BÁO CÁO BÀI TẬP LỚN

Môn Xử lý ngôn ngữ tự nhiên

Mô hình dịch máy

Nhật - Việt

Thành viên

Trần Hữu Trí

Phạm Minh Tiến

Lê Trọng Đức

Phạm Văn Thành

Giảng viên hướng dẫn : PGS. TS Lê Anh Cường

Mục lục

Lời mở đầu

Những năm gần đây, dịch máy đóng một vai trò quan trọng trong việc hỗ trợ con người cập nhật thông tin từ nhiều nguồn ngôn ngữ khác nhau một cách nhanh chóng. Trong sự phát triển của dịch máy có nhiều cách tiếp cận như sử dụng các kiến thức về ngôn ngữ học hoặc sử dụng các phương pháp thống kê. Những năm gần đây có một cách tiếp cận mới là sử dụng mạng nơ-ron nhân tạo để giải quyết bài toán dịch máy. Cách tiếp cận này hiện đang là một hướng phát triển đầy tiềm năng, thu hút được sự quan tâm của các nhà nghiên cứu.

Trong khuôn khổ khóa học NLP do viện AIAcademy giảng dạy, chúng em thực hiện đề tại xây dựng hệ thống dịch máy từ tiếng Nhật sang tiếng Việt sử dụng mô hình Transformer.Trong đồ án này, nhóm tập trung vào huấn luyện mô hình Transformer, triển khai một số mô hình đơn giản khác (như LSTM-Attention) để so sánh kết quả, cùng với đó triển khai thêm một số kĩ thuật, tinh chỉnh để cải thiện chất lượng mô hình

Báo cáo có bố cục gồm 3 phần:

Chương 1 giới thiệu bài toán và phương hướng tiếp cận.

Chương 2 trình bày một số kiến thức về ngôn ngữ Nhật-Việt và các mô hình mạng nơ ron sẽ áp dụng.

Chương 3 trình bày quá trình xây dựng

Chương 1: Giới thiệu bài toán

1.1 Tổng quan đề tài

Dịch máy là một mảng khá lâu đời trong Xử lý ngôn ngữ tự nhiên. Nhiệm vụ của nó là tìm cách để cho máy tính có thể dịch các văn bản hoặc tiếng nói từ một ngôn ngữ sang ngôn ngữ khác mà không có bất kì sự tác động nào của con người. Yêu cầu cơ bản của bài toán dịch máy cũng tương tự như các bài toán NLP khác, đó là máy tính phải hiểu ngôn ngữ hay tiếng nói của con người ở mức ngữ nghĩa.

Bài toán Dịch máy đã được nghiên cứu từ khá lâu và có nhiều phương pháp hiệu quả được công bố trong những thập kỉ qua bao gồm các phương pháp dùng luật hay các mô hình xác suất.Gần đây, với sự phát triển của DeepLearning, nhiều mô hình vượt trội đã được đề xuất như NMT, có khả năng tận dụng các phụ thuộc về ngữ nghĩa cho câu dài. Chính vì thế mà các mô hình dịch máy (như của Google) gần đây chỉ cần thời gian huấn luyện và lượng dữ liệu khổng lồ đã gần đạt tới mức hoàn hảo. Tuy nhiên có hàng trăm ngôn ngữ trên thế giới, mô hình dịch máy của Google là tốt, tuy nhiên với các cặp ngôn ngữ không phổ biến thì vẫn có thể đánh bại.

Tại thời điểm hiện tại, quan hệ giữa Việt Nam và Nhật Bản khá tốt, có khá nhiều người Nhật đang sinh sống và làm việc tại Việt Nam và ngôn ngữ luôn là một trong những rào cản lớn để họ hòa nhập cuộc sống trên một đất nước khác. Mô hình dịch máy của Google cho ngôn ngữ Nhật-Việt vẫn chưa đạt tới mức giới hạn và họ cũng không tập trung vào mô hình này. Nhóm chúng em muốn thực hiện bài toán Dịch máy để có các kĩ năng làm về các mô hình chuỗi-chuỗi trong xử lý ngôn ngữ tự nhiên và lựa chọn dịch từ tiếng Nhật sang tiếng Việt vì nhận thấy vẫn còn cơ hội vượt qua các mô hình tốt nhất hiện tại và nó cũng có như cầu lớn trong thực tiễn.

1.2 Phương hướng tiếp cận

Nhóm sẽ tiếp cận giải quyết bài toán này theo các bước sau

- Bước 1: Tìm hiểu về ngôn ngữ tiếng Nhật, sự khác nhau về hình thái từ, ngữ pháp giữa hai ngôn ngữ. Tìm hiểu về các kĩ thuật feature engineering trong tiếng Nhật. Từ đó nhóm có thể hiểu thêm về dữ liệu và có thể xử lý được nó.

- Bước 2: Tìm hiểu về các mô hình học sâu hiệu quả trong các bài toán học máy gần đây. Qua khảo sát, nhóm lựa chọn mô hình có kiến trúc encoder-decoder (cụ thể sẽ được giải thích ở chương sau). Nhóm sẽ triển khai lại một số mô hình để có thể so sánh kết quả với nhau.

- Bước 3: Huấn luyện mô hình, tinh chỉnh các tham số, để có thể lựa chọn mô hình tốt nhất dựa trên tập dev và kiểm thử so sánh các kết quả trên tập dev

Phương pháp đánh giá nhóm sử dụng là điểm BLEU-score, sử dụng script đánh giá multi-bleu.pl của OpenNMT.

1.3 Mục tiêu mong muốn

Nhóm chúng em muốn xây dựng hai mô hình hiệu quả và phổ biến trong thời gian gần đây cho các bài toán dịch máy đó là LSTM-Attention và Transformer để dịch các văn bản từ tiếng Nhật sang tiếng Việt. Trong quá trình xây dựng mô hình, nhóm sẽ phân tích các khó khăn trong bài toán dịch Nhật - Việt và tìm ra các xử lý. Đồng thời sử dụng các kĩ thuật trong xử lý ngôn ngữ tự nhiên như tiền xử lý hay feature engineering để cải thiện kết quả. Mục tiêu chính là xây dựng được mô hình có chất lượng tốt hơn so với mô hình học máy SMT.

Chương 2: Cơ sở lý thuyết

2.1 Kiến thức cơ bản của ngôn ngữ Nhật-Việt

Một số điểm khác nhau giữa ngôn ngữ Nhật và Việt

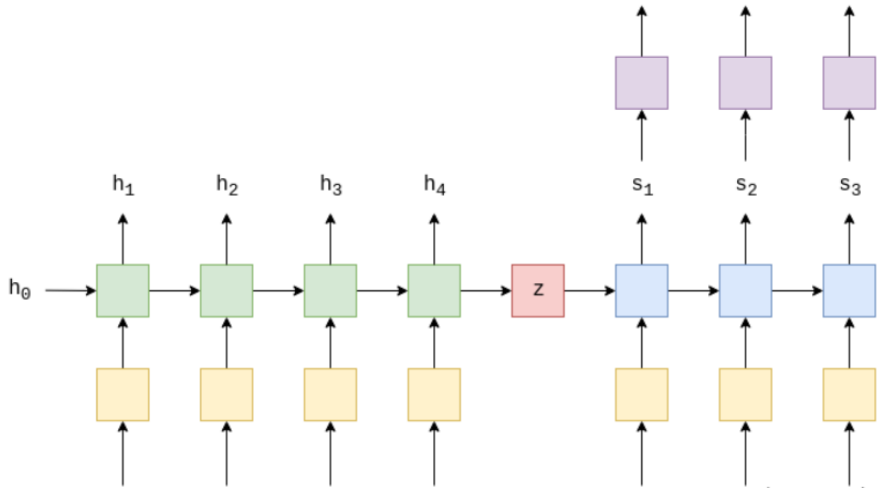
- Trong văn bản tiếng Nhật các âm tiết được viết liền với nhau còn tiếng Việt được ngăn nhau bởi dấu cách. Như vậy tách từ trong tiếng Nhật là tách các chuỗi âm cách ra trong mọt chuỗi liền mạch, còn ở tiếng Việt là tìm cách nối các âm lại với nhau.

- Về hình thái từ, tiếng Việt chỉ có duy nhất 1 bảng chữ guốc ngữ, trong khi đó ngôn ngữ Nhật có 3 bảng là Hiragana, Katakana và Kanji. Trong các ngữ cảnh ngữ pháp khác nhau thì tiếng Nhật có thể thay đổi cấu tạo từ hay tiếng Nhật có chua thì động từ (9 thì) trong khi tiếng Việt thì không. Hơn nữa, trong khi chia động từ ở tiếng Anh khá đơn giản (có quy tắc thêm s/es/ed/d hoặc 1 số lượng nhỏ bất quy tắc) thì trên tiếng Nhật có khoảng hơn 9 kiểu chia (tùy theo ý nghĩa), mỗi kiểu chia sẽ có 3 cách chia nhỏ (tùy theo động từ), mỗi cách chia nhỏ lại có thêm 2 dạng chia khẳng định - phủ định, hoặc một số có thể sẽ có thêm 2 dạng thân mật - lịch sự.

- Cấu trúc câu trong tiếng Nhật cũng khác hẳn với tiếng Việt và cả tiếng Anh. Trong tiếng Việt hay tiếng Anh, một câu có cấu trúc là <chủ ngữ> + <động từ> + <tân ngữ> nhưng ở tiếng Nhật thì như sau <chủ ngữ> + <tân ngữ> + <động từ>.

- ở mức nhỏ hơn là cụm từ, một cụm danh từ trong tiếng Việt thì vị trí danh từ chính là ở đầu , với tiếng Nhật thì nó được đặt ở cuối.

Có thể thấy ngôn ngữ Nhật có cấu trúc ngữ pháp và từ vựng vô cùng phức tạp, và tiếng Việt như chúng ta đã biết cũng có ngữ pháp phức tạp không kém đặc biệt là rất nhập nhằng về ngữ nghĩa. Việc xây dụng một mô hình dịch máy cho hai ngôn ngữ này sẽ khá thách thức.



Hình 2.1: Kiến trúc Transformer

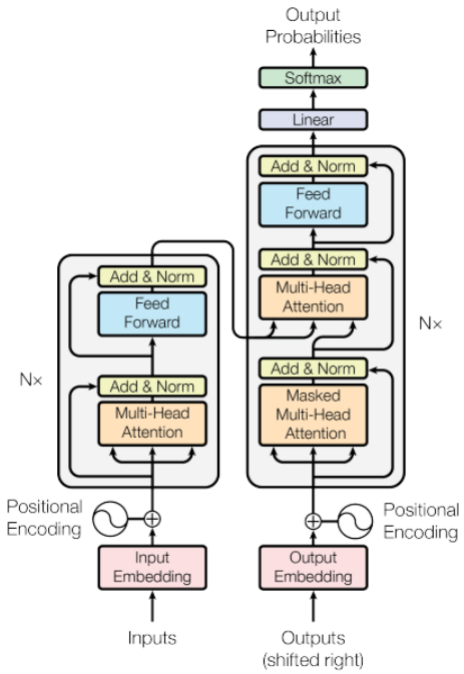
2.2 Sequence to Sequence

Mô hình sequence to sequence (seq2seq) là mô hình ánh xạ một chuỗi dữ liệu đầu vào với một chuỗi đầu ra thỏa mãn một mục tiêu nào đó. Các mô hình seq2seq thường có cấu trúc encoder-decoder. Cơ bản nhất, một mạng RNN sẽ mã hóa chuỗi đầu vào thành một véc tơ và véc tơ này sẽ được giải mã bằng một mạng RNN khác để có được chuỗi đầu ra mong muốn. Trong bài toán dịch máy thì hai chuỗi đầu vào và đầu ra sẽ là câu gốc và câu mục tiêu. Hình trên biểu diễn một ví dụ cho dịch máy. Câu nguồn sẽ được đưa qua một lớp embedding (màu vàng) và được đưa tiếp vào encoder (xanh lá). Các kí tự đặc biệt là '<sos>' và '<eos>' được thêm vào chuỗi đại diễn cho việc đánh dấu bắt đầu và kết thúc của câu. Mạng hồi quy là kiến trúc phổ biến nhất được lựa chọn cho các bộ encoder và decoder, khi đó tại mỗi bước huấn luyện, bộ mã hóa encoder nhận đầu vào là biểu diễn nhúng của từ hiện tại $e(x_t)$ và trạng thái ẩn có từ bước huấn luyện trước h_{t-1}

$$h_t = Encoder(e(x_t), h_{t_1})$$

Như vậy với chuỗi câu nguồn là $X = \{x_1, x_2, .., x_T\}$, khi từ cuối cùng của câu x_T được truyền qua tầng Encoder, ta có được trạng thái ẩn cuối cùng $h_T = z$ và coi nó là véc tơ biểu diễn cho toàn bộ câu nguồn. Véc tơ z sau đó sẽ được giải mã để ra câu đích mong muốn thông qua tầng Decoder. Tương tự, ta sử dụng các kí tự '<sos>' và '<eos>', mạng hồi quy ở tầng Decoder sẽ nhận đầu vào là biểu diễn nhúng của từ $d(y_t)$ và trạng thái ẩn ở bước huấn luyện trước s_{t-1}, với z đóng vai trò là véc tơ trạng thái ẩn khởi tạo.

$$s_t = Decoder(d(y_t), s_{t-1})$$



Hình 2.2: Kiến trúc Transformer

Ở tầng giải mã, ta cần biến đổi đưa trạng thái ẩn về một từ, do đó tại mỗi bước huấn luyện, ta sử dụng s_t để đoán từ tương ứng cho vị trí đó (có thể bằng cách đưa qua một lớp Linear) $\hat{y}_t = f(s_t)$

2.3 Mô hình transformer

Trong thời gian gần đây kể từ tháng 1 năm 2020, mô hình Transformers được coi là kiến trúc đột phá nhất trong NLP và được sử dụng để đạt được các kết quả state-of-the-art cho nhiều bài toán khác nhau. Một biến thể phổ biến nhất của Transformer là BERT và các tập BERT huấn luyện sẵn thường được sử dụng để thay thế lớp embedding hay toàn bộ mô hình.

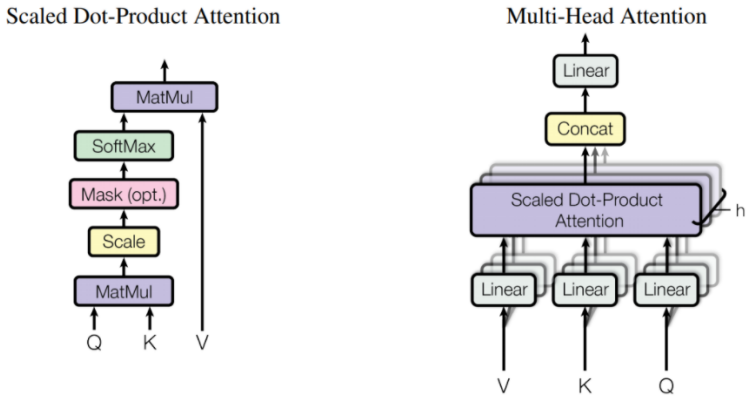
Tương tự, mô hình Transformer cũng sử dụng kiến trúc Encoder-Decoder. Trong đó, tầng mã hóa Encoder có nhiệm vụ mã hóa cầu đầu vào (câu nguồn) thành một véc tơ biểu diễn và tầng giải mã Decoder sẽ đưa véc tơ đó về câu gốc. Trong mô hình này, nổi bật nhất là kiến trúc Multi Head Attention

Multi Head Attention

Attention có các khái niệm về query, key và value. Tronng đó, query và key được sử dụng để sinh ra véc tơ attention để có tổng trọng số của value.

Mô hình Transformer sử dụng scaled dot-product attention, query và key được kết hợp lại sử dụng phép nhân vô hướng,kết quả đó được đưa qua toán tử softmax và được scale với hệ số d_k là số chiều của head (head dimension) để giảm thiểu việc tăng giá trị qua toán tử nhân vô hướng là quá lớn

$Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V$ Tuy nhiên, scaled dot-product attention không chỉ đơn ản áp dụng



Hình 2.3: Kiến trúc Transformer

cho query, key và value. Thay vì chỉ sử dụng một attention áp dụng cho các query, key và value, số chiều tầng ẩn được chia thành h đầu và được tính toán attention một cách song song. Sau đó ta lại kết hợp lại các đầu attention với nhau

$$MultiHead(Q, K, V) = Concat(heat_1, .., head_h)W^O$$
$$head_i = Attention(QW_i^Q, KWW_i^K, VW_i^V)$$

W^O, W^Q, W^K, W^V là các ma trận trọng số cho các lớp Linear tương ứng.

Encoder

Khác với mô hình sử dụng mạng hồi quy như đã nói ở trên, bộ encoder của Transformer sẽ không đưa cả chuỗi đầu vào $X = \{x_1, x_2, ..., x_n\}$ thành một véc tơ biểu diễn z. Thay vào đó, nó sẽ sinh ra một chuỗi các véc tơ $Z = \{z_1, z_2, ..., z_n\}$, các véc tơ này khác với trạng thái ẩn vì từng véc tơ được sinh ra khi mô hình quan sát được toàn bộ từ và vị trí của chúng trong chuỗi đầu vào chứ không phải chỉ một từ ở vị trí tương ứng

Đầu tiên, các token sẽ được đưa qua các lớp embedding. Tiếp đó, do mô hình không có tính hồi quy, nên nó không thể biết được thứ tự của các token trong chuỗi. Vấn đề này được giải quyết bằng một lớp embedding khác gọi là position embedding. Lớp này sẽ không nhúng nội ung từ đầu vào mà sẽ nhúng vị trí của từ trong chuỗi, đếm từ token '<sos>' mang giá trị 0. Kích thước "từ điển" của position embedding là độ dài lớn nhất của câu mà mô hình có thể nhận. Như vậy, trước khi được đưa vào mạng huấn luyện, ta sẽ có hai véc tơ embedding biểu diễn cho đầu vào, chúng sẽ được cộng lại để thu về một véc tơ bao gồm cả thông tin về token và vị trí của nó. Véc tơ đặc trưng này sẽ được truyền qua N lớp encoder để thu được đầu ra là Z và được truyền vào decoder.

Decoder

Mục tiêu của tầng decoderlà từ biểu diễn mã hóa của câu nguồn Z cho ra được chuỗi các từ trong câu đích $\hat{Y}$. Tầng giải mã khá giống tầng mã hóa, tuy nhiên nó có hai lớp multi-head attention. Một lớp dùng cho chuỗi câu đích làm đầu vào và một lớp cho chuỗi mã hóa. Ở lớp attention thứ hai, các giá trị biểu diễn mã hóa của câu nguồn được coi là key, value, đầu ra của lớp attention đầu tiên là query.

Tầng decoder cũng sử dụng position embedding và nó sẽ được kết hợp với biểu diễn nhúng của chuỗi câu đích đầu vào, kết quả cùng với biểu diễn mã hóa của chuỗi câu nguồn sẽ được đưa qua nhiều lớp decoder. Về lý thuyết thì số lớp encoder và decoder không nhất thiết phải bằng nhau, tuy nhiên ta nên để chúng bằng nhau để tạo sự cân bằng cho mô hình.
Kết quả khi đi qua các lớp decoder là một ma trận hay một chuỗi các véc tơ sẽ được đưa qua một lớp Linear để tính xác suất các từ có thể tương ứng với vị trí.

Chương 3: Thực ngiệm

3.1 Mô tả dữ liệu

Mô hình chúng em sử dụng yêu cầu bộ dữ liệu song ngữ, cụ thể, nhóm sử dụng bộ TED-talk của hai ngôn ngữ tiếng Việt và Nhật. Bộ dữ liệu được chia thành 3 tập train, dev, test, sau khi loại bỏ trùng lặp và các dòng trống nhóm thu được số lượng các cặp câu lần lượt là 106758, 568 và 1220.
Dữ liệu được tiền xử lý tách từ. Với tiếng Nhật nhóm sử dụng Kytea và với tiếng Việt là VnCoreNLP. Phân bố độ dài các câu trong tập huấn luyện ở hai ngôn ngữ như Hình 3.1 và 3.2
Một số vấn đề có thể nhận thấy trong tập dữ liệu

- Tập test cho ngôn ngữ tiếng Việt chỉ có một phiên bản dịch cho một câu, do đó đánh giá bằng BLEU score sẽ không thực sự có tính bao phủ rộng.

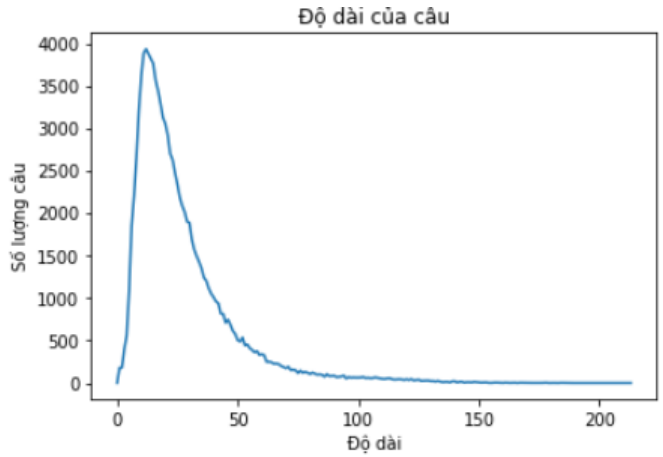
- Một số cặp câu, trong tiếng Nhật họ sử dụng số nhưng bên phía bản dịch tiếng Việt thì số đó viết ở dạng chữ. Ví dụ như ở tập Japan sang Vietnam thì "2" chuyển thành "hai". Nhóm đã khắc phục bằng cách lựa chọn các cặp câu mà chỉ chứa số trong một ngôn ngữ, sử dụng một từ điển để chuyển số về dạng chữ trong ngôn ngữ đó, tuy nhiên chỉ có thể xử lý các số có một chữ số và vẫn bỏ sót các trường hợp lúc số lúc chữ trong câu.

- Độ chênh lệch về độ dài giữa các cặp câu là khá cao, có cặp câu lệch nhau 40 từ. Do kiến thức về tiếng Nhật còn hạn chế, nhóm chưa thể kiểm chứng được liệu dữ liệu có chính xác hay không, nhưng nó khá ảnh hưởng đến việc huấn luyện cho tầng decoder. Đây là một vấn đề cần xử lý để cải thiện mô hình về sau.

3.2 Mô hình triển khai

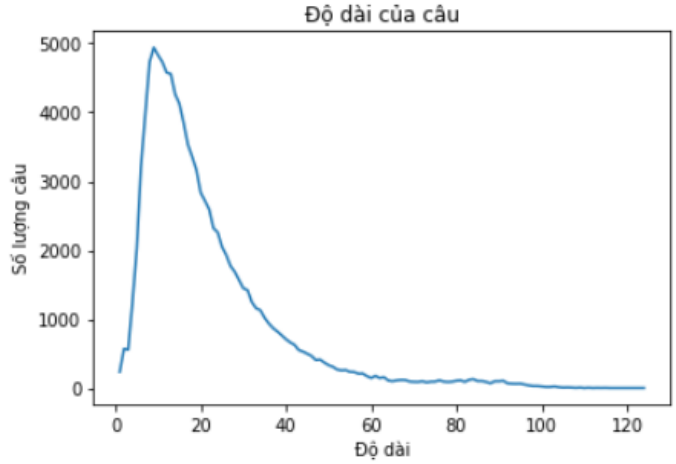
3.2.1 Mô hình LSTM-Attention

Nhóm triển khai mô hình sử dụng mạng LSTM, trong đó tầng encoder sẽ mã hóa câu nguồn sử dụng một lớp Bi-LSTM. Tầng decoder sẽ sử dụng attention cho véc tơ ẩn có từ encoder và cũng sử dụng một lớp




```
Độ dài trung bình:  27.05454392176699
MAX:  213
MIN:  1
```

Hình 3.1: Độ dài các câu tiếng Nhật




```
Độ dài trung bình:  21.42368704614664
MAX:  124
MIN:  1
```

Hình 3.2: Độ dài các câu tiếng Việt

```
(encoder): Encoder(
  (tok_embedding): Embedding(30000, 300, padding_idx=3)
  (emb2hid): Linear(in_features=300, out_features=128, bias=True)
  (pos_embedding): Embedding(100, 128)
  (layers): ModuleList(
    (0): EncoderLayer(
      (self_attn_layer_norm): LayerNorm((128,), eps=1e-05, elementwise_affine=True)
      (ff_layer_norm): LayerNorm((128,), eps=1e-05, elementwise_affine=True)
      (self_attention): MultiHeadAttentionLayer(
        (fc_q): Linear(in_features=128, out_features=128, bias=True)
        (fc_k): Linear(in_features=128, out_features=128, bias=True)
        (fc_v): Linear(in_features=128, out_features=128, bias=True)
        (fc_o): Linear(in_features=128, out_features=128, bias=True)
        (dropout): Dropout(p=0.2, inplace=False)
      )
      (positionwise_feedforward): PositionwiseFeedforwardLayer(
        (fc_1): Linear(in_features=128, out_features=512, bias=True)
        (fc_2): Linear(in_features=512, out_features=128, bias=True)
        (dropout): Dropout(p=0.2, inplace=False)
      )
      (dropout): Dropout(p=0.2, inplace=False)
      ..................
  )
  (dropout): Dropout(p=0.2, inplace=False)
)
```

Hình 3.3: Kiến trúc Encoder

LSTM để biểu diễn đặc trưng cho câu đích.

Với các tham số cơ bản *hid_dim = 512, max_vocab=30000, max_sent_len = 50, optimizer = Adam, learning_rate = e10-3, dropout = 0.2, num_epochs = 20* . Kết quả khi chạy một lần chưa tinh chỉnh tham số đạt được là **BLEU-score = 4.90**. So với các mô hình basline thì kết quả còn kém hơnkhá nhiều.

Các câu tiếng Việt khi được dịch từ mô hình này từ tập test có hiện tượng bị lặp chữ, có lẽ do phía decoder sử dụng argmax trên véc tơ nên lúc giải mã ra từ thì mô hình vẫn còn khá ngây thơ. Nhóm sẽ không tìm hiểu kĩ hơn về mô hình này mà sẽ triển khai giải quyết bài toán này trên mô hình Transformer sẽ được trình bày ở mục tiếp theo.

3.2.2 Mô hình Transformer

Mô hình Transformer nhóm triển khai tương tự như trong mô hình của Ashish Vaswan [2] với một chút thay đổi.

- Sử dụng bộ word embedding pretrained (các bộ pretrained được huấn luyện bằng mô hình word2vec, trong đó bộ tiếng Nhật lấy của Kyubyong và bộ tiếng Việt là của Xuan-Son Vu) số chiều các bộ embedding này nhóm lựa chọn là 300.

- Thay vì cộng tổng từ nhúng và vị trí nhúng, nhóm sử dụng toán tử concat cho 2 biểu diễn này, do đó các thành phần véc tơ sẽ tách vai trò rõ rệt hơn và mô hình học các trọng số riêng cho chúng.

- Postion embedding sẽ được train và cập nhật trọng số chứ không bị đóng băng.

Cụ thể kiến trúc mô hình như Hình 3.3 và 3.4 Các tham số cơ bản nhóm sử dụng là *optimizer=Adam,*

```
(decoder): Decoder(
  (tok_embedding): Embedding(30000, 300, padding_idx=3)
  (emb2hid): Linear(in_features=300, out_features=128, bias=True)
  (pos_embedding): Embedding(100, 128)
  (layers): ModuleList(
    (0): DecoderLayer(
      (self_attn_layer_norm): LayerNorm((128,), eps=1e-05, elementwise_affine=True)
      (enc_attn_layer_norm): LayerNorm((128,), eps=1e-05, elementwise_affine=True)
      (ff_layer_norm): LayerNorm((128,), eps=1e-05, elementwise_affine=True)
      (self_attention): MultiHeadAttentionLayer(
        (fc_q): Linear(in_features=128, out_features=128, bias=True)
        (fc_k): Linear(in_features=128, out_features=128, bias=True)
        (fc_v): Linear(in_features=128, out_features=128, bias=True)
        (fc_o): Linear(in_features=128, out_features=128, bias=True)
        (dropout): Dropout(p=0.2, inplace=False)
      )
      (encoder_attention): MultiHeadAttentionLayer(
        (fc_q): Linear(in_features=128, out_features=128, bias=True)
        (fc_k): Linear(in_features=128, out_features=128, bias=True)
        (fc_v): Linear(in_features=128, out_features=128, bias=True)
        (fc_o): Linear(in_features=128, out_features=128, bias=True)
        (dropout): Dropout(p=0.2, inplace=False)
      )
      (positionwise_feedforward): PositionwiseFeedforwardLayer(
        (fc_1): Linear(in_features=128, out_features=512, bias=True)
        (fc_2): Linear(in_features=512, out_features=128, bias=True)
        (dropout): Dropout(p=0.2, inplace=False)
      )
      (dropout): Dropout(p=0.2, inplace=False)
    )
      ..................
    )
  )
  (fc_out): Linear(in_features=128, out_features=30000, bias=True)
  (dropout): Dropout(p=0.2, inplace=False)
)
```

Hình 3.4: Kiến trúc Decoder

learning_ rate=1e-4, dropout=0.2, num_ epochs = 100. Số lượng tham số của mô hình là 23M tổng thời gian huấn luyện là 20 tiếng.

Kết quả

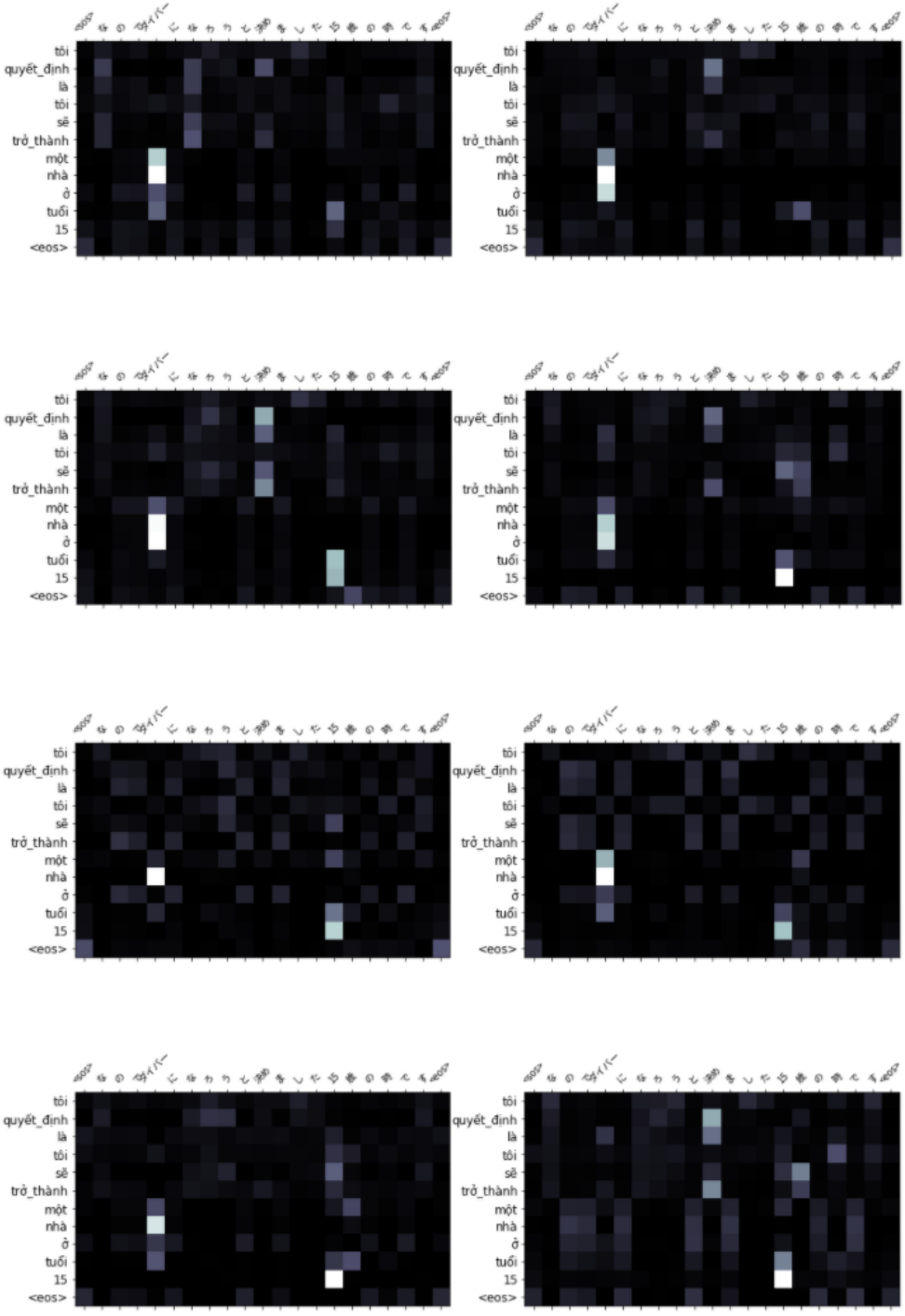
Mô hình Transformer sử dụng word2vec của nhóm đạt kết quả BLEU-score = 9.07, đây là một kết quả cao hơn các mô hình baseline khá, tuy nhiên vẫn kém mô hình SOTA hiện tại.

	BLEU
NMT + JPBPE + VNBPE + Back Translation + Mix-Source	9.64
NMT-baseline	8.18
SMT-baseline	7.73
LSTM-Att	4.90
Transformer+w2v	9.07

Bảng 3.1: Bảng so sánh kết quả

Nhận xét

Nhìn chung thì bài toán này còn khá nhiều thách thức, so với các ngôn ngữ phổ biến như tiếng Anh hay Pháp, điểm BLEU khi dịch từ tiếng Nhật (tính cả SOTA) còn khá khiêm tốn. Để có thể ứng dụng mô hình này trong thực tiễn, nhóm cần cố gắng để cải thiện kết quả lên nhiều. Các việc cần làm là cải thiện dữ liệu và triển khai các kĩ thuật phức tạp khác như back-translation hay sử dụng BPE.



Hình 3.5: Thể hiện Attention

Tổng kết

Kết quả đạt được sau bài tập lớn môn học này là nhóm đã xây dựng và huấn luyện được hai mô hình dịch máy từ tiếng Nhật sang tiếng Việt. Qua đánh giá, mô hình LSTM-Attention của nhóm có kết quả BLEU-score là 7.90, đây có lẽ chưa phải điểm số cao nhất khi sử dụng mô hình này vì trong thời gian thực hiện bài tập lớn nhóm không tập trung trên LSTM-Attention. Với mô hình Transformer, kết quả đạt được là 9.07, cao hơn kết quả sử dụng NMT được report trong [3].

Kết quả cuối cùng nhóm đạt được vẫn đang thấp hơn so với state-of-the-art là 9.64. Nhóm cho rằng bước xử lý dữ liệu chưa thực hiện kĩ, hoặc chưa áp dụng hết các kĩ thuật được chứng minh là hiệu quả trong dịch máy như sử dụng tách âm, back translation và sử dụng mix-source. Việc chưa triển khai được các kĩ thuật này trong mô hình do hạn chế về thời gian có trong bài tập lớn này và nhóm sẽ tiếp tục phát triển để cải thiện kết quả trong tương lai.

Chúng em xin cảm ơn PGS. TS Lê Anh Cường đã tham gia hướng dẫn, giúp đỡ nhóm hoàn thành được bài tập lớn này. Nhóm nhận thấy kết quả vẫn cần phải cải thiện nên các thành viên sẽ tiếp tục nghiên cứu, tìm hiểu nhiều phương pháp để có thể nâng cao được hiệu năng của mô hình hơn và có thể đưa mô hình vào sử dụng trong thực tiễn.

Tài liệu tham khảo

- *AI Academy* - NLP course slides [1]
- *A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N Gomez, L. Kaiser, and I. Polosukhinl* - Attention Is All You Need [2]
- *Thi-Vinh Ngo, Thanh-Le Ha, Phuong-Thai Nguyen, Le-Minh Nguyen* - Combining Advanced Methods in Japanese-Vietnamese Neural Machine Translation [3]
- *Shuoheng Yang, Yuxin Wang, Xiaowen Chu* - A Survey of Deep Learning Techniques for Neural Machine Translation [4]
- *D. Bahdanau, K. Cho, and Y. Bengio* - Neural Machine Translation by Jointly Learning to Align and Translate [5]
- *G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush* - Opennmt: Open-source toolkit for neural machine translation [6]
- *D. Do, M. Utiyama, and E. Sumita* - Machine translation from japanese and french to vietnamese, the difference among language families [7]
- *Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W... Klingner, J* - Google's neural machine translation system: Bridging the gap between human and machine translation [8]
- *Xuan-Son Vu* - Pre-trained Word2Vec models for Vietnamese [9]
- *G. Neubig, Y. Nakata, and S. Mori* - Pointwise prediction for robust, adaptable japanese morphological analysis [10]
- *T.-L. Ha, J. Niehues, and A. Waibel* - "Toward multilingual neural machine translation with universal encoder and decoder [11]