TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CUỐI KỲ HK1 NĂM HỌC 2022 - 2023**

**BÁO CÁO CUỐI KỲ MÔN**

**NHẬP MÔN XỬ LÝ NGÔN NGỮ TỰ NHIÊN**

*Người hướng dẫn: Giảng viên :* **LÊ ANH CƯỜNG**

*Người thực hiện:* **ĐÀO NHỰT QUI – 52000510**

**NGUYỄN MINH CHÍ – 52000189**

**NGUYỄN MINH HOÀNG CHƯƠNG - 52000744**

*Lớp :* **20050301**

*Khóa* : **24**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CUỐI KỲ HK1 NĂM HỌC 2022 - 2023**

**BÁO CÁO CUỐI KỲ MÔN**

**NHẬP MÔN XỬ LÝ NGÔN NGỮ TỰ NHIÊN**

*Người hướng dẫn: Giảng viên :* **LÊ ANH CƯỜNG**

*Người thực hiện:* **ĐÀO NHỰT QUI – 52000510**

**NGUYỄN MINH CHÍ – 52000189**

**NGUYỄN MINH HOÀNG CHƯƠNG - 5200744**

*Lớp :* **20050301**

*Khóa* : **24**

**THÀNH PHỐ HỒ** **CHÍ MINH, NĂM 2023**

**LỜI CẢM ƠN**

Trước tiên, chúng tôi xin gửi lời cảm ơn chân thành và lòng biết ơn sâu sắc đến Giảng viên Lê Anh Cường. Thầy là người đã luôn hỗ trợ và hướng dẫn tận tình cho chúng tôi trong suốt quá trình giảng dạy và là nền tảng để tôi hoàn thành bài tập với đề tài cuối kỳ môn “Nhập môn xử lý ngôn ngữ tự nhiên".

Tiếp theo, chúng tôi xin gửi lời cảm ơn đến khoa Công Nghệ Thông Tin trường Đại học Tôn Đức Thắng vì đã tạo điều kiện cho tôi được học tập và nghiên cứu môn học này. Khoa đã luôn sẵn sàng chia sẻ các kiến thức bổ ích cũng như chia sẻ các kinh nghiệm tham khảo tài liệu, giúp ích không chỉ cho việc thực hiện và hoàn thành đề tài nghiên cứu mà còn giúp ích cho việc học tập và rèn luyện trong quá trình thực hành tại trường Đại học Tôn Đức Thắng nói chung.

Cuối cùng, sau khoảng thời gian học tập trên lớp chúng tôi đã hoàn tất bài báo cáo nhờ vào sự hướng dẫn, giúp đỡ và những kiến thức học hỏi được từ Quý thầy cô. Do giới hạn về mặt kiến thức và khả năng lý luận nên tôi vẫn còn nhiều thiếu sót và hạn chế, kính mong sự chỉ dẫn và đóng góp của Quý thầy cô giáo để bài báo cáo của tôi được hoàn thiện hơn. Hơn nữa, nhờ những góp ý từ thầy cô và các bạn hữu, tôi sẽ hoàn thành tốt hơn ở những bài nghiên cứu trong tương lai. Tôi mong Quý thầy cô và các bạn bè – những người luôn quan tâm và hỗ trợ chúng tôi – luôn tràn đầy sức khỏe và sự bình an.

TP. Hồ Chí Minh, ngày tháng năm 2023

Tác giả(Ký tên và ghi rõ họ tên)

**ĐỒ ÁN / BÁO CÁO ĐƯỢC HOÀN THÀNH TẠI  
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong Khóa luận/Đồ án tốt nghiệp còn sử dụng một số nhận xét, đánh  
giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào chúng tôi xin hoàn toàn chịu trách nhiệm về nội dung Khóa luận/Đồ án tốt nghiệp của mình**. Trường Đại học Tôn Đức Thắngkhông liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày háng năm 2023

*Tác giả  
(Ký tên và ghi rõ họ tên)*

**PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN**

Phần xác nhận của GV hướng dẫn

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

TP. Hồ Chí Minh, ngày háng năm 2023

*(kí và ghi họ tên)*

Phần đánh giá của GV chấm bài  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

TP. Hồ Chí Minh, ngày háng năm 2023

*(kí và ghi họ tên)*

**MỤC LỤC**

[**CHƯƠNG 1: TÌM HIỂU CÁC MÔ HÌNH TỔNG QUÁT CHO TEXT GENERATION.** 7](#_Toc133361455)

[**1.1** **Lý thuyết LSTM** 7](#_Toc133361456)

[**1.1.1** **Mô hình LSTM:** 7](#_Toc133361457)

[**1.1.2** **So sánh giữa LSTM và RNN:** 7](#_Toc133361458)

[**1.2** **LSTM Model cho Chatbot.** 9](#_Toc133361459)

[**1.2.1** **Khái quát về LSTM Model cho Chatbot** 9](#_Toc133361460)

[**1.2.2** **Cách hoạt động của Chatbot sử dụng model LSTM:** 9](#_Toc133361461)

[**CHƯƠNG 2: CÁC BƯỚC ĐỂ ÁP DỤNG MÔ HÌNH** 18](#_Toc133361462)

[**2.1** **Thu thập dữ liệu:** 18](#_Toc133361463)

[**2.2** **Tiền xử lý dữ liệu:** 18](#_Toc133361464)

[**2.3** **Xây dựng model:** 18](#_Toc133361465)

[**2.4** **Giải quyết Overfiting:** 19](#_Toc133361466)

[**CHƯƠNG 3: HUẤN LUYỆN MÔ HÌNH** 20](#_Toc133361467)

[**3.1** **Độ đo Bleu** 20](#_Toc133361468)

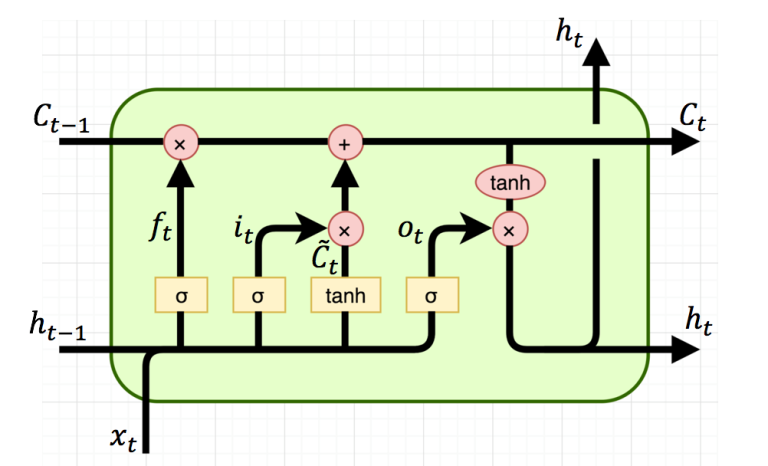
[**3.2** **Độ đo Rouge** 20](#_Toc133361469)

[**3.3** **Kết quả chạy model** 21](#_Toc133361470)

[**3.4** **So sánh với các model học máy khác** 21](#_Toc133361471)

[**TÀI LIỆU THAM KHẢO** 22](#_Toc133361472)

# **CHƯƠNG 1: TÌM HIỂU CÁC MÔ HÌNH TỔNG QUÁT CHO TEXT GENERATION.**

1. **Lý thuyết LSTM**
2. **Mô hình LSTM:**

Hình 48: LSTM

Out put:

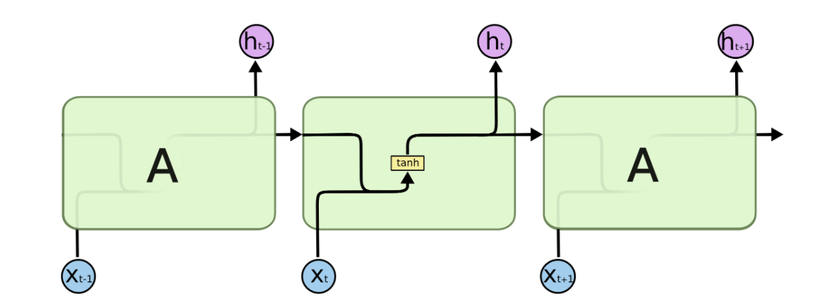
Input: . Trong đó là input ở state thứ t của model là output của layer trước. h đóng vai trò khá giống như s ở RNN, trong khi c là điểm mới của LSTM.

tương ứng với Forget rate, input rate, output rate:

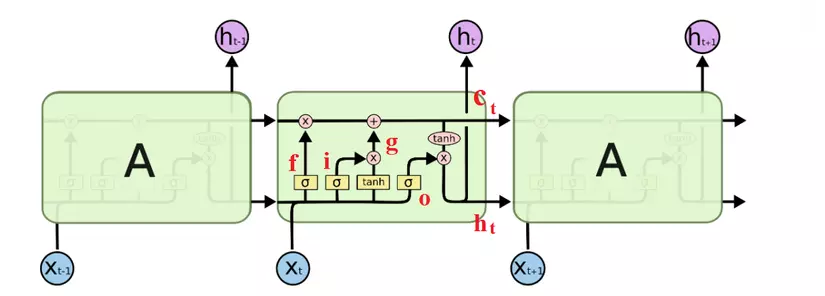
* Forget rate: . Forget rate quyết định những thông tin nào cần bỏ mà không quan trọng của bước trước
* Input rate: . Input rate quyết định thông tin nào được chấp nhận dựa vào độ ảnh hưởng tới bước hiện tại.
* Output rate: . Output rate cho phép thông tin chấp nhận từ input rate ảnh hưởng tới output ở bước hiện tại.

1. **So sánh giữa LSTM và RNN:**

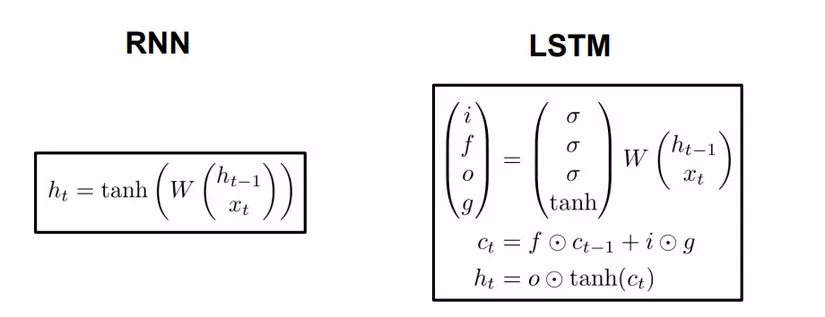
Hai sơ đồ hoạt động của RNN và LSTM ta có như sau:

 **RNN:**

Hình 49: RNN

 **LSTM**:

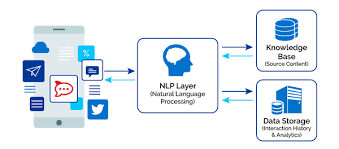
Hình 50: LSTM chi tiết

Ý tưởng của LSTM và RNN khá giống nhau, điểm khác biệt giữa 2 model này nằm ở các công thức tính toán.

Hình 51: So sánh giữa RNN và LSTM

1. **LSTM Model cho Chatbot.**
2. **Khái quát về LSTM Model cho Chatbot**

LSTM là một trong những kiến trúc mạng nơ-ron đặc biệt được sử dụng để xử lý dữ liệu chuỗi, giúp mô hình có khả năng học và ghi nhớ thông tin lâu hơn. LSTM được sử dụng rộng rãi trong các ứng dụng AI, bao gồm cả chatbot.



Trong một mô hình chatbot sử dụng LSTM, dữ liệu đầu vào là chuỗi văn bản từ người dùng. Để xử lý dữ liệu đầu vào, các bước tiền xử lý như chuẩn hóa, tách từ và vector hóa được thực hiện trước đó. Sau đó, các vector đại diện cho các từ được đưa vào mạng LSTM để học các đặc trưng của dữ liệu.

Mô hình chatbot sử dụng LSTM cần được huấn luyện với một tập dữ liệu lớn các câu hội thoại để học cách trả lời cho các câu hỏi của người dùng. Một phương pháp phổ biến để huấn luyện mô hình chatbot sử dụng LSTM là huấn luyện theo kiểu supervised learning, trong đó mô hình được cung cấp các cặp câu hỏi và câu trả lời. Mô hình sẽ học cách ánh xạ từ các câu hỏi đến các câu trả lời tương ứng.

Trong quá trình hoạt động, mô hình chatbot sử dụng LSTM sẽ nhận dữ liệu đầu vào là câu hỏi từ người dùng. Sau đó, mô hình sẽ sử dụng mạng LSTM để dự đoán câu trả lời phù hợp nhất với câu hỏi đó dựa trên kiến thức đã học được từ tập dữ liệu huấn luyện.

1. **Cách hoạt động của Chatbot sử dụng model LSTM:**

Khi sử dụng LSTM Model trong chatbot, mô hình sẽ được huấn luyện trên một tập dữ liệu lớn các câu hội thoại để học cách trả lời cho các câu hỏi của người dùng. Quá trình huấn luyện mô hình LSTM cho chatbot thường được thực hiện bằng cách sử dụng phương pháp supervised learning, trong đó các cặp câu hỏi và câu trả lời tương ứng được cung cấp cho mô hình.

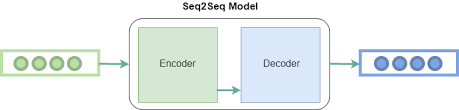
Khi chatbot nhận được câu hỏi từ người dùng, câu hỏi sẽ được xử lý và chuyển thành một vector số hóa trước khi được đưa vào mô hình LSTM để dự đoán câu trả lời tương ứng. Mô hình sẽ sử dụng các giá trị trọng số và đặc trưng học được trong quá trình huấn luyện để dự đoán câu trả lời cho câu hỏi đó.

Trong quá trình dự đoán câu trả lời, mô hình LSTM cho chatbot sẽ trả về một câu trả lời gợi ý. Câu trả lời này có thể được chatbot sử dụng trực tiếp để trả lời người dùng hoặc được sử dụng để tạo ra các câu trả lời khác thông qua các kỹ thuật như beam search hoặc sampling.

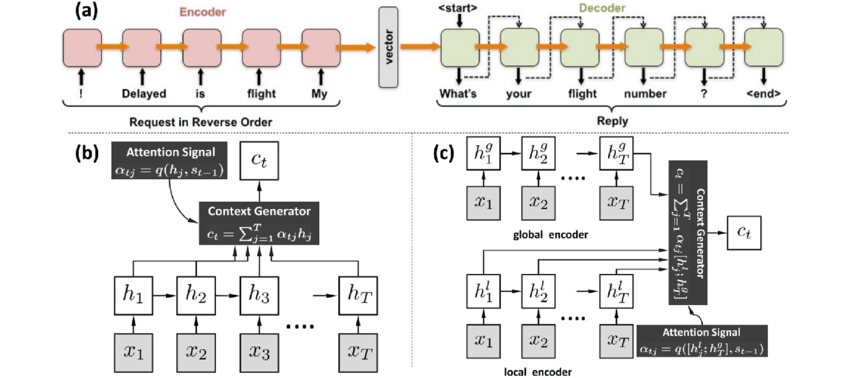
Tuy nhiên, mô hình LSTM cho chatbot không phải là hoàn hảo và có thể trả về các câu trả lời không phù hợp. Để cải thiện chất lượng câu trả lời, các kỹ thuật như fine-tuning, transfer learning và kết hợp nhiều kiến trúc mạng nơ-ron có thể được sử dụng.

1. **Mô hình Seq2seq**

Mô hình seq2seq là một mô hình dựa trên mạng nơ-ron, trong đó, một mạng nơ-ron truyền thẳng (encoder) được sử dụng để mã hóa thông tin từ chuỗi đầu vào, và một mạng nơ-ron giải mã (decoder) được sử dụng để tạo ra chuỗi đầu ra. Với mỗi chuỗi đầu vào, mô hình seq2seq tạo ra một chuỗi đầu ra tương ứng.



Mô hình seq2seq thường được xây dựng bằng cách sử dụng các kiến trúc LSTM (Long Short-Term Memory), một dạng của mạng nơ-ron hồi quy được thiết kế để xử lý dữ liệu chuỗi bằng cách lưu giữ thông tin quan trọng và bỏ qua thông tin không cần thiết. LSTM là một trong những phương pháp phổ biến trong việc xây dựng các mô hình seq2seq, do khả năng xử lý được dữ liệu chuỗi với độ dài khác nhau và giảm thiểu hiện tượng mất mát thông tin trong quá trình xử lý.



Trong mô hình seq2seq sử dụng LSTM, mạng encoder sẽ sử dụng một bộ LSTM để mã hóa chuỗi đầu vào thành một vector ẩn, trong khi mạng decoder sử dụng một bộ LSTM khác để giải mã chuỗi đầu ra từ vector ẩn đó. Việc sử dụng LSTM cho cả hai phần của mô hình sẽ giúp mô hình xử lý được các dữ liệu chuỗi có độ dài khác nhau và lưu giữ thông tin quan trọng để tạo ra chuỗi đầu ra chính xác hơn.

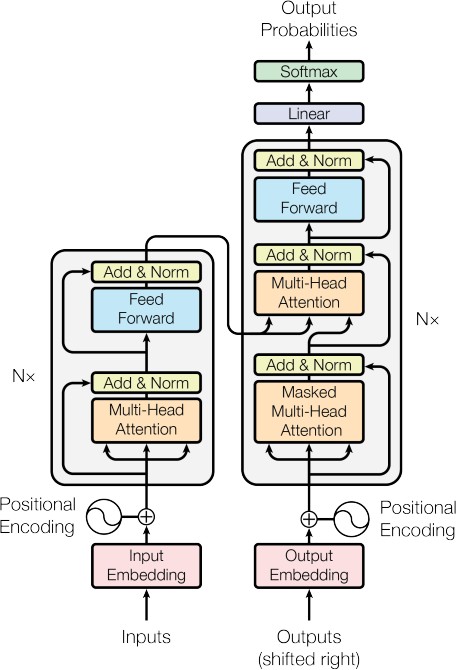
1. **Tranformer**
   * 1. **Motivation**

Các mô hình deep learning như CNN, RNN đã giải quyết được tốt các bài toán seq2seq có các ưu nhược điểm như:

* Các mạng CNN có thể dễ dàng thực hiện tính toán song song ở mỗi layer nhưng không có khả năng nắm bắt các phụ thuộc chuỗi có khả năng thay đổi.
* Các mạng RNN có khả năng nắm bắt các thông tin ở xa nhau trong chuỗi có độ dài biến thiên, nhưng không thể thực hiện song song trong một chuỗi mà phải tính toán lần lượt từng token.

Để kết hợp các ưu điểm của hai mô hình trên, vào 2017, Attention is all you need, đã đưa ra mô hình Transformer sử dụng cơ chế tập trung. Thực hiện song song bằng cách học chuỗi hồi tiếp với cơ chế tập trung, đồng thời mã hóa vị trí của từng token trong chuỗi.

Mô hình Transformer được thiết kế dựa trên kiến trúc encode-decode. Đầu vào của hai khối là embedding của chuỗi, kết hợp với các multi-head attention, thông tin vị trí qua positional encoding và áp dụng layer normalization.



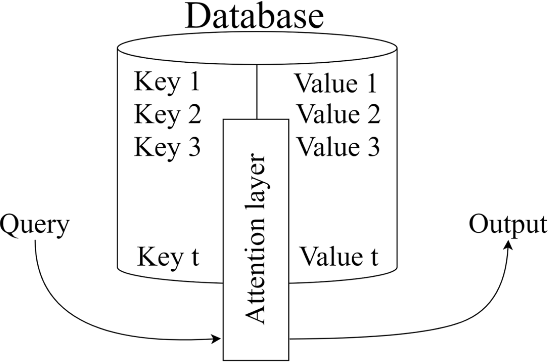
* + 1. **Cơ chế tập trung**

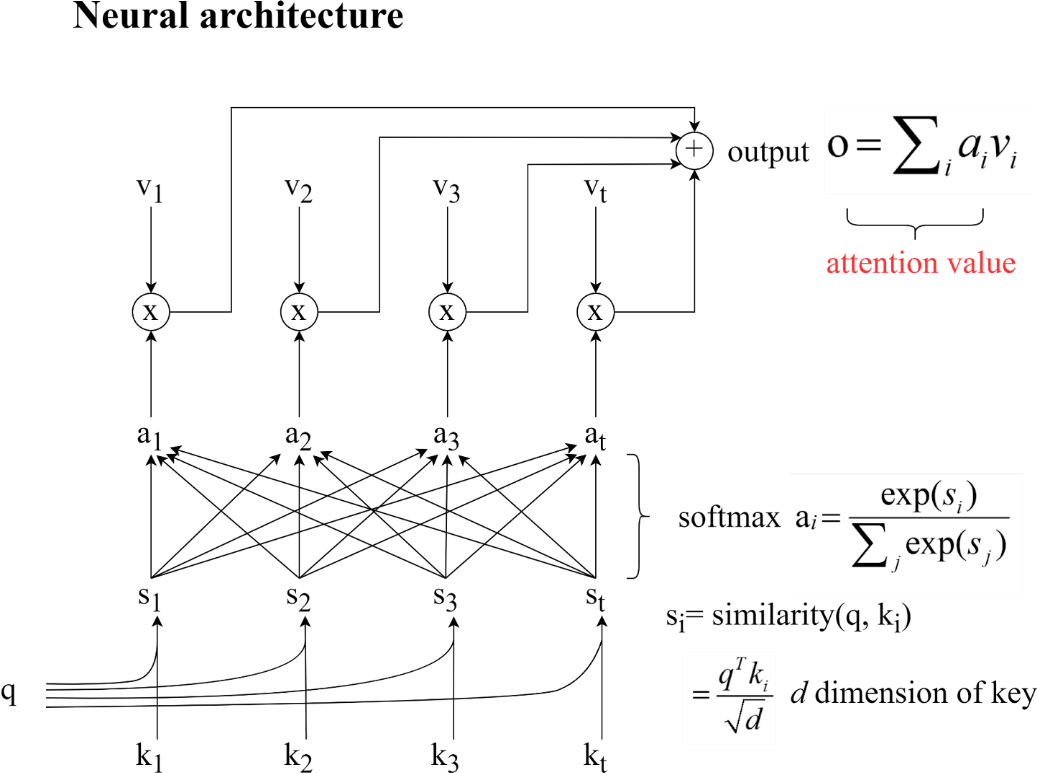
Attention mechanism (cơ chế tập trung) là một cơ chế mô phỏng theo sự tập trung ý thức thần kinh của con người dành cho mạng neural máy tính. Ở đây, cơ chế tập trung có thể xem như là phép gộp tổng quát theo trọng số trên mỗi giá trị đầu vào

Trong các mô hình seq2seq, các chuỗi input được encode thành các hidden state sau đó được đưa đến decoder sinh ra chuỗi output. Do đó, một token trong output chỉ liên quan đến một vài token chứ không phải toàn bộ token trong chuỗi nguồn. Các decoder có thể ngầm chọn thông tin tương ứng từ hidden state từ encoder. Tuy nhiên, cơ chế tập trung có thể thực hiện phép chọn này một cách tường min.

Cơ chế tập trung có thể được coi là một phép gộp tổng quát. Nó gộp các đầu vào dựa trên các trọng số khác nhau. Phần lõi của cơ chế này nằm ở attention layer (tầng tập trung). Đầu vào của tầng tập trung là các query. Với mỗi query, sẽ được tầng tập trung trả về đầu ra dựa trên bộ nhớ là tập các key-value được encode trong tầng tập trung này.

Ví dụ một bộ nhớ chứa n cặp vector key-value (k1, v1,)…, (kn, vn), với mỗi vector query q, tầng tập trung sẽ trả về một output o có cùng chiều kích thước với vector value.



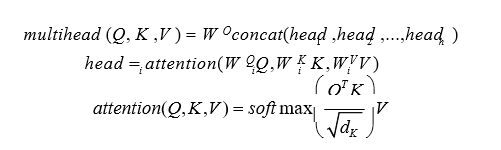


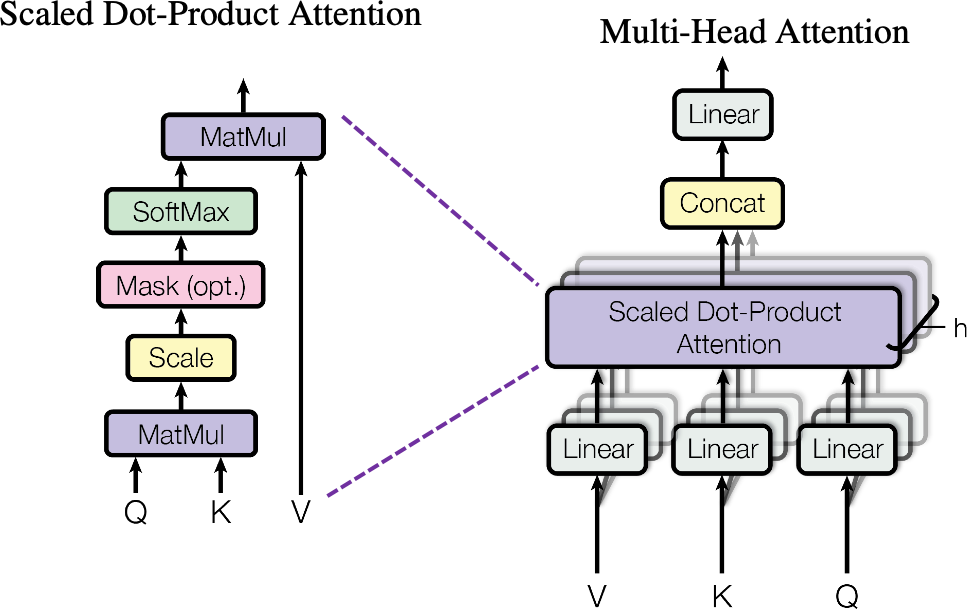
Cách lựa chọn scoring function khác nhau sẽ tạo ra các tầng tập trung khác nhau. Trong phạm vi bài tiểu luận này sẽ đề cập đến hai loại thường được sử dụng: dot product attention và additive attention.

* + 1. **Self-attention, Multi-head attention**

Kiến trúc self-attention (tự tập trung) sẽ biểu thị quan hệ giữa chính token đang xét với các token trong chuỗi.

Bằng cách sử dụng nhiều tầng self-attention song song với nhau ta được tầng multi-head attention.





* 1. **Positional encoding**

Không giống mô hình RNN, Transformer tính toán các đầu ra của từng phần tử trong chuỗi tại các tầng một cách độc lập. Nhờ đó ta có thể thực hiện tính toán song song nhưng lại không giữ lại được thông tin tuần tự trong chuỗi đầu vào. Để khắc phục điều này, Transformer đã sử dụng positional encoding để giữ lại được thông tin tuần tự một cách hiệu quả

*PE*( *position*, 2*i*) = sin( *position* / 100002*i*/*d* )

*PE*( *position*, 2*i* +1) = cos( *position* / 100002*i*/*d* )

Trong đó, position là vị trí hiện tại của token, i là chỉ số của phần tử nằm trong vector encoding, d là kích thước của các vector positional embedding.

* 1. **Mô hình GPT**

**1.5.1 GPT là gì?**

GPT (Generative Pre-trained Transformer) là một mô hình ngôn ngữ tự động sinh văn bản dựa trên kiến trúc Transformer. Transformer là một kiến trúc mạng nơ-ron tái cấu trúc cho các tác vụ xử lý ngôn ngữ tự nhiên, trong đó các phần tử đầu vào và đầu ra được xử lý dưới dạng chuỗi. GPT là một trong những mô hình nổi tiếng và hiệu quả nhất được xây dựng trên kiến trúc Transformer để giải quyết các tác vụ xử lý ngôn ngữ tự nhiên, như dịch máy, tóm tắt văn bản, hoặc sinh văn bản tự động.

GPT được xây dựng dựa trên kiến trúc Transformer và được huấn luyện trên một lượng lớn dữ liệu ngôn ngữ tự nhiên. Quá trình huấn luyện của GPT sử dụng một lớp đơn giản của kiến trúc Transformer gọi là "Transformer Decoder" để học cấu trúc ngôn ngữ tự nhiên và kết hợp các từ thành các câu hoàn chỉnh

Transformer và GPT có liên quan mật thiết với nhau. Transformer là kiến trúc mạng nơ-ron tái cấu trúc cho các tác vụ xử lý ngôn ngữ tự nhiên, trong khi GPT là một trong những mô hình thành công nhất được xây dựng trên kiến trúc Transformer.

**1.5.2 GPT 2 là gì?**

GPT-2 là mô hình ngôn ngữ được đào tạo bằng cách sử dụng Transformer, một kiến trúc mạng neural. Nó được huấn luyện trên 40GB văn bản được lấy từ Internet. Mô hình này có thể được sử dụng để tạo ra văn bản giống con người ở quy mô lớn.

GPT-2 bao gồm một encoder và một decoder. Encoder đọc các token đầu vào và chuyển đổi chúng thành các vector. Sau đó, decoder sử dụng các vector này để tạo ra các token đầu ra tiếp theo dựa trên ngữ cảnh. Quá trình này được lặp lại cho đến khi decoder tạo ra dấu hiệu kết thúc.

GPT-2 có thể mô phỏng ngôn ngữ tự nhiên và tạo ra văn bản liền mạch. Nó cũng có thể thực hiện các nhiệm vụ khác như tóm tắt văn bản, trả lời các câu hỏi, dịch ngôn ngữ. Mô hình này có kích thước lớn, có số lượng tham số lên tới hàng tỷ, do đó có thể mô phỏng ngôn ngữ phức tạp.

# **CHƯƠNG 2: CÁC BƯỚC ĐỂ ÁP DỤNG MÔ HÌNH**

1. **Thu thập dữ liệu:**

* Việc thu thập dữ liệu là bước quan trọng để xây dựng một chatbot chất lượng. Dữ liệu được sử dụng để huấn luyện chatbot và giúp chatbot hiểu được ngôn ngữ tự nhiên của con người. Có nhiều cách để thu thập dữ liệu cho chatbot, bao gồm:
  + Thu thập dữ liệu từ các nguồn công khai trên Internet: Điều này bao gồm các trang web, diễn đàn, mạng xã hội, v.v. có nội dung liên quan đến chủ đề mà chatbot sẽ hỗ trợ.
  + Thu thập dữ liệu từ các cuộc trò chuyện giữa con người: Điều này có thể được thực hiện bằng cách thu thập các cuộc trò chuyện qua email, chat, hoặc các cuộc trò chuyện trực tiếp.
  + Thu thập dữ liệu từ các bộ sưu tập câu hỏi và câu trả lời trên Internet: Các trang web như Quora, Stack Exchange, Yahoo Answers, v.v. có các bộ sưu tập câu hỏi và câu trả lời có thể được sử dụng để thu thập dữ liệu cho chatbot.

1. **Tiền xử lý dữ liệu:**

* Sau khi thu thập dữ liệu, việc tiền xử lý dữ liệu là bước quan trọng tiếp theo. Tiền xử lý dữ liệu bao gồm các bước sau:
* Loại bỏ các ký tự không cần thiết như dấu câu, ký tự đặc biệt, v.v.
* Chuyển đổi các từ thành chữ thường để giảm sự phức tạp của dữ liệu.
* Tách từ (tokenization) là quá trình chuyển đổi văn bản thành các từ riêng lẻ để tạo ra các vector đặc trưng.
* Xóa các từ không có ý nghĩa (stop words) như "và", "là", v.v. để giảm kích thước của tập từ vựng.
* Sử dụng phương pháp mã hóa để chuyển đổi các từ thành vector.

1. **Xây dựng model:**

Việc xây dựng model là bước quan trọng để tạo ra một chatbot hiệu quả. Có nhiều mô hình khác nhau để xây dựng chatbot, bao gồm mô hình truyền thống và mô hình học sâu (deep learning). Một số mô hình phổ biến để xây dựng chatbot bao gồm:

* Mô hình Markov: Mô hình Markov là một mô hình xác suất để

dự đoán từ tiếp theo trong chuỗi dữ liệu. Mô hình Markov thường được sử dụng cho các chatbot đơn giản và yêu cầu ít dữ liệu.

* Mô hình RNN (Recurrent Neural Network): Mô hình RNN là một mô hình học sâu phổ biến trong việc xây dựng chatbot. Mô hình này có thể hiểu được ngữ cảnh và chuỗi dữ liệu và có khả năng sinh ra các câu trả lời dựa trên các thông tin đã học được từ dữ liệu huấn luyện.
* Mô hình Transformer: Mô hình Transformer là một mô hình học sâu mới nhất trong lĩnh vực xử lý ngôn ngữ tự nhiên. Mô hình này được sử dụng cho các chatbot thông minh và yêu cầu nhiều dữ liệu.

1. **Giải quyết Overfiting:**

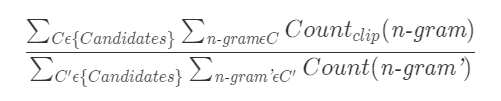
* Overfitting là một vấn đề trong huấn luyện mô hình chatbot khi mô hình quá tập trung vào dữ liệu huấn luyện và không thể tổng quát hóa được với dữ liệu mới. Có một số cách để giải quyết vấn đề này, bao gồm:
* Thu thập thêm dữ liệu huấn luyện: Nếu mô hình chatbot quá tập trung vào dữ liệu huấn luyện hiện có, việc thu thập thêm dữ liệu huấn luyện có thể giúp giảm overfitting.
* Sử dụng kỹ thuật Regularization: Kỹ thuật Regularization giúp giảm overfitting bằng cách thêm các ràng buộc vào mô hình để hạn chế việc mô hình quá tập trung vào dữ liệu huấn luyện.
* Sử dụng kỹ thuật Dropout: Kỹ thuật Dropout giúp giảm overfitting bằng cách ngẫu nhiên loại bỏ một số đơn vị trong quá trình huấn luyện mô hình.
* Sử dụng kiểm định chéo (Cross-validation): Kiểm định chéo giúp đánh giá khả năng tổng quát hóa của mô hình và giúp giảm overfitting bằng cách chia dữ liệu thành các tập huấn luyện và kiểm tra khác nhau.

**CHƯƠNG 3: HUẤN LUYỆN MÔ HÌNH**

1. **Độ đo Bleu**

Độ đo BLEU (BiLingual Evaluation Understudy) là một độ đo đánh giá chất lượng kết quả dịch thuật bằng cách so sánh văn bản dịch với các văn bản tham chiếu (hay còn gọi là ngữ liệu đích). Nó được sử dụng phổ biến trong các bài toán dịch máy và tổng hợp văn bản.

BLEU còn được dùng để đánh giá một **corpus** (tập hợp của các sentence, hay một đoạn văn) khá là tốt. Đầu tiên là tính số match với từng câu. Cộng các số này rồi chia cho tổng số n-gram từ các câu là ra **modified precision score** cho test corpus.



BLEU tính độ chính xác của bộ phân loại ngôn ngữ (language model) dựa trên sự khớp n-gram giữa đoạn văn bản đích (reference text) và văn bản được sinh ra (hypothesis text) bởi bộ phân loại. Kết quả BLEU được đưa ra dưới dạng tỷ lệ phần trăm, từ 0 đến 100. Khi giá trị BLEU càng cao thì chất lượng đoạn văn bản sinh ra càng tốt.

Để tính toán độ đo BLEU, trước tiên phải xác định các n-gram (các chuỗi liên tiếp gồm n từ) trong đoạn văn bản đích và đoạn văn bản sinh ra. Sau đó, tính toán số lượng các n-gram trong đoạn văn bản sinh ra có trong đoạn văn bản đích và chia cho tổng số các n-gram trong đoạn văn bản sinh ra. Cuối cùng, áp dụng trọng số của mỗi n-gram để tính toán điểm số của đoạn văn bản sinh ra. Trong BLEU, thường sử dụng các trọng số đơn giản như trọng số đều hoặc trọng số tỷ lệ tổng quan.

1. **Độ đo Rouge**

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) là một hệ thống đánh giá hiệu quả cho các bài tóm tắt văn bản (text summarization) hoặc các bài toán sinh câu (text generation). ROUGE sử dụng độ đo Precision, Recall và F1-score để đánh giá độ chính xác của các tóm tắt được tạo ra bởi một mô hình so với tóm tắt thực tế.

Hai loại đánh giá được sử dụng phổ biến: ROUGE-N, ROUGE-L:

* ROUGE-N : Nó đo lường sự trùng lặp của n-gram giữa bản tóm tắt được tạo tự động và bản tóm tắt tham chiếu. Trong n-gam giá trị của N có thể thay đổi từ 1 đến n nhưng khi giá trị của n tăng thì chi phí tính toán cũng tăng nhanh. Các số liệu n-gram được sử dụng chủ yếu là uni và bi-gram.
* ROUGE-L : Tính toán chuỗi dài nhất khớp nhau giữa tóm tắt tham khảo và tóm tắt dự đoán. Mỗi câu trong văn bản được coi là một chuỗi các từ. Hai văn bản tóm tắt có chuỗi các từ phổ biến dài hơn thì giống nhau hơn.

ROUGE được sử dụng rộng rãi trong nhiều cuộc thi và bài báo liên quan đến xử lý ngôn ngữ tự nhiên và máy học.

1. **Kết quả chạy model**
2. **So sánh với các model học máy khác**

# **TÀI LIỆU THAM KHẢO**

* + 1. <https://vinbigdata.com/cong-nghe-giong-noi/tong-quan-ve-he-thong-sinh-van-ban-tu-dong-natural-language-generation-nlg.html>
    2. <https://trituenhantao.io/kien-thuc/bleu-phep-do-trong-dich-may/>