NEURAL MACHINE TRANSLATION

Mục lục

1. Tổng quan kiến trúc mô hình Neural Machine Translation và phương pháp đánh giá mô hình NMT (Bleu Score)

1.1 Tổng quan kiến trúc mô hình NMT

a. Trước NMT

Cách dịch truyền thống: Chia nhỏ các câu thành các cụm từ và tiến hành dịch trên từng cụm từ một. Kết quả cuối cùng là một câu ghép lại các cụm từ đã được dịch. Cách tiếp cận này gọi là dịch theo cum pharase-based.
Nhược điểm: Kết quả không giống với cách con người sử dụng trong dịch thuật là đọc toàn bộ câu, tùy thuộc vào văn cảnh để nắm ý nghĩa của câu đưa ra ccaau dịch tương ứng.

b. Neural Machine Translation (NMT)

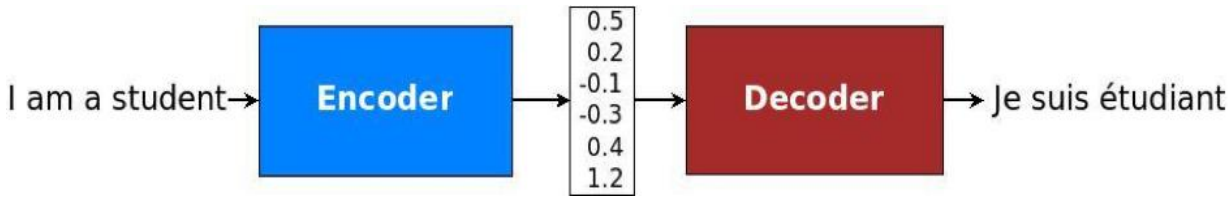
NMT (Neural Machine Translation) là sự kết hợp giữa dịch máy (Machine Translation – MT) và mạng nơ-ron nhân tạo (Artificial Neural Network – NN).
Cụ thể, mạng NN được sử dụng trong mô hình NMT là mạng nơ-ron hồi quy (hoặc truy hồi) (Recurrent Neural Network – RNN) và mô hình NMT được sử dụng là mô hình xây dựng theo kiến trúc NMT của Google, trên nền tảng thư viện Tensorflow dành cho Python theo Source code (https://github.com/tensorflow/nmt).

Một kiến trúc NMT được xây dựng dựa trên sự kết hợp của 2 thành phần chính là Sequence-to-Sequence (Seq2Seq) và Attention.

1.1.1 Sequence to Sequence (Seq2Seq)

Kiến truc seq2seq trong bài luận này dựa trên bài nghiên cứu Sequence to Sequence Learning with Neural Networks (Sutskerver et Alcohol., 2014).

Ý tưởng chính của mô hình Seq2Seq là sử dụng bộ mã hóa (Encoder) và bộ giải mã (Decoder) nối tiếp nhau để dịch một ngôn ngữ nguồn (seq) sang một ngôn ngữ đích (seq).

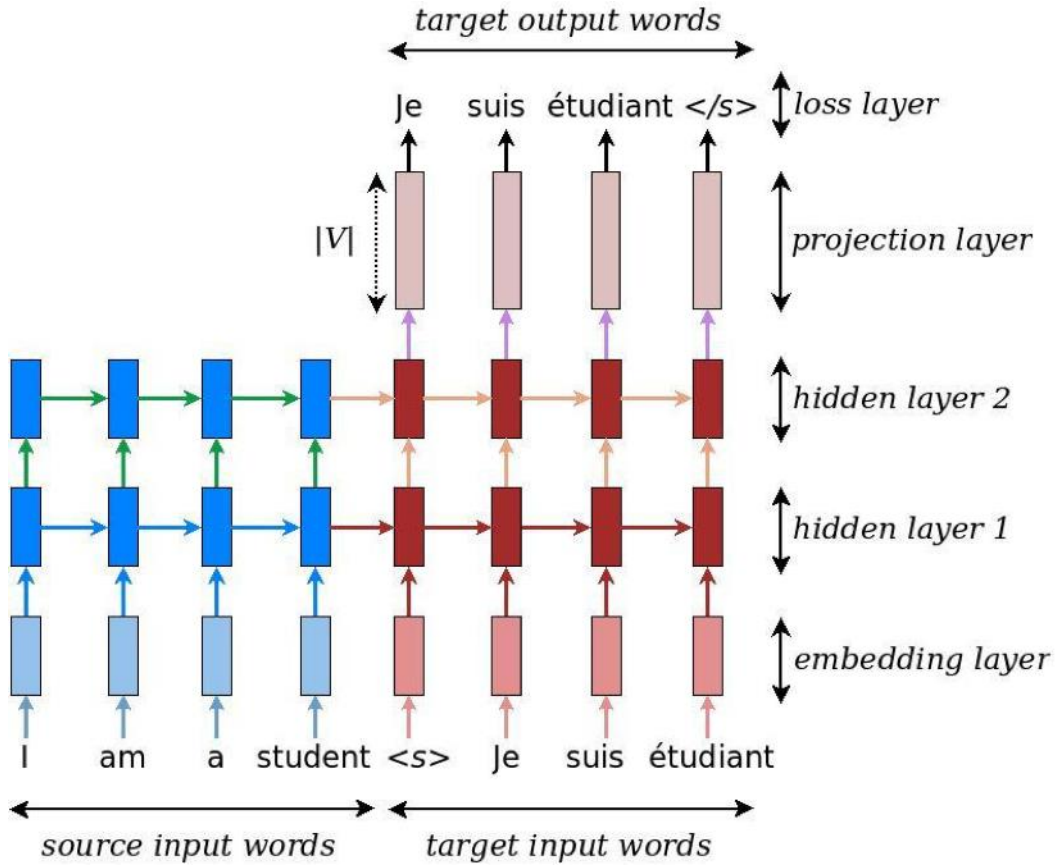


Hình 1. Kiến trúc tổng quan Encoder – Decoder
(Nguồn: https://github.com/tensorflow/nmt)

Cụ thể:

- NMT sẽ đưa toàn bộ câu ở ngôn ngữ gốc vào Encoder để nén ý nghĩa của câu thành một vector gọi là context (hoặc throught)
- Đưa vector đó sang cho bộ giải mã (Decoder) để chuyển vector thành câu thuộc ngôn ngữ khác (ngôn ngữ đích) mang ý nghĩa tương ứng với câu thuộc ngôn ngữ gốc.

Bài luận này sẽ sử dụng Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU) hoặc Long Short-term Memory (LSTM) để làm Encoder và Decoder.



Hình 2. Ví dụ cấu trúc NMT.

Bộ Encoder (màu xanh) và Decoder (màu đỏ) đều được tạo từ 2 lớp RNN cùng chiều, chồng lên nhau. Kí hiệu <s> để báo hiệu bắt đầu quá trình Decoder và </s> để báo hiệu dừng quá trình Decoder. Ngoài ra còn các lớp:

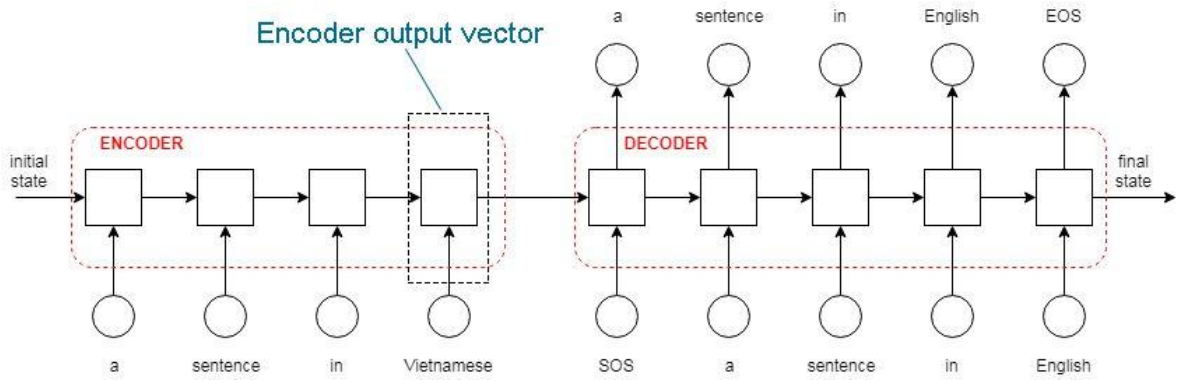
- 1 lớp Embedding ở Encoder
- 1 lớp Embedding và 1 lớp projection ở Decoder (2 lớp này dung chung bộ trọng số, chỉ số ngược chiều)

Đây là các lớp NN embedding, ứng với mỗi ngôn ngữ sẽ có một bộ NN Embedding riêng biệt. Vai trò của các thành phần:

- Embedding: chuyển một từ trong không gian từ điển (vocab) của ngôn ngữ sang không gian vector – có chiều tương ứng với không gian của vector context.
- Projection: chuyển ngược lại một từ thuộc không gian vector sang không gian từ điển (vocab) của ngôn ngữ
- Encoder: nén ngữ nghĩa của một câu (tập hợp từ thuộc không gian vector) của ngôn ngữ gốc thành một vector context
- Decoder: giải nén vector context thành một câu (tập hợp các từ thuộc không gian vector) của ngôn ngữ đích.

Các chiều không gian vector đều có chung chiều là chiều của vector context (kích thước mạng RNN)

** Vấn đề của kiến trúc Seq2Seq **

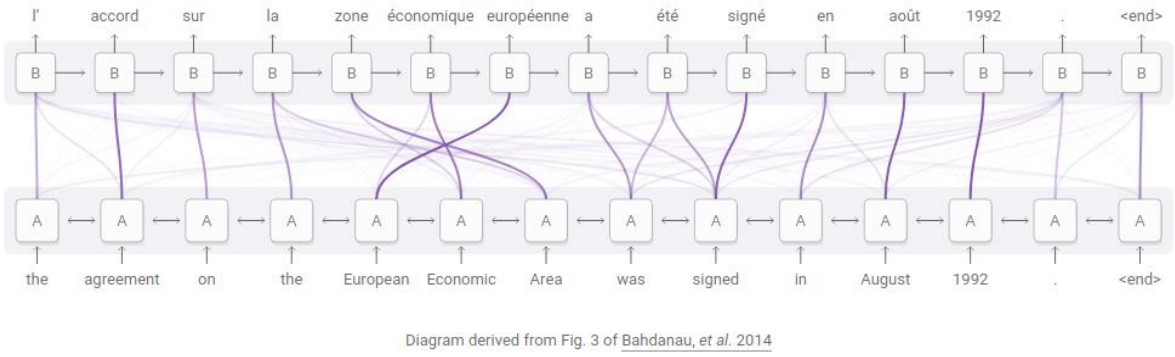


Hình 3. Vấn đề của kiến trúc Encoder – Decoder

Khi đưa một câu nguồn vào bộ decoder thì các từ sẽ đi vào các cell tuần tự theo thời gian, vector output của cell phía trước cùng với vector biểu diễn của từ tiếp theo sẽ là input của cell tiếp theo. Điều này dẫn đến output của cell cuối cùng cần phải chứa tất cả thông tin về câu nguồn cần dịch. Do đó dẫn đến việc vector output sẽ bị quá tải về mặt thông tin

1.1.2 Attention

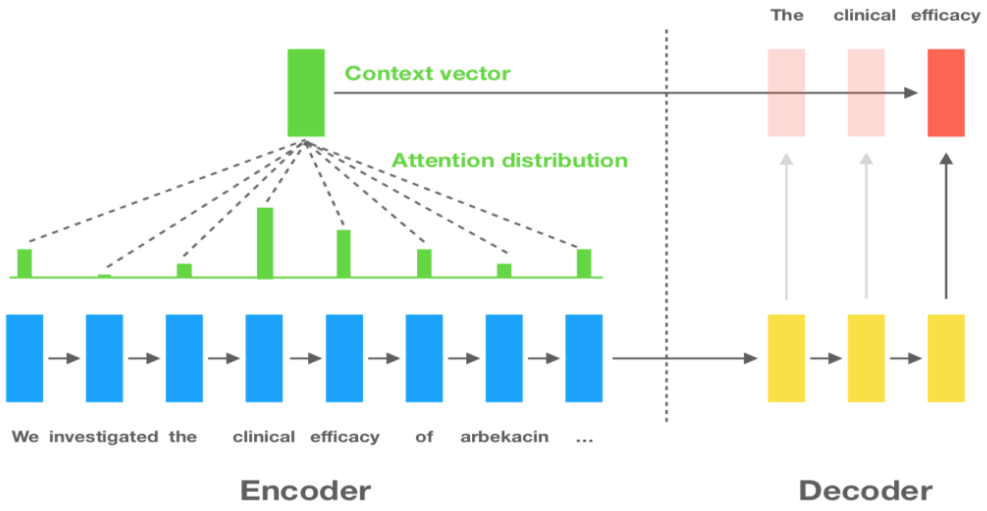
Thay vì mong muốn phần Encoder nén toàn bộ thông tin, ta cho phép phần Decoder được quan sát toàn bộ output của Encoder



Hình 4. Biểu diễn cơ chế Attention(Nguồn: Bahdanau, et al.2014)

Việc Encoder toàn bộ thông tin từ source vào 1 vector cố định khiến việc mô hình khi thực hiện các câu dài (long sentence) không thực sự tốt, mặc dù sử dụng LSTM (BiLSTM, GRU) để khắc phục điểm yếu của mạng RNN truyền thống với hiện tượng Vanishing Gradient, nhưng như thế có vẻ vẫn chưa đủ, đặc biệt đối với những câu dài hơn trong training data.

Từ đó, tác giả Bahdanau đề xuất 1 cơ chế cho phép mô hình có thể chú trọng vào những phần quan trọng (word liên kết với word từ source đến target), và thay vì chỉ sử dụng context layer được tạo ra từ layer cuối cùng của Encoder, tác giả sử dụng tất cả output của từng cell qua từng timestep, kết hợp với hidden state của từng cell để tổng hợp ra một context vector (attention vector) và dung nó làm đầu vào cho từng cell trong Decoder. Với NMT và Seq2Seq, việc này giúp mô hình hiểu được độ liên quan giữa input và từ cần predict tiếp theo tại Decoder.



Nguồn: https://pgigioli.github.io/project/2017/11/15/abstractive-title-generation.html

Từ nghiên cứu trên, có một số cách tính alignment score khác nhau như sau:

- Nghiên cứu của Bahdanau et al., 2015 có Additive Attention

$$score(s_{i-1}, h_j) = va^T \tanh\,(W_a s_{i-1} + U_a h_j)$$

- Nghiên cứu của Luong et al., 2015
  + Multiplicate Attention hay General Attention

$$score(s_{i-1}, h_j) = s_{i-1}^T W_a h_j$$

+ Dot Product (simple mechanism)

$$score(s_{i-1}, h_j) = s_{i-1}^T h_j$$

1.2 Phương pháp đánh giá mô hình NMT (Bleu Score)

Bleu Score (Bilingual Evaluation Understudy Score) là một thang điểm được dung phổ biến để đánh giá MT. BLEU được Kishore Papineni và cộng sự để xuất lần đầu vào năm 2002 qua bài nghiên cứu "A method for Automatic Evaluation of Machine Translation".
BLEU được tính dựa trên số lượng n-grams giống nhau giữa câu dịch của mô hình (output) với các câu tham chiếu tương ứng (lable) có xét tới yếu tố độ dài của câu. Thông thường n-grams được chọn là 4.

BLEU là một phương pháp dung để đánh giá chat lượng bản dịch được đề xuất bởi IBM tại hội nghị ACL ở Philadelphie vào tháng 7/2001. Ý tưởng chính là so sánh kết quả bản dịch tự động bằng máy với một bản dịch chuẩn dung làm bản đối chiếu. Việc so sánh được thực hiện thông qua việc thống kê sự trùng khớp của các từ trong hai bản dịch có tính đến thứ tự của chúng trong câu. (phương pháp n-grams theo từ). Phương pháp này dựa trên hệ số tương quan giữa bản dịch máy và bản dịch chính xác được thực hiện bởi con người để đánh giá chất lượng của một hệ thống dịch.
Việc đánh giá được thực hiện trên kết quả thống kê mức độ trùng khớp các n-grams (dãy ký tự gồm n từ hoặc ký tự) từ kho dữ liệu của kết quả dịch và kho các bản dịch tham khảo có chất lượng cao.

Giải thuật của IBM đánh giá chất lượng của hệ thống dịch qua việc trùng khớp các n-grams đồng thời nó cũng dựa trên việc so sánh độ dài của các bản dịch. Công thức để tính điểm đánh giá của IBM như sau:

$$score = \exp\left\{\sum_{i=1}^{N} w_i \log(p_i) - \max\left(\frac{L_{ref}}{L_{tra}} - 1{,}0\right)\right\} \qquad (1)$$

- $P_i = \dfrac{\sum_j NR_j}{\sum_j NT_j}$
- NR_j: là số lượng các n-grams trong phân đoạn j của bản dịch dùng để tham khảo.
- NT_j: là số lượng các n-grams trng phân đoạn j của bản dịch bằng máy.
- $w_i = N^{-1}$
- L_{ref}: là số lượng các từ trong bản dịch tham khảo, độ dài của nó thường là gần bằng độ dài của bản dịch bằng máy.
- L_{tra}: là số lượng các từ trong bản dịch bằng máy.

Giá trị score đánh giá mức độ tương ứng giữa hai bản dịch và nó được thực hiện trên từng phân đoạn, ở đây phân đoạn được hiểu là đơn vị tối thiểu trong các bản dịch, thông thường mỗi phân đoạn là một câu hoặc một đoạn. Việc thống kê đồ trùng khớp của các n-grams dựa trên tập hợp các ngrams trên các phân đoạn, trước hết là nó được tính trên từng phân đoạn, sau đó tính lại giá trị này trên tất cả các phân đoạn.

2. Kết quả và nhận xét về sự ảnh hưởng của siêu tham số đối với mô hình NMT

2.1Kết quả

Thông số mặc định

```
attention=None
unit_type=lstm
encoder_type=bi (2 chiều)
num_layers=2
num_units=128
num_train_steps=12000
optimizer=sgd
learning_rate=0.001
warmup_scheme=t2t
infer_mode=greedy

vi > en
```

Optimizer	Learning rate	Attention	Num layers	Num units	BLUE Score	Time
Adam	0.001	None	2	128	4.3	56’ (5)
SGD	0.001	None	2	128	0.0	52’
Adam	1.0	None	2	128	0.0	Overflow (stop early – 2’)
SGD	1.0	None	2	128	4.6	39’
Adam	0.001	scaled_luong	4	512	21.2	2h37’
SGD	1	scaled_luong	4	512	21.5	2h36’
Adam	0.001	scaled_luong	2	128	16.9	1h2’
SGD	1	scaled_luong	2	128	16.8	57’

Bảng 1: Kết quả train tiếng Việt – tiếng Anh

Nhận xét :

- Adam với lr = 0.001 sẽ cho kết quả tốt hơn lr = 1.0. SGD với lr = 1.0 sẽ tốt hơn lr=0.01
- Việc nâng cao số num_layers và num_units có làm tăng giá trị của BLUE_score tuy nhiên nếu điều chỉnh thích hợp thì sử dụng SGD với lr=1, num_layers=2, num_units=128 cho kết quả tương đối cao.
- Việc sử dụng attention có làm tăng giá của BLUE_score

**Sử dụng beam_search:

```
--infer_mode=beam_search \
--beam_width=10 \
--decay_scheme=luong234 \
```

Kết quả: sau khi sử dụng beam_search

Optimizer	Learning rate	Attention	Num layers	Num units	BLUE Score	Time
SGD	1	scaled_luong	2	128	22.8	1h25’
Adam	0.001	Scaled_luong	4	512	22.0	2h49’

Bảng 2: Kết quả sau khi sử dụng beam_search

Như vậy thì để hiệu quả nhất thì việc sử dụng mô hình với optimizer=sgd, num_units=2, num_layers=128, learning_rate=1 có thể cho kết quả tương đối trong thời gian ngắn với tập dữ liệu nhỏ.

2.2 Ảnh hưởng của siêu tham số đối với mô hình NMT

Việc lựa chọn các siêu tham số cho mô hình NMT sẽ tác động lớn đến độ phức tạp và độ nặng của hệ thống khi huấn luyện và chạy mô hình

a. Cấu trúc mô hình

- Loại RNN (unit_type): LSTM< GRU
- Số lớp (num_layers): độ sâu của mạng RNN, có thể là 2 lớp, 4 lớp hoặc thâm chí là 8 lớp RNN chồng lên nhau. Tuy nhiên trong nhiều trường hợp, mô hình quá sâu (4,8,.. lớp) không đem lại sự thay đổi đáng kể trong kết quả huấn luyện.(Reimers & Gurevych, 2017 và Britz et al.,2017)
- Số đơn vị (num_units): đặc trưng cho kích thước của mạng RNN, kích thước của vector context. Kích thước **2048** mang lại kết quả tốt nhất, nhưng khó áp dụng rộng rãi vì làm cho mô hình cực kỳ nặng. Kích thước **128** mang lại kết quả tương đối tốt với tốc độ huấn luyện nhanh và nhẹ hơn tới 16 lần. (Britz et al.,2017)

- Chiều của Encoder (encoder_type): 1 chiều, 2 chiều hoặc kết hợp một lớp 2 chiều và nhiều lớp 1 chiều. Và Encoder 2 chiều đã được nghiên cứu mang lại hiệu quả tốt hơn nhiều so với một chiều. (sutskever et al., 2014 & Britz et al., 2017)

b. Cơ chế tối ưu hóa

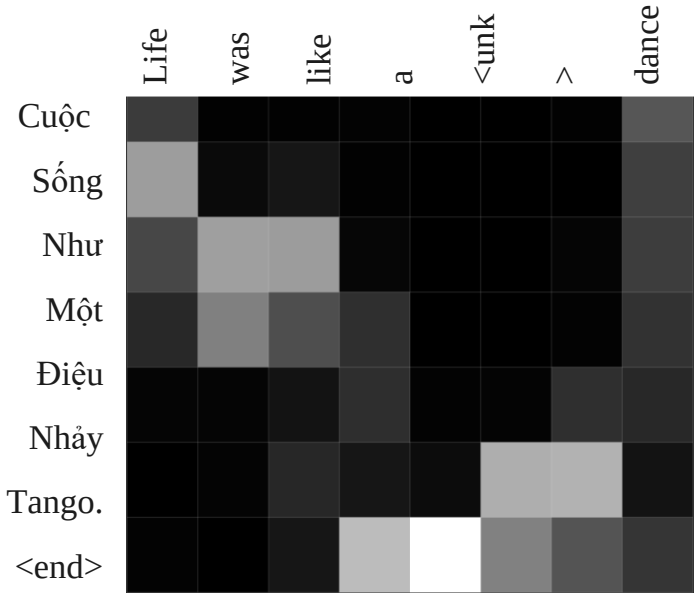
- Thuật toán tối ưu hóa (optimizer): có 2 lựa chọn là Adam và SGD (stochastic Gradient Descent). Adam mới được ra mắt song được sử dụng rộng rãi trong nghiên cứu NLP vì sự vượt trội rõ rang của Adam so với SGD. Một số nghiên cứu cho thấy với những tùy chỉnh hợp lý thì SGD đã có thể nhỉnh hơn Adam một chút (Ưu et.al, 2016) hoặc vượt trội hơn hẳn việc áp dụng momentum (Zhang & Mitliagkas, 2017).
- Cơ chế khởi động tốc độ học ban đầu và giảm dần khi về cuối (warmup_chêm & decay_scheme). Mô hình cho phép lựa chọn một số cơ chế tích hợp sẵn như :
    - Tensor2Tensor's warmup_scheme, khởi động với tốc độ học learning_rate nhỏ hơn 100 lần và tăng dần cho đến khi đạt được con số mong muốn.
    - Luong234 decay_scheme, sau 2/3 bước huấn luyện, bắt đầu giảm tốc độ học learning_rate 4 lần, mỗi lần 50%
    - Luong5 decay_scheme, sau ½ bước huấn luyện, bắt đầu giảm tốc độ học learning_rate 5 lần, mỗi lần 50%
    - Luong10 decay_scheme, sau ½ bước huấn luyện, bắt đầu giảm tốc độ học learning_rate 10 lần, mỗi lần 50%
- Bước huấn luyện (training_step): Dao động phụ thuộc vào độ phức tạp và khả năng huấn luyện, thường từ 10000 lần cho tới hàng trăm nghìn lần.

3. Ma trận Attention

Các Attention Matrix dưới đây được tạo bởi mô hình với BLUE score = 22.8

Ma trận 1

Input: “Cuộc sống như một điệu nhảy Tango.”
Output: “Life was like a <unk> dance.”

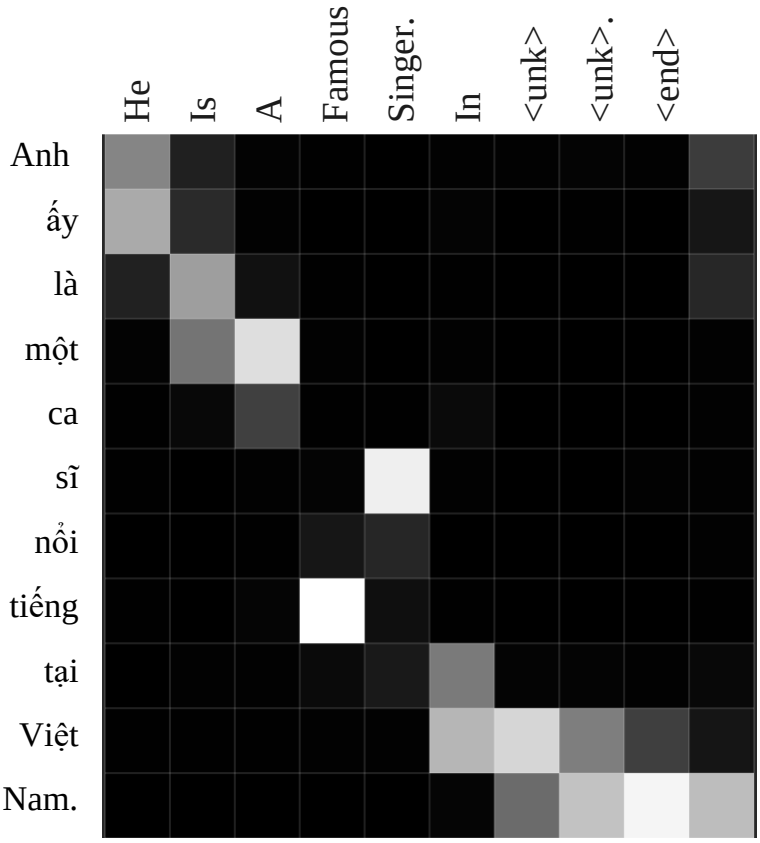


Hình 5: Biểu diễn ma trận Attention

Ma trận 2

Input: “Anh ấy là một ca sĩ nổi tiêng tại Việt Nam.”

Output: “He is a famous singer in <unk> <unk>.”

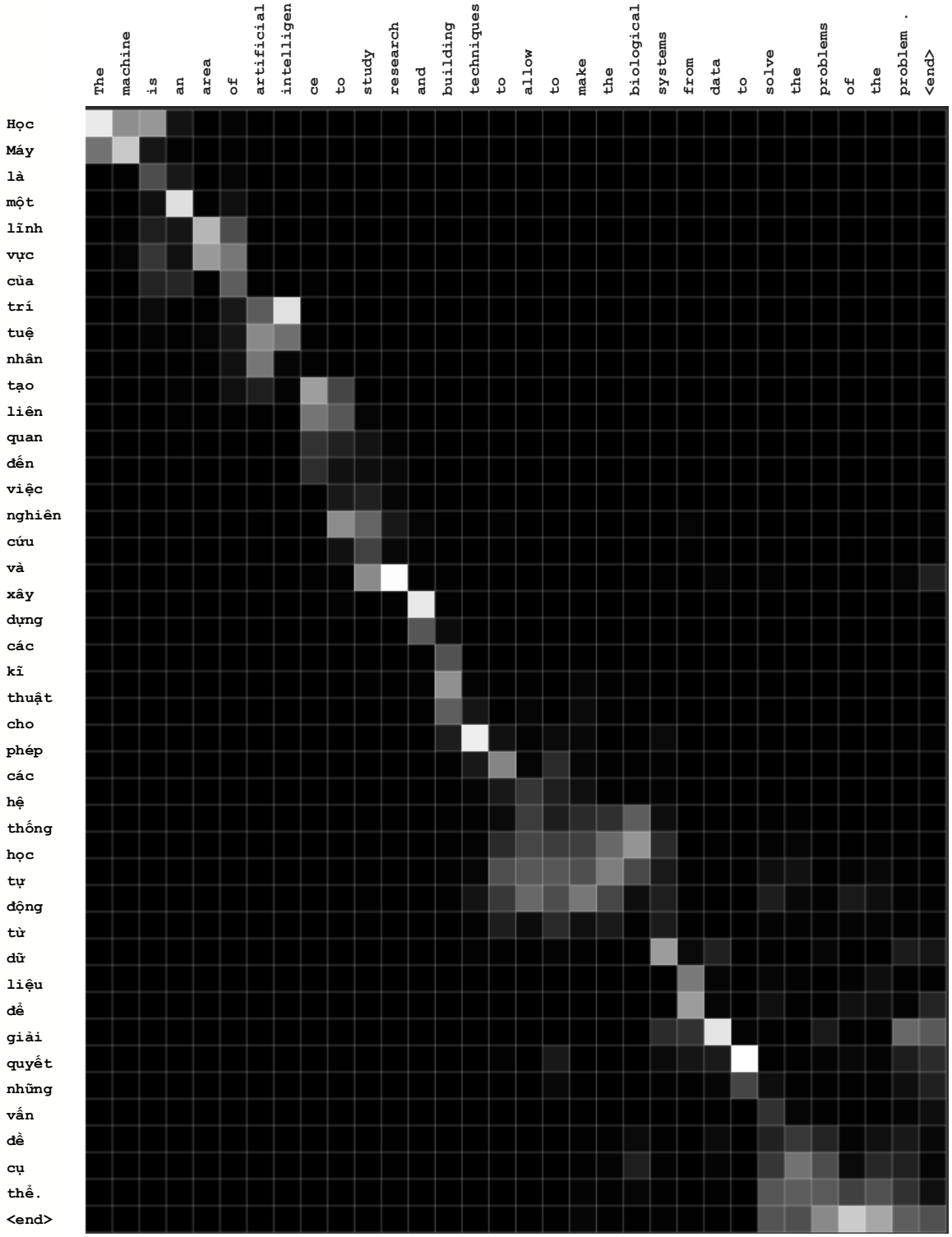


Hình 6 : Biểu diễn ma trận Attenton

Ma trận 3

Input:
“Học máy là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc nghiên cứu và xây dựng ác kĩ t huật cho phép các hệ thống học tự động từ dữ liệu để giải quyết những vấn đề cụ thể.”

Output: “The machine is an area of artificial intelligence to study research and building techniques to allow to make the biological systems from data to solve the problems of the problem .”



Hình 7: Biểu diễn ma trận Attention

Tài liệu tham khảo

1. Neural Machine Translation, mô hình dịch máy và đánh giá mô hình dịch máy

   (https://viblo.asia/p/neural-machine-translation-mo-hinh-dich-may-va-danh-gia-mo-hinh-dich-may-Qbq5QkVRZD8?fbclid=IwAR0VNFRS10y_sbqA10B3GPc57RZrUusUPKvSNIW-uDFh83WYEq0Rfn0plHU)

2. [Machine Learning] Attention, Attention, Attention, ...!

   (https://viblo.asia/p/machine-learning-attention-attention-attention-eW65GPJYKDO)

3. Neural Machine Translation (seq2seq) Tutorial

   (https://github.com/tensorflow/nmt)