TRƯỜNG ĐẠI HỌC THỦY LỢI

KHOA CÔNG NGHỆ THÔNG TIN

BÀI TẬP LỚN

HỌC PHẦN: XỬ LÝ NGÔN NGỮ TỰ NHIÊN

ĐỀ TÀI: XÂY DỰNG MÔ HÌNH DỊCH MÁY

Giảng viên hướng dẫn:	Nguyễn Thị Kim Ngân
Nguyễn Trung Tuyến	2151260823
Bùi Trung Quốc	2151260827
Nguyễn Đức Tuấn	2151264692
Nguyễn Anh Tuấn	2151264691

Hà Nội, tháng 06 năm 2024

MỤC LỤC

CHƯƠNG 1. LÝ THUYẾT

1.1. Lý thuyết dịch máy

Bài toán dịch máy, hay Machine Translation (MT), là một lĩnh vực trong khoa học máy tính và trí tuệ nhân tạo liên quan đến việc sử dụng phần mềm để dịch văn bản hoặc lời nói từ ngôn ngữ này sang ngôn ngữ khác. MT được bắt đầu nghiên cứu từ những năm 50 bằng những chiếc máy tính khổng lồ nhưng lúc đó mới chỉ xây dựng dựa trên những nguyên tắc cơ bản nên có rất nhiều hạn chế. Những năm 1990 cho đến nay đã có thêm nhiều nghiên cứu/ phát triển đặc biệt là MT (Statistical Machine Translation) hay NMT(Neural Machine Translation)[i]. Những phương pháp này đã đạt được những kết quả lớn và càng ngày nhận được sự tin cậy từ người dùng.

Dịch máy chính là quá trình chuyển đổi ngữ nghĩa của một ngôn ngữ nguồn (source language) sang một ngôn ngữ đích (target language) một cách tự động mà không cần sự can thiệp của con người.

Ví dụ:

Câu đầu vào: I have a cat

Câu đầu ra: Tôi có một con mèo

Trong bối cảnh ứng dụng thực tiễn, dịch máy có thể được sử dụng trong nhiều lĩnh vực khác nhau như dịch tài liệu, dịch các đoạn văn bản trên trang web, hỗ trợ giao tiếp trong các ứng dụng di động, và nhiều hơn nữa. Tuy nhiên, mặc dù đã có những tiến bộ lớn, dịch máy vẫn còn phải đối mặt với nhiều thách thức, đặc biệt là trong việc xử lý các ngôn ngữ phức tạp và ngữ cảnh đặc thù của từng ngôn ngữ.

1.2. Neural Machine Translation

Trong đề tài này, nhóm thực hiện nghiên cứu, tìm hiểu lý thuyết của phương pháp Neral Machine Translation cho bài toán dịch Tiếng Anh sang Tiếng Việt. NMT (Neural Machine Translation) là sự kết hợp của dịch máy (Machine Translation - MT) và mạng nơ-ron nhân tạo (Artificial Neural Network - NN). Với đề tài này của nhóm sẽ phát triển 2 mô hình khác nhau của mạng nơ-ron nhân tạo đó là GRU và LSTM. Đây đều là các biến thể của mạng nơ-ron RNN và được các nhà nghiên cứu tin dùng vì những ưu điểm của nó.

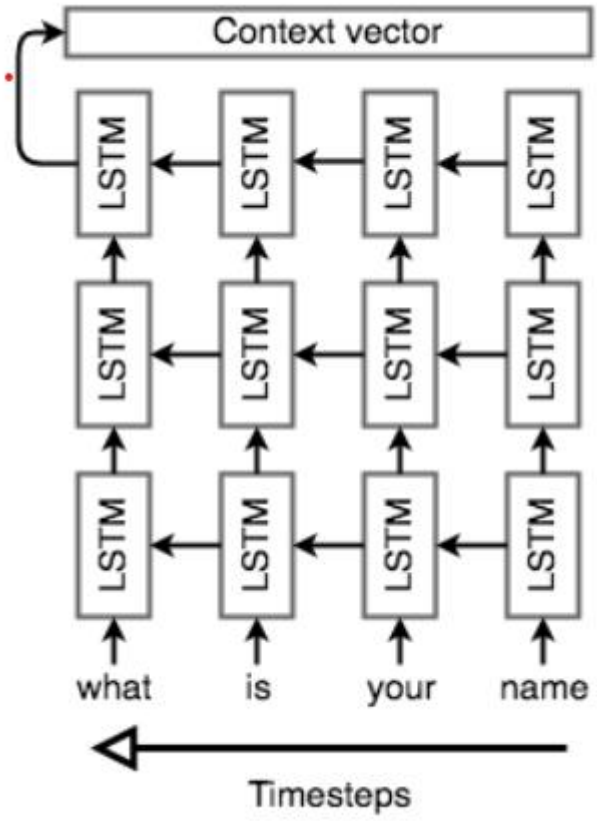
Kiến trúc NMT còn được gọi là sequence-to-sequence (seq2seq), Seq2Seq là một mô hình được đề xuất từ năm 2014 cho việc dịch tiếng Anh-tiếng Pháp. Seq2Seq cũng là một Conditional Language Model. Ở một cấp độ cao hơn, một mô hình Seq2Seq là một mô hình end-to-end với 2 mạng RNN (hoặc các biến thể của RNN):

- Encoder: Bộ mã hóa lấy câu nguồn làm đầu vào của mô hình và encode nó thành một context-vector có kích thước cố định.

- Decoder: Bộ giải mã sử dụng context-vector của encoder làm khởi tạo cho hidden-state đầu tiên.

1.2.1. Kiến trúc Seq2Seq - Encoder

Encoder là một thành phần quan trọng trong mô hình Seq2Seq, chịu trách nhiệm mã hóa thông tin từ câu nguồn (ngôn ngữ nguồn) thành một biểu diễn cố định để sử dụng trong quá trình giải mã. Encoder thường được xây dựng bằng các mạng nơ-ron tái hồi (Recurrent Neural Networks - RNN), mạng nơ-ron LSTM (Long Short-Term Memory), hoặc GRU (Gated Recurrent Unit).



Hình 1.1. Kiến trúc minh họa của Encoder

Cấu trúc và cách hoạt động của Encoder:

Bước 1: Embedding Layer (Lớp nhúng):

Đầu vào của encoder là một chuỗi các từ được biểu diễn dưới dạng các chỉ số từ vựng. Các chỉ số này được chuyển đổi thành các vector nhúng thông qua một lớp nhúng (embedding layer). Vector nhúng chứa thông tin ngữ nghĩa của từ và có chiều không gian thấp hơn nhiều so với số lượng từ vựng ban đầu.

Bước 2: Recurrent Layers (Lớp tái hồi):

Sau khi qua lớp nhúng, các vector nhúng được đưa vào các lớp RNN, LSTM hoặc GRU. Các lớp này xử lý thông tin tuần tự, từng bước một, và lưu giữ trạng thái ngữ cảnh của toàn bộ câu nguồn.

Mỗi bước, một từ từ chuỗi đầu vào sẽ được xử lý và cập nhật trạng thái ẩn của mạng (hidden state).

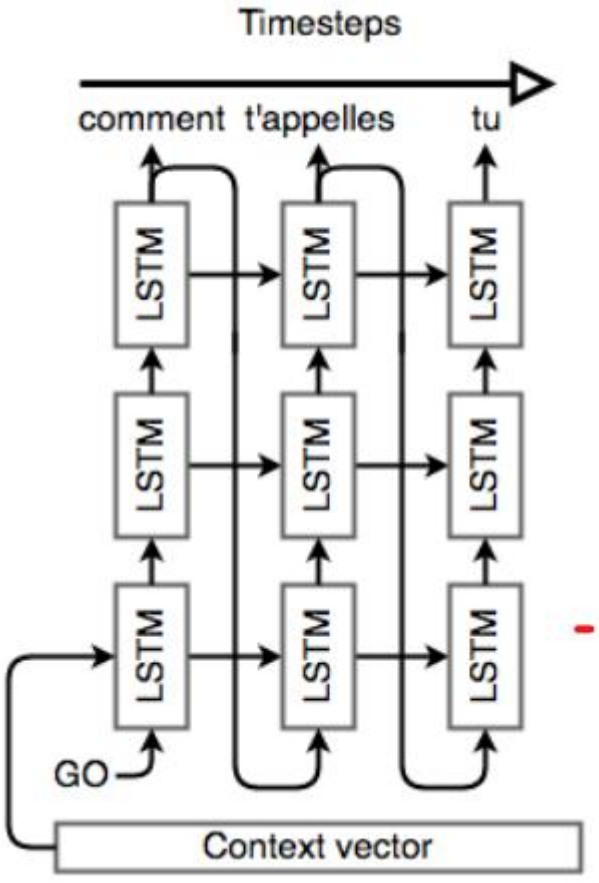
Bước 3: Hidden States (Trạng thái ẩn):

Trạng thái ẩn của mạng tại mỗi bước giữ thông tin ngữ cảnh từ các từ trước đó trong câu nguồn. Trạng thái ẩn cuối cùng của encoder chứa thông tin ngữ nghĩa tổng hợp của toàn bộ câu nguồn và được truyền sang decoder để bắt đầu quá trình giải mã.

Encoder hoạt động như một bộ mã hóa chuỗi, chuyển đổi chuỗi đầu vào thành một biểu diễn ngữ cảnh. Biểu diễn này sau đó được sử dụng để tạo ra các bản dịch tương ứng trong ngôn ngữ đích thông qua decoder.

1.2.2. Kiến trúc Seq2Seq - Decoder

Decoder là thành phần của mô hình Seq2Seq chịu trách nhiệm chuyển đổi biểu diễn ngữ cảnh từ encoder thành chuỗi đầu ra (ngôn ngữ đích). Tương tự như encoder, decoder cũng thường được xây dựng bằng các mạng RNN, LSTM, hoặc GRU.



Hình 1.2. Kiến trúc minh họa của Decoder

Cấu trúc và cách hoạt động của Decoder:

Bước 1: Embedding Layer (Lớp nhúng):

Đầu vào của decoder là các từ đã được dịch trước đó (từ bắt đầu hoặc các từ đã dịch từ các bước trước đó). Các từ này cũng được chuyển đổi thành các vector nhúng thông qua lớp nhúng.

Bước 2: Recurrent Layers (Lớp tái hồi):

Các vector nhúng sau đó được đưa vào các lớp RNN, LSTM hoặc GRU của decoder. Trạng thái ẩn từ encoder (biểu diễn ngữ cảnh) được sử dụng làm trạng thái khởi tạo cho decoder.

Ở mỗi bước, decoder sử dụng từ đã dịch trước đó và trạng thái ẩn hiện tại để dự đoán từ tiếp theo trong chuỗi đích.

Bước 3: Output Layer (Lớp đầu ra):

Lớp đầu ra của decoder là một lớp fully-connected (kết nối đầy đủ) hoặc một lớp softmax, chịu trách nhiệm chuyển đổi trạng thái ẩn hiện tại thành xác suất của các từ trong từ vựng đích. Từ có xác suất cao nhất được chọn làm từ tiếp theo trong bản dịch.

Decoder hoạt động bằng cách sử dụng biểu diễn ngữ cảnh từ encoder và từng từ đã dịch để dự đoán từ tiếp theo trong chuỗi đầu ra. Quá trình này tiếp tục cho đến khi gặp một từ kết thúc (ví dụ: <EOS>) được tạo ra hoặc khi đạt đến độ dài tối đa cho phép của câu dịch.

CHƯƠNG 2: XÂY DỰNG MÔ HÌNH

2.1. Mô tả tập dữ liệu

Tập dữ liệu mà nhóm đang sử dụng được thu thập từ các trang website truyện, báo song ngữ Anh - Việt. Một số website nhóm đã thu thập dữ liệu đó là:

1. https://truyensongngu.net/ : Đây là website chứa các câu truyện ngụ ngôn, đời sống được dịch song ngữ Anh - Việt.

2. https://websongngu.com/ : Đây là website chứa các bài báo nước ngoài/ trong nước đã được dịch song ngữ Anh - Việt.

Dữ liệu nhóm thu thập và tiền xử lý để tách thành các câu đơn bao gồm 2069 câu đã được dịch song ngữ. Trong quá trình xây dựng nhóm đã thực hiện chia tập train và test theo tỉ lệ 80:20, tập train được sử dụng để huấn luyện các mô hình còn tập test sẽ được sử dụng để cho mô hình dịch và đánh giá các mô hình đó.

2.2. Xây dựng mô hình dịch máy

Để giúp đánh giá được các mô hình dịch máy mà nhóm xây dựng thì nhóm đã xây dựng 02 mô hình và sử dụng thêm 02 mô hình đã được pretrain sẵn để dịch cho tập test.

Trong quá trình xây dựng và đánh giá mô hình, nhóm sử dụng 02 độ đo đó là BLEU và ACC:

1. *BLEU (Bilingual Evaluation Understudy):* là một chỉ số đánh giá chất lượng của các hệ thống dịch máy bằng cách so sánh kết quả đầu ra của hệ thống dịch với một hoặc nhiều bản dịch chuẩn (reference translations) do con người thực hiện. BLEU là một trong những độ đo phổ biến nhất và được sử dụng rộng rãi trong lĩnh vực dịch máy.

2. *ACC (Accuracy):* là một độ đo đánh giá mức độ chính xác của các hệ thống phân loại hoặc dịch máy. Trong ngữ cảnh dịch máy, ACC thường được sử dụng để đánh giá xem bản dịch có đúng theo bản dịch chuẩn hay không trên cơ sở từ vựng hoặc cụm từ.

2.2.1. Mô hình dịch sử dụng thư viện VinAI

Vinai-translate-en2vi-v2 là một mô hình dịch máy tiên tiến được phát triển bởi VinAI Research, một tổ chức nghiên cứu hàng đầu tại Việt Nam. Mô hình này được thiết kế để dịch từ tiếng Anh sang tiếng Việt với độ chính xác và chất lượng cao. Đây là phiên bản thứ hai của dòng mô hình dịch máy vinai-translate, được cải tiến để mang lại hiệu suất tốt hơn so với phiên bản trước đó.

Bằng việc sử dụng thư viện này, kết quả dịch trên tập test cũng khả quan và tương đối chính xác. Với kết quả thu được sau khi dịch các câu trong tập test là như sau: ***BLEU = 0.415, ACC = 0.669***

```
5 câu đầu tiên:

Câu tiếng Anh: i want to rest a bit.
Câu tiếng Việt gốc: tôi muốn nghỉ một chút .
Câu tiếng Việt dịch bởi thư viện: Tôi muốn nghỉ ngơi một chút.

Câu tiếng Anh: everyone has their good and bad things.
Câu tiếng Việt gốc: người nào cũng có cái hay và cái dở của họ .
Câu tiếng Việt dịch bởi thư viện: mọi người đều có những điều tốt và xấu.

Câu tiếng Anh: i used to go to church on sunday before.
Câu tiếng Việt gốc: trước đây tôi đã từng đi lễ nhà thờ ngày chủ nhật .
Câu tiếng Việt dịch bởi thư viện: Tôi đã từng đi nhà thờ vào chủ nhật trước.

Câu tiếng Anh: when i got home, my younger brother had finished his homework.
Câu tiếng Việt gốc: khi tôi về đến nhà , em trai tôi đã làm xong bài tập về nhà .
Câu tiếng Việt dịch bởi thư viện: Khi tôi về nhà, em trai tôi đã làm xong bài tập về nhà.

Câu tiếng Anh: i like this.
Câu tiếng Việt gốc: mình thích cái này .
Câu tiếng Việt dịch bởi thư viện: Tôi thích điều này.
```

Hình 2.1. Các câu trong tập test được dịch bởi thư viện VinAI

2.2.2. Mô hình dịch sử dụng mô hình T5

Mô hình T5 là một mô hình mở trên Huggingface, nên nhóm đã quyết định fine-tuning lại mô hình này với dữ liệu của nhóm.

```python
batch_size = 8
model_name = model_checkpoint.split("/")[-1]
args = Seq2SeqTrainingArguments(
    f"{model_name}-finetuned-{source_lang}-to-{target_lang}",
    evaluation_strategy = "epoch",
    learning_rate=2e-5,
    per_device_train_batch_size=batch_size,
    per_device_eval_batch_size=batch_size,
    weight_decay=0.01,
    save_total_limit=3,
    num_train_epochs=5,
    predict_with_generate=True,
#     fp16=True,
#     push_to_hub=True,
)
```

Hình 2.2. Thực hiện fine-tuning mô hình T5

Sau khi fine-tuning nhóm sử dụng mô hình đó dịch các câu trong tập test và thu được kết quả các độ đo như sau: ***BLEU: 0.556, ACC: 0.57***

2.2.3. Mô hình Seq2Seq + GRU

Đầu vào của mô hình:

Các cặp câu tiếng Anh và tiếng Việt trong tập train đã được tiền xử lý như sau:

- Tiền xử lý câu: Loại bỏ khoảng trắng thừa, chuyển đổi tất cả ký tự thành chữ thường.

- Tách từ và thêm token đặc biệt: Tách các câu thành các từ, và thêm các token đặc biệt <bos> (bắt đầu) và <eos> (kết thúc) cho cả câu tiếng Anh và tiếng Việt.

- Chuyển đổi thành chỉ số từ vựng: Sử dụng bộ từ vựng để chuyển đổi các từ trong câu thành các chỉ số (word indices).

- Padding: Thêm padding để các chuỗi có độ dài bằng nhau.

Đầu ra của quá trình huấn luyện:

1. Mô hình đã được huấn luyện (Trained Model):

- Encoder Model: Mô hình này học cách mã hóa các câu tiếng Anh thành các vector trạng thái (state vectors).

- Decoder Model: Mô hình này học cách sử dụng các vector trạng thái từ Encoder để giải mã và tạo ra các câu tiếng Việt tương ứng.

2. Trọng số (Weights) đã được tối ưu hóa:

- Các trọng số của tất cả các lớp trong cả Encoder và Decoder, bao gồm lớp Embedding, lớp GRU, và lớp Dense. Các trọng số này đã được tối ưu hóa thông qua quá trình huấn luyện để giảm hàm mất mát (loss).

3. Thông số huấn luyện (Training Metrics):

- Loss: Giá trị của hàm mất mát (categorical cross-entropy loss) sau mỗi epoch huấn luyện, chỉ ra mức độ tốt của mô hình trong việc dự đoán các từ trong câu đích.

- Accuracy: Độ chính xác của mô hình trong việc dự đoán đúng từ trong câu đích sau mỗi epoch huấn luyện.

Kết quả cụ thể sau quá trình huấn luyện:

- Mô hình hoàn chỉnh: Có khả năng dịch các câu mới từ tiếng Anh sang tiếng Việt.

- Lưu trữ mô hình: Mô hình và các trọng số đã được tối ưu hóa có thể được lưu trữ để sử dụng lại sau này.

Kiến trúc của mô hình Seq2Seq sử dụng GRU:

Mô hình này gồm các thành phần chính như sau:

Thứ nhất: Embedding Layer (Lớp nhúng):

- Có tác dụng chuyển đổi các chỉ số từ vựng thành các vector nhúng (embedding vectors) có kích thước cố định.

- Các tham số:

+ *input_dim:* Kích thước của từ vựng tiếng anh: 1677 + 1 = 1678

+ *hidden_dim:* Kích thước của vector nhúng: 256

Thứ hai: Encoder (Lớp mã hóa):

- Mã hóa câu nguồn (tiếng Anh) thành một biểu diễn ngữ cảnh (context vector).

- Cấu trúc: Gồm 01 lớp GRU với 02 tầng (layers)

- Tham số truyền vào:

+ *hidden_dim:* Kích thước của vector ẩn: 256

+ *n_layers:* Số lượng lớp GRU: 02

Thứ ba: Decoder (Lớp giải mã):

- Giải mã biểu diễn ngữ cảnh từ encoder để tạo ra câu đích (tiếng Việt).

- Cấu trúc: Gồm 01 lớp GRU với 02 tầng (layers)

- Tham số truyền vào:

+ *hidden_dim:* Kích thước của vector ẩn: 256

+ *n_layers:* Số lượng lớp GRU: 02

Thứ tư: Fully-Connected Layer:

- Chuyển đổi vector ẩn của decoder thành xác suất của các từ trong từ vựng đích.

- Tham số truyền vào:

+ *hidden_dim:* Kích thước của vector ẩn: 256

+ *output_dim:* Kích thước của từ vựng tiếng Việt

Ngoài những tham số kể trên, mô hình còn sử dụng phương pháp tối ưu hóa là Adam, hàm mất mát CrossEntropy. Dưới đây là các đoạn code thiết kế, khai báo mô hình:

```
class Seq2Seq(nn.Module):
    def __init__(self, input_dim, output_dim, hidden_dim, n_layers):
        super(Seq2Seq, self).__init__()
        self.embedding = nn.Embedding(input_dim, hidden_dim)
        self.encoder = nn.GRU(hidden_dim, hidden_dim, n_layers, batch_first=True)
        self.decoder = nn.GRU(hidden_dim, hidden_dim, n_layers, batch_first=True)
        self.fc = nn.Linear(hidden_dim, output_dim)

    def forward(self, source, target, teacher_forcing_ratio=0.5):
        embedded_source = self.embedding(source)
        embedded_target = self.embedding(target)
        encoder_outputs, hidden = self.encoder(embedded_source)
        outputs = []
        decoder_input = embedded_target[:, 0].unsqueeze(1)
        for t in range(1, target.size(1)):
            decoder_output, hidden = self.decoder(decoder_input, hidden)
            prediction = self.fc(decoder_output.squeeze(1))
            outputs.append(prediction.unsqueeze(1))
            teacher_force = np.random.random() < teacher_forcing_ratio
            decoder_input = embedded_target[:, t].unsqueeze(1) if teacher_force else
self.embedding(prediction.argmax(1)).unsqueeze(1)
        outputs = torch.cat(outputs, dim=1)
        return outputs
```

Hình 2.3. Khai báo cấu trúc Seq2Seq

```
input_dim = len(en_vocab) + 1
output_dim = len(vn_vocab) + 1
hidden_dim = 256
n_layers = 2
else:
    model = Seq2Seq(input_dim, output_dim, hidden_dim, n_layers).to(device)
    optimizer = optim.Adam(model.parameters())
    criterion = nn.CrossEntropyLoss(ignore_index=vn_vocab['<PAD>'])

    # Huấn luyện mô hình
    n_epochs = 300
    for epoch in range(n_epochs):
        model.train()
        epoch_loss = 0
        for batch_idx, batch in enumerate(train_loader):
            source, target = batch
            source, target = source.to(device), target.to(device)

            optimizer.zero_grad()
            output = model(source, target)

            output = output.view(-1, output.shape[-1])
            target = target[:, 1:].contiguous().view(-1)

            loss = criterion(output, target)
            loss.backward()
            optimizer.step()
            epoch_loss += loss.item()
        print(f'Epoch {epoch+1}, Loss: {epoch_loss/len(train_loader)}')
```

Hình 2.4. Khai báo và huấn luyện mô hình Seq2Seq với GRU

Sau khi huấn luyện mô hình với bộ dữ liệu train, nhóm đã thực hiện dịch trên tập test và tính độ đo BLEU, ACC được kết quả như sau:

BLEU score: 0.735

Accuracy: 0.794

2.2.4. Mô hình Seq2Seq + LSTM

Các cặp câu tiếng Anh và tiếng Việt trong tập train đã được tiền xử lý như sau:

- Tiền xử lý câu: Loại bỏ khoảng trắng thừa, chuyển đổi tất cả ký tự thành chữ thường.

- Tách từ và thêm token đặc biệt: Tách các câu thành các từ, và thêm các token đặc biệt <bos> (bắt đầu) và <eos> (kết thúc) cho cả câu tiếng Anh và tiếng Việt.

- Chuyển đổi thành chỉ số từ vựng: Sử dụng bộ từ vựng để chuyển đổi các từ trong câu thành các chỉ số (word indices).

- Padding: Thêm padding để các chuỗi có độ dài bằng nhau.

Đầu ra của quá trình huấn luyện:

1. Mô hình đã được huấn luyện (Trained Model):

- Encoder Model: Mô hình này học cách mã hóa các câu tiếng Anh thành các vector trạng thái (state vectors).

- Decoder Model: Mô hình này học cách sử dụng các vector trạng thái từ Encoder để giải mã và tạo ra các câu tiếng Việt tương ứng.

2. Trọng số (Weights) đã được tối ưu hóa:

- Các trọng số của tất cả các lớp trong cả Encoder và Decoder, bao gồm lớp Embedding, lớp LSTM, và lớp Dense. Các trọng số này đã được tối ưu hóa thông qua quá trình huấn luyện để giảm hàm mất mát (loss).

3. Thông số huấn luyện (Training Metrics):

- Loss: Giá trị của hàm mất mát (categorical cross-entropy loss) sau mỗi epoch huấn luyện, chỉ ra mức độ tốt của mô hình trong việc dự đoán các từ trong câu đích.

- Accuracy: Độ chính xác của mô hình trong việc dự đoán đúng từ trong câu đích sau mỗi epoch huấn luyện.

Kết quả cụ thể sau quá trình huấn luyện:

- Mô hình hoàn chỉnh: Có khả năng dịch các câu mới từ tiếng Anh sang tiếng Việt.

- Lưu trữ mô hình: Mô hình và các trọng số đã được tối ưu hóa có thể được lưu trữ để sử dụng lại sau này.

Kiến trúc của mô hình Seq2Seq sử dụng LSTM:

Encoder

- Embedding Layer:

+ Chuyển đổi các chỉ số từ vựng thành các vector có kích thước nhỏ hơn và liên tục.

- LSTM Layer:

+ Xử lý các vector nhúng và lưu lại trạng thái cuối cùng (hidden state và cell state).

```
import tensorflow as tf
Input = tf.keras.layers.Input
Embedding = tf.keras.layers.Embedding
LSTM = tf.keras.layers.LSTM
Model = tf.keras.models.Model

# Kích thước từ vựng tiếng Anh và kích thước vector embedding
en_size = len(en_tokenizer.word_index)
latent_dim = 256

# Định nghĩa input và Embedding layer
encoder_inputs = Input(shape=(None,))
enc_emb = Embedding(en_size + 1, latent_dim, mask_zero=True)
(encoder_inputs)

# LSTM layer: xử lý các dense vector từ lớp Embedding
encoder_lstm = LSTM(latent_dim, return_state=True)
encoder_outputs, state_h, state_c = encoder_lstm(enc_emb)

# Lưu lại các trạng thái cuối cùng của LSTM để truyền vào Decoder
encoder_states = [state_h, state_c]

# Xây dựng mô hình encoder
encoder_model = Model(encoder_inputs, encoder_states)

# In ra summary của mô hình encoder
encoder_model.summary()
```

Hình 2.5. Khai báo kiến trúc encoder

Decoder:

- Embedding Layer:

+ Chuyển đổi các chỉ số từ vựng thành các vector có kích thước nhỏ hơn và liên tục.

- LSTM Layer:

+ Sử dụng trạng thái cuối cùng từ bộ mã hóa làm trạng thái ban đầu.

- Dense Layer:

+ Chuyển đổi đầu ra của LSTM thành xác suất của các từ trong từ vựng tiếng Việt.

```
Dense = tf.keras.layers.Dense

# Thiết lập decoder, sử dụng `encoder_states` làm trạng thái khởi
đầu.
decoder_inputs = Input(shape=(None,))
dec_emb_layer = Embedding(vi_size + 1, latent_dim, mask_zero=True)
dec_emb = dec_emb_layer(decoder_inputs)

# Thiết lập decoder để trả về toàn bộ chuỗi đầu ra và các trạng thái
nội bộ.
decoder_lstm = LSTM(latent_dim, return_sequences=True,
return_state=True)
decoder_outputs, _, _ = decoder_lstm(dec_emb,
initial_state=encoder_states)

# Lớp Dense với softmax để tạo xác suất cho các từ trong từ vựng.
decoder_dense = Dense(vi_size + 1, activation='softmax')
decoder_outputs = decoder_dense(decoder_outputs)
```

Hình 2.6. Khai báo kiến trúc decoder

Các Tham Số Mô Hình:

Trong quá trình xây dựng và huấn luyện mô hình Seq2Seq với LSTM, các tham số quan trọng sau đây được sử dụng:

- Batch Size: Đây là số lượng mẫu được sử dụng để cập nhật mô hình trong mỗi bước huấn luyện: 16

- Epochs: Đây là số lần toàn bộ tập dữ liệu huấn luyện được sử dụng để cập nhật các tham số của mô hình: 200.

- Latent Dimension (latent_dim): Đây là số chiều của vector ẩn trong các tầng LSTM và các vector embedding: 128

- Embedding Dimension: Đây là số chiều của vector embedding, tương ứng với latent_dim. Các vector embedding giúp chuyển đổi các từ thành các vector dense có ý nghĩa hơn đối với mô hình: 128

- Loss Function: Hàm mất mát được sử dụng để đánh giá sự khác biệt giữa phân phối xác suất dự đoán của mô hình và phân phối thực tế. categorical_crossentropy là hàm mất mát thường dùng cho các bài toán phân loại nhiều lớp.

- Optimizer: Thuật toán tối ưu hóa được sử dụng để cập nhật các tham số của mô hình. RMSprop là một thuật toán tối ưu hóa phù hợp cho các bài toán xử lý ngôn ngữ tự nhiên. Learning rate điều chỉnh tốc độ cập nhật các tham số: RMSprop với learning rate 0.001

```
batch_size = 16
epochs = 200
latent_dim = 128
RMSprop = tf.keras.optimizers.RMSprop
from tensorflow.keras.utils import to_categorical

# Xây dựng mô hình với các đầu vào và đầu ra của encoder và decoder
model = Model([encoder_inputs, decoder_inputs], decoder_outputs)

# Compile mô hình với các tham số đã định nghĩa
model.compile(optimizer=RMSprop(learning_rate=0.001),
              loss='categorical_crossentropy',
              metrics=['acc'])

# Huấn luyện mô hình với dữ liệu huấn luyện và các tham số đã định
nghĩa
model.fit(generate_batch(trainX, trainY, batch_size=batch_size),
          steps_per_epoch=train_samples // batch_size,
          epochs=epochs,
          validation_data=generate_batch(testX, testY,
batch_size=batch_size),
          validation_steps=val_samples // batch_size)
```

Hình 2.7. Huấn luyện mô hình Seq2Seq với LSTM

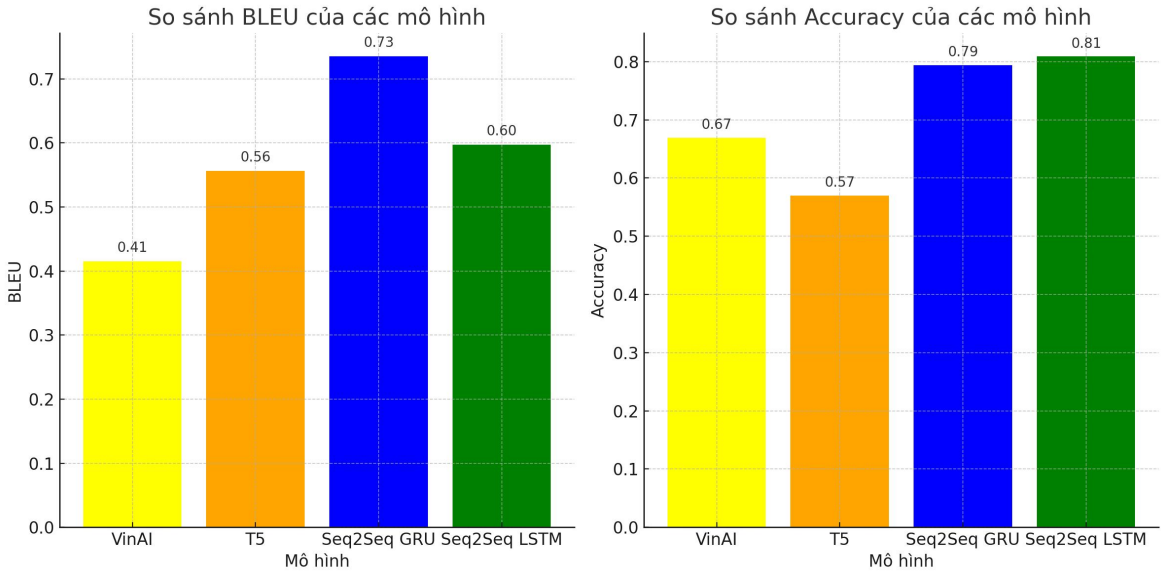
Sau khi huấn luyện mô hình với bộ dữ liệu train, nhóm đã thực hiện dịch trên tập test và tính độ đo BLEU, ACC được kết quả như sau:

BLEU score: 0.597

Accuracy: 0.809

2.3. Kết luận

Sau khi triển khai xây dựng và chạy thực nghiệm được 4 mô hình khác nhau đó là: thư viện VinAI, mô hình T5, mô hình Seq2Seq với GRU, mô hình Seq2Seq với LSTM nhóm đã thu được các kết quả của các độ đo đánh giá như sau:



Hình 2.8. Biểu đồ so sánh độ đo của các mô hình

Quan sát hình 2.8 thấy rằng:

- Mô hình Seq2Seq GRU cho ra kết quả khá là tốt ở cả 2 độ đo BLEU và ACC.

- Khi dịch bằng thư viện VinAI cho ra kết quả thấp nhất bởi vì nó chưa được huấn luyện, học các đặc trưng của dữ liệu mà nhóm sử dụng.

- Mô hình T5 cũng được học bằng dữ liệu của nhóm, nhưng có thể do nó đã được học các đặc trưng từ nhiều nguồn khác nhau nên khi dịch về ngữ nghĩa thì đúng nhưng không đúng về các từ. Nên có thể đây là một lí do mà mô hình cho kết quả không tốt lắm.

- Mô hình Seq2Seq LSTM có kết quả độ đo ACC cao nhất nhưng độ đo BLEU lại thấp hơn mô hình Seq2Seq GRU.

Từ những nhận xét trên, mô hình Seq2Seq GRU và LSTM đều phù hợp với dữ liệu mà nhóm đang sử dụng.

TÀI LIỆU THAM KHẢO

[i] Viblo Nguyen Dinh Thien, "*Tổng quan về Neral Machine Translation*", https://viblo.asia/p/tong-quan-ve-neural-machine-translation-E375zrMd5GW , truy cập ngày 10/06/2024.