Xây dựng mô hình dịch máy English-Vietnamese bằng phương pháp Transformer

Văn Kim Ngân 19520177

Nguyễn Thị Bảo Hân 19520071

Tiêu Kim Hảo 19521480

Abstract

Trong bài báo cáo này, chúng tôi xây dựng một mô hình dịch máy chuyển từ ngôn ngữ tiếng Anh sang tiếng Việt trên bộ dữ liệu 'IWSLT'15 English-Vietnamese Datasets' bằng phương pháp Transformers với độ đo được sử dụng là BLEU. Mô hình nhóm xây dựng cho kết quả với BLEU score đạt 33.08.

1 Introduction

Nhu cầu dịch thuật từ ngôn ngữ này sang ngôn ngữ khác ngày càng tăng do sự bùng nổ của Internet và sự trao đổi thông tin giữa nhiều khu vực sử dụng các ngôn ngữ khu vực khác nhau. Dịch máy từ lâu đã trở thành một vấn đề lớn trong lĩnh vực Xử lý ngôn ngữ tự nhiên (NLP). Neural Machine Translation (NMT) gần đây đã được đưa vào nghiên cứu và đã tạo ra những cải tiến rất lớn cho hệ thống dịch máy. Hầu hết các hệ thống NMT đều dựa trên kiến trúc bộ encoder-decoder bao gồm hai mạng nơ-ron. Encoder nén các chuỗi nguồn thành một vectơ, được decoder sử dụng để tạo chuỗi đích. Trước khi Transformer ra đời, hầu như các tác vụ xử lý ngôn ngữ tự nhiên, đặc biệt trong mảng Machine Translation (dịch máy) đều sử dụng kiến trúc Recurrent Neural Networks (RNNs). Do phải xử lý câu đầu vào một cách tuần tự nên nhược điểm của RNNs là tốc độ xử lý chậm và hạn chế trong việc biểu diễn sự phụ thuộc xa giữa các từ trong một câu.

Khi Transformer xuất hiện, các vấn đề mà RNNs gặp phải gần như đã được giải quyết triệt để. Transformer là một kiến trúc mạng dựa trên cơ chế self-attention. Self-Attention là cơ chế giúp encoder nhìn vào các từ khác trong lúc mã hóa một từ cụ thể, vì vậy, Transformers có thể hiểu được sự liên quan giữa các từ trong một câu, kể cả khi chúng có khoảng cách xa. Các decoder cũng có kiến trúc giống như vậy nhưng giữa chúng có một lớp attention để nó có thể tập trung vào các phần liên quan của đầu vào. Transformer làm tốt trong việc dịch máy và nhiều tác vụ NLP bởi vì chúng hoàn toàn tránh đệ quy, bằng cách xử lý toàn bộ các câu và học các mối quan hệ giữa các từ nhờ cơ chế multi-head attention và positional embeddings.

Ở bài báo cáo này, chúng tôi đề xuất một mô hình dịch máy lên bộ dữ liệu 'IWSLT'15 English-Vietnamese Datasets' được trích xuất từ trang Hugging face và cung cấp bởi Stanford NLP group. Mô hình dịch chính được chúng tôi sử dụng trong này là Transformer.

Phần còn lại của báo cáo được tổ chức như sau. Phần 2 sẽ trình bày nguồn gốc bộ dữ liệu cũng như các phương pháp tiền xử lý, đồng thời trình bày thuật toán Transformer. Phần 3 sẽ nói về độ đo đánh giá và mô tả Transformer setup. Phần 4 trình bày kết quả thực nghiệm và quan sát kết quả. Và cuối cùng phần 5 sẽ trình bày những kết luận thu được qua bài nghiên cứu này.

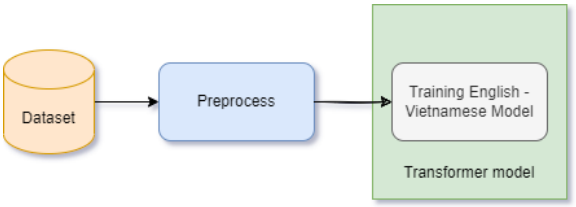
2 Methodology

2.1 Our proposed system architecture

Hình 1 dưới đây minh họa kiến trúc hệ thống được đề xuất của chúng tôi. Trong bài báo này, chúng tôi đã sử dụng Transformer làm mô hình dịch chính. Bộ dữ liệu phải được làm sạch và xử lý trước trước khi cấp cho mô hình Transformer, điều này sẽ được trình bày trong tiểu mục 2.2.

Để xây dựng mô hình dịch máy cuối cùng, có hai giai đoạn chính cần phải được thực hiện:

- **Giai đoạn 1:** Pre-process Data.
- **Giai đoạn 2:** Training English-Vietnamese translation model bằng Transformer.



Hình 1: The proposed system architecture.

2.2 Text Pre-processing

2.2.1 IWSLT'15 English-Vietnamese Datasets

Chúng tôi sử dụng bộ dữ liệu IWSLT'15 English-Vietnamese Datasets được cung cấp bởi Stanford NLP group. Đây là bộ dữ liệu song ngữ Anh-Việt, chúng tôi sẽ sử dụng dữ liệu này để phát triển mô hình của mình. Bộ dữ liệu được chia làm ba tập, với lần lượt kích thước là train: 133K, validation: 1.2K, test: 1.2K. Ngoài việc tokenize bằng cách sử dụng SentencePiece thì còn sử dụng MarianMT[1] pre-trained model . Kích thước từ vựng là 53K cho tiếng Anh và tiếng Việt.

2.2.2 Data cleaning and Pre-processing

Bộ dữ liệu song ngữ được dán nhãn thủ công nên rất ít xảy ra các vấn đề về chất lượng dịch thuật thấp. Do đó, chúng ta chỉ cần loại bỏ các unicode trong tập dữ liệu này và các dòng có ô trống (không có text).

Dataset	Size (sentences)
train	133318
validation	1268
test	1268

Bảng 1: Kích thước bộ dữ liệu.

Sau khi làm sạch và pre-processing dữ liệu thì dữ liệu được làm sạch, chuẩn hóa, sau đó được mã hóa bằng cách sử dụng SentencePiece.[2]

Tất cả các thuật toán tokenization được mô tả cho đến nay đều có cùng một vấn đề: Giả định rằng văn bản đầu vào sử dụng dấu cách để phân tách các từ. Tuy nhiên, không phải tất cả các ngôn ngữ đều sử dụng dấu cách để phân tách các từ. Một giải pháp khả thi là sử dụng language specific pre-tokenizers.

2.3 The Transformer Model

Thành phần Transformer bao gồm encoder và decoder. Đầu vào của encoder đầu tiên sẽ đi qua một lớp seft-attention, đầu ra được truyền vào một mạng nơ ron truyền thẳng (feed-forward). Decoder cũng có hai thành phần đó nhưng nằm giữa chúng là một lớp attention giúp decoder tập trung vào phần quan trọng của câu đầu vào.

Ý tưởng cốt lõi đằng sau mô hình Transformer đó chính là self-attention, một lớp giúp cho encoder nhìn vào các từ khác khi đang mã hóa một từ cụ thể. Transformer tạo các ngăn xếp là các lớp self-attention để xây dựng cho cả encoder và decoder thay vì các lớp RNNs hay CNNs. Kiến trúc chung này giúp cho Transformer model tính toán một cách song song, thay vì một chuỗi như RNNs. Kiến trúc của transformer được thể hiện ở Hình 2.

Transformer có cơ chế mang tên "multi-headed" attention. Cơ chế này cải thiện hiệu năng của lớp attention theo hai khía cạnh. Nó mở rộng khả năng của mô hình trong việc tập trung vào các vị trí khác nhau. Nó mang lại cho lớp attention nhiều không gian con để biểu diễn.

Ở Transformer không sử dụng recurrence hay convolution, mà ở đây chúng ta xử lý thứ tự của các từ trong chuỗi đầu vào bằng cách sử dụng position encoding function.

Do đó, đầu vào của lớp dưới cùng cho mỗi mạng có thể được biểu thị bằng *Input* = *Embedding* + *PositionalEncoding*. Transformer thêm một véc tơ vào mỗi embedding đầu vào. Các véc tơ này tuân theo một mẫu cố định mà mô hình học được, giúp nó xác định vị trí của từng từ hoặc khoảng cách của các từ khác nhau trong chuỗi. Ý tưởng ở đây là việc thêm các giá trị đó sẽ cung cấp thông tin về khoảng cách giữa các véc tơ embedding khi chúng được phản ánh thông qua các véc tơ Q/K/V và thông qua phép lấy tích vô hướng.

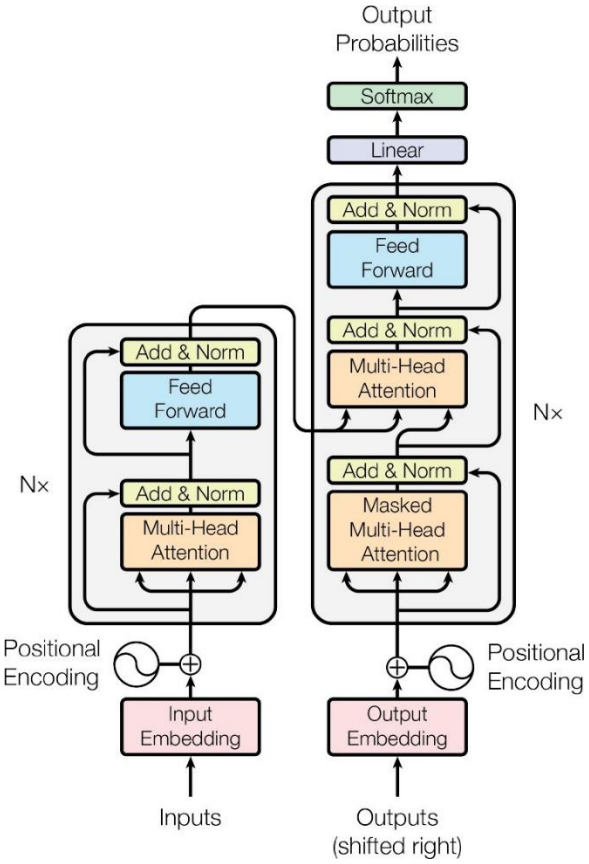
Encoder gồm có nhiều lớp chồng lên nhau. Ở mỗi lớp gồm có cơ chế multi-head self-attention và fully connected feed-forward network. Cơ chế multi-head self-attention giúp cho mô hình có thể "chú ý" đến các phần nội dung nhất định của đầu vào.

Decoder cũng là một ngăn xếp với lớp giống nhau, ở mỗi lớp bao gồm ba lớp con. Ở vị trí cuối cùng chính là masked multi-head self-attention, lớp self-attention chỉ cho phép chú ý lên các vị trí

[1]https://huggingface.co/docs/transformers/model_doc/marian

[2]https://huggingface.co/docs/transformers/tokenizer_summary#sentencepiece

phía trước của chuỗi đầu ra. Điều này được thực hiện bằng cách che đi các vị trí phía sau thông qua masking (đặt chúng về -inf) trước bước softmax khi tính self-attention. Ở giữa chính là lớp "Encoder-Decoder Attention", lớp này hoạt động như multi-head attention, ngoại trừ việc nó tạo các ma trận Q từ lớp phía dưới và lấy các ma trận K và V từ đầu ra của ngăn xếp encoder. Và lớp trên cùng của ngăn xếp decoder là các lớp con feed-forward. Đầu ra của decoder là véc tơ của các số thực, đi qua Linear transform và lớp Softmax sẽ chuyển số thành các từ.



Hình 2: Transformer architecture.

3 Experiment

3.1 Transformer setup

Chúng tôi sử dụng Transformer model, đã được train bởi Jörg Tiedemann, sử dụng the Marian C++ library[3], giúp cho việc training và translation trở nên nhanh hơn. Tất cả các thử nghiệm đều dựa trên Big Transformer architecture với 6 block encoder và decoder. Chúng tôi đã sử dụng cùng một siêu tham số cho tất cả các thử nghiệm, kích thước của word representations là 1024, lớp feed-forward có kích thước bên trong là 4096.

Chúng tôi sử dụng 16 attention heads và tính trung bình checkpoint trên 3 epoch, với batch_size bằng 4. Mô hình đã được tối ưu hóa bằng thuật toán Adam với $\alpha = 1e\text{-}4$, $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\varepsilon = 1e\text{-}8$.

3.2 Automatic Evaluation

BLEU score là thang đo đánh giá một câu được tạo so với 1 câu tham chiếu khác. BLEU thường được sử dụng trong bài toàn dịch máy (Machine Translation) như một độ đo hay một hệ số điểm khi so sánh một bản dịch (candidate translation) với một hay nhiều bản dịch tham khảo (reference translation).

BLEU có giá trị trong khoảng (0.0, 1.0). 1.0 tức là perfect match, đúng hoàn toàn, còn 0.0 tức là perfect mismatch. BLEU có thể tính bằng phương pháp đếm số matching n-grams của candidate và reference, kết quả sẽ là số match chia cho số từ của candidate. Các match này không phụ thuộc vào vị trí, do vậy BLEU không sử dụng word order.

Do đó, khi đếm matching n-grams cần chú ý cả số lần xuất hiện của từ trong reference, một từ trong reference khi được match rồi thì không nên match nữa. Công thức của BLEU là:

$$\frac{\sum_{C\epsilon\{Candidates\}} \sum_{n-gram\epsilon C} Count\ clip(n-gram)}{\sum_{C'\epsilon\{Candidates\}} \sum_{n-gram'\epsilon C'} Count(n-gram')}$$

4 Experimental Results

4.1 IWSLT'15 Experimental Results

Sau khi xây dựng hoàn thành mô hình Transformer, kết quả chúng tôi đạt được sau khi thử nghiệm trên bộ dữ liệu IWSLT'15 English-Vietnamese test set là 33.08 BLEU score.

Model	BLEU score
Transformer	33.08

Bảng 2: Kết quả thực nghiệm của Transformer trên IWSLT'15 English-Vietnamese test set.

4.2 Observations

Chúng ta sẽ tiến hành quan sát kết quả đầu ra của mô hình Transformer để đánh giá kết quả dịch thuật của mô hình một cách khách quan.

[3]https://huggingface.co/docs/transformers/model_doc/marian

Input: Instead, Conor came home one Friday evening and he told me that he had quit his job that day, his dream job.
Target: Thay vì vậy, Conor **đã** về nhà vào một tối thứ sáu và anh ấy nói với tôi rằng anh ấy nghỉ việc ngày hôm nay công việc mơ ước của anh ta.
Prediction: Thay vào đó , Conor trở về nhà vào một buổi tối thứ sáu và nói với tôi rằng anh ấy đã bỏ việc hôm đó , công việc mơ ước của anh ấy.

Bảng 3: Quan sát kết quả Prediction của mô hình so với Input và Target của bộ dữ liệu.

Ta có thể thấy, kết quả prediction của mô hình có kết quả tạm ổn vì khi đọc ta có thể hiểu được nội dung chính của input. Tuy nhiên nhìn vào kết quả, ta thấy được mô hình vẫn chưa tốt. Vì so với target, prediction vẫn chưa đúng ngữ cảnh, chưa chia thì chính xác, câu văn còn lủng củng. Ở ví dụ này, câu input sử dụng thì Simple Past, tuy nhiên kết quả xuất ra lại không có từ chỉ quá khứ "**đã**" như target (Bảng 3).

5 Conclusion

Trong bài báo này, nhóm chúng tôi đã sử dụng mô hình Transformer cho bài toán dịch máy English-Vietnamese. Ở đây chúng tôi sử dụng bộ dữ liệu liệu IWSLT'15 English-Vietnamese Datasets được cung cấp bởi Stanford NLP group, ở bước pre-processing, nhóm loại bỏ các unicode trong tập dữ liệu này và các dòng có ô trống, sau đó sử dụng MarianMT pre-trained model và tokenize bằng SentencePiece.

Mô hình được sử dụng ở đây là Transformer, trước khi Google công bố bài báo về Transformer (Attention Is All You Need), hầu hết các tác vụ xử lý ngôn ngữ tự nhiên, đặc biệt là dịch máy (Machine Translation) sử dụng kiến trúc Recurrent Neural Networks (RNNs). Điểm yếu của phương pháp này là rất khó bắt được sự phụ thuộc xa giữa các từ trong câu và tốc độ huấn luyện chậm do phải xử lý input tuần tự. Transformers sinh ra để giải quyết 2 vấn đề này. Các biến thể của nó như BERT, GPT-2 tạo ra state-of-the-art mới cho các tác vụ liên quan đến NLP.

Kết quả thực nghiệm trên IWSLT'15 English-Vietnamese test cho kết quả là 33.08 BLEU score. Vì mô hình không huấn luyện nhiều nên cho kết quả vẫn chưa tốt nhưng đây là kết quả chấp nhận được. Theo quan sát kết quả đầu ra, mô hình dịch cho kết quả khá tốt tuy nhiên vẫn cần khắc phục và tối ưu hóa hơn để đạt kết quả như mô hình dịch máy của Google.

References

MarianMT.

Jay Alammar. 2018. The Illustrated Transformer.

Jason Brownlee. 2019. A Gentle Introduction to Calculating the BLEU Score for Text in Python.

Trung Nghia Nguyen. 2019. BLEU.

Minh-Thang Luong, Christopher D Manning, et al. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the international workshop on spoken language translation*, IWSLT. Da Nang, Vietnam.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Trang Nguyen Thi Thu et al. 2020. Vietnamese-english translation with transformer and back translation in vlsp 2020 machine translation shared task. In *Proceedings of the 7th International Workshop on Vietnamese Language and Speech Processing*, pages 64– 70.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017a. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. 2019. Learning deep transformer models for machine translation. *arXiv preprint arXiv:1906.01787*.