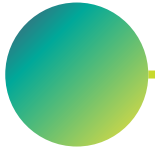




VietAI

# Bài 15: Cơ Chế Attention Trong Mô Hình Seq2Seq

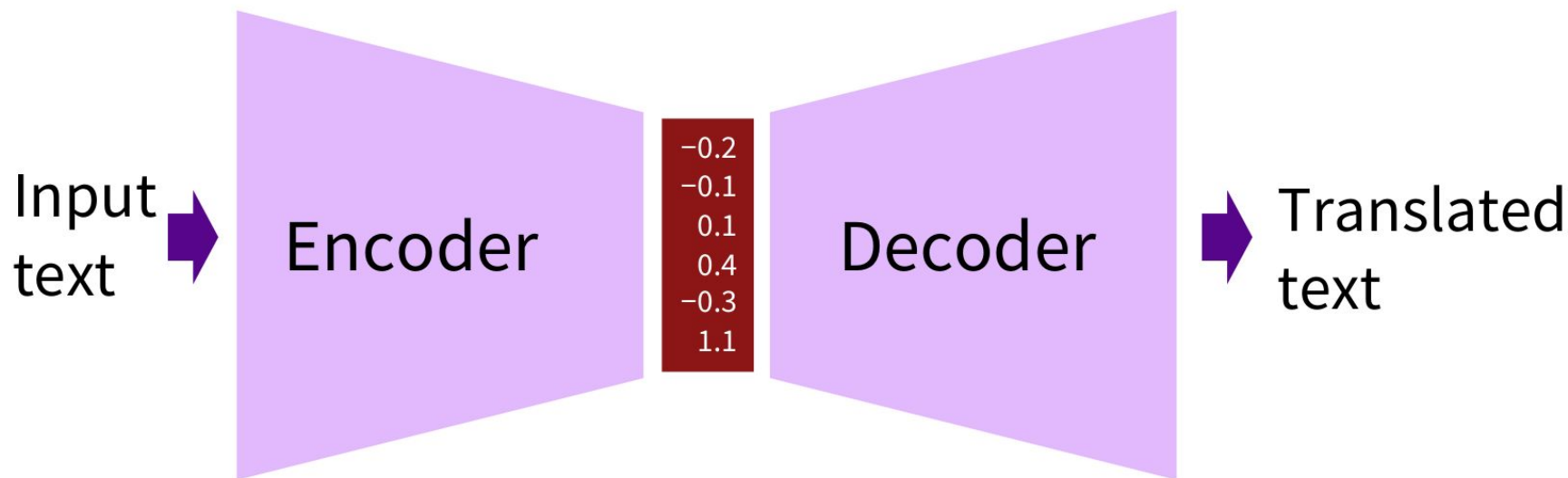


VietAI Teaching Team

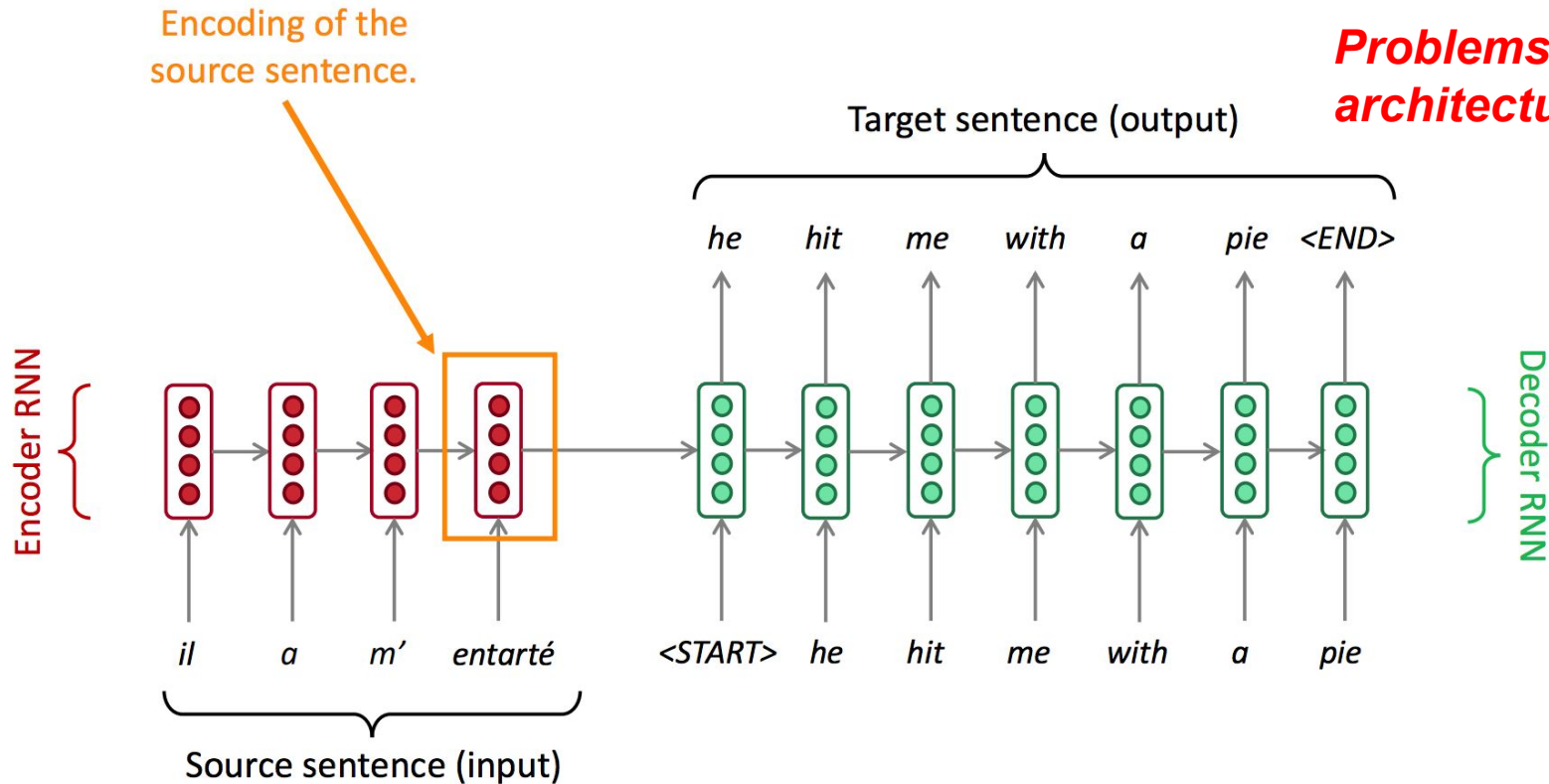
## Nội dung

- Kiến trúc Encoder - Decoder
- Cơ chế Attention
- Beam Search

# 1 Kiến trúc Encoder - Decoder

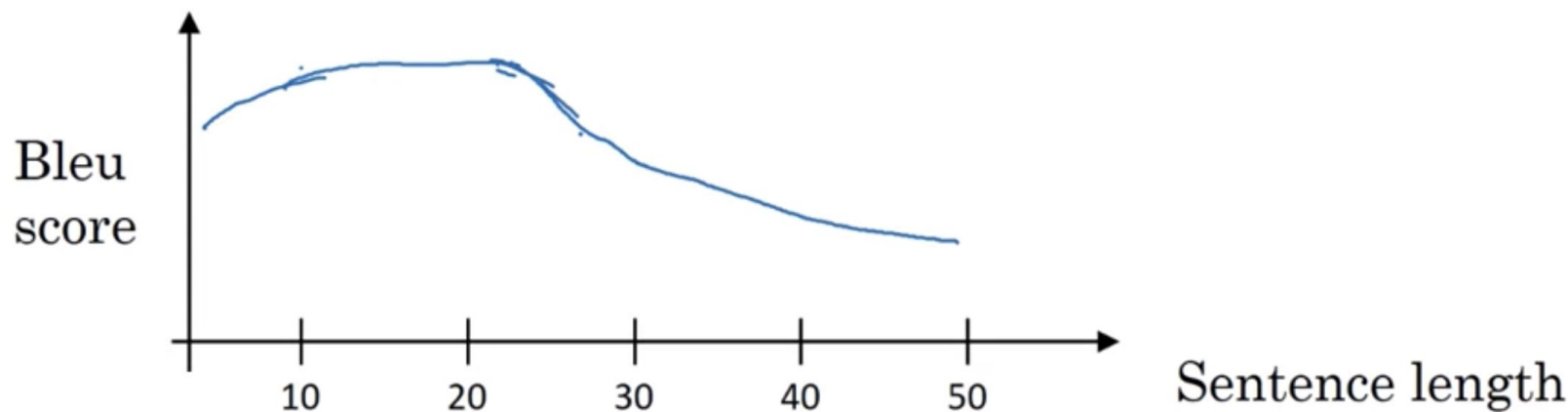


# 1 Kiến trúc Encoder - Decoder



# 1 Kiến trúc Encoder - Decoder (Basic)

Vấn đề: Khi input quá dài, Encoder khó có thể encode tất cả thông tin vào một vector có chiều dài cố định (fixed length vector)



<https://www.coursera.org/lecture/nlp-sequence-models/attention-model-intuition-RDXpX>

## 2 Cơ chế Attention - Ý tưởng

Thay vì để encoder nén toàn bộ thông tin, ta cho decoder được quan sát toàn bộ output của encoder.

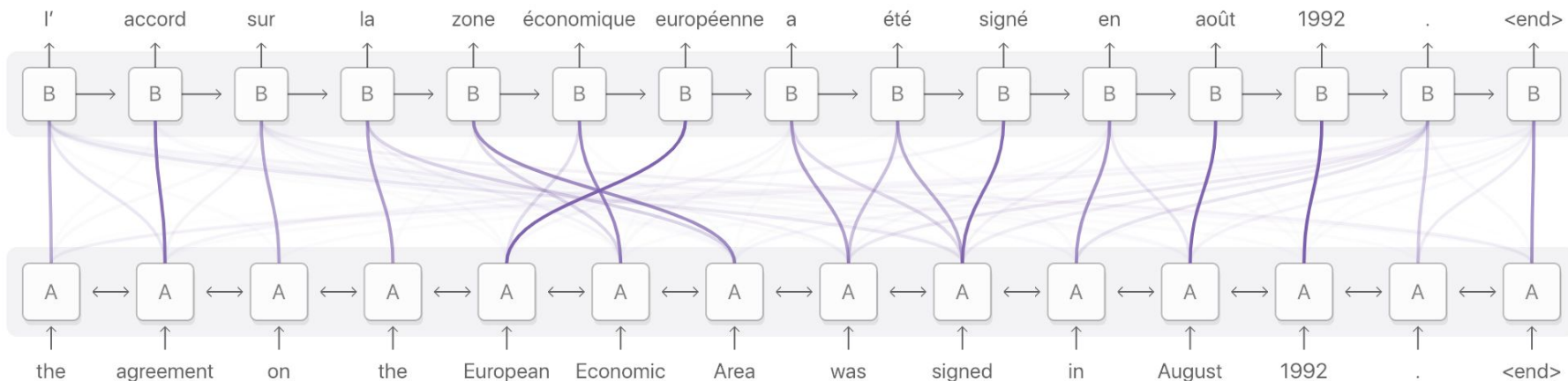
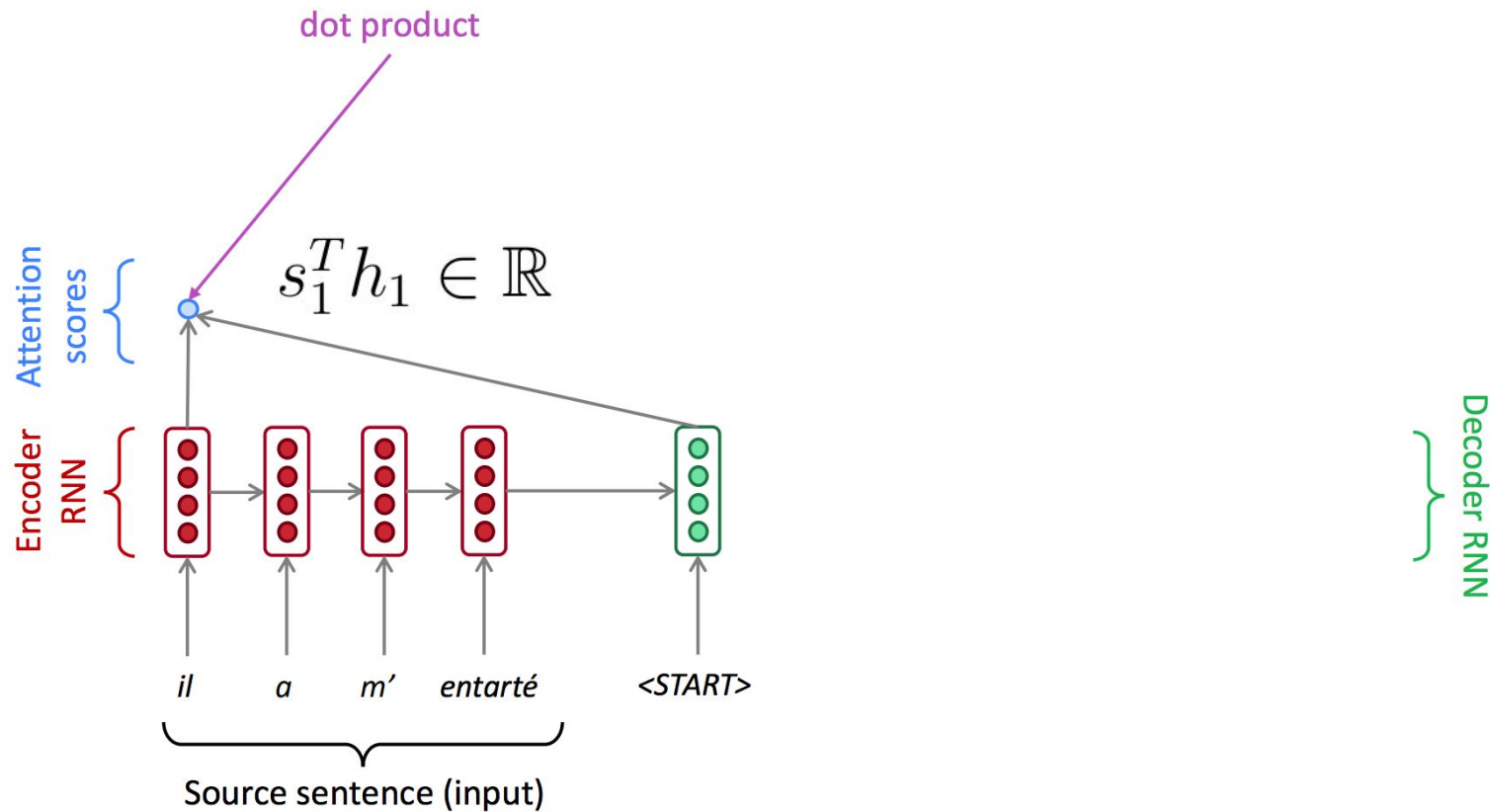


Diagram derived from Fig. 3 of [Bahdanau, et al. 2014](#)

## 2 Cơ chế Attention

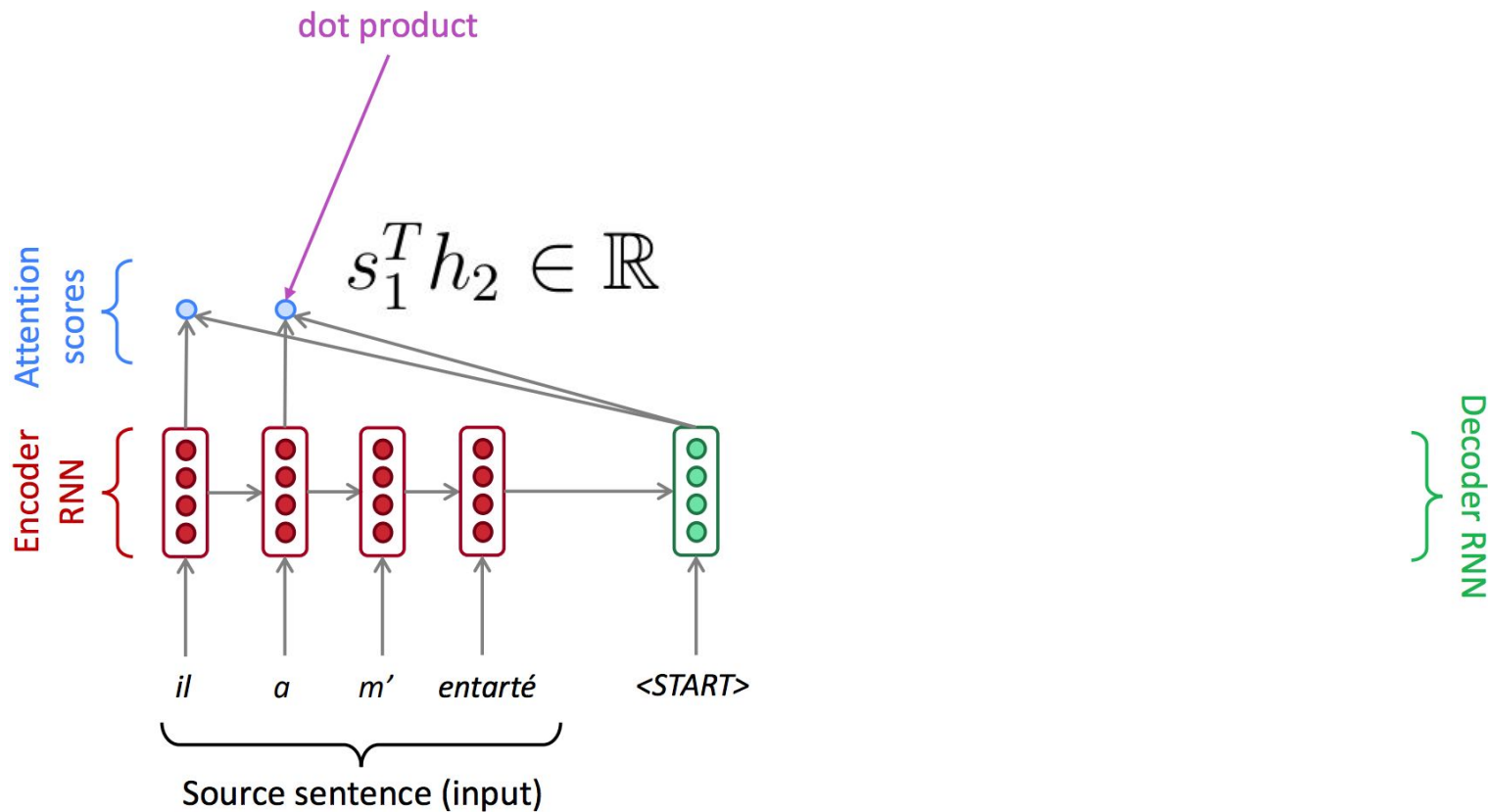
- Encoder hidden states:  $h_1, \dots, h_N \in \mathbb{R}^h$
- Tại thời điểm  $t$ , decoder hidden state:  $s_t \in \mathbb{R}^h$

# Cơ chế Attention

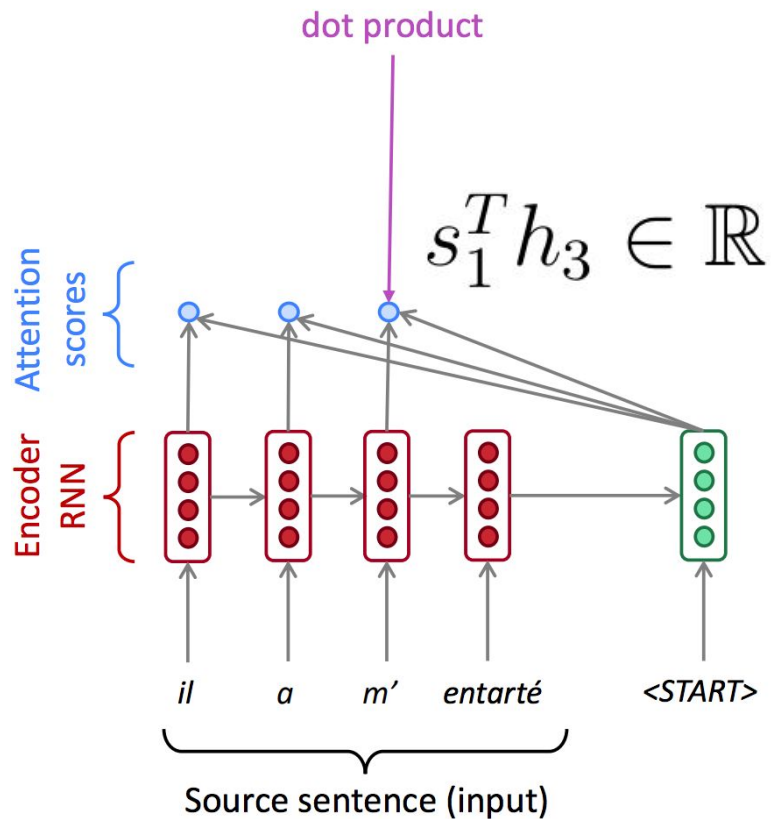




## 2 Cơ chế Attention

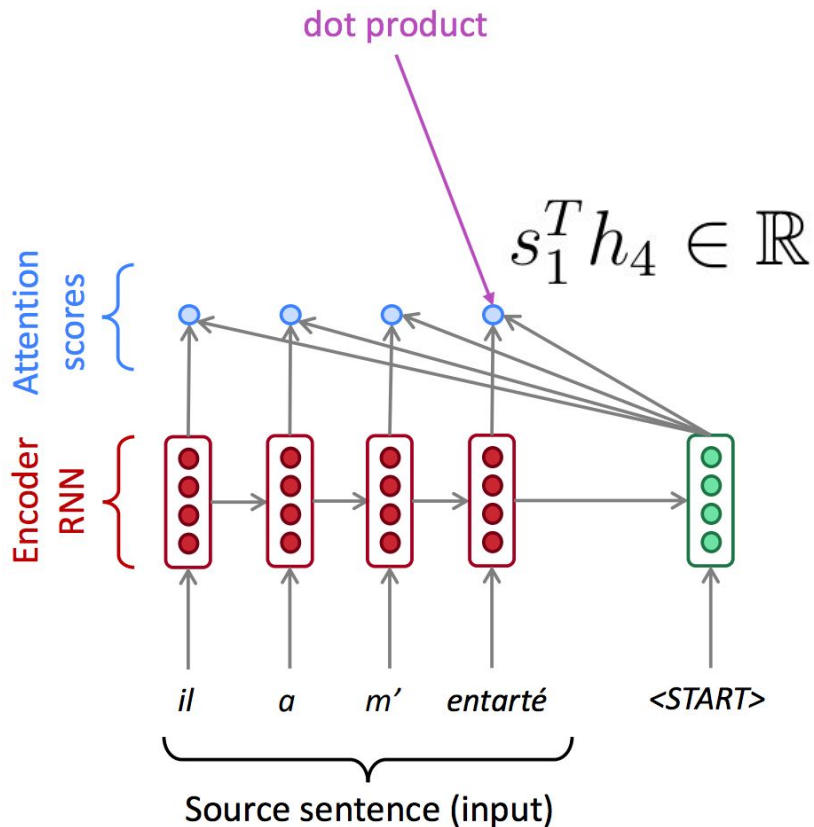


## 2 Cơ chế Attention



## 2

# Cơ chế Attention



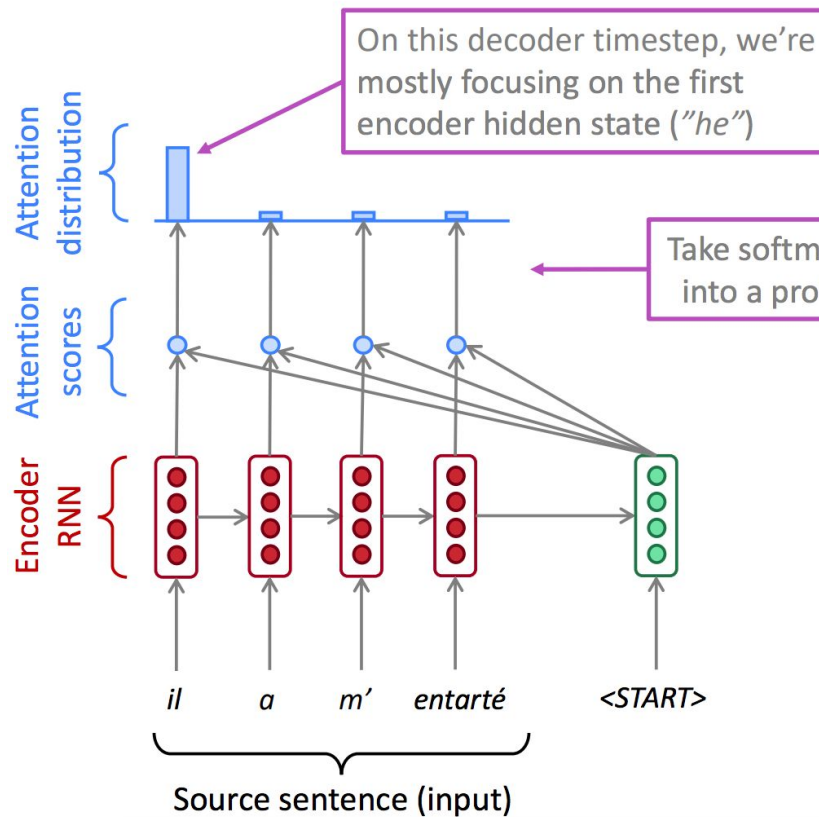
$$e^1 = [s_1^T h_1, \dots, s_1^T h_4] \in \mathbb{R}^4$$

↓

$$e^t = [s_t^T h_1, \dots, s_t^T h_N] \in \mathbb{R}^N$$

Decoder RNN

## 2 Cơ chế Attention



**Softmax dùng để làm gì?**

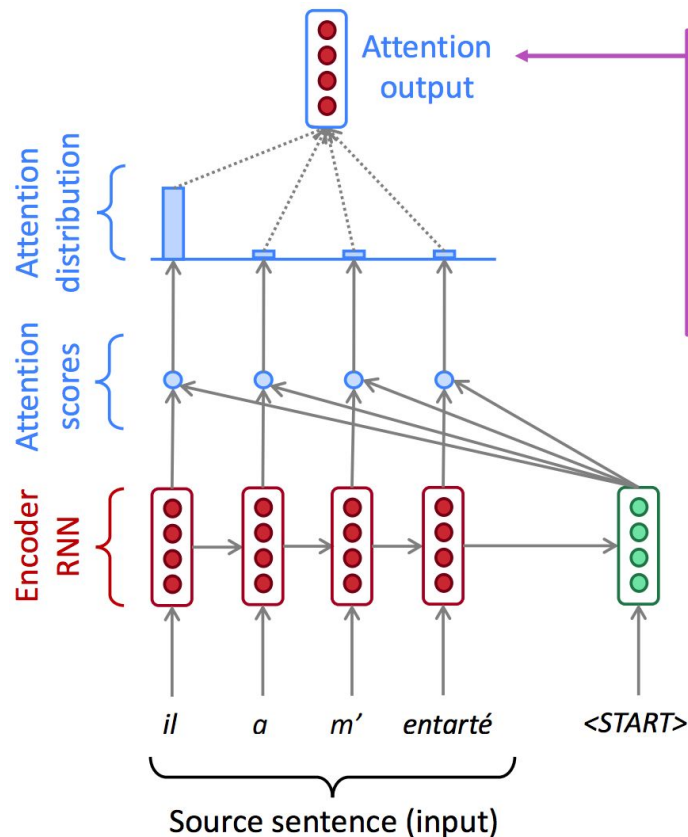
$$\alpha^1 = [\alpha_1^1, \alpha_2^1, \alpha_3^1, \alpha_4^1]$$

$$\alpha^1 = \text{softmax}(e^1) \in \mathbb{R}^N$$

$$\alpha^t = \text{softmax}(e^t) \in \mathbb{R}^N$$

Decoder RNN

## 2 Cơ chế Attention



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

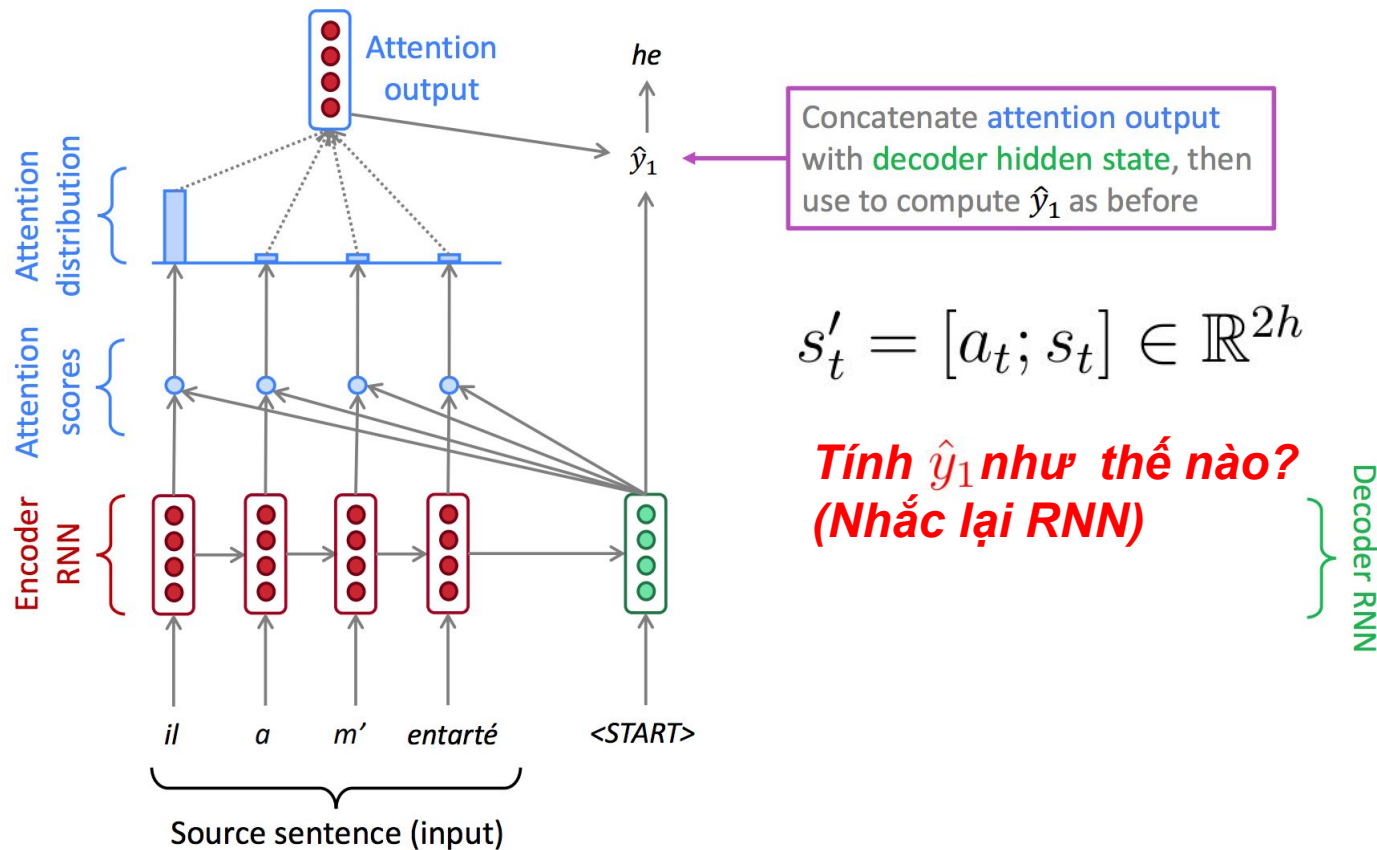
The **attention output** mostly contains information from the **hidden states** that received **high attention**.

$$a_1 = \sum_{i=1}^N \alpha_i^1 h_i \in \mathbb{R}^h$$

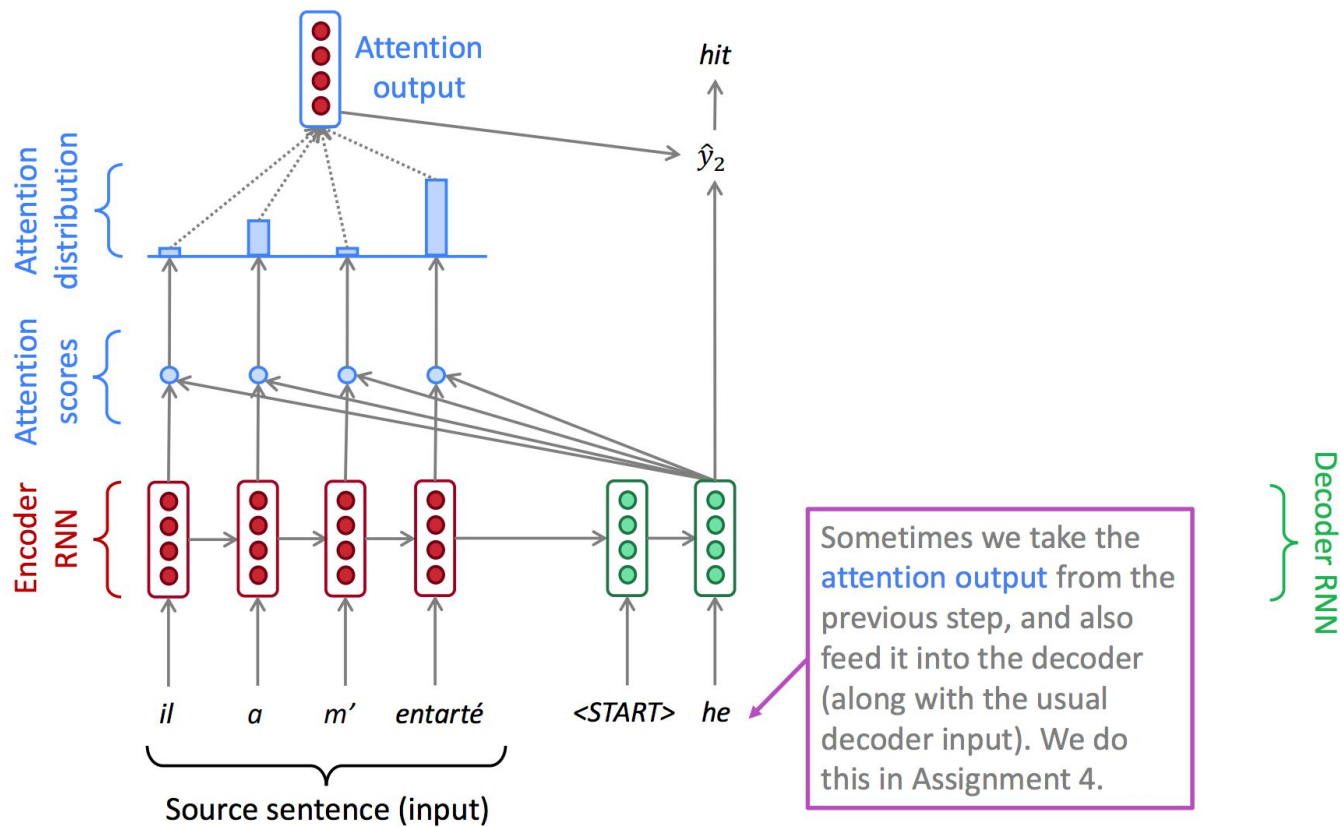
$$a_1 = \alpha_1^1 h_1 + \alpha_2^1 h_2 + \alpha_3^1 h_3 + \alpha_4^1 h_4$$

$$a_t = \sum_{i=1}^N \alpha_i^t h_i \in \mathbb{R}^h$$

## 2 Cơ chế Attention

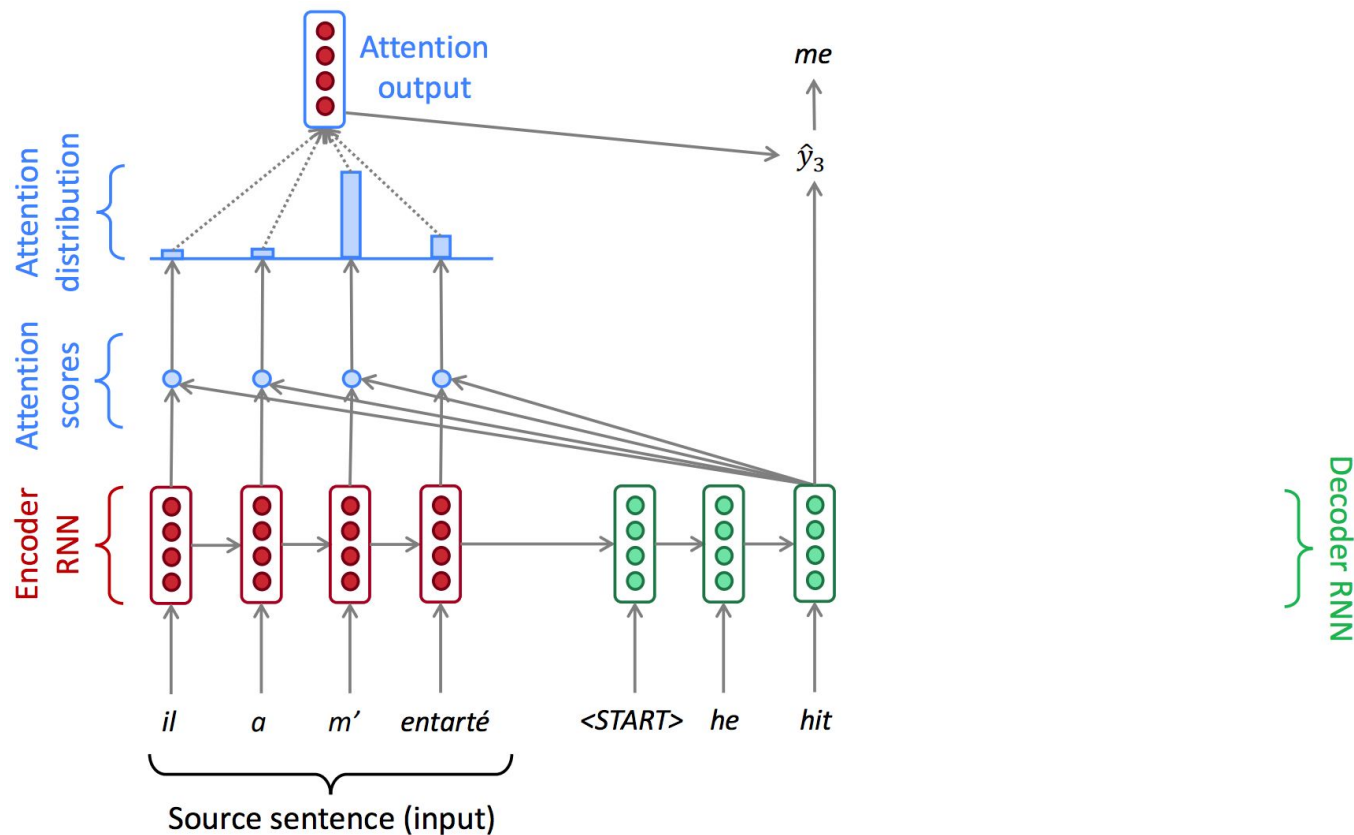


## 2 Cơ chế Attention



## 2

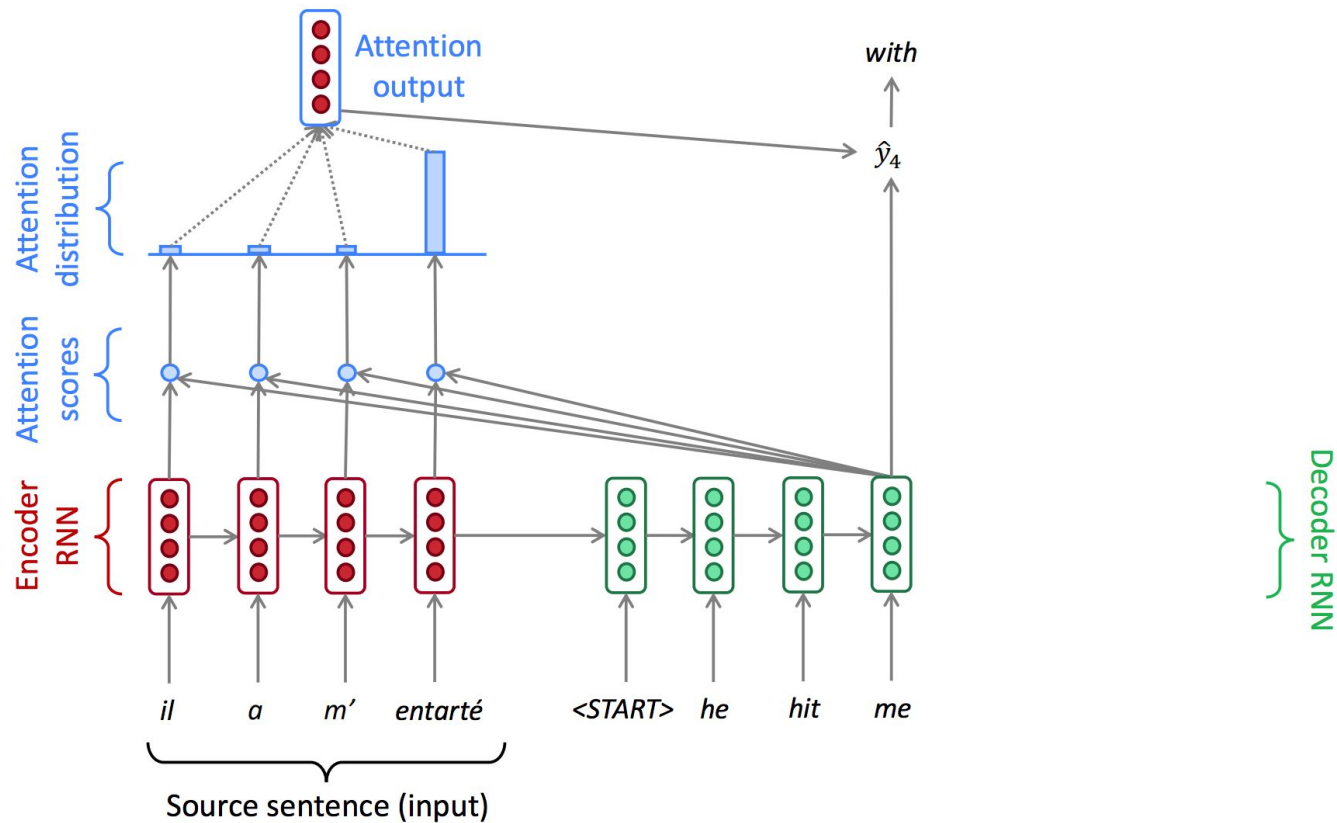
# Cơ chế Attention



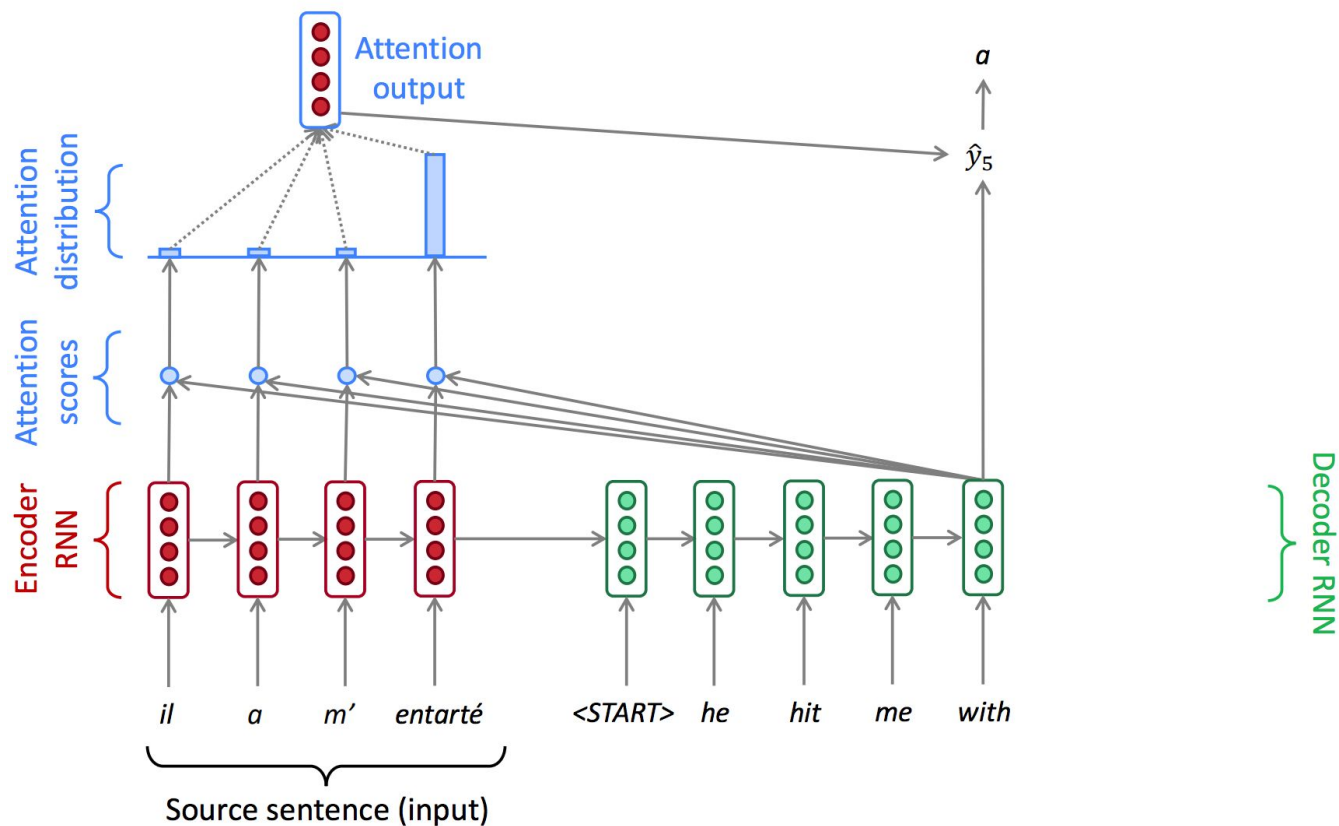


## 2

# Cơ chế Attention

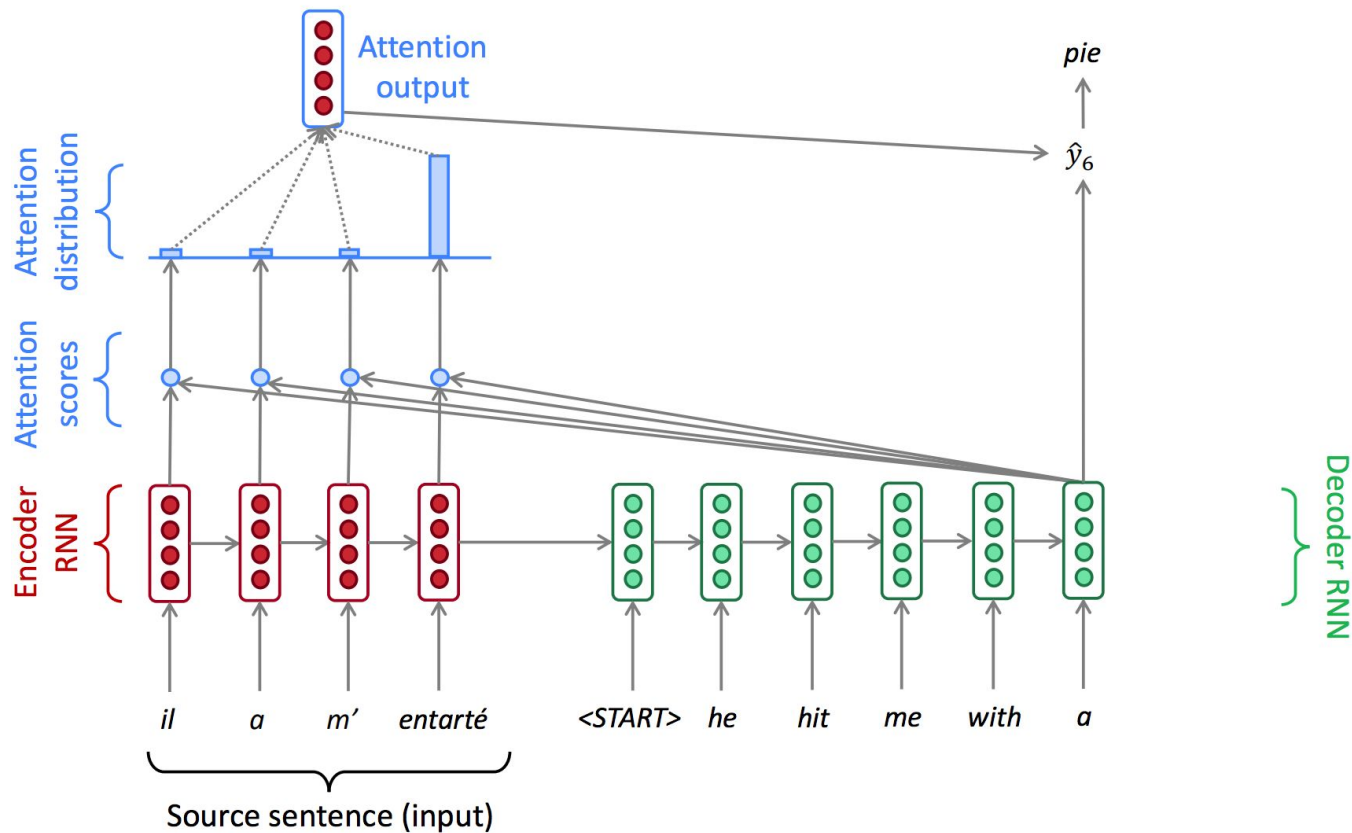


## 2 Cơ chế Attention



## 2

# Cơ chế Attention



## 2 Cơ chế Attention

- Encoder hidden states:  $h_1, \dots, h_N \in \mathbb{R}^h$
- Tại thời điểm  $t$ , decoder hidden state:  $s_t \in \mathbb{R}^h$
- Tại thời điểm  $t$ , attention scores:
$$e^t = [s_t^T h_1, \dots, s_t^T h_N] \in \mathbb{R}^N$$
- Attention distribution:
$$\alpha^t = \text{softmax}(e^t) \in \mathbb{R}^N$$
- Attention output:
$$a_t = \sum_{i=1}^N \alpha_i^t h_i \in \mathbb{R}^h$$
- Concatenate attention output and decoder hidden state:
$$s'_t = [a_t; s_t] \in \mathbb{R}^{2h}$$

## 2 Cơ chế Attention


Chúng ta có:

1. Some values:  $h_1, \dots, h_N \in \mathbb{R}^{d_1}$
2. A query:  $s \in \mathbb{R}^{d_2}$

- More general definition of attention:
  - Given a set of vector *values*, and a vector *query*, attention is a technique to compute a weighted sum of the values, dependent on the query.

## 2 Cơ chế Attention

Các bước khi tính attention như sau:

1. Tính **attention scores**:  $e \in \mathbb{R}^N$   *There are many ways to do this*
2. Tính **attention distribution**:

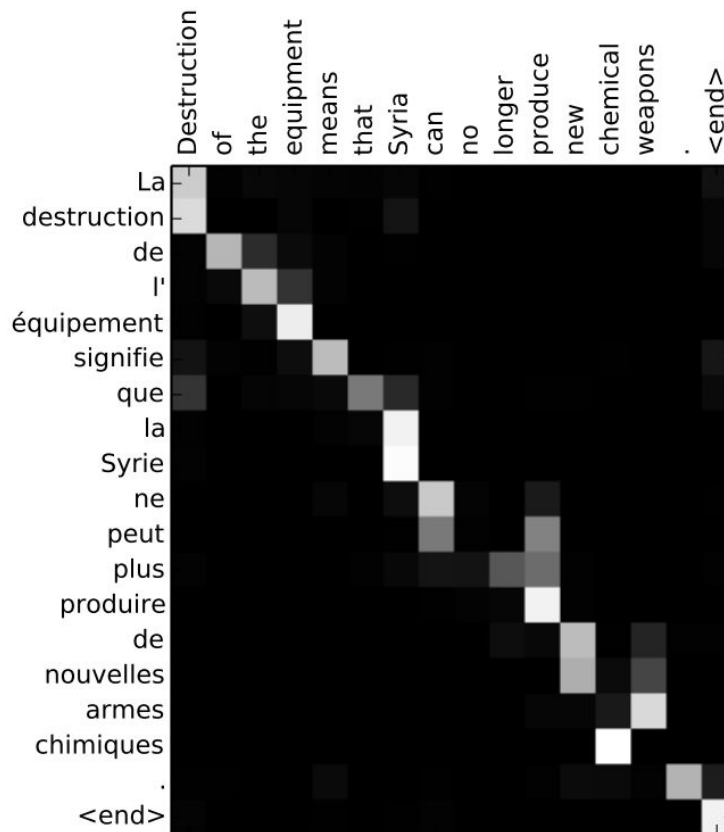
$$\alpha = \text{softmax}(e) \in \mathbb{R}^N$$

3. Tính **attention output** (hay còn gọi là context vector):

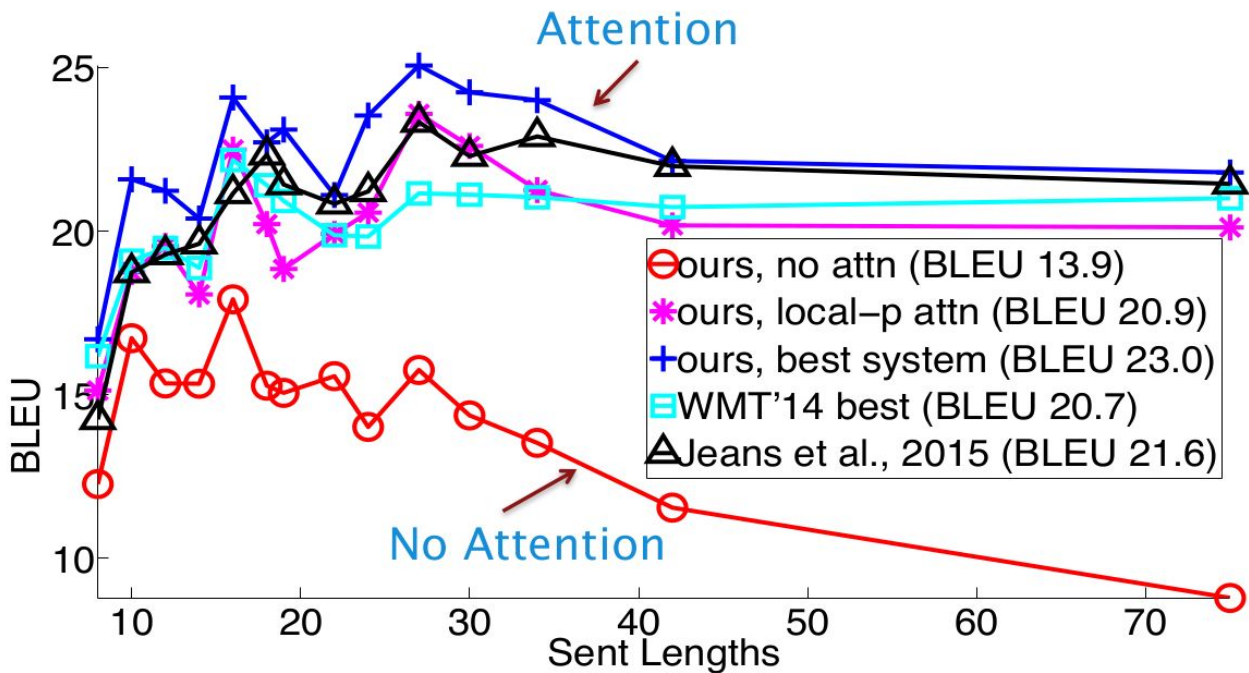
$$a = \sum_{i=1}^N \alpha_i h_i \in \mathbb{R}^{d_1}$$

## 2 Cơ chế Attention

Cơ chế Attention tự động xây dựng sự tương ứng (alignment) giữa các từ tiếp theo phải sinh ra trong câu đích và các từ trong câu nguồn từ đó giúp việc dịch chính xác hơn, đặc biệt với các câu dài.



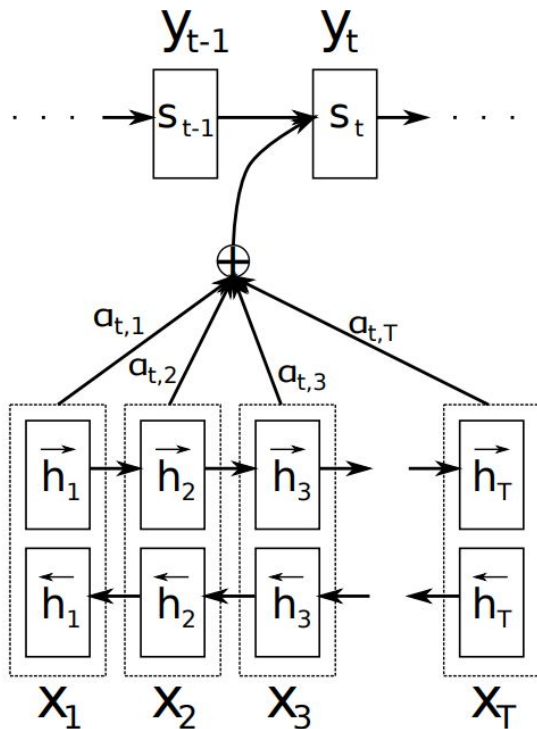
## 2 Cơ chế Attention



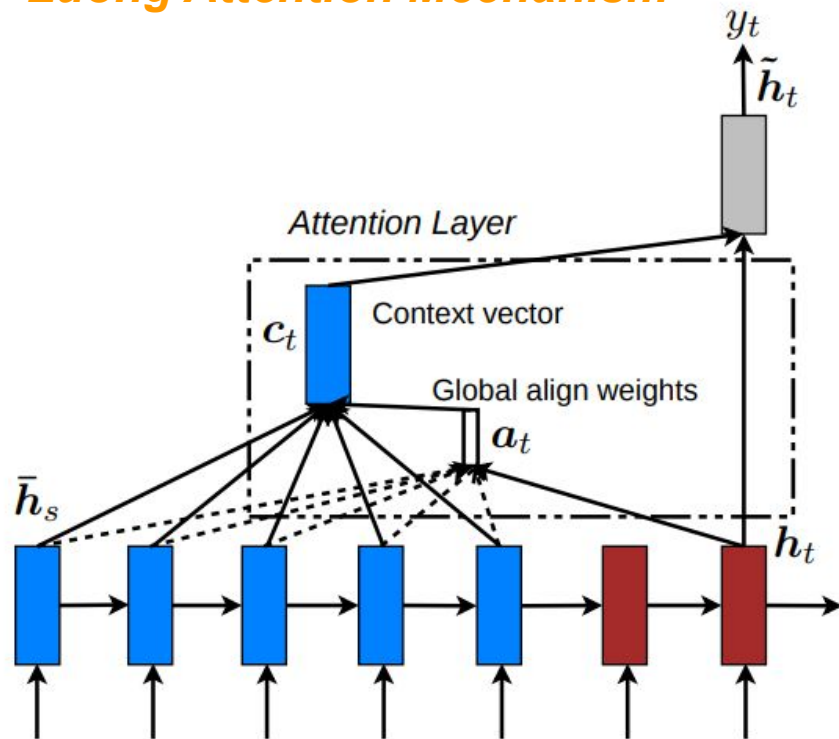


## 2 Attention Models

### Bahdanau Attention Mechanism



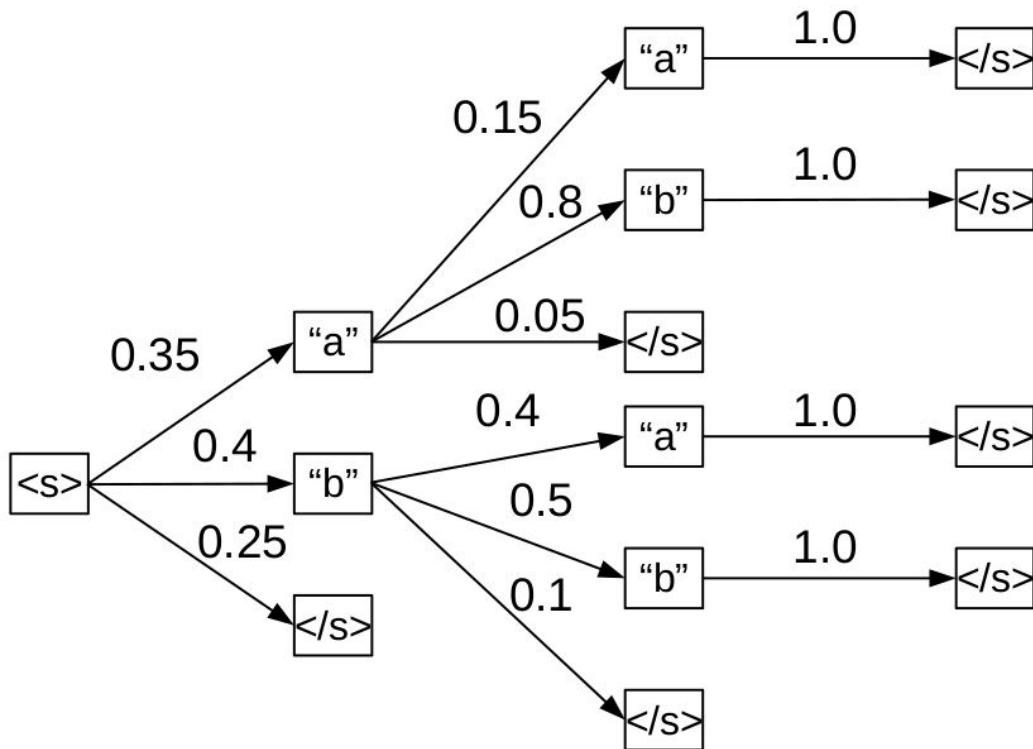
### Luong Attention Mechanism



### 3 Beam Search

- Để tìm được chuỗi  $y$  cho xác suất  $P(y_1 y_2 \dots y_N | x)$  cao nhất, ta phải tìm thông qua các xác suất  $P(y_i | y_1 \dots y_{i-1}, x)$
- Có nhiều cách để tìm  $y$ :
  - **Random Sampling (Acestral Sampling)**: tại bước thứ  $i$ , chọn ngẫu nhiên  $y_i$  theo phân phối xác suất được cho từ mạng decoder.
  - **Greedy Search**: tại bước thứ  $i$ , chọn  $y_i$  là từ cho giá trị  $P(y_i | y_1 \dots y_{i-1}, x)$  cao nhất.
  - **Beam Search**: tại bước thứ  $i$ , chỉ giữ lại  $k$  chuỗi con độ dài  $i$  cho xác suất  $P(y_1 y_2 \dots y_i)$  ( $k$  được gọi là *beam size* hay *beam width*) cao nhất để thực hiện việc tìm kiếm tại bước  $i + 1$  tiếp theo.

### 3 Beam Search



- Theo hình bên, chuỗi "a b </s>" cho ta xác suất cao nhất.
- Nếu áp dụng Greedy Search sẽ cho ta chuỗi "b b </s>".
- Nếu áp dụng Beam Search với  $k = 2$  sẽ cho ta chuỗi: "a b </s>"

# Tài liệu tham khảo

1. CS224N, Stanford University
2. Bahdanau et al., [Neural machine translation by jointly learning to align and translate](#) - ICLR15.
3. Sutskever et al., [Sequence to sequence learning with neural networks](#) - NIPS14.
4. Luong et al., [Effective approaches to attention-based neural machine translation](#) - EMNLP15.
5. [Neural Machine Translation \(seq2seq\) Tutorial with TensorFlow](#)