Task 2

a) *Tại sao nói cơ chế attention là tất cả những gì chúng ta cần. Hãy giải thích về vấn đề này.*
b) *Tại sao kiến trúc Transformer lại là kiến trúc tốt nhất hiện nay trong các mô hình học máy, hãy giải thích.*

A. *Tại sao nói cơ chế attention là tất cả những gì chúng ta cần. Hãy giải thích về vấn đề này.*

Answer:

Attention mechanism mang lại nhiều ưu điểm so với cách trích xuất thông tin theo thống kê cổ điển và học máy. Cơ chế Attention ra mắt lần đầu vào 2014 do **Bahdanau et al.**, trong paper "*Neural Machine Translation by Jointly Learning to Align and Translate*" cho phép model tập trung vào 1 phần của input sequence trong lúc sinh văn bản tương ứng.

Trước đó, các phương pháp học máy và học sâu cũ cho các tác vụ sinh văn bản (seq2seq) thường bị phụ thuộc vào độ dài câu input, các vector biểu diễn hay vector ngữ cảnh được sinh ra nhờ tính toán last hidden state của một bộ mã hóa (encoder), hoặc tính tổng có trọng số tất cả của hidden state. Kĩ thuật này **có hiệu quả đối với đầu vào ngắn và vừa, tuy nhiên lại mất đi khả năng thu tóm ngữ nghĩa, thông tin nằm ở phần văn bản cách trước đó quá xa.** Đây là hệ quả phụ quả việc xử lí tuần tự, dẫn đến thông tin bị mất đi theo thời gian và không hoàn toàn nắm bắt được chính xác ngữ cảnh của từng từ (nhập nhằng về nghĩa của các từ đồng âm), dù nằm trong các ngữ cảnh khác nhau thì các model RNN, LSTM cũ khó phân biệt được từ ngữ mang ý nghĩa nào.

Nhược điểm này được khắc phục bởi Attention mechanism nhờ tính toán trọng số các token của đầu vào, cho phép sinh ra context vector có sự tập trung vào chỉ 1 phần hoặc nhiều phần của văn bản. Kĩ thuật này cho phép model có thêm thông tin ngữ cảnh đa chiều tại mỗi bước sinh output, giúp đạt được kết quả phù hợp hơn trong quá trình decode. Qua đó đạt được hiệu suất cao hơn trên các câu văn dài và chính xác hơn trong dịch thuật.

Bên cạnh đó, cơ chế self-attention, hay intra-attention được giới thiệu vào 2017, giúp tính toán biểu diễn sự tập trung thông tin của 1 token đối với các phần còn lại của 1 chuỗi đầu vào. Điểm mới của self-attention là không chỉ áp dụng attention trên output seq đối với input sequence, mà còn là attention của từng từ trong chính câu input, self-attention được áp dụng trên cả quá trình mã hóa và giải mã, giúp đạt thêm một mức độ nâng cao mới về biểu diễn ngữ nghĩa. Quá trình này được hiểu là

đánh giá thông tin 1 token mang ý nghĩa trả lời câu hỏi gì, cho phần nào của chính câu văn chứa nó.

"Attention is All You Need" là tiêu đề bài báo ra mắt self-attention, nhấn mạnh sự linh hoạt của attention có thể sử dụng, và attention là đủ để đạt được kết quả tốt, thậm chí đã vượt qua toàn bộ các kĩ thuật khác trên tác vụ mô hình hóa thông tin chuỗi. **Ngoài ra trong bài báo cũng chỉ ra computation cost của Attention thấp hơn so với các kĩ thuật truyền thống như Recurrent hay Convolution**, mà vẫn đạt kết quả cao hơn đáng kể, điều này chính là điểm mấu chốt của tiêu đề, khi mà "*chỉ cần có attention là đủ*"

Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

Bảng thống kê độ phức tạp thuật toán trích từ bài báo "Attention is All You Need"

B. Tại sao kiến trúc Transformer lại là kiến trúc tốt nhất hiện nay trong các mô hình học máy, hãy giải thích.

Answer:

Có vài lí do mà transformer thể hiện tốt hơn so với các model học sâu học máy truyền thống:

- **Khả năng xử lí song song** -> nhanh và không phụ thuộc độ dài input: Các mạng RNN cũ thường xử lí chuỗi 1 cách tuần tự, việc này diễn ra khá chậm và hạn chế. Trong khi đó transformer chỉ sử dụng self-attention, như đã nói ở câu 1, đạt được khả năng xử lí song song tất cả các token trong câu.
- **Không bị phụ thuộc độ dài đầu vào, nắm bắt ngữ cảnh xa**: RNN bị mất thông tin khi văn bản đầu vào có độ dài lớn. Mặt khác, kiến trúc transformer cho phép 1 token tập trung ý nghĩa vào 1 phần bất kì khác của văn bản, bất kể độ dài hay vị trí, khắc phụ được nhược điểm long-term dependency của RNN
- **Khả năng phát triển kích thước kiến trúc**: Transform có chi phí tính toán rẻ hơn, nhanh hơn, vì có thể xử lí song song nên có thể nhận đầu vào lớn mà không đẩy chi phí hoạt động vượt qua ngân sách. Bên cạnh đó, thực nghiệm đã chứng minh các kiến trúc transformer càng lớn, có độ sâu (số lượng stack

encoder/decoder) càng lớn thì model càng hiểu ngôn ngữ tốt hơn. RNN không có khả năng này vì tính chất xử lí tuần tự, việc nhận văn bản đầu vào quá dài có thể khiến chi phi tính toán tăng theo cấp số nhân mà lại không đạt được kết quả kì vọng.

- **Khả năng học chuyển tiếp và chuyển đổi tác vụ** (Transfer Learning và Fine-tuning): các Transformer có chiến thuật training khác đi nhiều so với RNN truyền thống, chúng thường được train qua 2 giai đoạn, pretraining trên tập dữ liệu lớn để đạt NLU (natural language understanding) và fine-tuning cho 1 tác vụ cụ thể. Đây là điểm vượt trội so với RNN, LSTM khi mà 1 cấu hình transformer có thể được fine-tune cho nhiều nhiệm vụ khác nhau (vd: BERT, GPT, BART, PEGASUS, T5, …) như text classification, sentiment analysis, and question answering. Điều này chỉ có thể đạt được vì transformer có khả năng khát quát và học được ngữ cảnh của ngôn ngữ xuất sắc giúp chúng hoạt động hiệu quả mà trên mọi loại data.
- **Hoạt động đa lĩnh vực**: Ngoài ra, dù transformer được phát triển ban đầu dựa trên NLP tasks, đã có những áp dụng thành công của của kiến trúc này trên cả Computer Vision, Speech Processing, Multimodal Transformer với đa dạng khả năng như text-to-speech, text-to-image, image-to-text, … Vì transformer có kiến trúc đa dạng, encoder-only, decoder-only, encoder-decoder và self-attention giúp mở rộng khả năng “tập trung” thông tin không chỉ trên văn bản mà còn là dữ liệu ảnh, voice.