

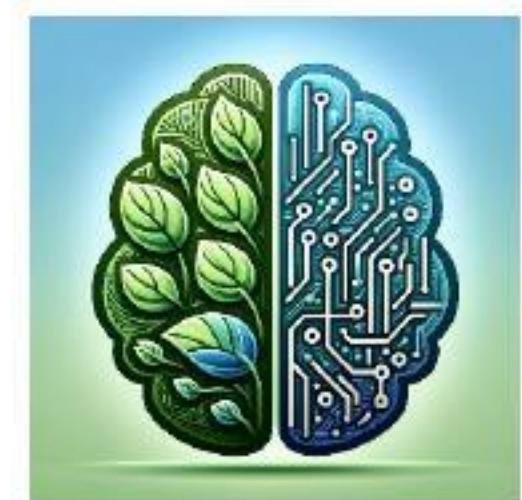


Natural Language Processing with Deep Learning

Bài 9: TRANSFORMER ARCHITECTURE



https://csc.edu.vn/data-science-machine-learning/natural-language-processing-with-deep-learning_293



ATTENTION BASED NETWORK – LSTM



I. LSTM Encoder – Decoder

II. Attention Mechanism

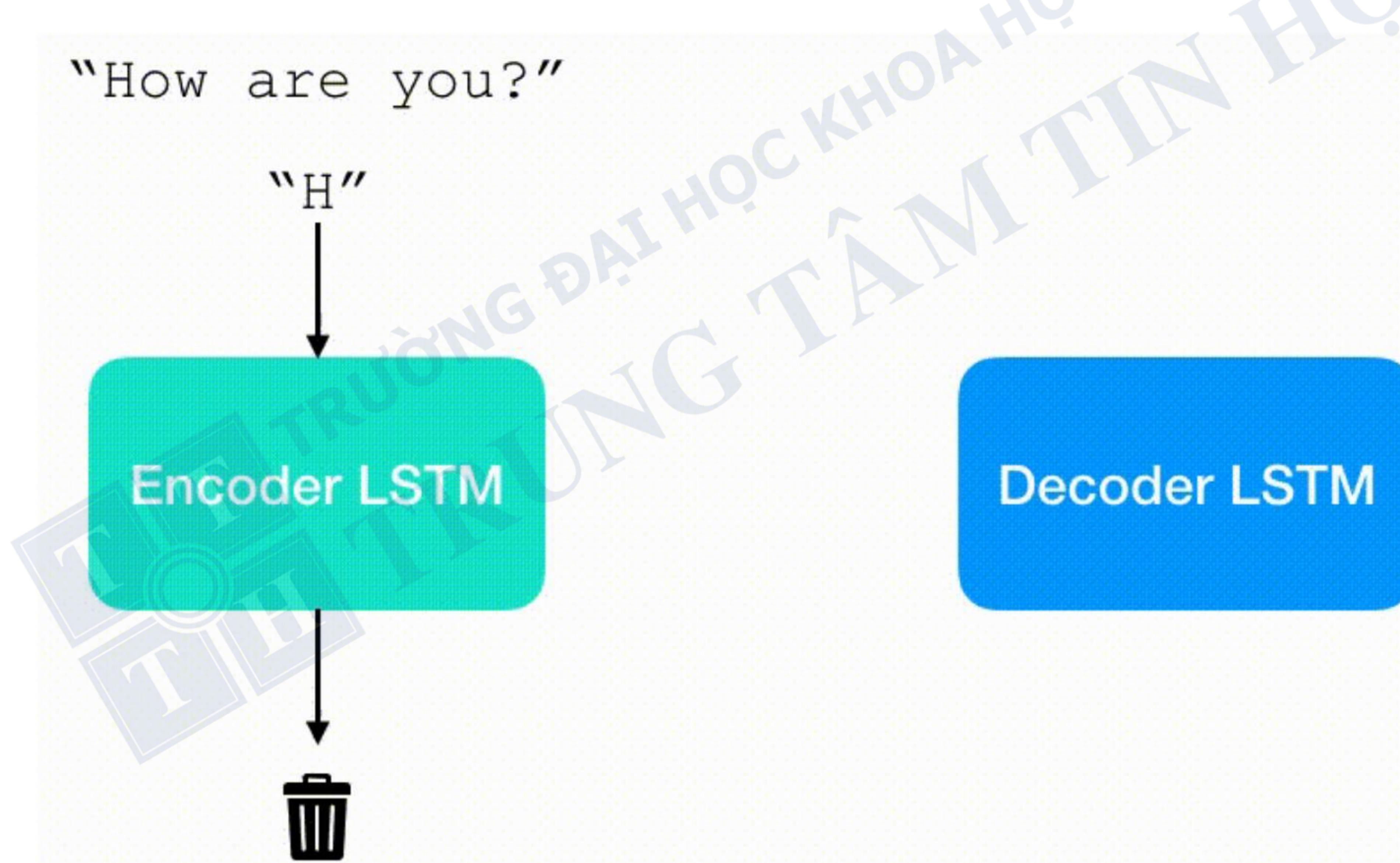
III. Transformer Model





Tổng quan về LSTM Encoder - Decoder

- **LSTM Encoder-Decoder** là mạng neural network về **xử lý ngôn ngữ tự nhiên**, dùng để chuyển đổi input sang output với một độ dài khác.



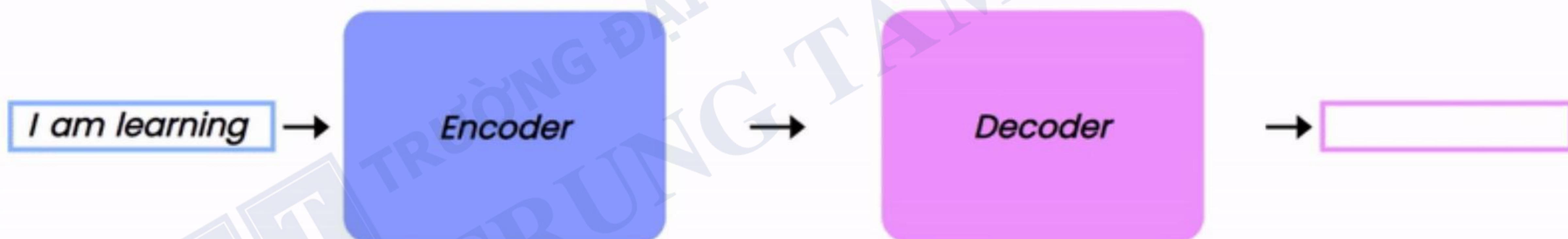


Tổng quan về LSTM Encoder - Decoder

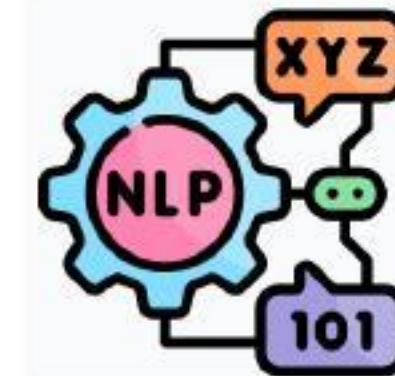
- LSTM Encoder-Decoder bao gồm hai phần chính:

LSTM Encoder nhận dữ liệu input và tạo ra một vector trạng thái ẩn.

LSTM Decoder sử dụng vector trạng thái ẩn này để tạo ra dữ liệu output.

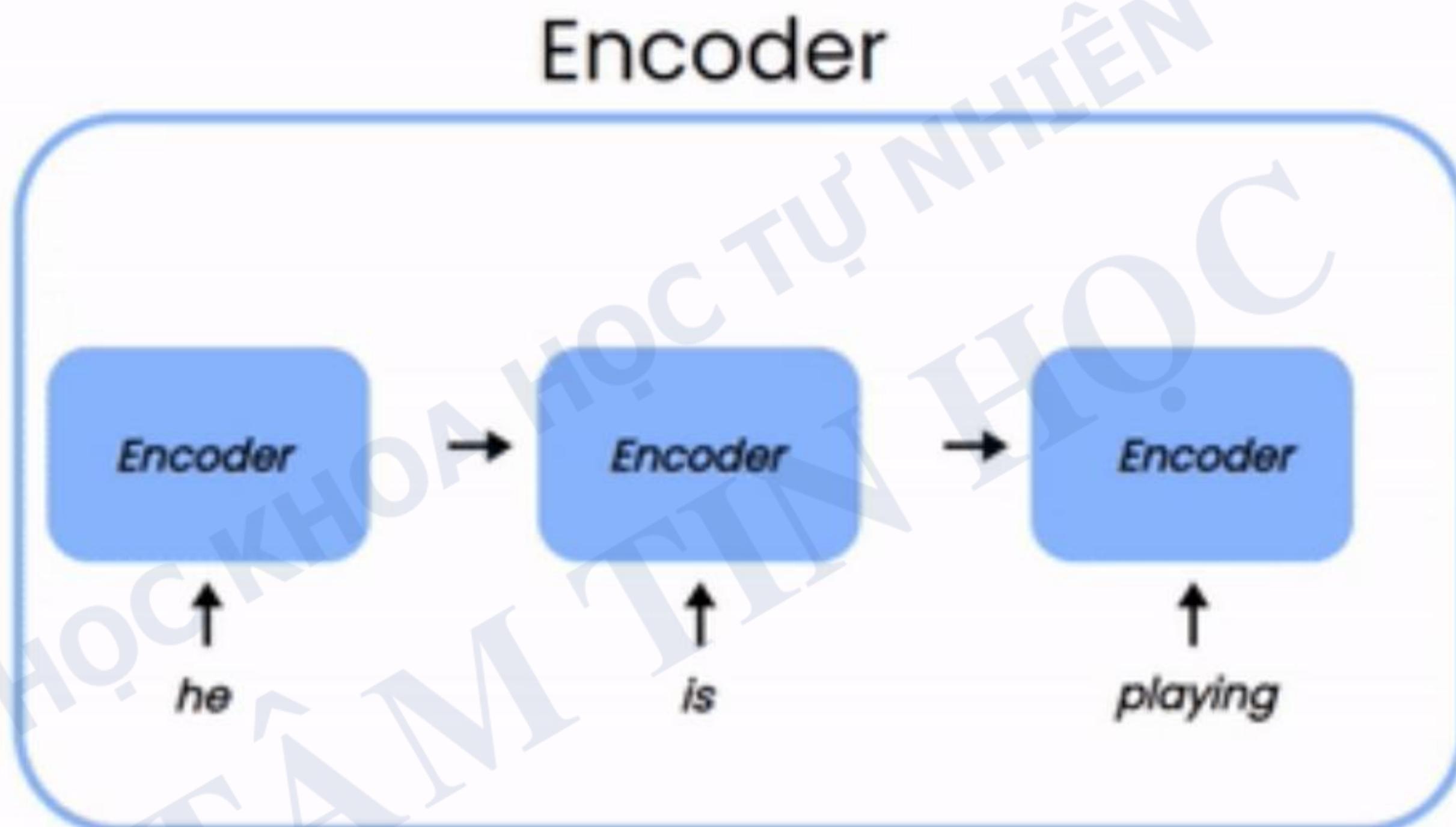


Ví dụ: LSTM Encoder nhận đầu vào là 1 câu tiếng Anh và tạo ra một vector trạng thái ẩn. LSTM Decoder sử dụng vector trạng thái ẩn này để tạo ra câu tiếng Việt tương ứng.

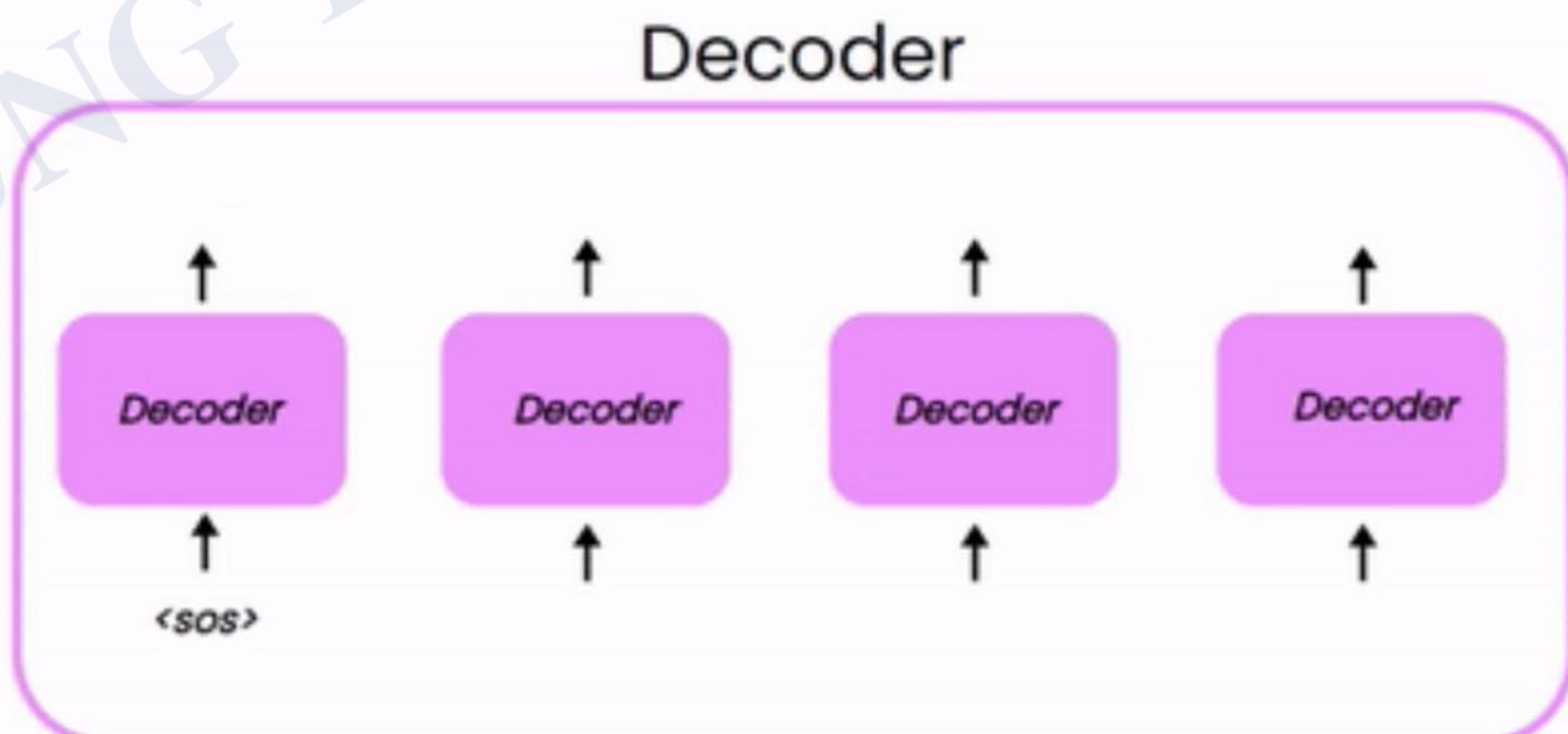


LSTM Encoder - Decoder

Encoder mã hóa ý nghĩa tổng quát của câu input.



Decoder lấy state của encoder để câu output.

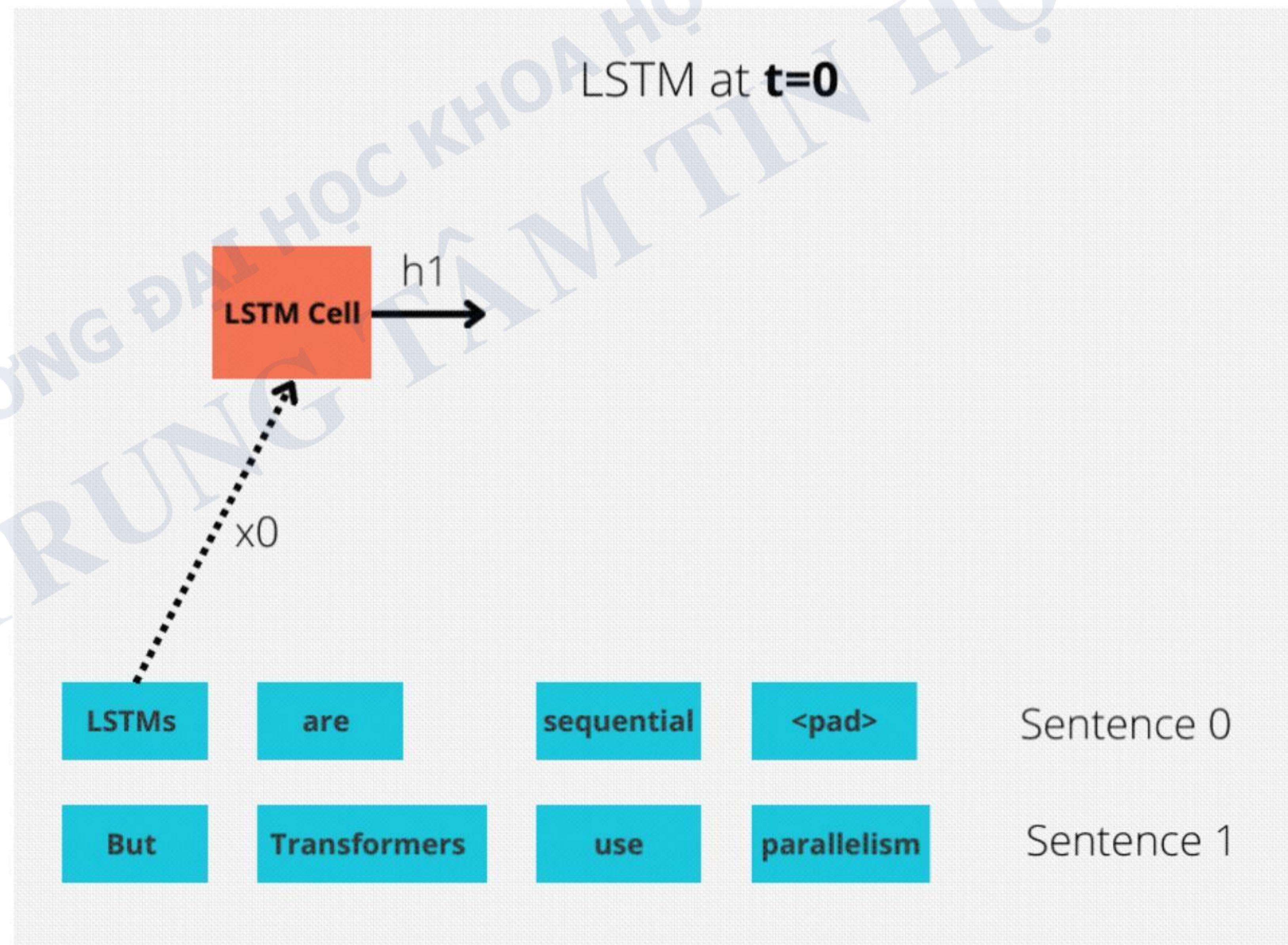




Hạn chế của LSTM Encoder - Decoder

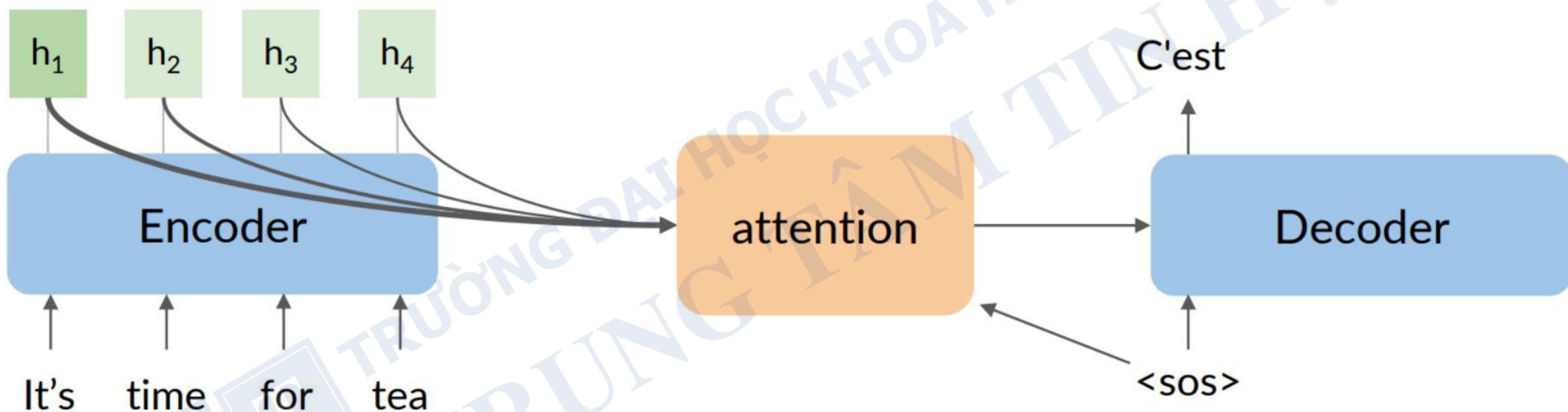
- Câu có độ dài thay đổi + bộ nhớ có độ dài cố định
- LSTM đi qua tất cả các state ẩn của encoder

→ Giảm hiệu
xuất mô hình,
chậm hơn.



Hạn chế của LSTM Encoder - Decoder

- **Giải pháp:** Chỉ tập trung vào những state ẩn quan trọng trong mỗi bước.



→ **Attention-based Network**

Attention cho phép truy xuất bộ nhớ (state ẩn) linh hoạt hơn.



Self-Attention Mechanism

Còn gọi là **intra Attention**, là cơ chế tập trung vào mối quan hệ giữa các giá trị khác nhau trong chuỗi.

Attention : What part of the input should we focus?

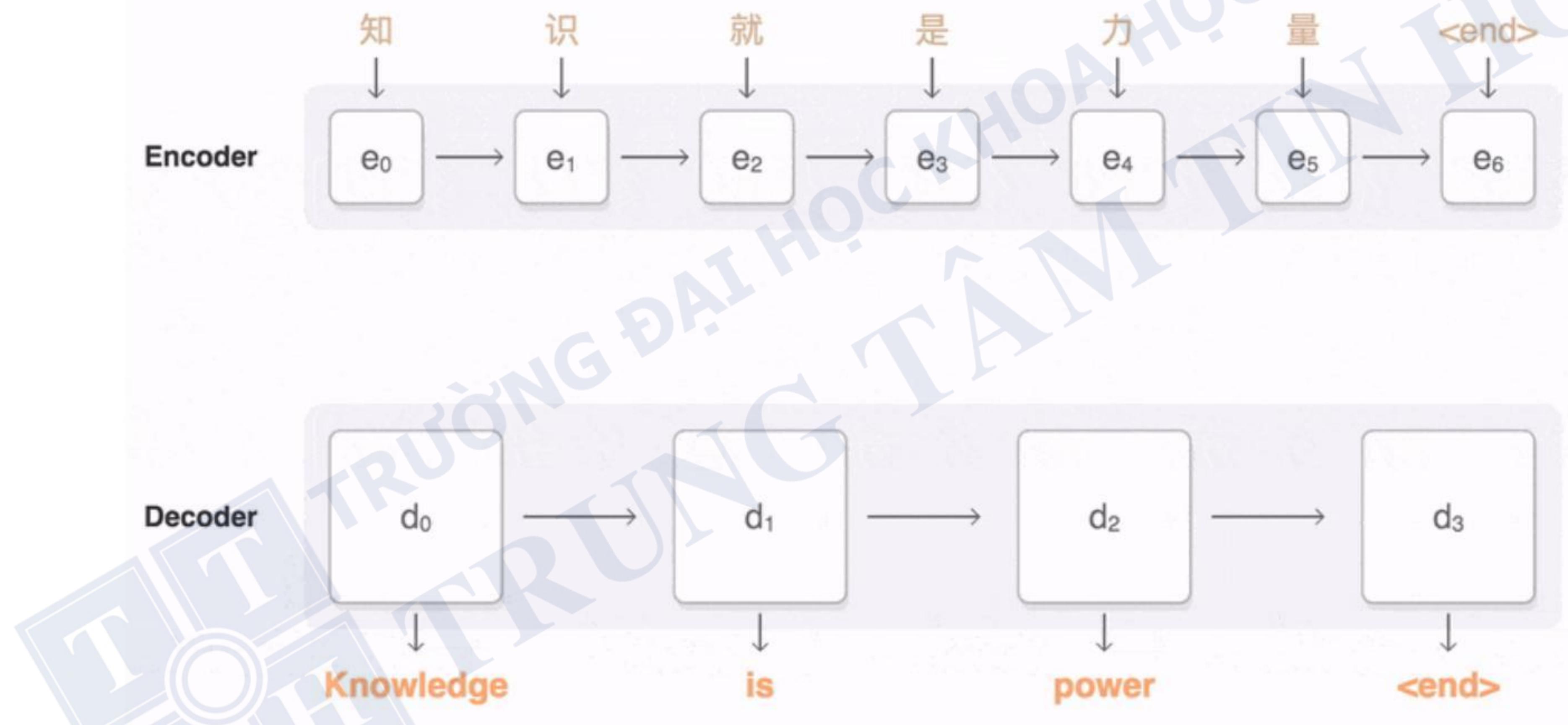
Focus	Attention Vectors
The	[0.71 0.04 0.07 0.18] ^T
big	[0.01 0.84 0.02 0.13] ^T
red	[0.09 0.05 0.62 0.24] ^T
dog	[0.03 0.03 0.03 0.91] ^T

→ Hữu ích trong **machine reading, abstractive summarization, và image description**.



Tổng quan Self-Attention Mechanism

□ **Attention Mechanism** xem mỗi từ là một thành phần trong một chuỗi giá trị.



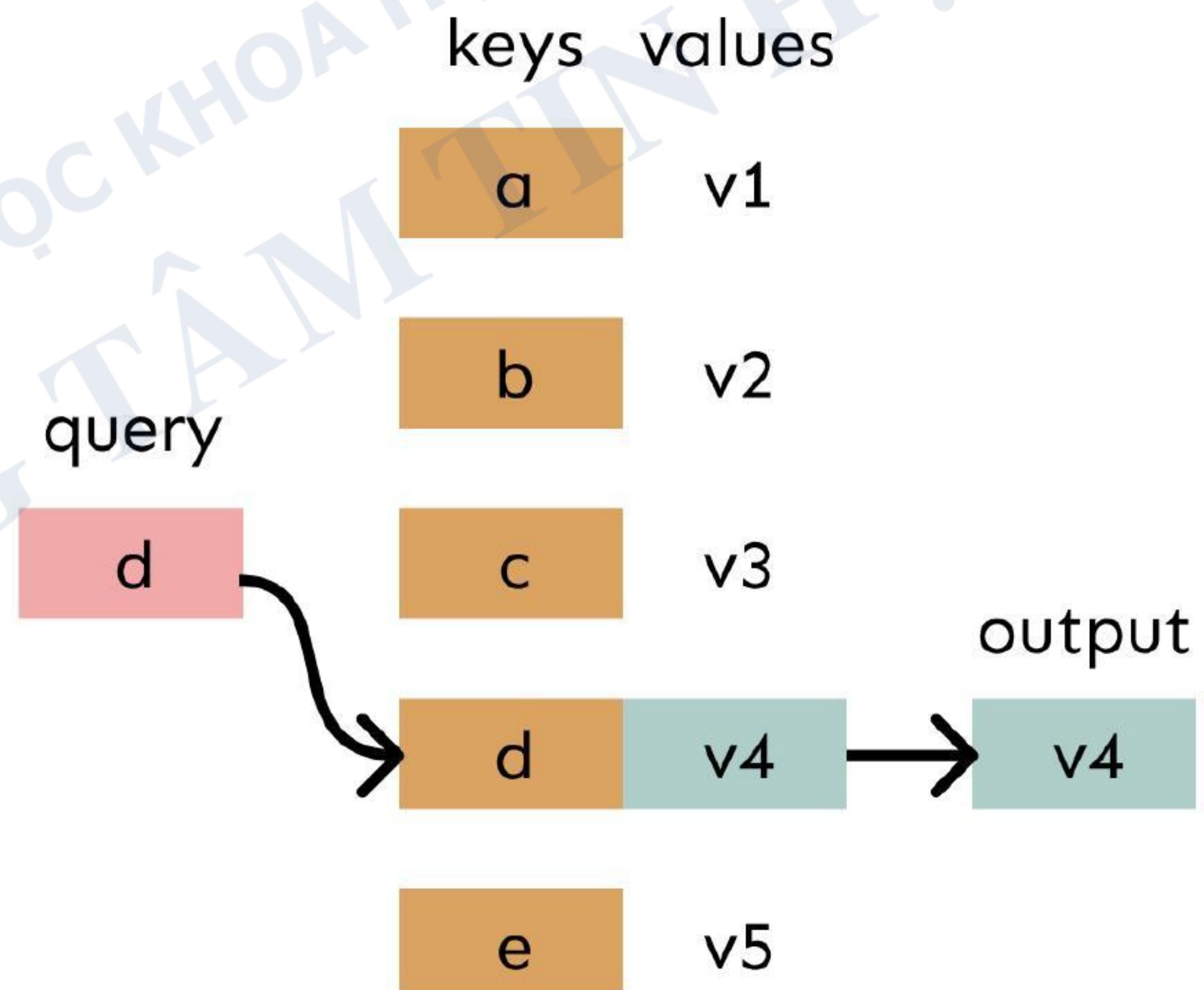
- Attention đi từ bộ **decoder sang encoder**.
- Tất cả các từ của layer sau được áp dụng cho tất cả các từ ở layer trước → **Giảm số lần lặp**.

Tổng quan Self-Attention Mechanism

- Có thể xem attention đang thực hiện **hàm tra cứu lookup** cho một kho lưu trữ key-value.

Trong 1 bảng tra cứu, chúng ta có các **key** tương ứng với các **value**.

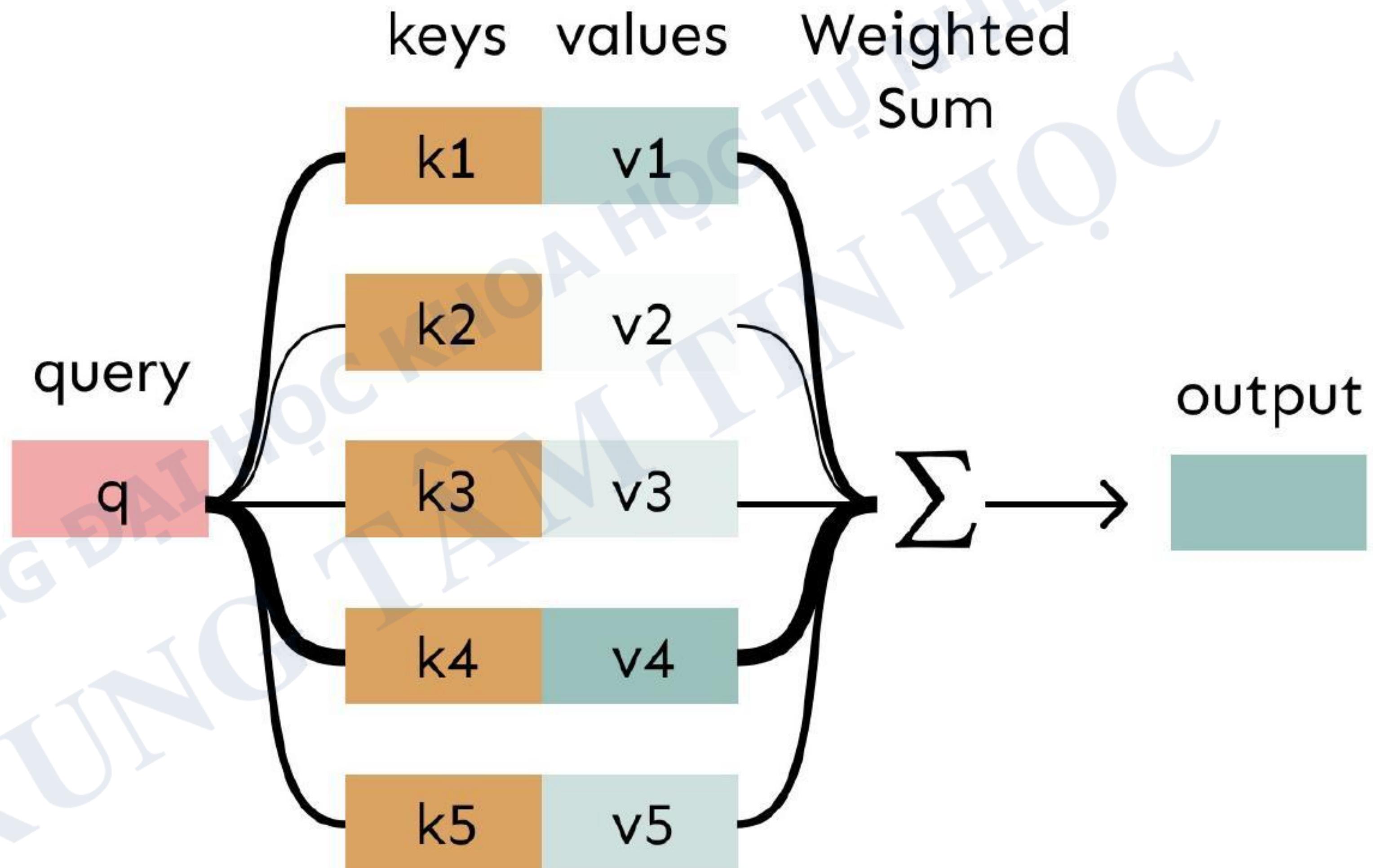
Khi truy vấn **query** có giá trị khớp với một **key**, **value** tương ứng sẽ được trả về.





Tổng quan Self-Attention Mechanism

Với attention mechanism, **query** khớp với tất cả các **key** với trọng số từ 0 - 1.

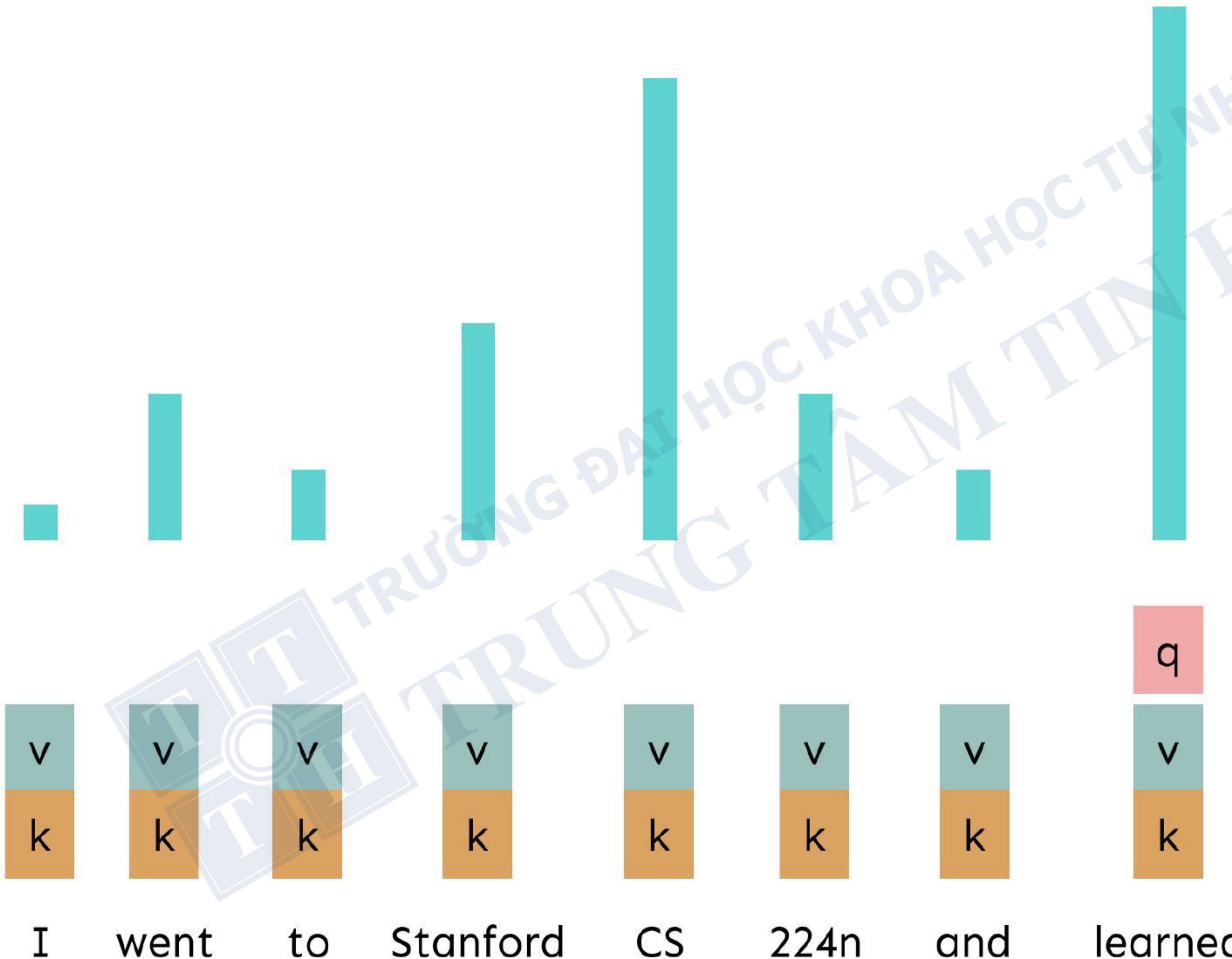


$$\text{Output} = \sum (\text{Value của các key} * \text{Trọng số})$$



Ví dụ về Self-Attention

Trọng số
attention
cho chữ
“learned”





Tổng quan Self-Attention Mechanism

□ Biết $w_{1:n}$ là một chuỗi các từ trong vocabulary V , ví dụ:

Paul made his uncle tea.

Với mỗi w_i , biết $x_i = Ew_i$, trong đó $E \in \mathbb{R}^{d \times |V|}$ là một embedding matrix.

1. Nhân mỗi word embedding với các ma trận Q, K, V, mỗi ma trận thuộc $\mathbb{R}^{d \times d}$

$$\mathbf{q}_i = Q\mathbf{x}_i$$

$$\mathbf{k}_i = K\mathbf{x}_i$$

$$\mathbf{v}_i = V\mathbf{x}_i$$

2. Tính sự tương đồng giữa mỗi cặp **key** và **query**, chuẩn hóa với hàm softmax

$$e_{ij} = \mathbf{q}_i^\top \mathbf{k}_j$$

$$a_{ij} = \frac{\exp(e_{ij})}{\sum_{i'} \exp(e_{ij'})}$$

3. Tính output cho mỗi từ dưới dạng tổng các giá trị có trọng số

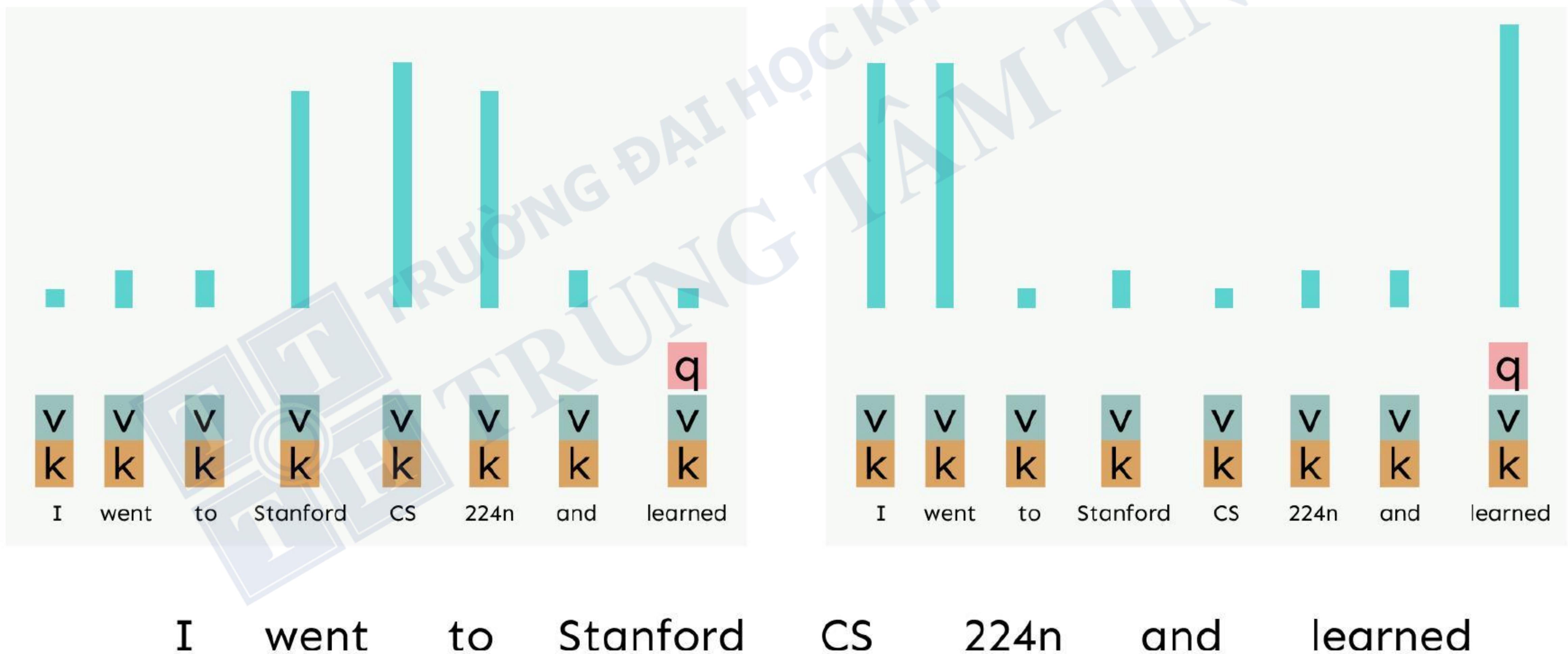
$$\text{output}_i = \sum_j a_{ij} \mathbf{v}_j$$



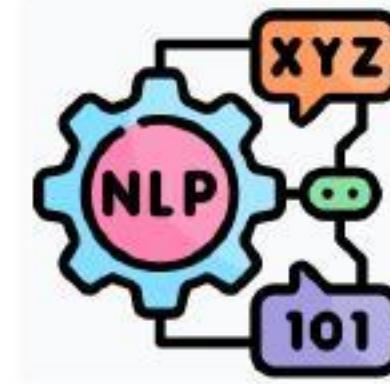
Tổng quan Multi-Head Attention

Attention head 1 tập trung vào các **từ quan trọng**.

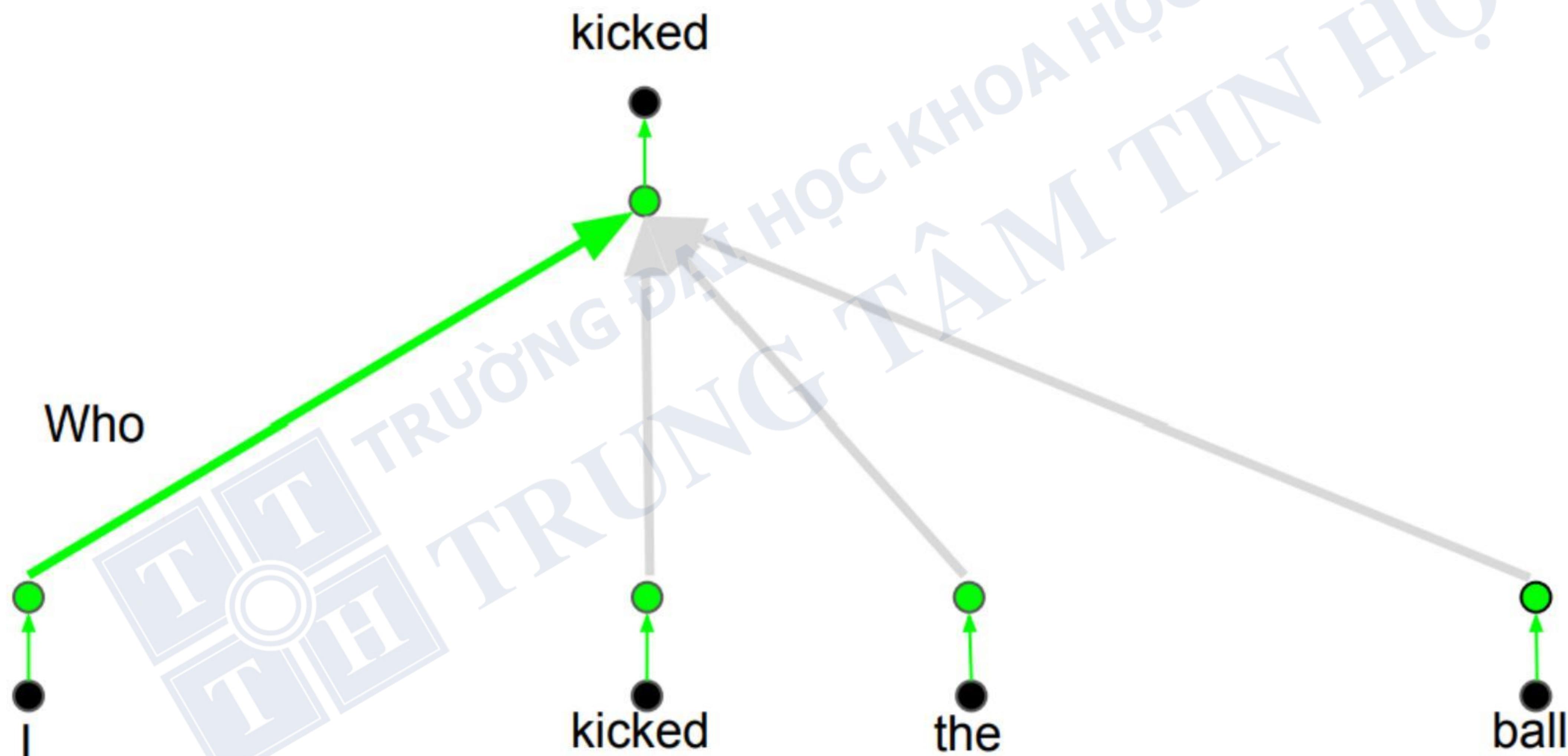
Attention head 2 tập trung vào các **từ liên quan về mặt cú pháp**.



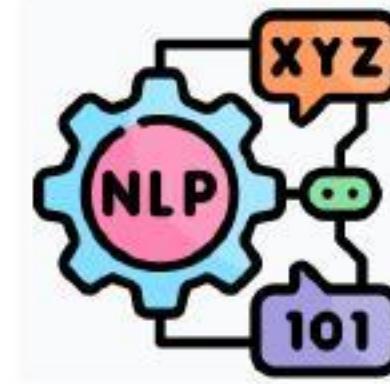
Self-Attention và Multi-Head Attention



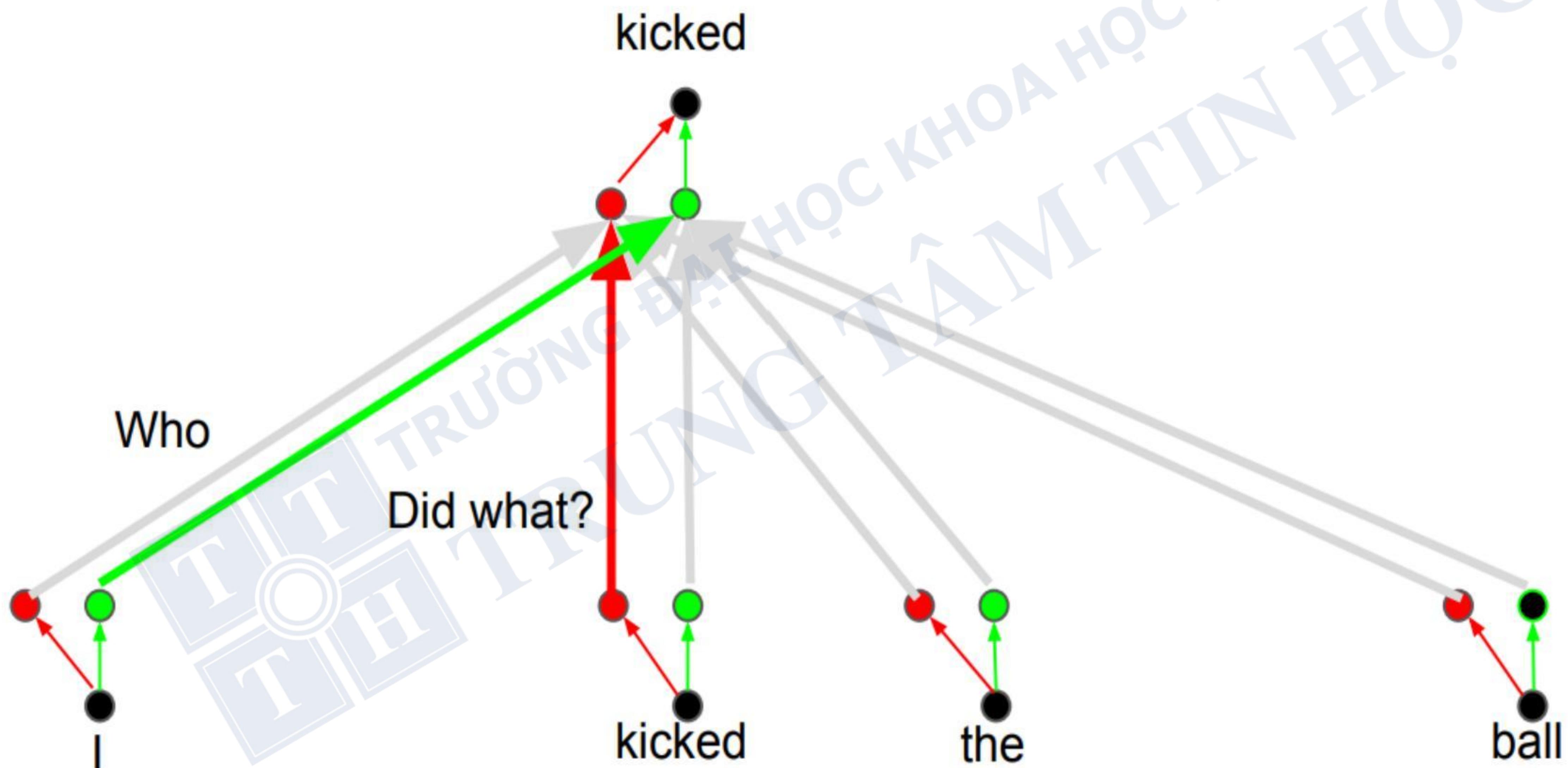
Attention head: WHO



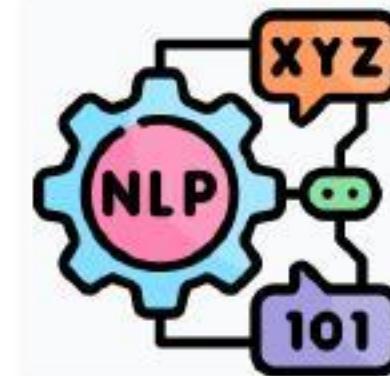
Self-Attention và Multi-Head Attention



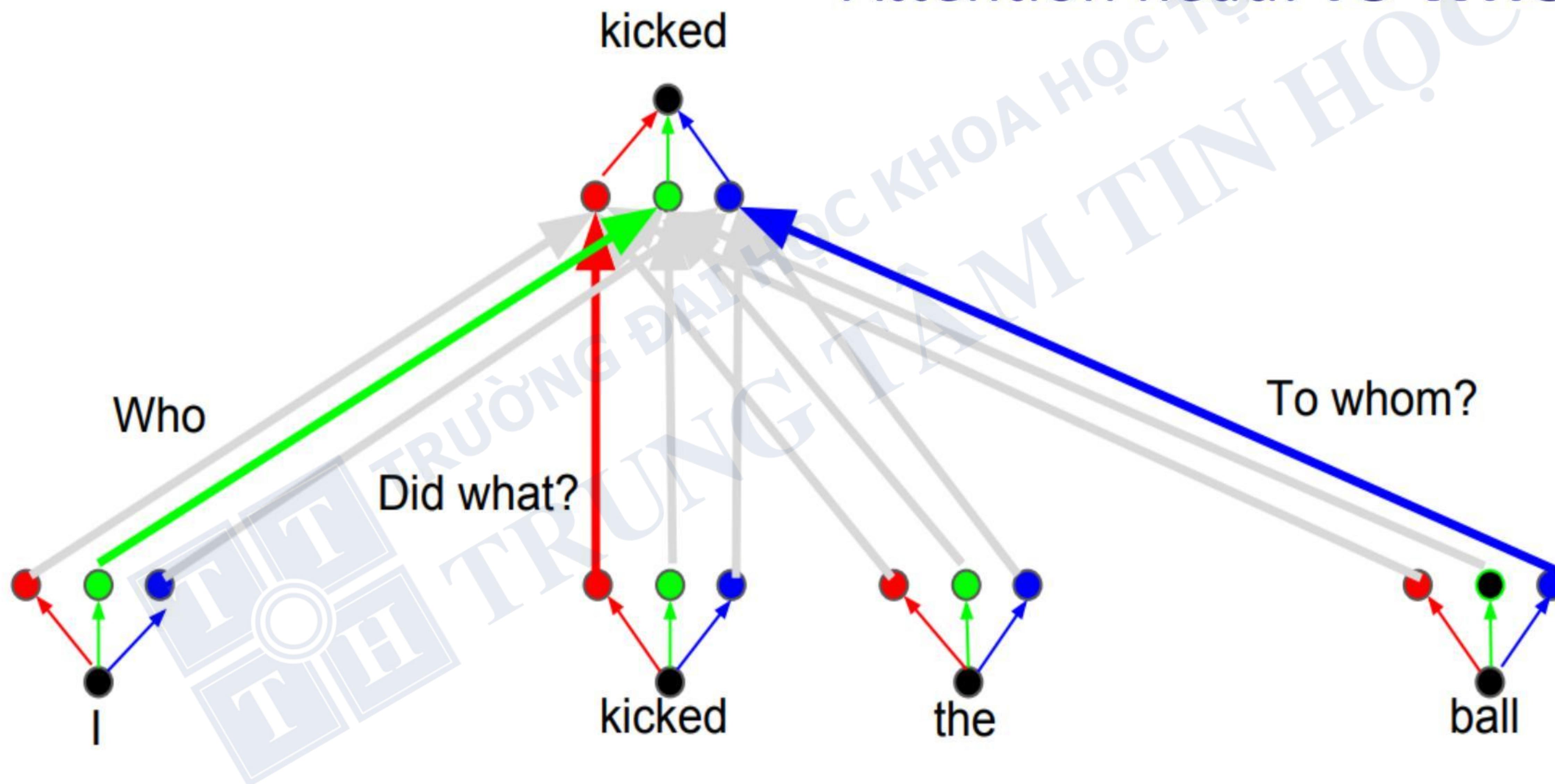
Attention head: DID WHAT



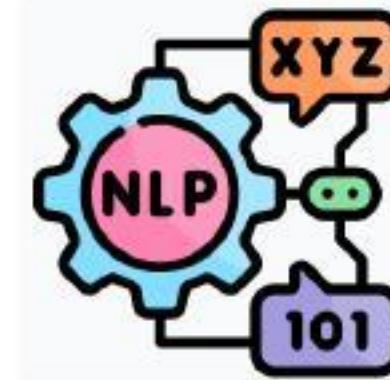
Self-Attention và Multi-Head Attention



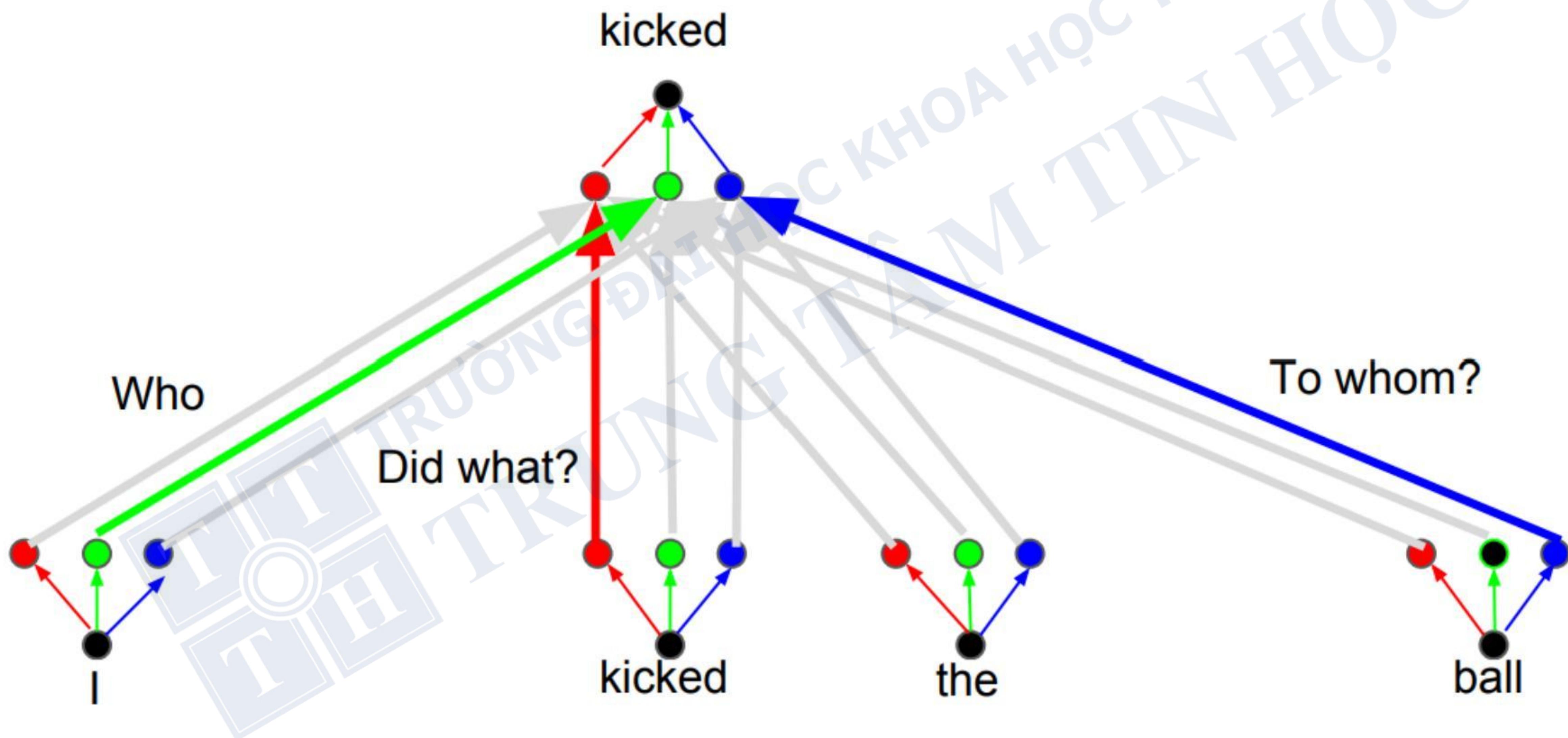
Attention head: TO WHOM



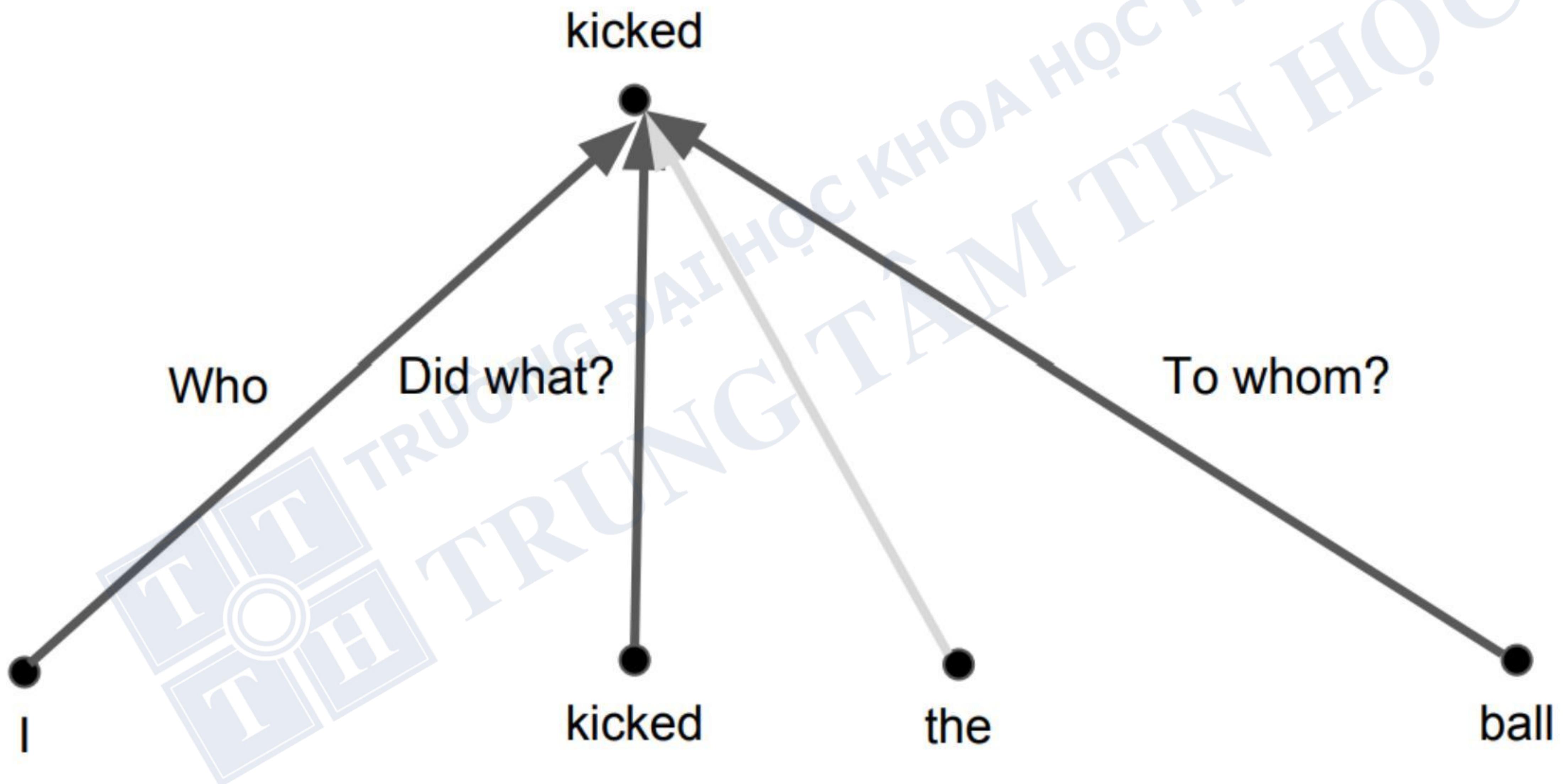
Self-Attention và Multi-Head Attention



Multi-Head Attention



Self-Attention: Average





Ví dụ về Attention Mechanism

TT TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TH TRUNG TÂM TIN HỌC



Bài 4: ATTENTION BASED NETWORK – LSTM



I. LSTM Encoder – Decoder

II. Attention Mechanism

III. Transformer Model



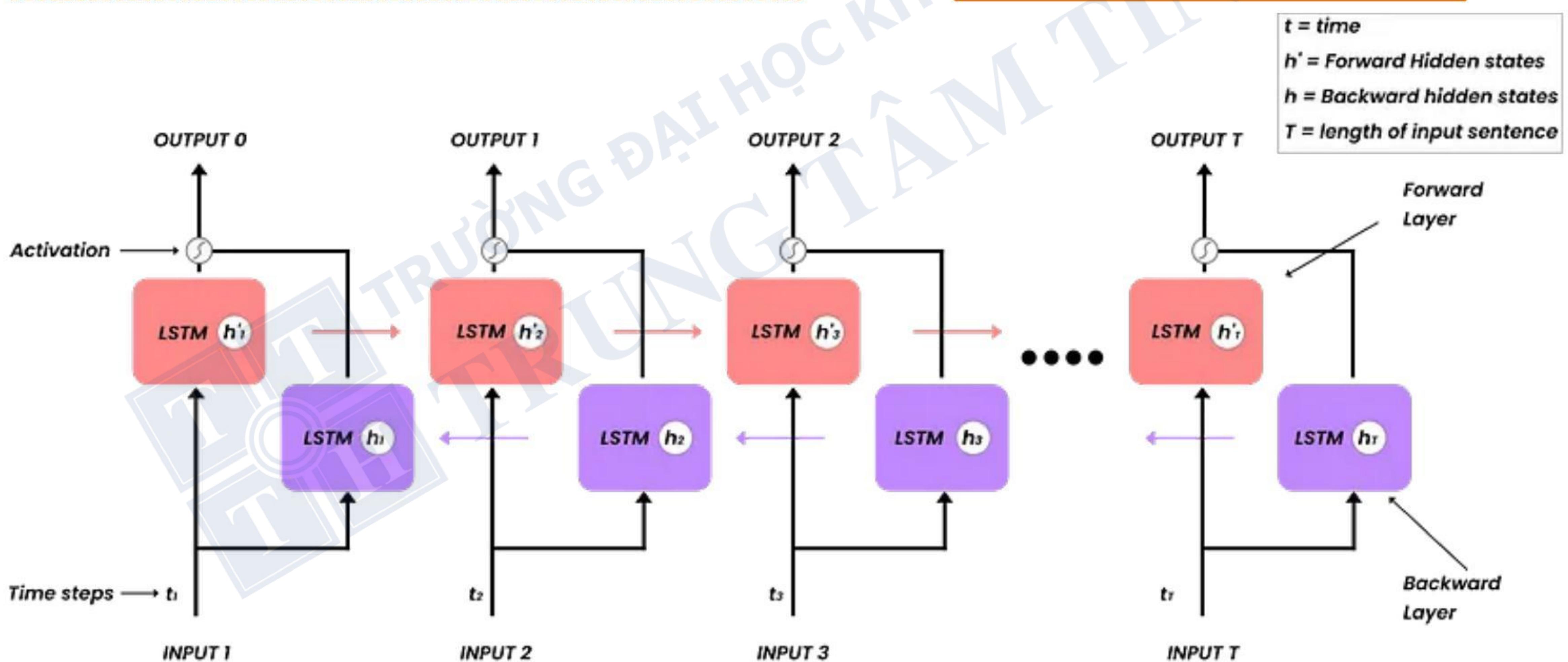


Transformer Model

- Mạng RNN như LSTM tồn tại **Bottleneck Problem**.

Ở **output layer** ẩn hiện tại phải encode tất cả thông tin đã được xử lý từ đầu cho đến thời điểm hiện tại.

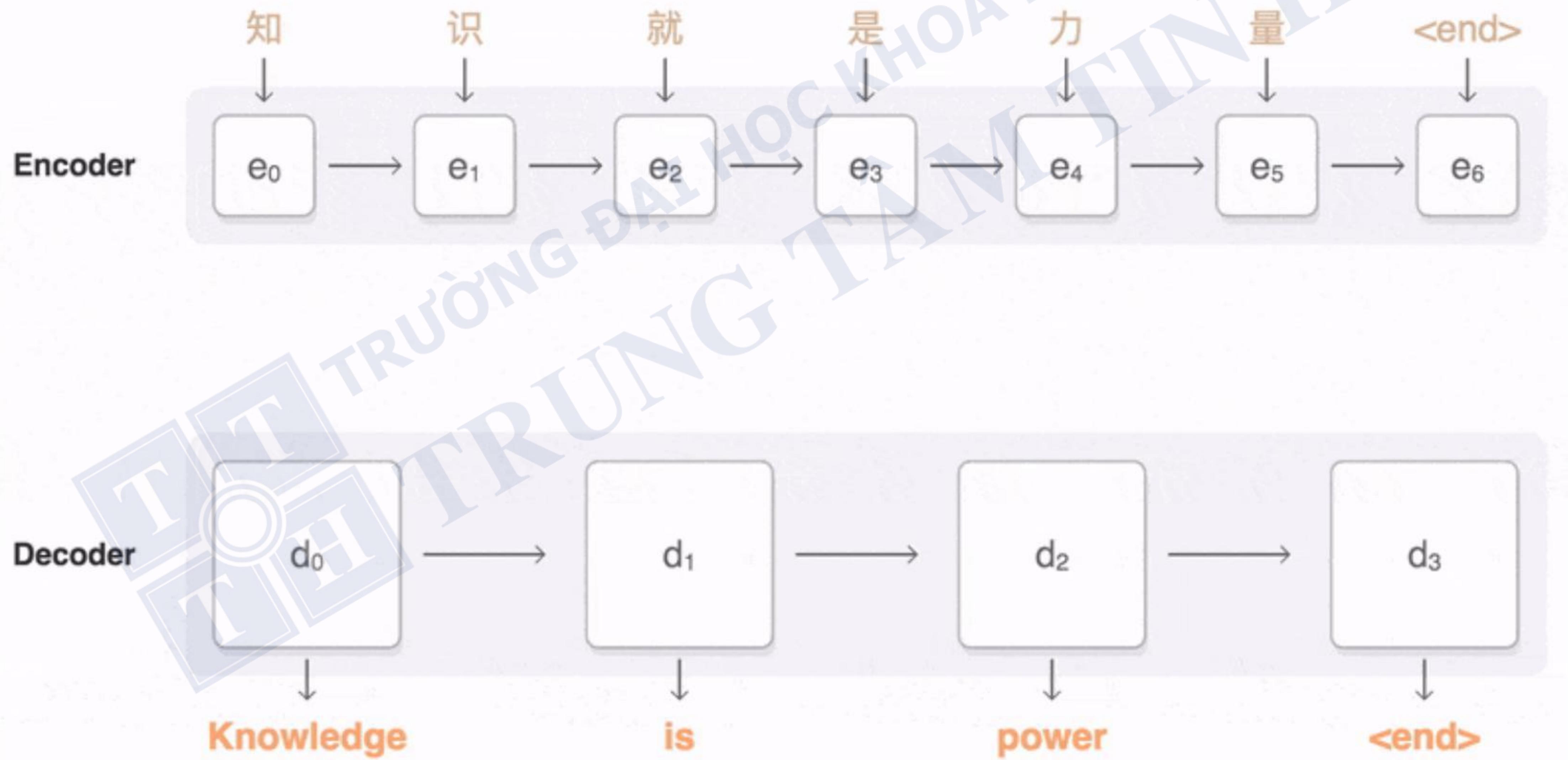
Chuỗi văn bản càng dài, quá trình tính toán càng chậm và khó hơn.



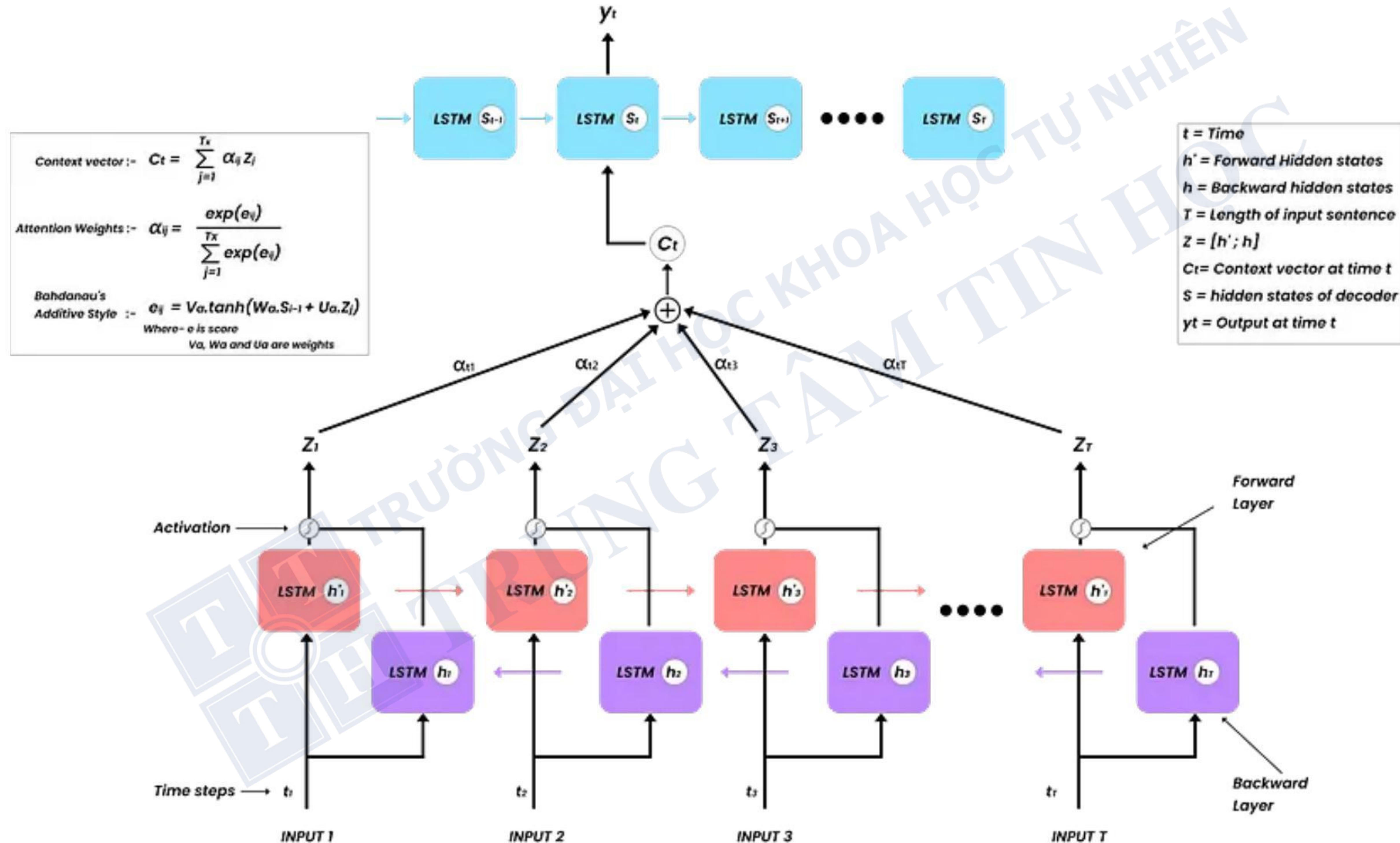


Transformer Model

→ Thay vì lưu trữ toàn bộ thông tin lên output của layer cuối.
Ta giữ các state ẩn của các layer cũ bằng **Attention Mechanism**.



Transformer Model





Ví dụ về Transformer Model

TT TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TTH TRUNG TÂM TIN HỌC



Code Demo



→ Truy cập vào:

[LINK](#)



Q&A

