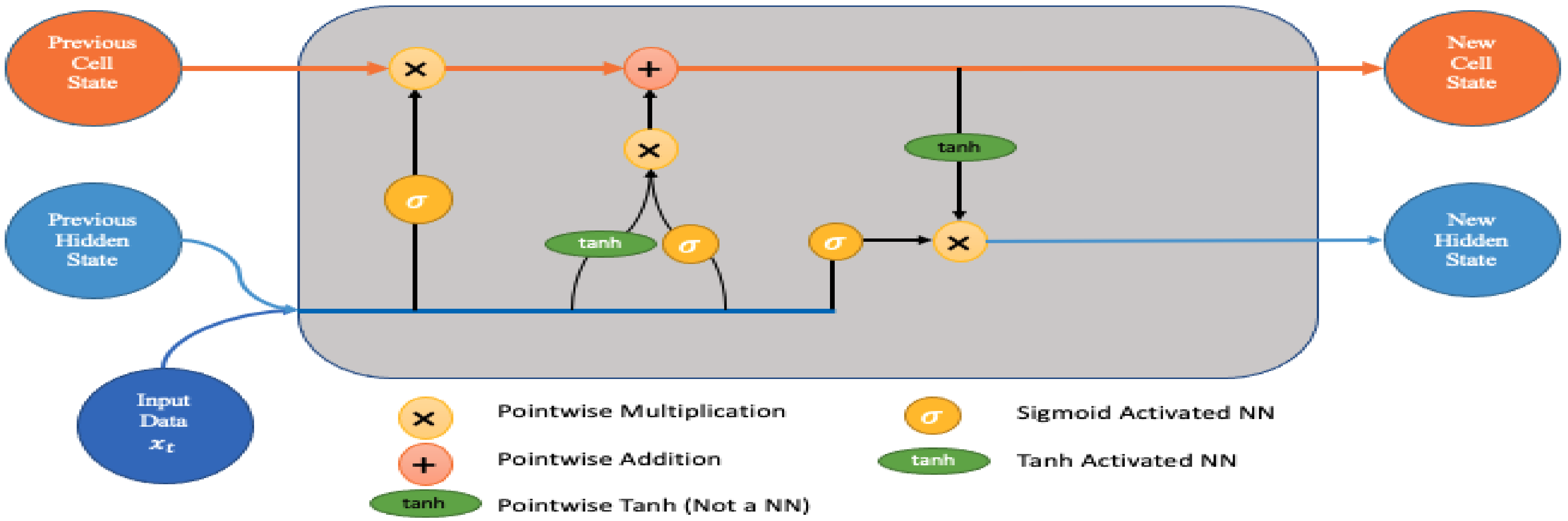
Designed to handle sequential data by maintaining a **hidden state that captures information from previous time steps.**



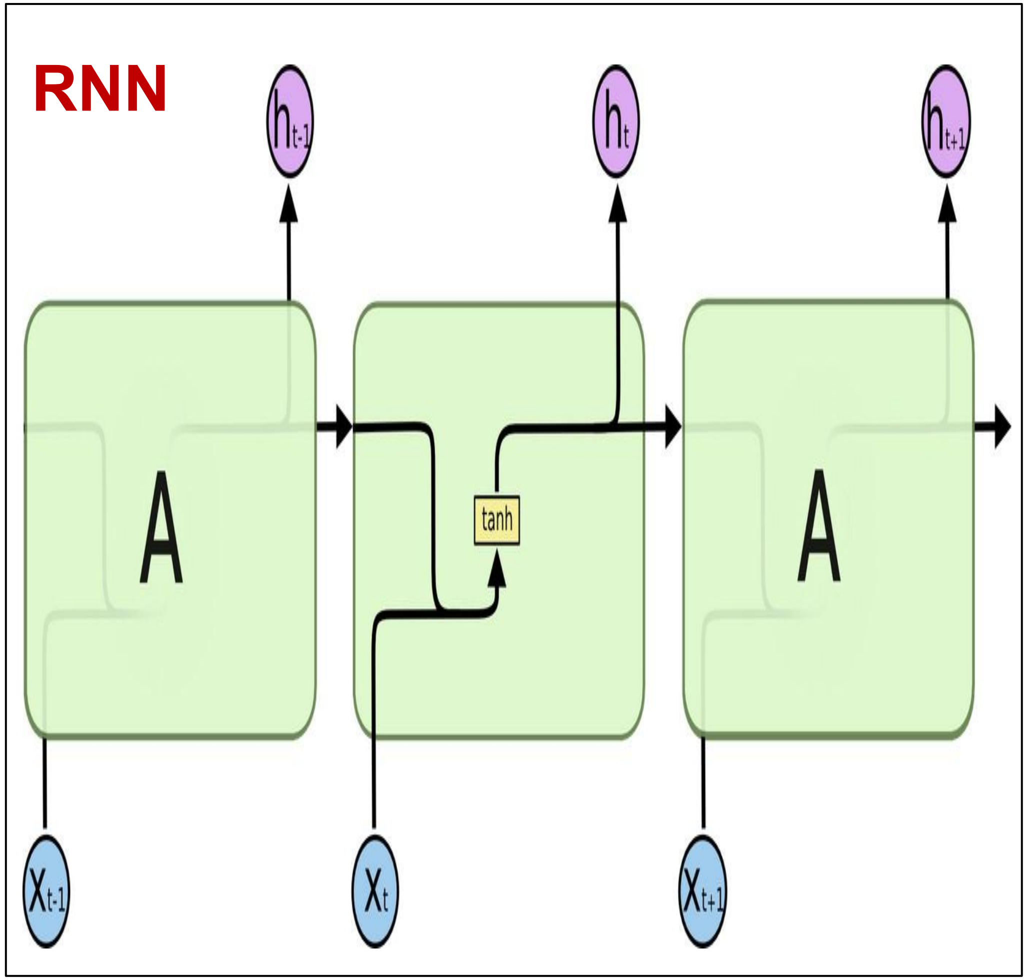
Source of image: deeplearning.ai

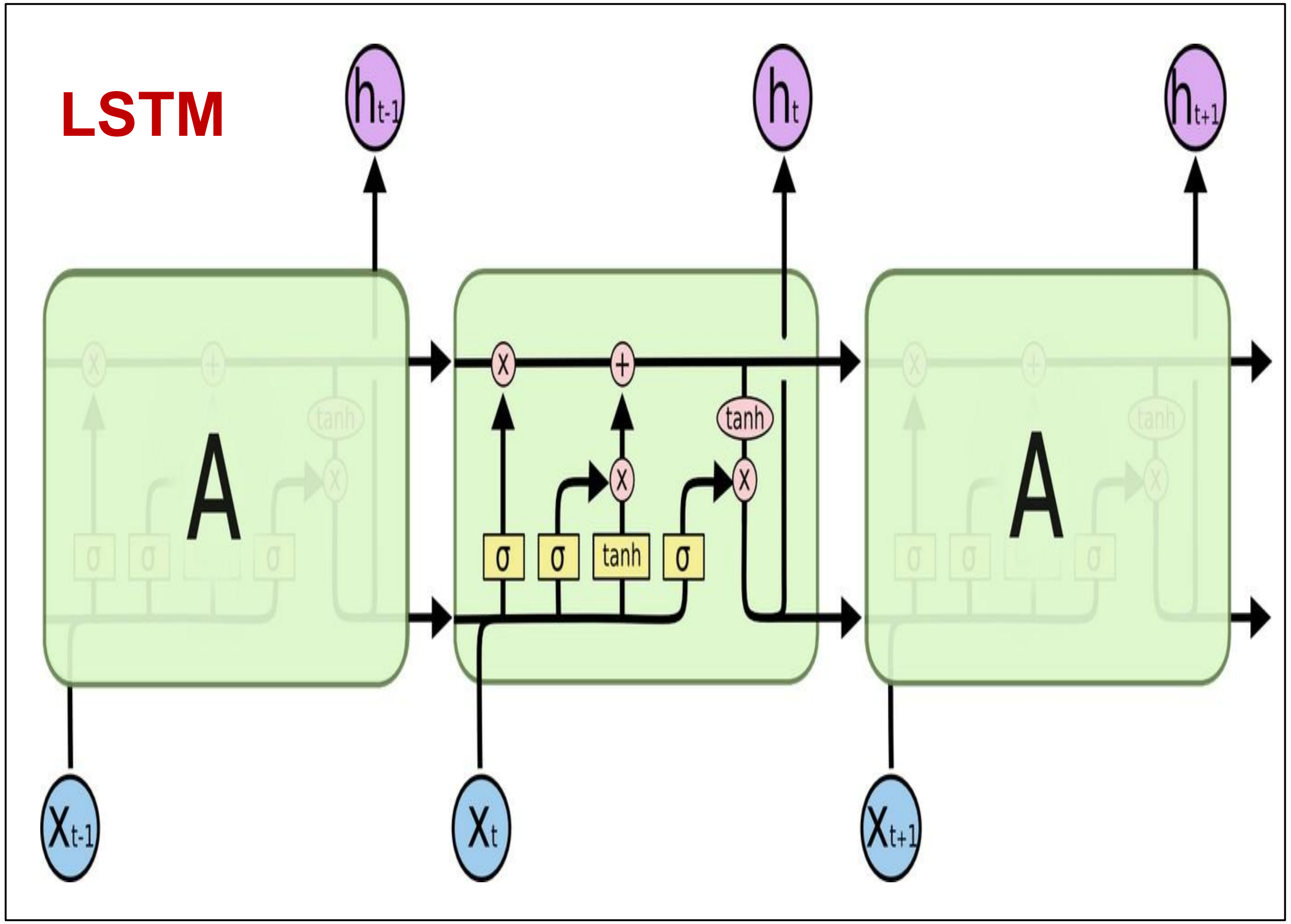
Recap: LSTM

LSTM networks are a type of RNN that can learn long-term dependencies. They use gates (input, forget, and output gates)to **control the flow of information**, making them effective for tasks requiring memory over long sequences.

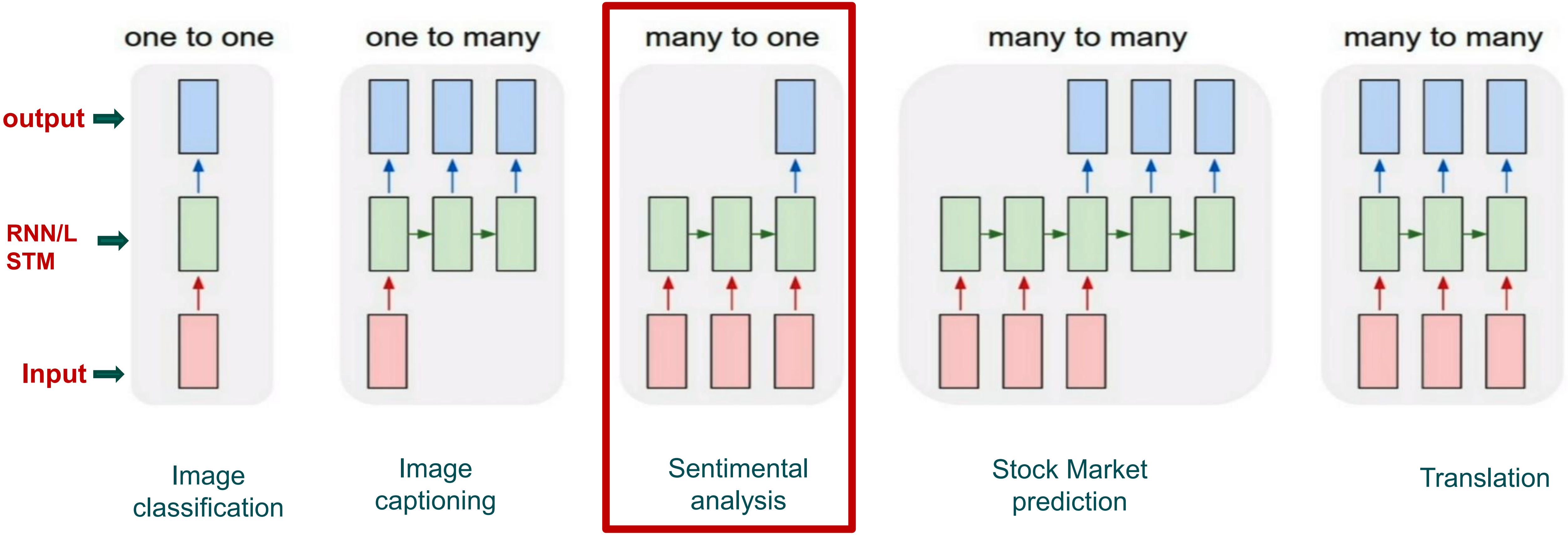


RNN vs LSTM cell



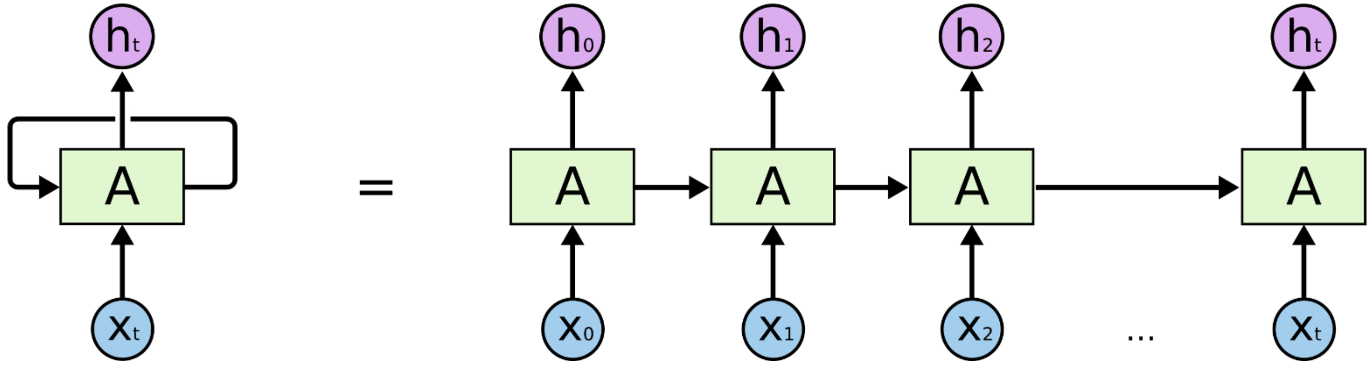


Types Of Sequence Problems in NLP Task



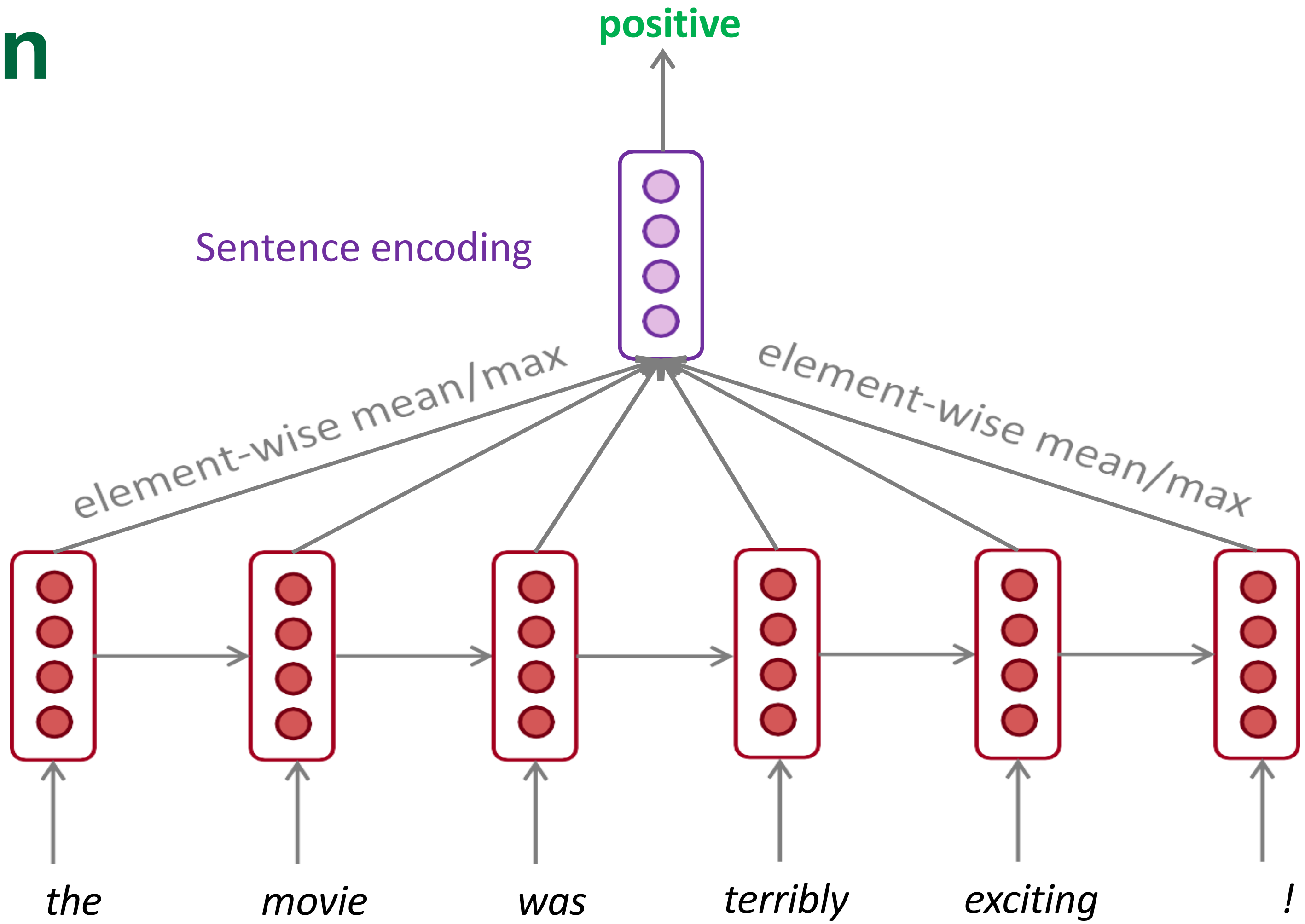
(source: http://karpathy.github.io/2015/05/21/rnn-effectiveness/)

Bidirectional Long Short-Term Memory (Bi-LSTM): Motivation

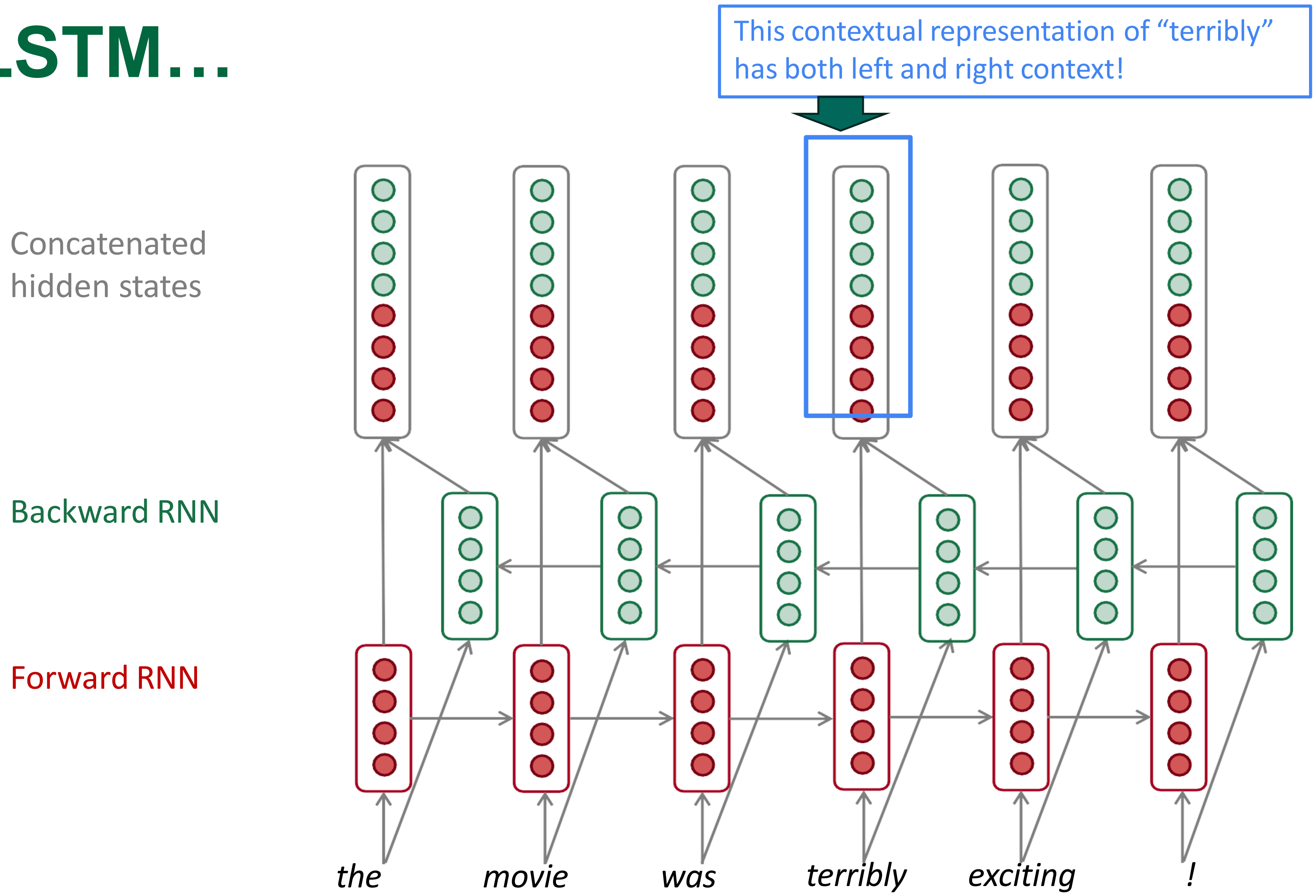


The movie was terribly exciting!

Bidirectional Long Short-Term Memory (Bi-LSTM): Motivation

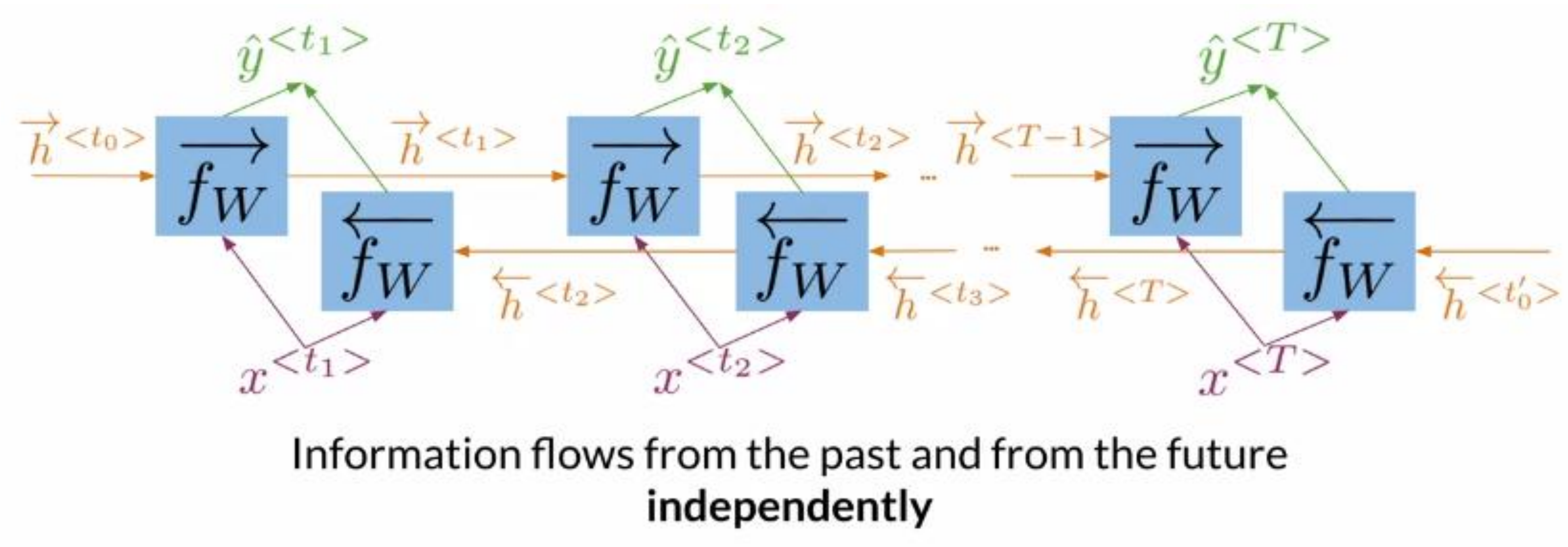


Bi-LSTM…



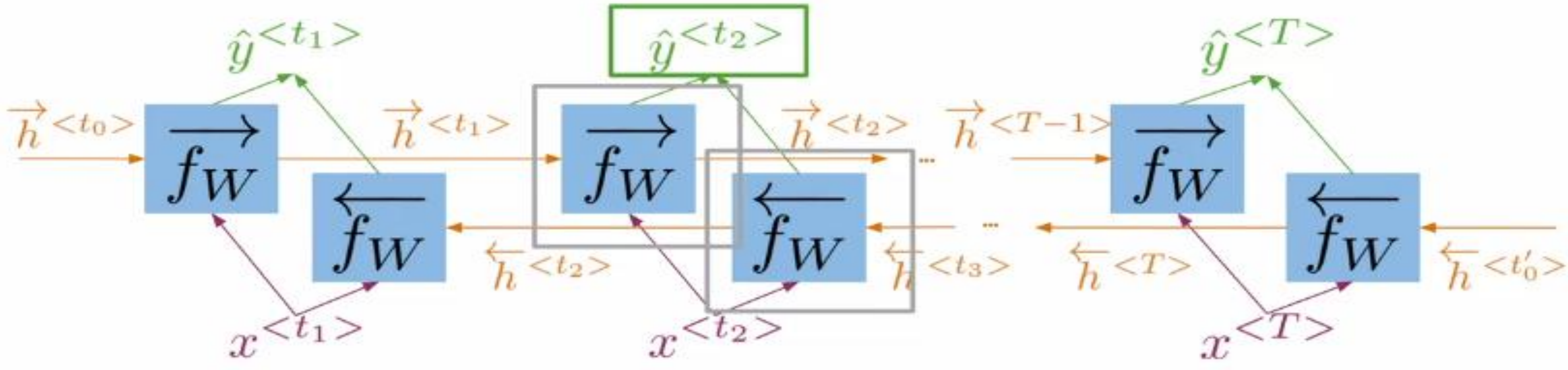
Slide credit: Daniel Jurafsky

Bi-LSTM…



Source of image: deeplearning.ai

Bi-LSTM…



$$\hat{y}^{<t>} = g(W_y[\overrightarrow{h}^{<t>}, \overleftarrow{h}^{<t>}] + b_y)$$

Bidirectional RNNs

On timestep *t*:

This is a general notation to mean "compute one forward step of the RNN" – it could be a vanilla, LSTM or GRU computation.

Forward RNN $\overrightarrow{\boldsymbol{h}}^{(t)} = \boxed{\mathrm{RNN}_{\mathrm{FW}}}(\overrightarrow{\boldsymbol{h}}^{(t-1)}, \boldsymbol{x}^{(t)})$

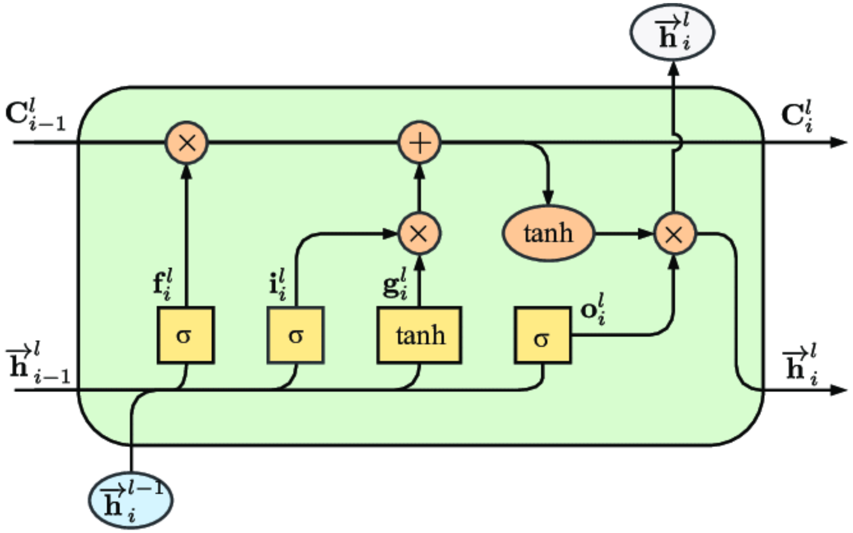
Backward RNN $\overleftarrow{\boldsymbol{h}}^{(t)} = \mathrm{RNN}_{\mathrm{BW}}(\overleftarrow{\boldsymbol{h}}^{(t+1)}, \boldsymbol{x}^{(t)})$

Generally, these two RNNs have separate weights

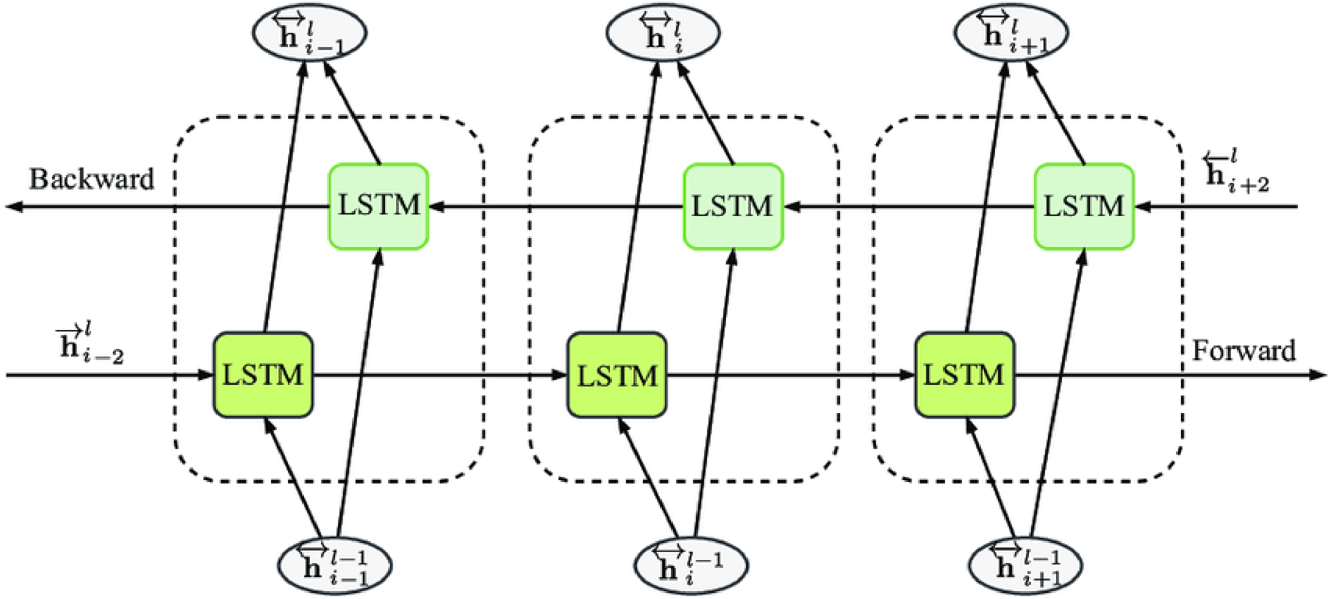
Concatenated hidden states $\boxed{\boldsymbol{h}^{(t)}} = [\overrightarrow{\boldsymbol{h}}^{(t)}; \overleftarrow{\boldsymbol{h}}^{(t)}]$

We regard this as "the hidden state" of a bidirectional RNN. This is what we pass on to the next parts of the network.

Bidirectional Long Short-Term Memory (Bi-LSTM)…



Single forward LSTM layer



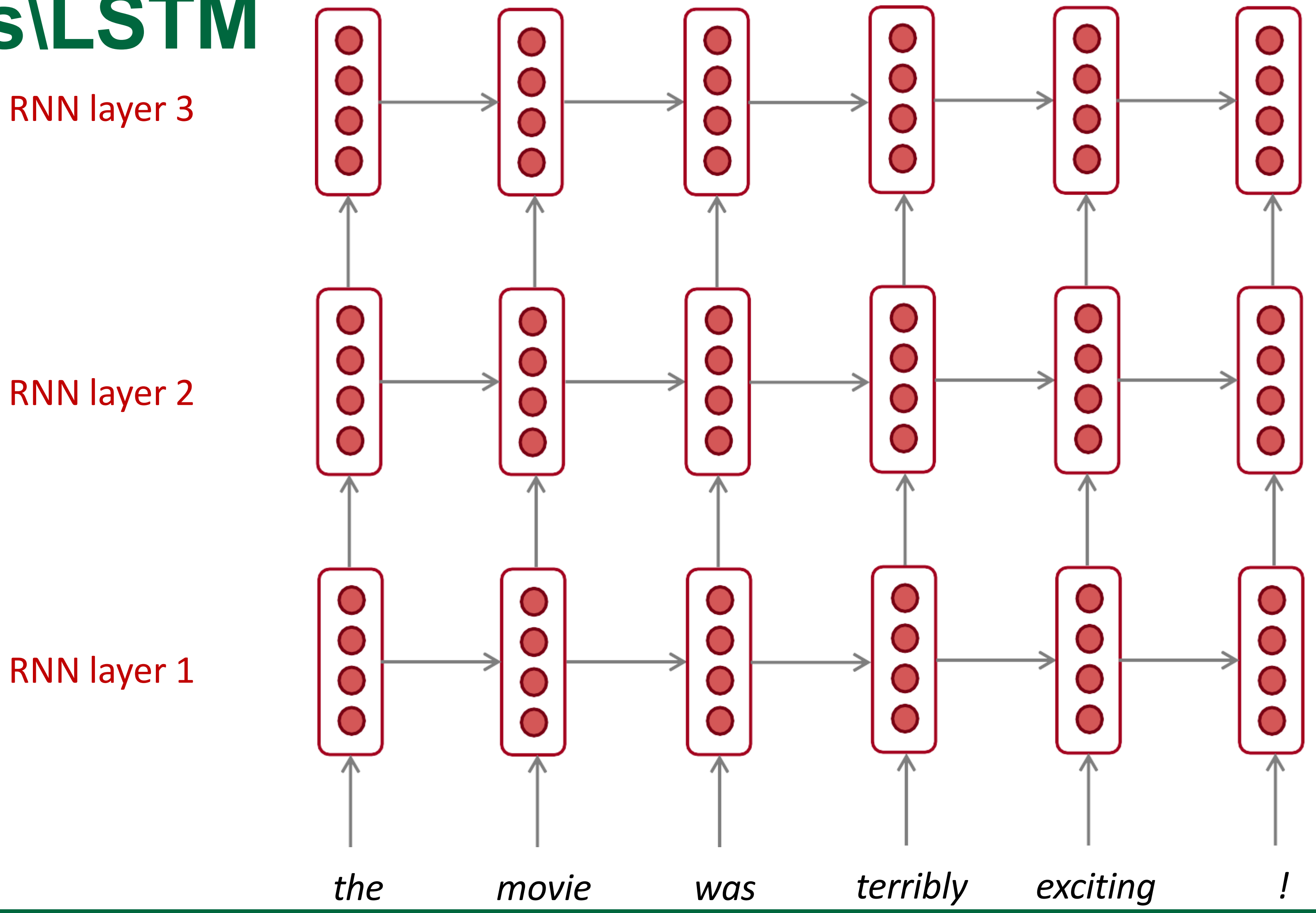
Bi-LSTM model

Bi-LSTM model Architecture for Classification

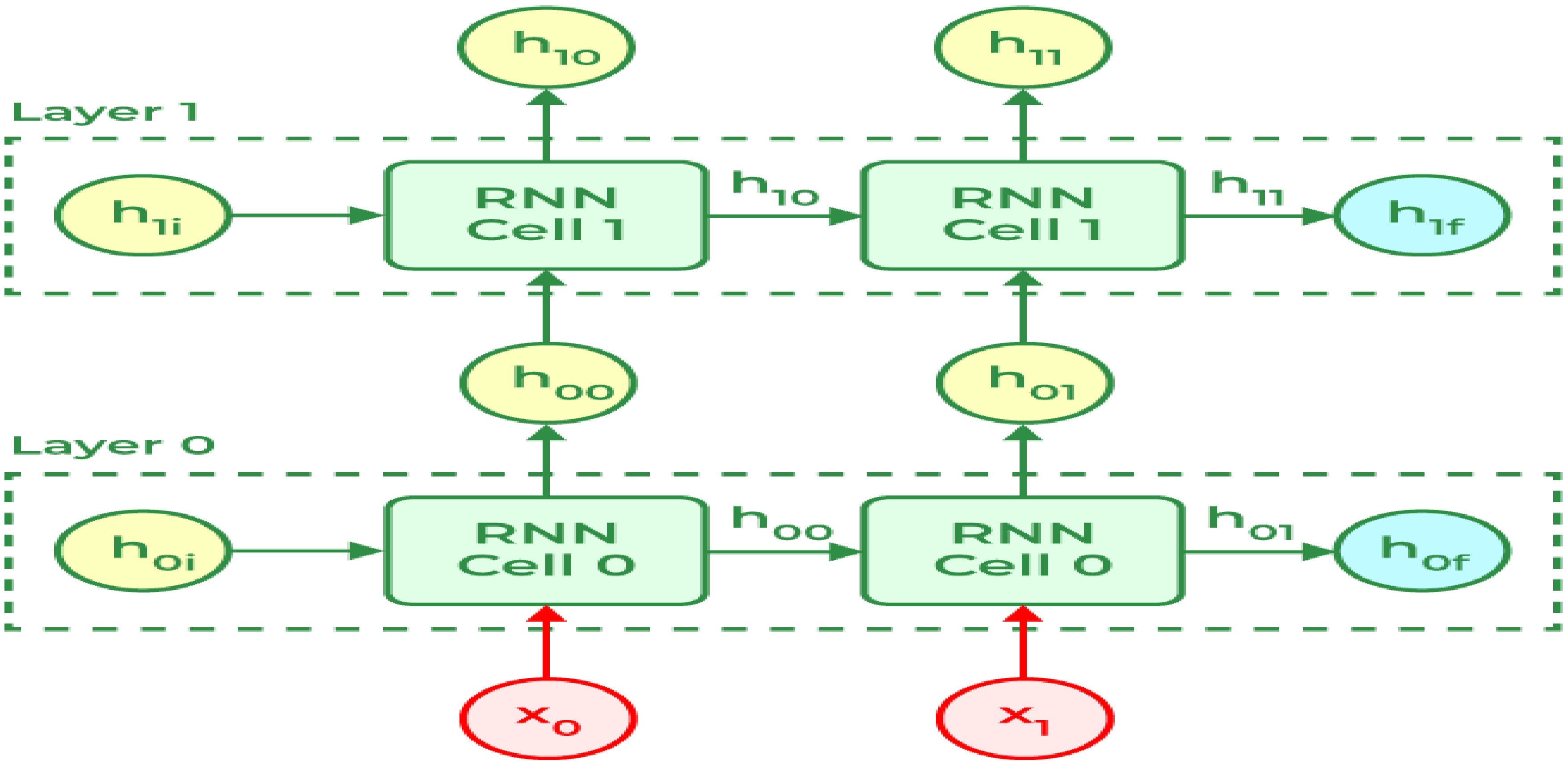
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, Bidirectional,
LSTM, Dense
model = Sequential([
    Embedding(input_dim=vocab_size,
              output_dim=embedding_dim,
              input_length=max_len),
    Bidirectional(LSTM(n_lstm)),
    Dense(1, activation='sigmoid')
])
```

Multi-layer RNNs\LSTM

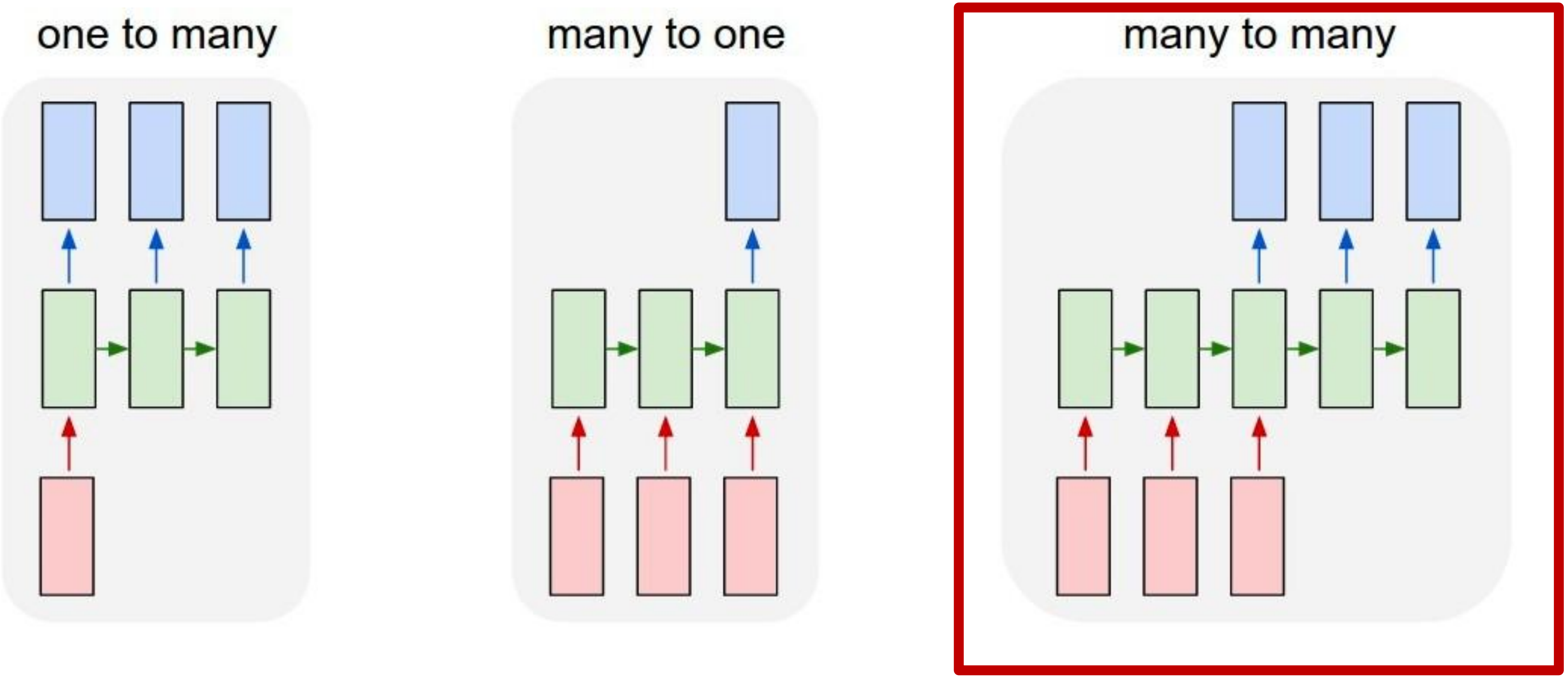
The hidden states from RNN layer *i* are the inputs to RNN layer *i+1*



Multi-layer RNNs…



Types of Sequence Problems



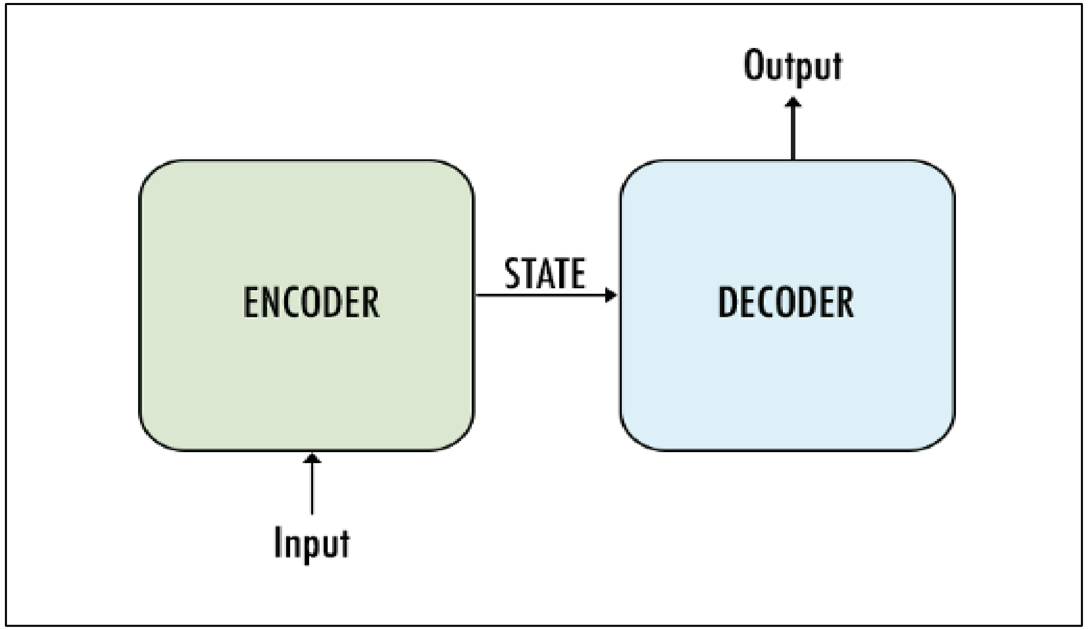
(source: http://karpathy.github.io/2015/05/21/rnn-effectiveness/)

Introduction to Sequence-to-Sequence (Seq2Seq)

- Seq2Seq is a type of model used to transform one sequence into another sequence.
- Commonly used in tasks where the input and output are sequences of varying lengths.

(Encoder-Decoder) Model

solution to Seq2Seq task



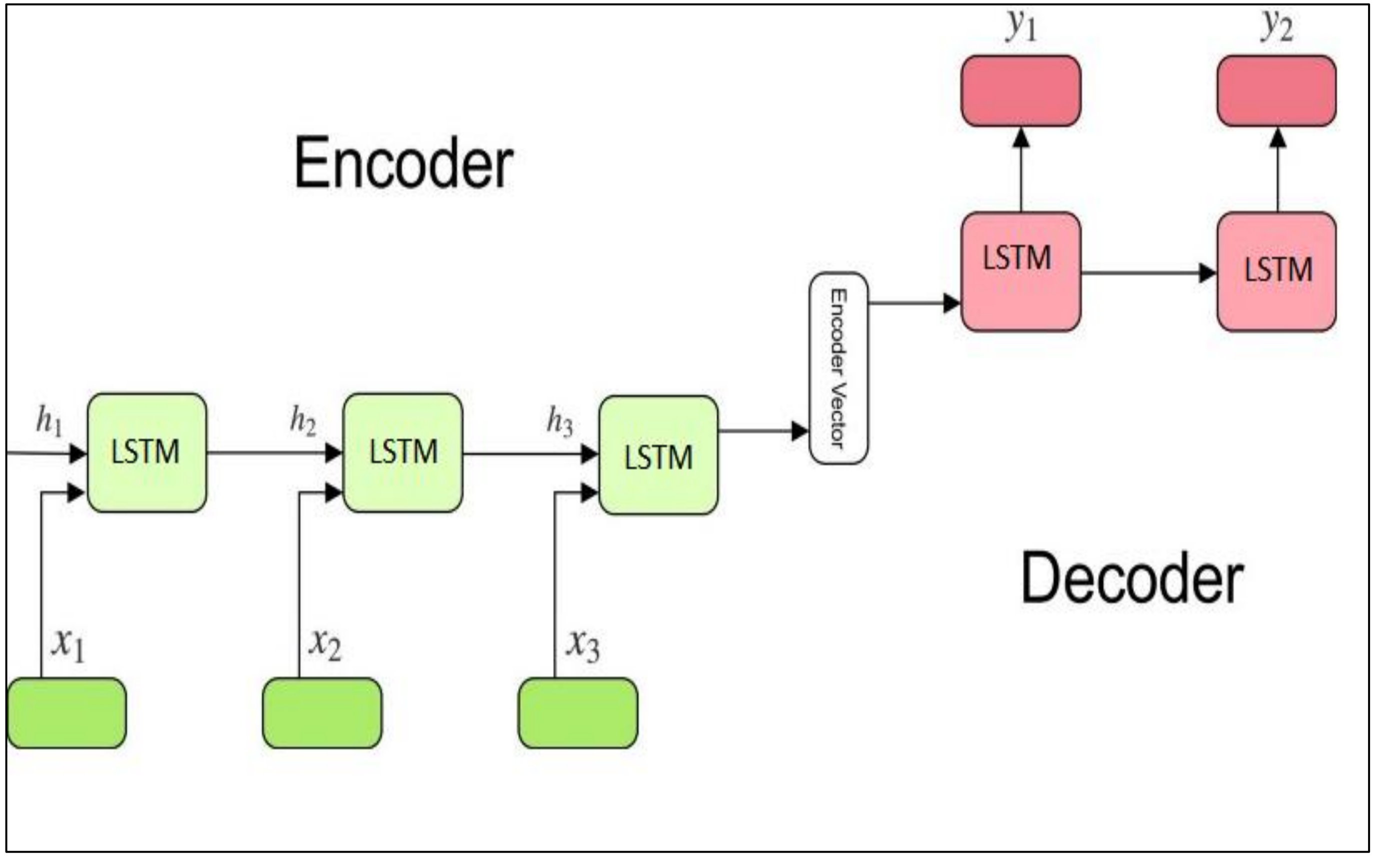


Image source: https://pradeep-dhote9.medium.com/seq2seq-encoder-decoder-lstm-model-1a1c9a43bbac

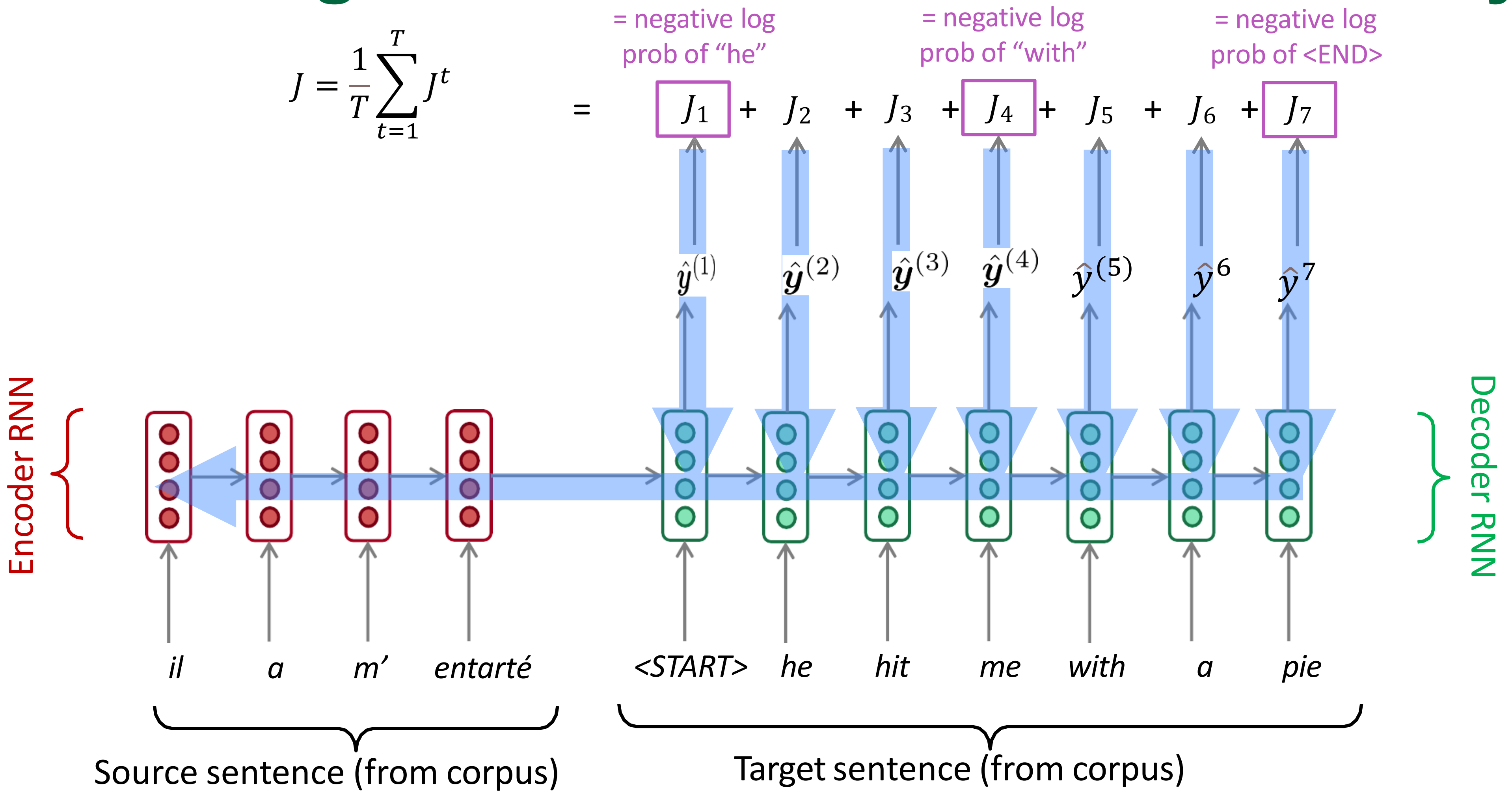
Machine Translation (MT)

- The sequence-to-sequence model is an example of a **Conditional Language Model**.
  - **Language Model :** task is predicting the next word of the target sentence *y*
  - **Conditional** :predictions are *also* conditioned on the source sentence *x*

- MT directly calculates : $P(y|x)$

$$P(y|x) = P(y_1|x)\, P(y_2|y_1, x)\, P(y_3|y_1, y_2, x) \ldots \underbrace{P(y_T|y_1, \ldots, y_{T-1}, x)}_{\text{Probability of next target word, given target words so far and source sentence } x}$$

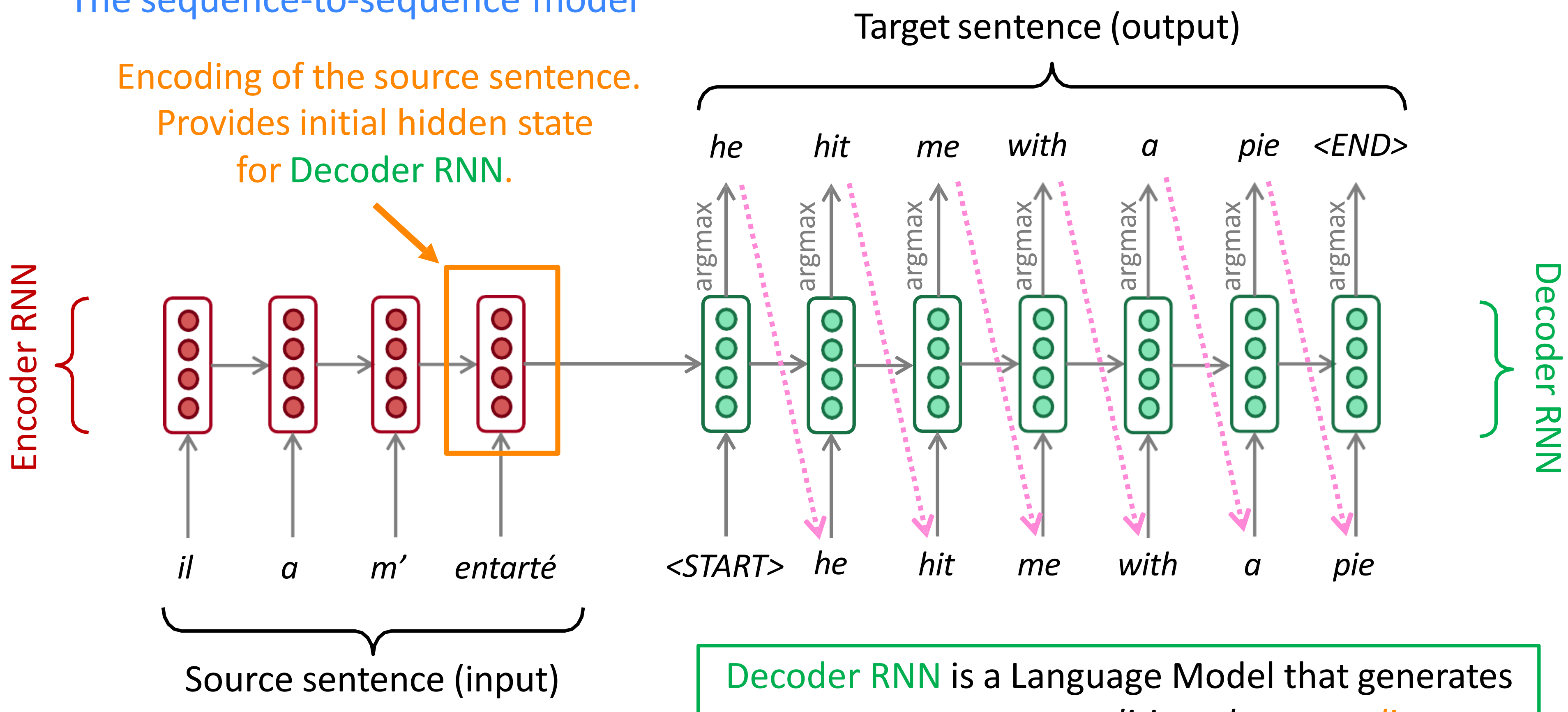
Training a Neural Machine Translation system



Seq2seq is optimized as a **single system.**
Backpropagation operates *"end-to-end"*.

Neural Machine Translation(Testing)

The sequence-to-sequence model



Decoder RNN is a Language Model that generates target sentence, *conditioned on encoding*.

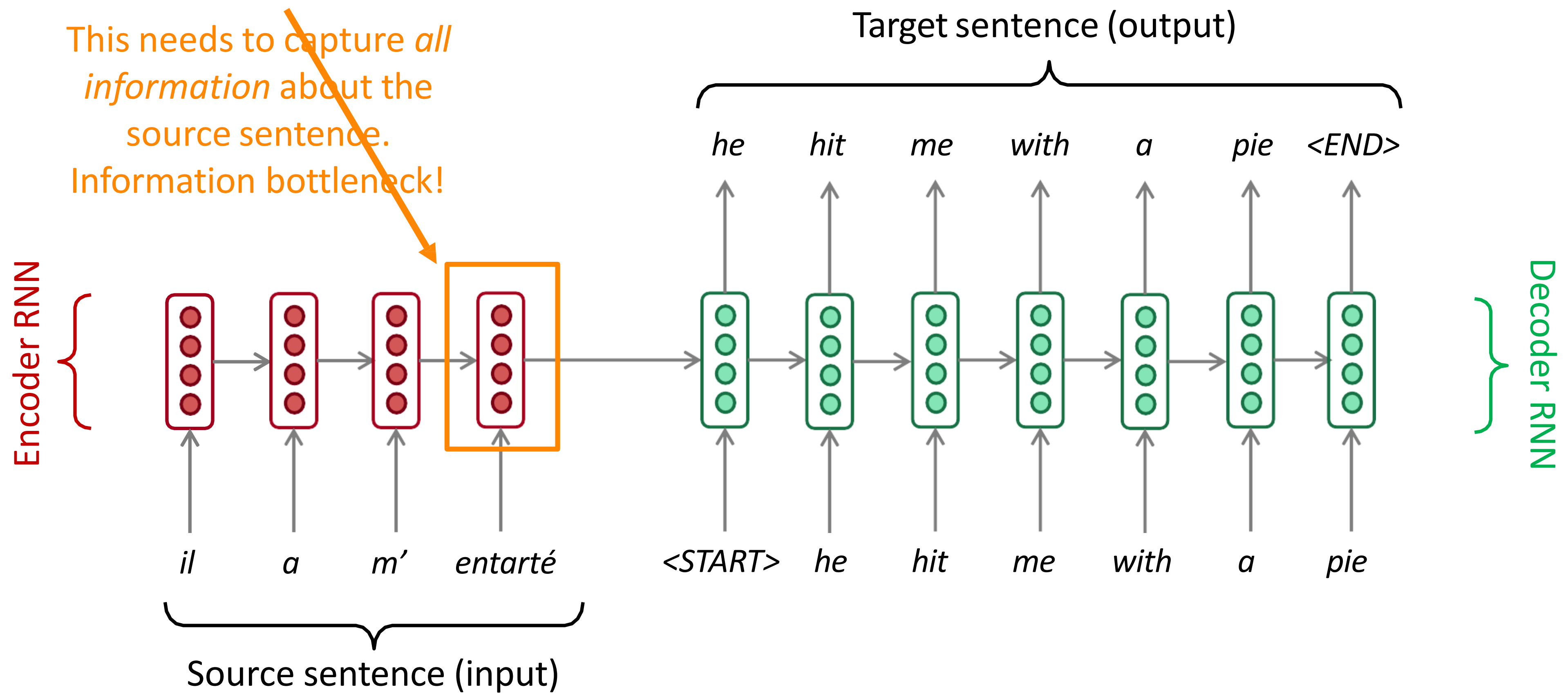
Encoder RNN produces an encoding of the source sentence.

Note: This diagram shows **test time** behavior: decoder output is fed in ·····> as next step's input

Sequence-to-sequence: bottlenecks problem

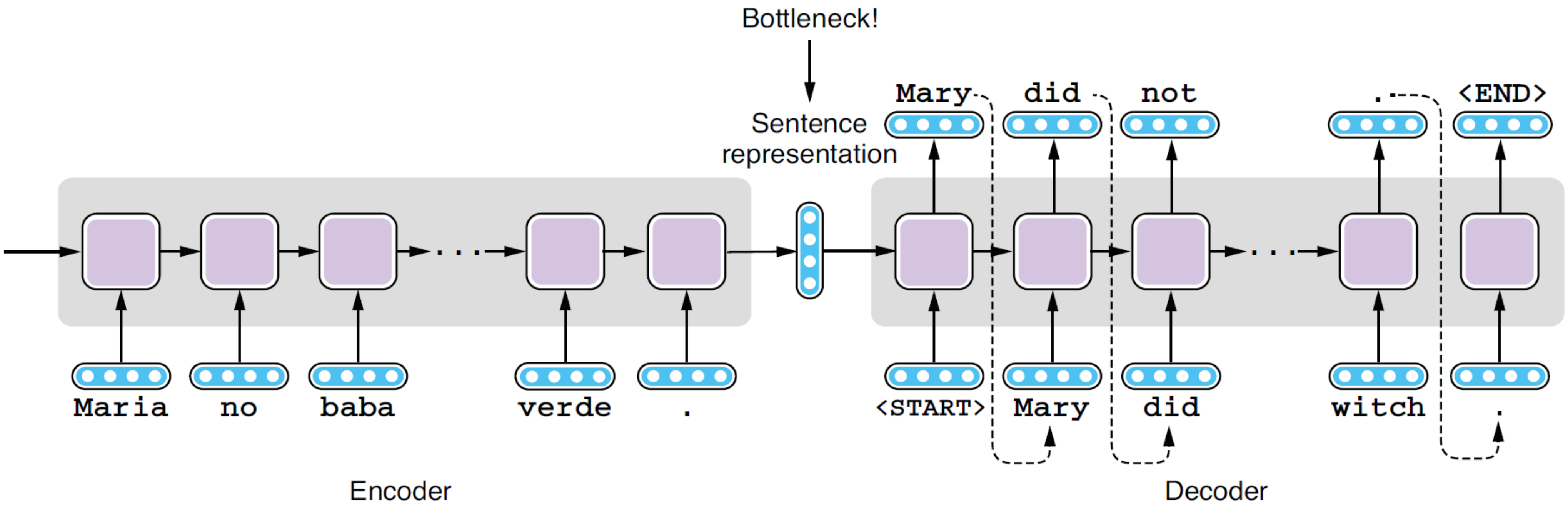
Encoding of the source sentence.

This needs to capture *all information* about the source sentence. Information bottleneck!



Problems with this architecture?

Sequence-to-sequence: Limitations



Pair of RNN used for translation

Solution with Attention

Image source:https://www.directenergyregulatedservices.com/blog/kw-vs-kwh-whats-difference

What is attention?

- Attention is a **weighted average over a set of inputs**

- How should we compute this weighted average?

  Compute pairwise similarity between each encoder hidden state and decoder hidden state.

  Convert pairwise similarity scores to probability **distribution** (using softmax) over encoder hidden states and compute weighted average

ALGONQUIN COLLEGE

Attention

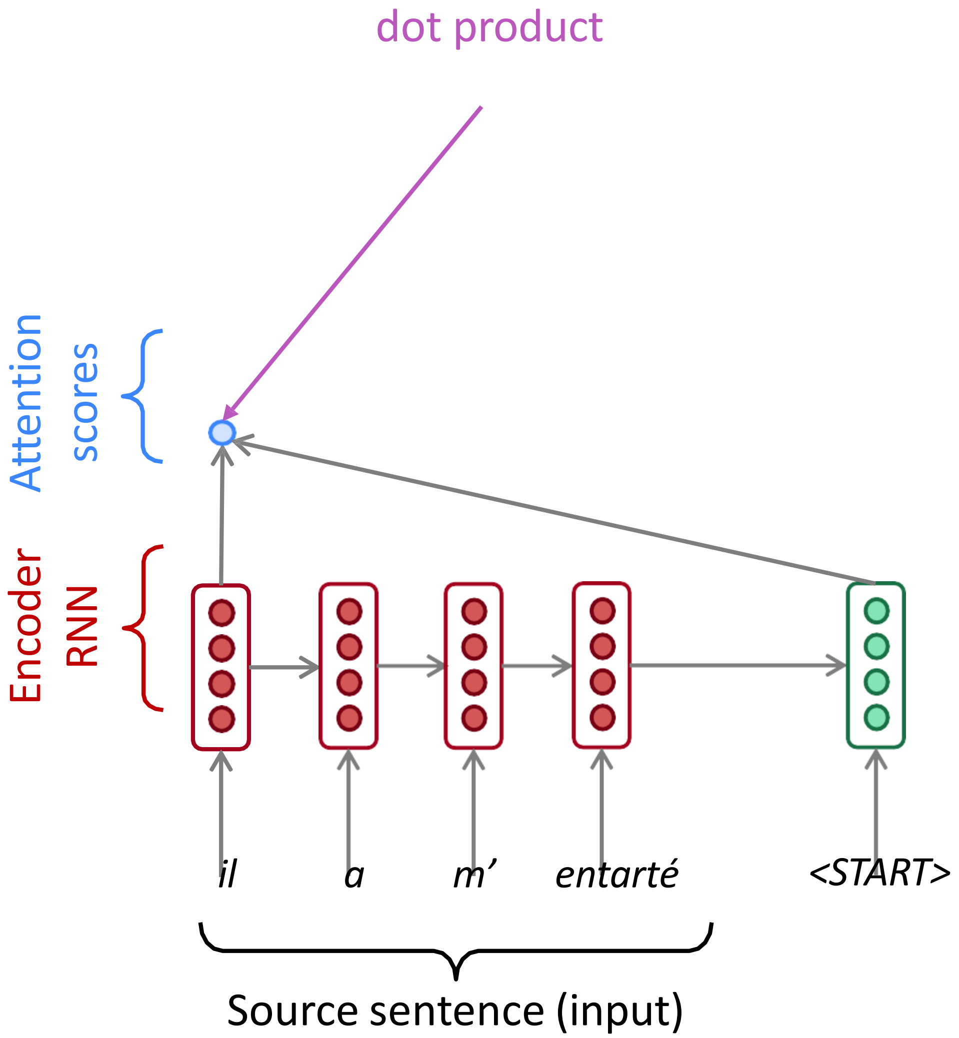
Solution to the **bottleneck problem**.

Benefits

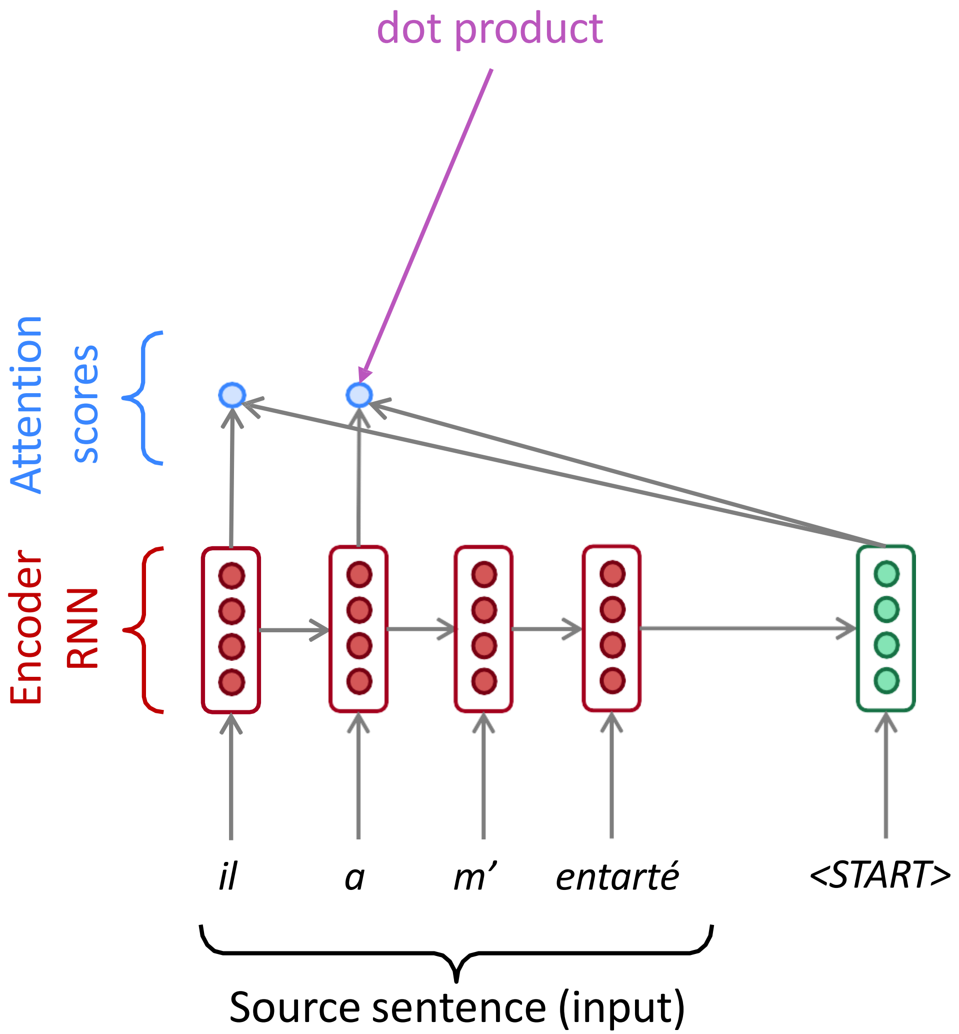
- Improved handling of **variable-length input sequences**.
- Enhanced modeling of **long-range dependencies**.
- Better performance in tasks where certain parts of the input sequence are **more relevant** to specific parts of the output sequence.

Core idea: on **each step** of the decoder, **use direct connection** to the encoder to focus on a particular part of the source sequence

Sequence-to-sequence with attention

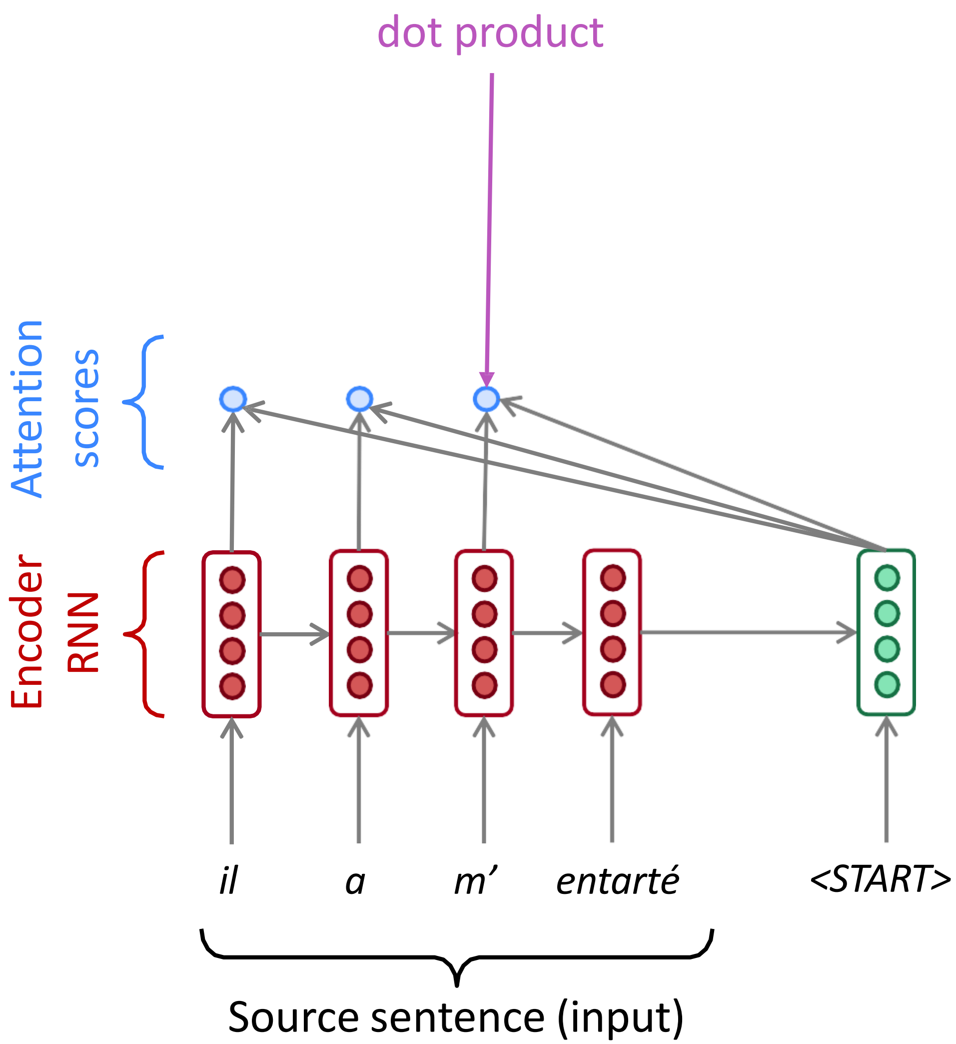


Sequence-to-sequence with attention…

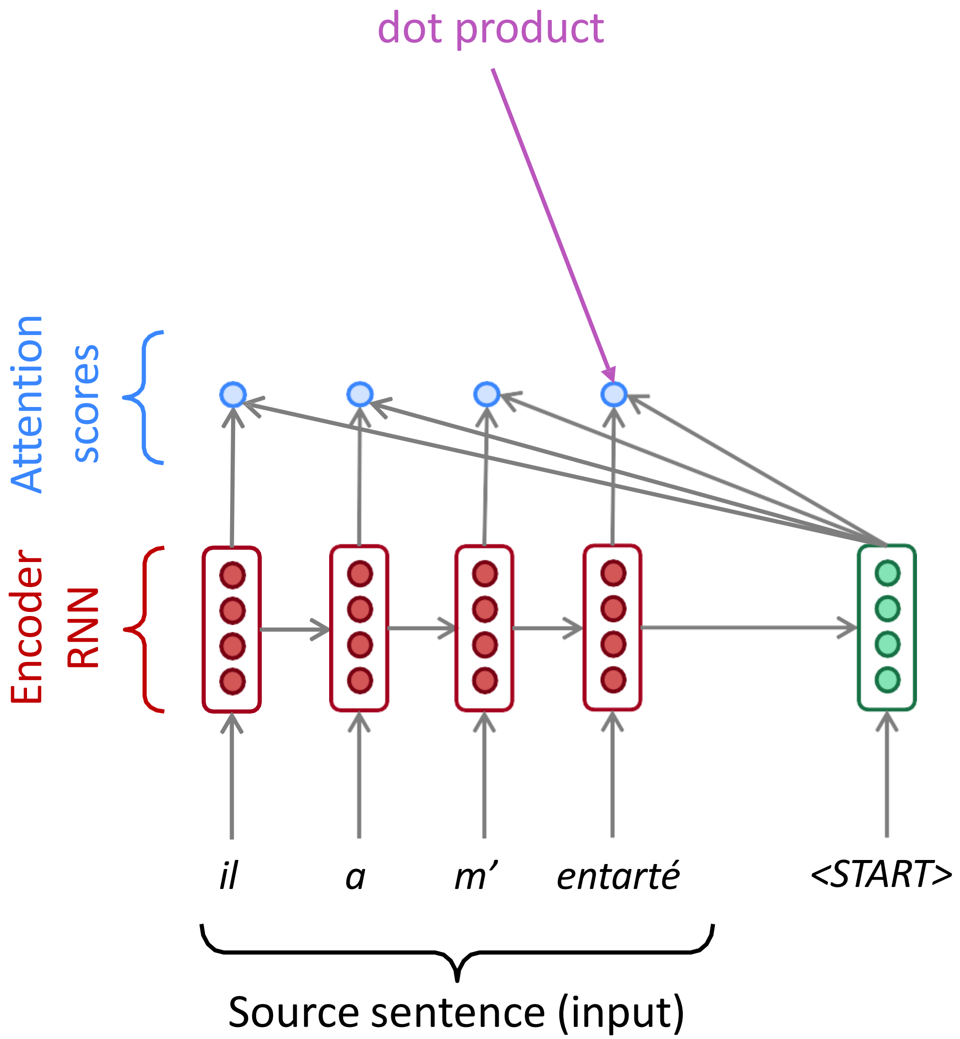


Slide credit: Daniel Jurafsky

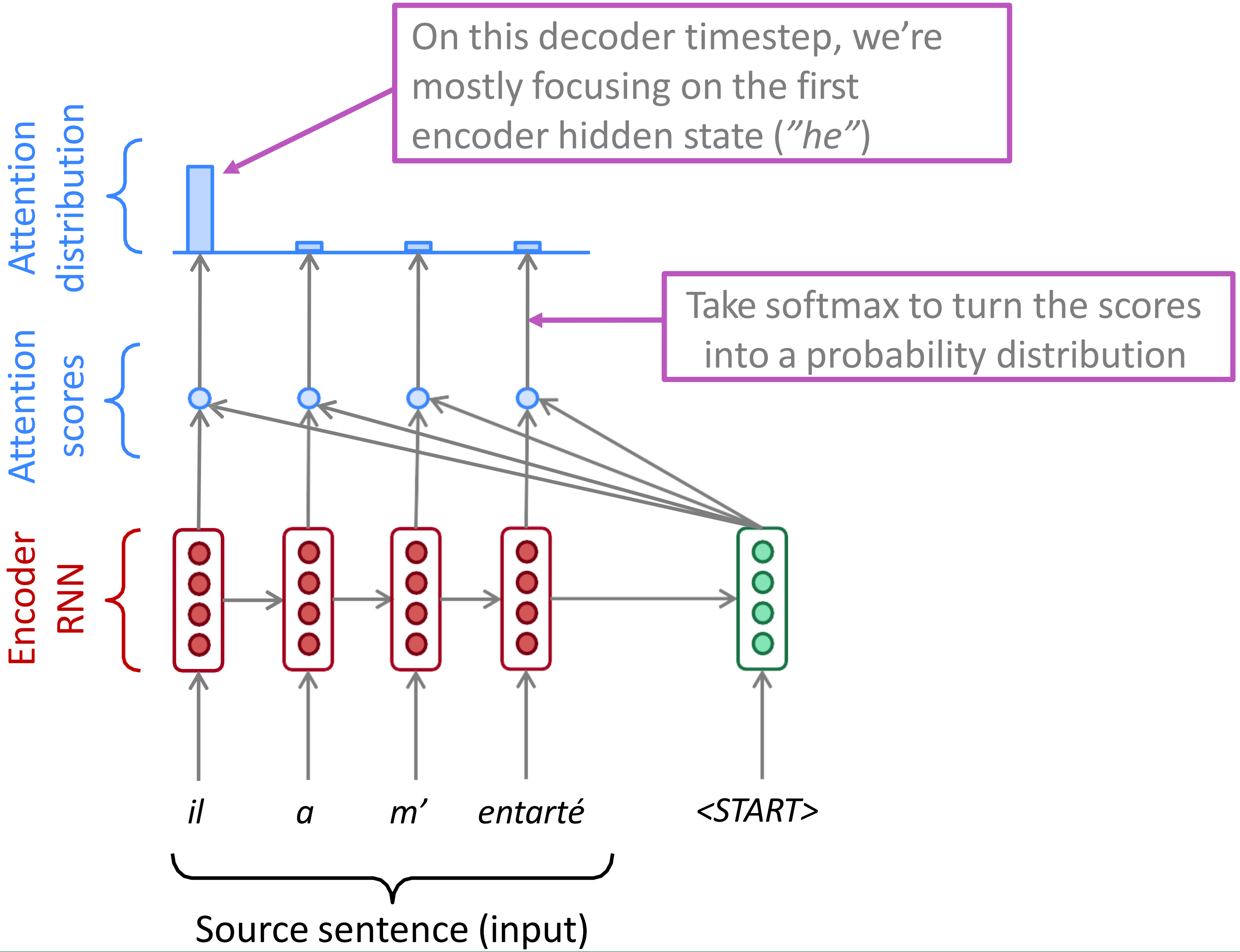
Sequence-to-sequence with attention…



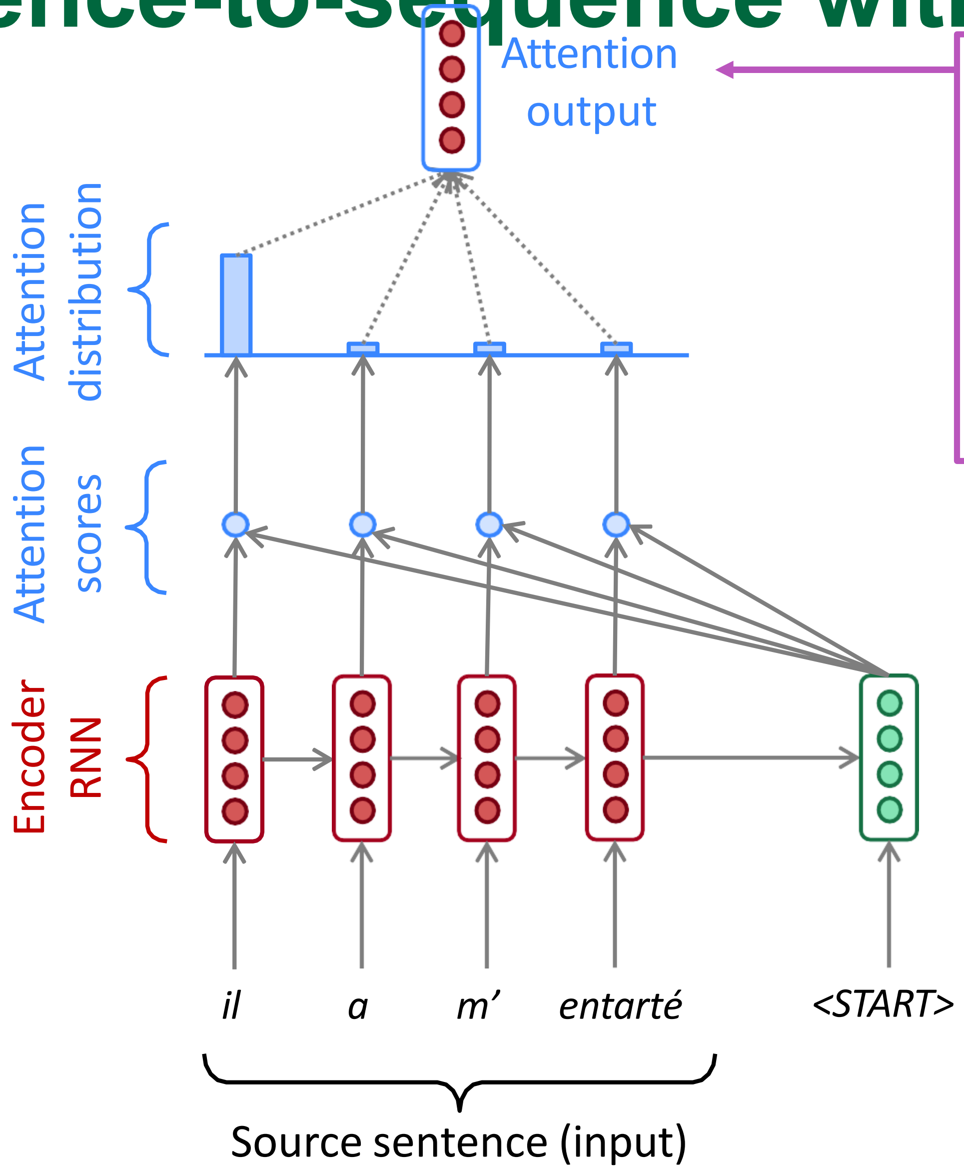
Sequence-to-sequence with attention…



Sequence-to-sequence with attention…



Sequence-to-sequence with attention…

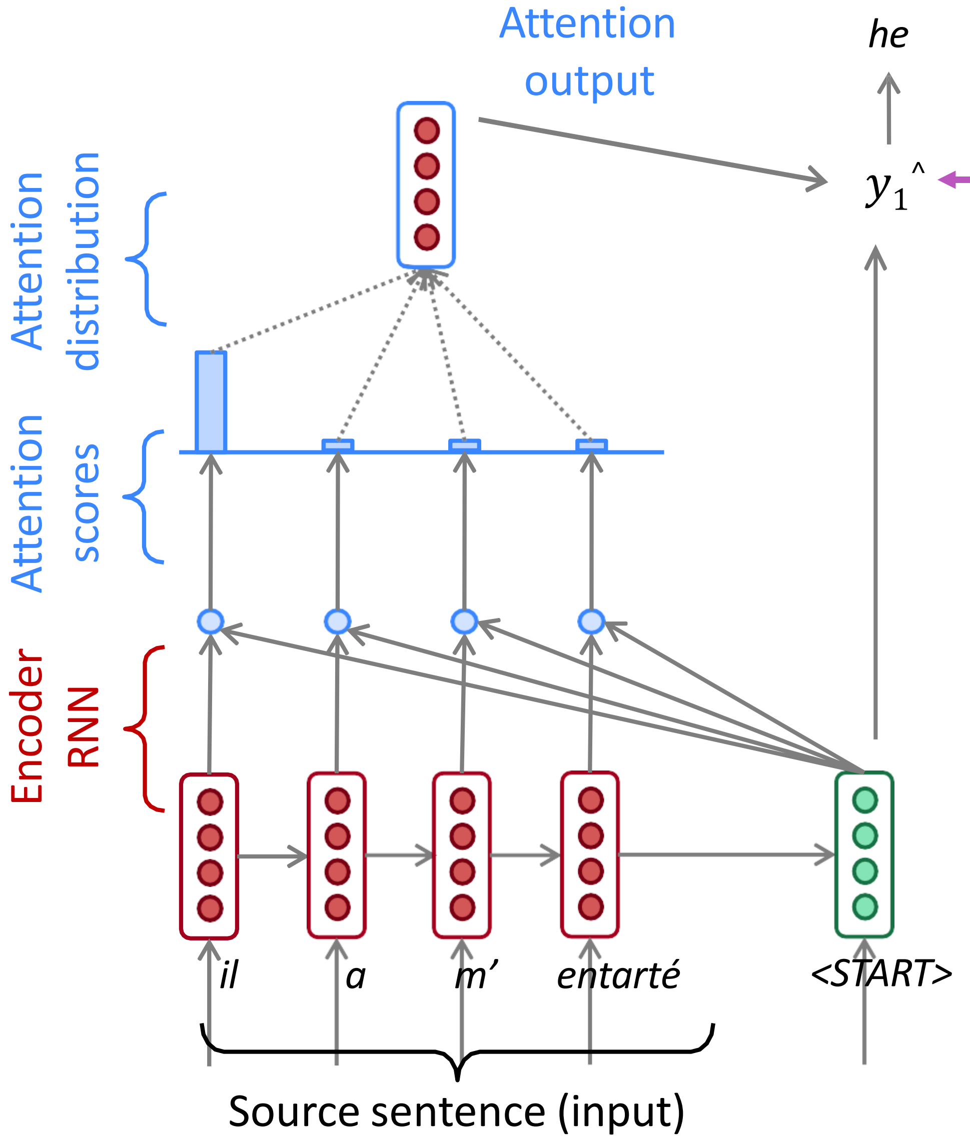


Use the attention distribution to take a **weighted sum** of the encoder hidden states.

The attention output mostly contains information from the hidden states that received high attention.

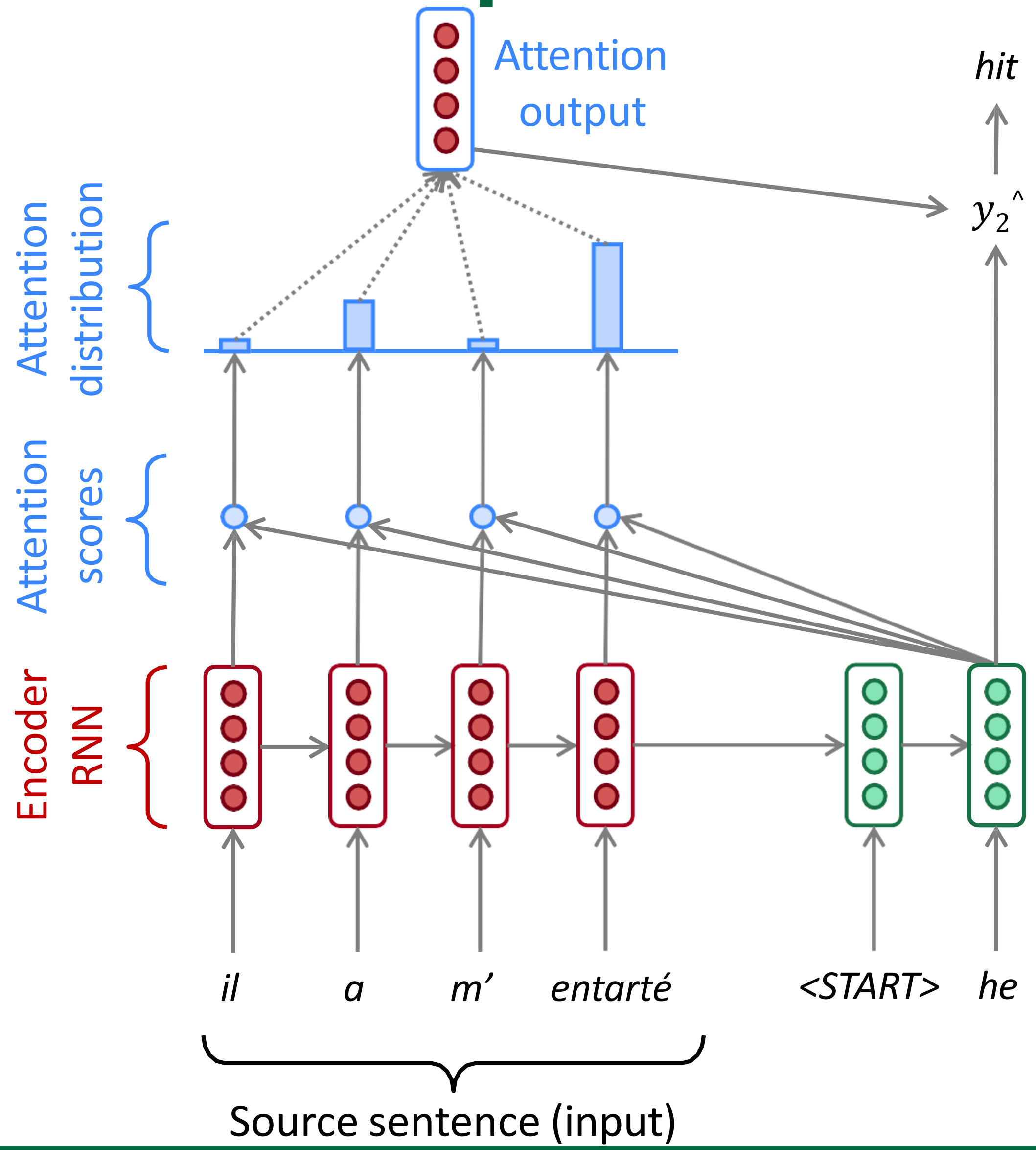
Slide credit: Daniel Jurafsky

Sequence-to-sequence with attention…



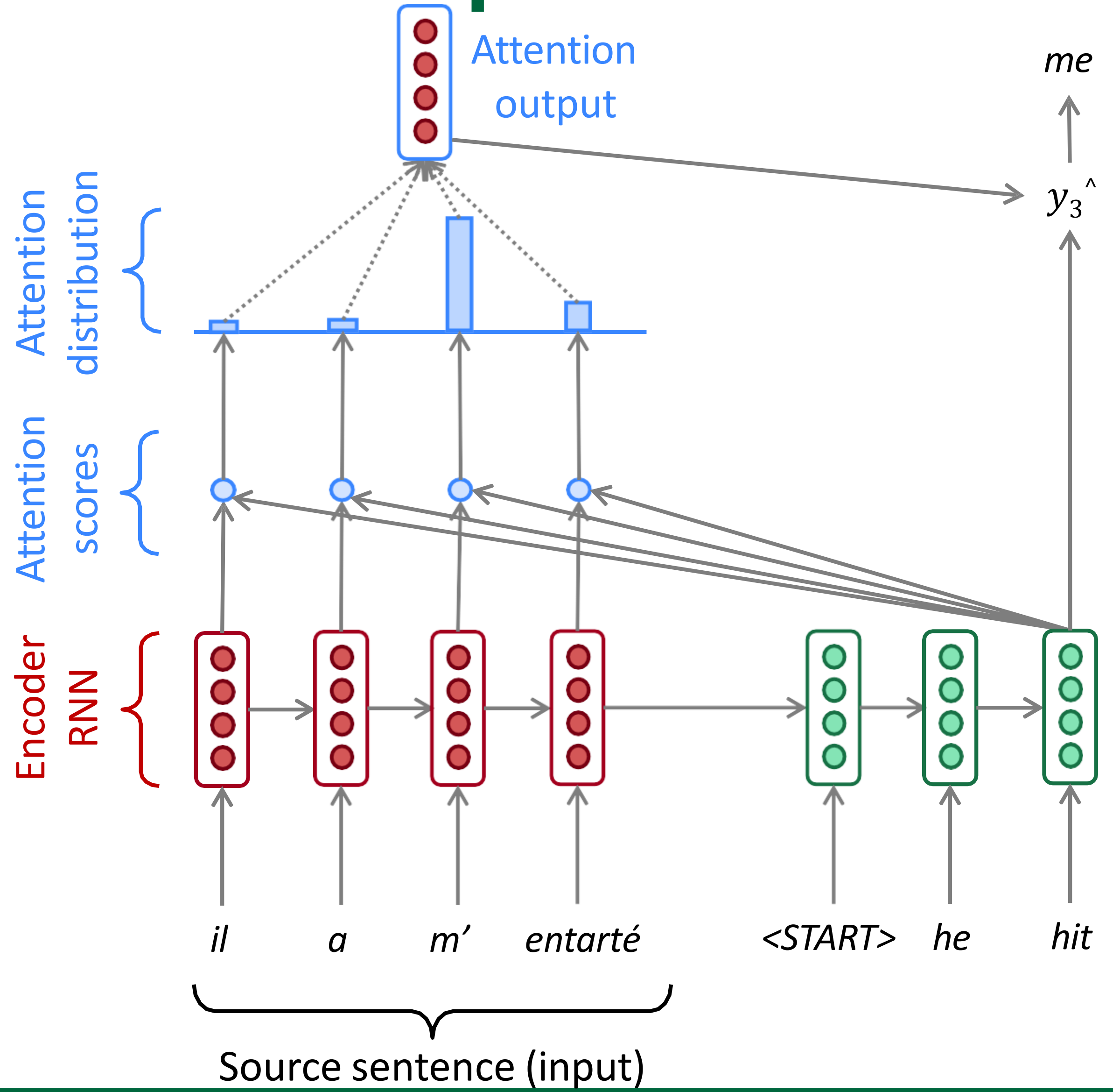
Concatenate attention output with decoder hidden state, then use to compute y_1^ as before

Sequence-to-sequence with attention



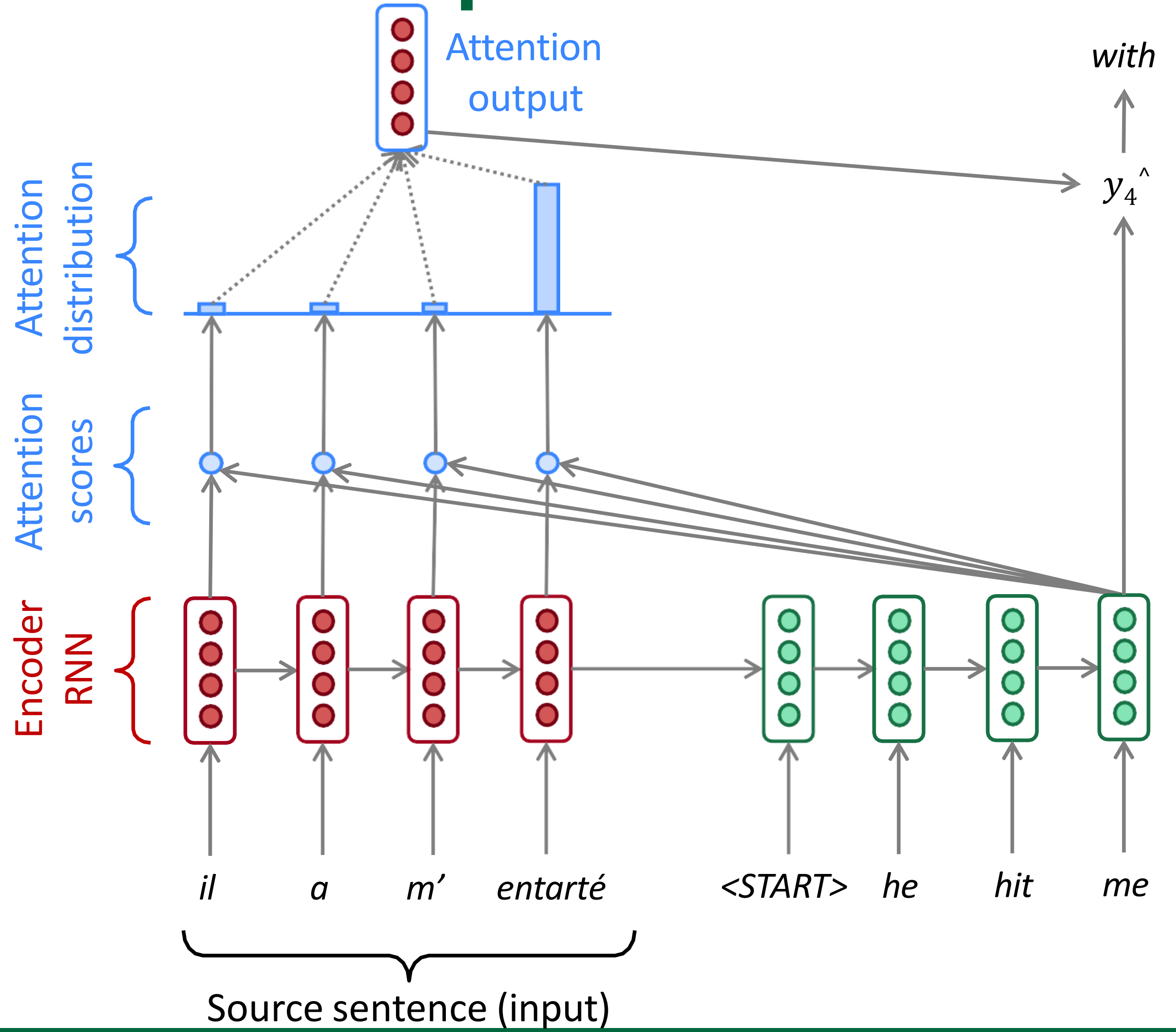
Slide credit: Daniel Jurafsky

Sequence-to-sequence with attention…



Slide credit: Daniel Jurafsky

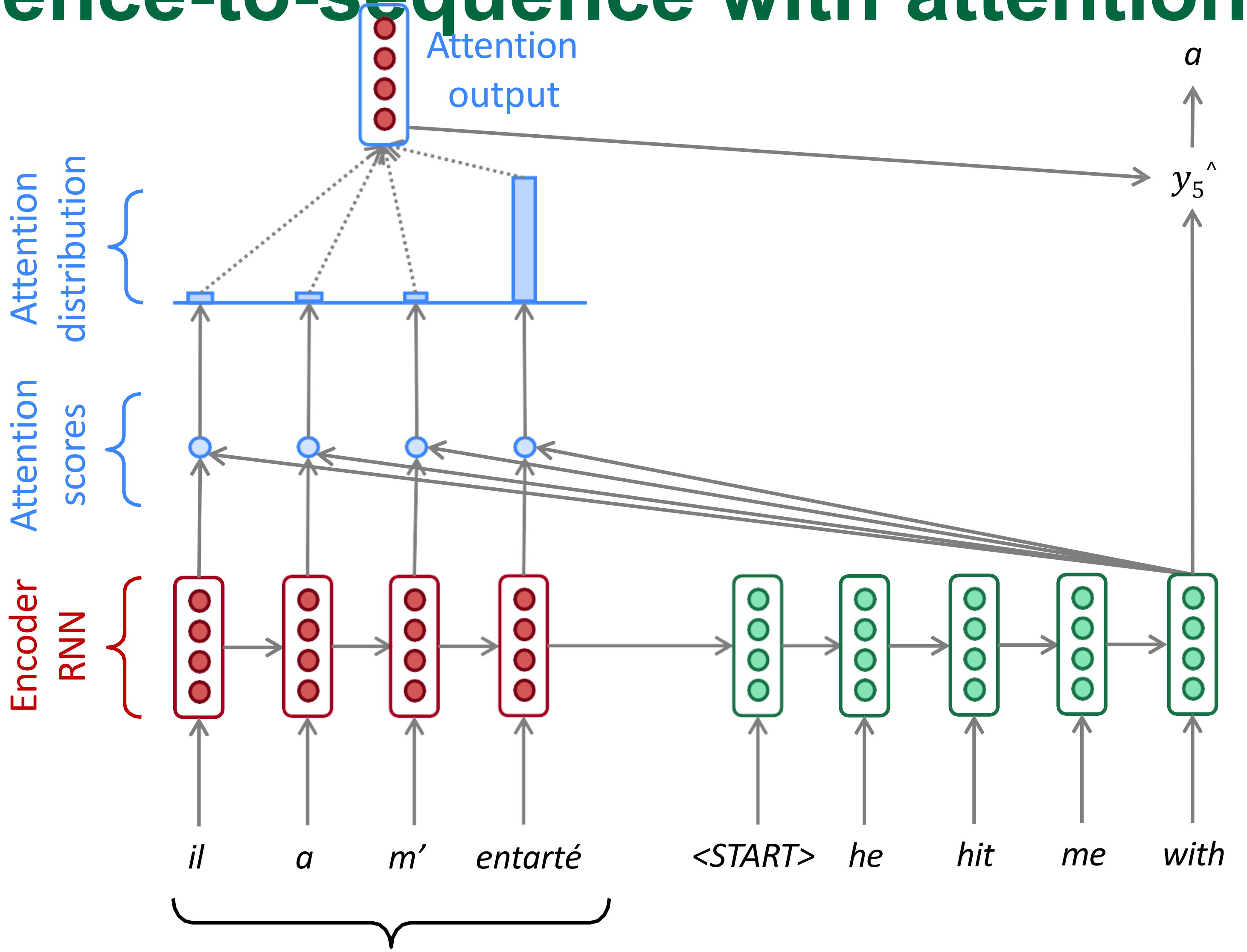
Sequence-to-sequence with attention…



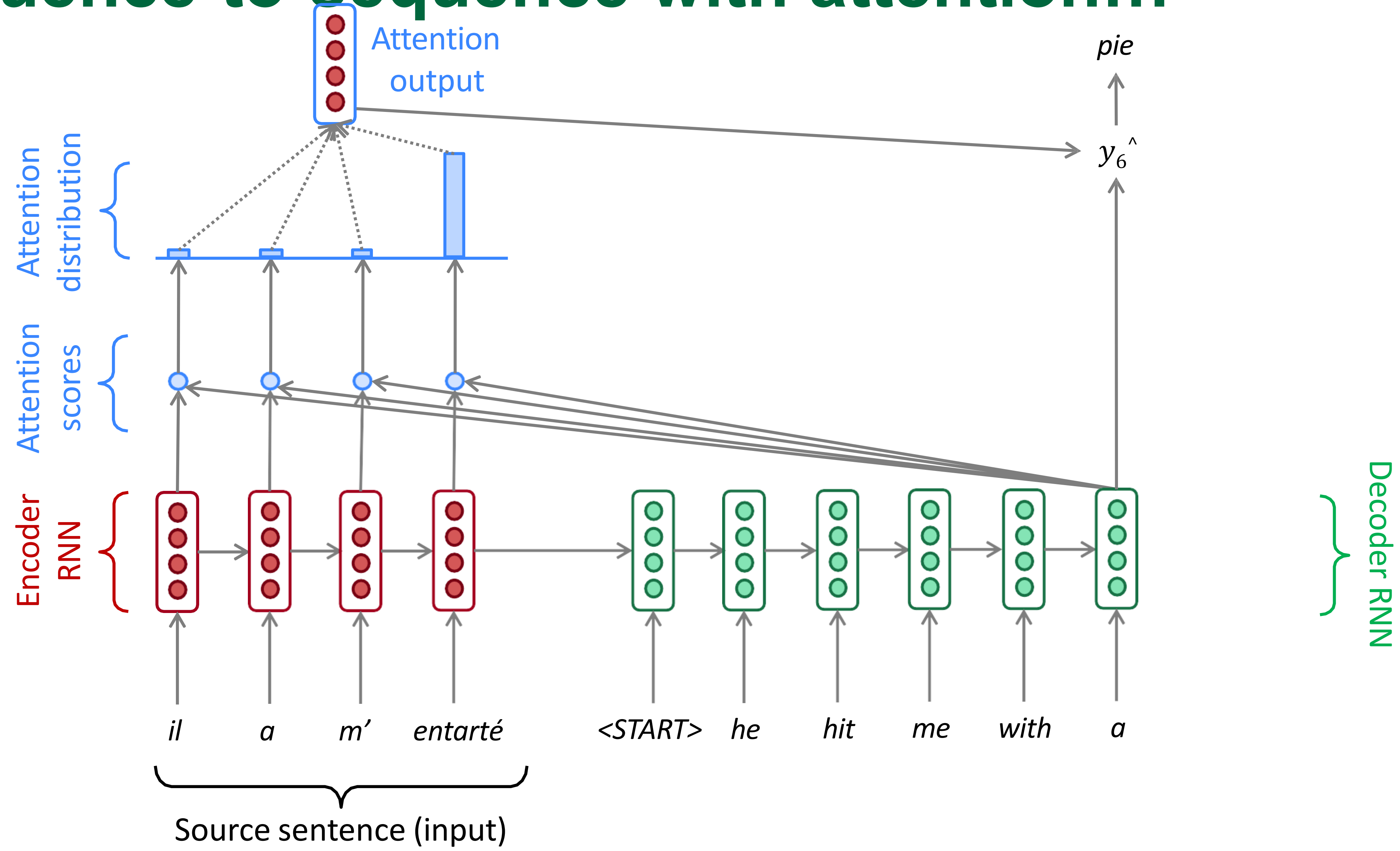
Slide credit: Daniel Jurafsky

ALGONQUIN COLLEGE

Sequence-to-sequence with attention…



Sequence-to-sequence with attention…



Slide credit: Daniel Jurafsky

ALGONQUIN COLLEGE

Attention Mechanism Benefits vs Challenges

How does attention address the temporal bottleneck in sequence-to-sequence models?

Transformers(2017)

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

*https://arxiv.org/abs/1706.03762

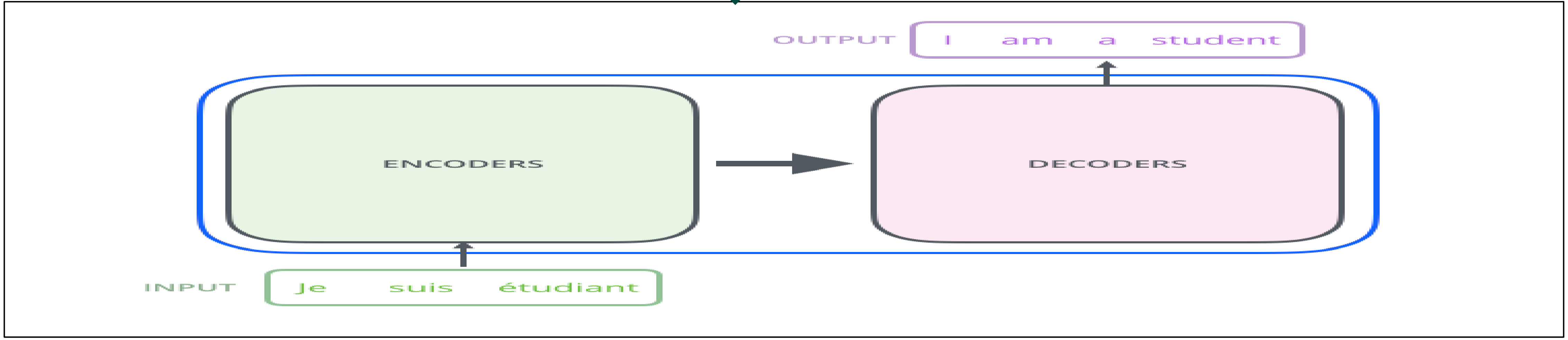
What is Transformer

- The Transformer in NLP is a novel architecture that aims to solve sequence-to-sequence tasks while handling long-range dependencies with ease.
- The Transformer was proposed in the paper ***Attention Is All You Need*** *.

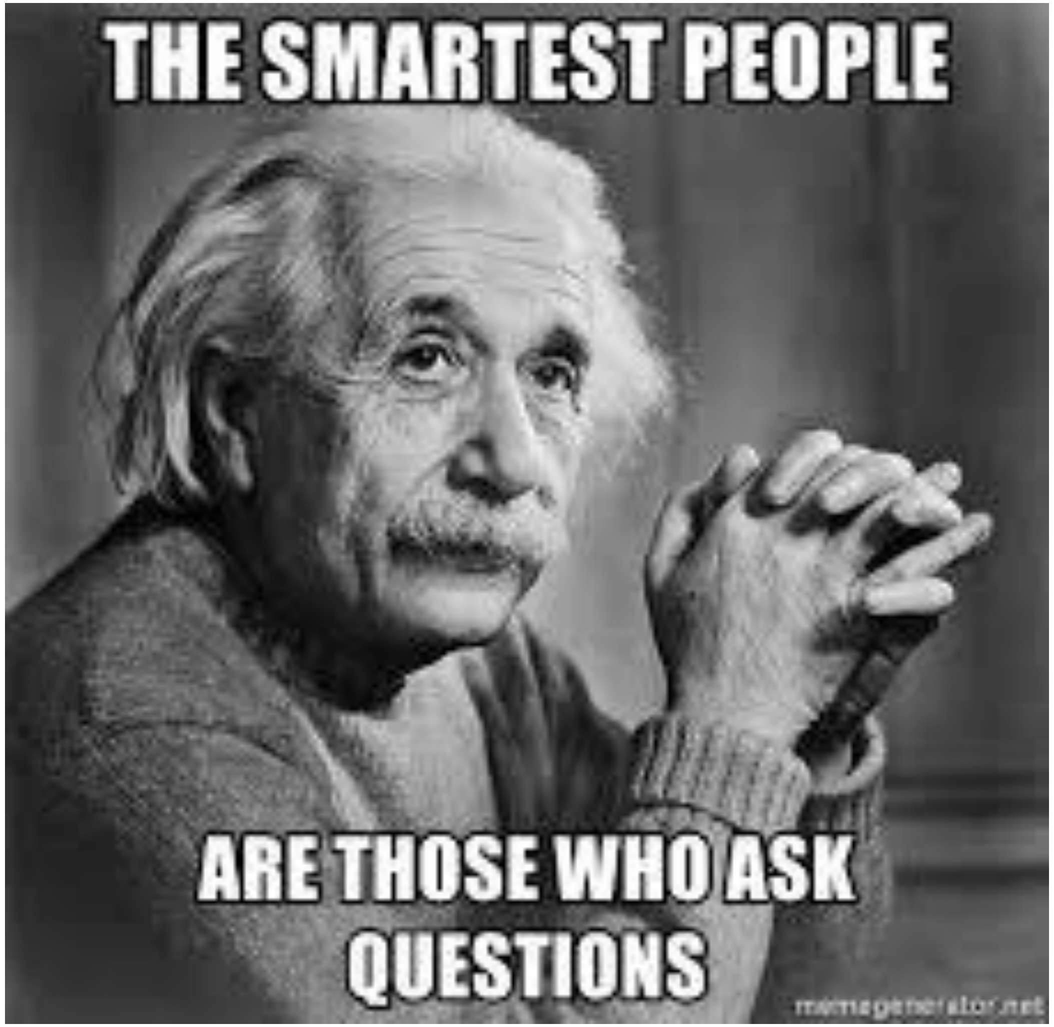
- Relying entirely on self-attention to compute representations of its input and output.

*https://arxiv.org/abs/1706.03762

Transformer Architecture



Q&A


THE SMARTEST PEOPLE
ARE THOSE WHO ASK
QUESTIONS
memegenerator.net