**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

BÁO CÁO ĐỒ ÁN XỬ LÝ NGÔN NGỮ TỰ NHIÊN

**Đề tài: XÂY DỰNG MÔ HÌNH DỊCH MÁY CHO CẶP NGÔN NGỮ HÀN – VIỆT**

**Giảng viên hướng dẫn:** TS. Đặng Thị Phúc

**Lớp học phần:** DHKHDL15A

**Sinh viên thực hiện:** Trịnh Thị Bảo Bảo – 19514491

Hồ Quang Huy - 19443561

Thái Thị Hiền – 19527801

# **LỜI CẢM ƠN**

Lời đầu tiên, nhóm chúng em xin chân thành cảm ơn cô TS. Đặng Thị Phúc – giảng viên bộ môn Xử lý ngôn ngữ tự nhiên đã tận tình hướng dẫn và chỉ bảo trong thời gian làm đồ án.

Xin cảm ơn anh chị, bạn bè đã có nhiều góp ý đến nhóm trong quá trình học tập và nghiên cứu đồ án này.

Do kiến thức và kinh nghiệm của các thành viên trong nhóm có hạn chế nên đề tài không tránh khỏi sai sót. Mong sẽ nhận được những ý kiến nhận xét, góp ý của thầy cô và các bạn để đề tài được hoàn thiện hơn trong thời gian sắp tới.

Chúng em xin chân thành cảm ơn!

Thành phố HCM, ngày 16 tháng 05 năm 2022.

# **Mục lục**

[**Giới thiệu**](#_heading=h.gjdgxs)5

[**CHƯƠNG 1. TỔNG QUAN VỀ BÀI TOÁN**](#_heading=h.quqmicokjxtr)6

[1.Khái niệm dịch máy](#_heading=h.rzxm8hjozo8c) 6

[2.Cách tiếp cận](#_heading=h.u3tk844prlj6) 6

[**CHƯƠNG 2. THU THẬP VÀ XỬ LÝ DỮ LIỆU**](#_heading=h.2efm7wx8i3dv)7

[1.Thu thập dữ liệu](#_heading=h.c95nxtpook5i) 7

[2.Xử lý dữ liệu](#_heading=h.q45d75q9dy2l) 7

[2.1Tập hợp dữ liệu](#_heading=h.2qoheuetfap7) 7

[2.2 Làm sạch dữ liệu](#_heading=h.uwllvvsm7b4r) 7

[2.3 Lưu dữ liệu](#_heading=h.91rnncb774xl) 8

[3. Chuẩn bị dữ liệu cho mô hình](#_heading=h.emrnc6niiwcs) 8

[3.1 Thêm mã thông báo](#_heading=h.jq3vxhxt08yi) 8

[3.2 Tách từ](#_heading=h.gzkd6ncpnzq6) 8

[3.3 Tạo từ điển và vectơ hóa](#_heading=h.gzodaf18o58q) [8](#_heading=h.gzodaf18o58q)

[3.4 Padding](#_heading=h.nc3bs333erni) [8](#_heading=h.nc3bs333erni)

[3.5 Chia dữ liệu train-test-validation](#_heading=h.nf71ja349eql) [8](#_heading=h.nf71ja349eql)

[**CHƯƠNG 3. TRỰC QUAN HÓA DỮ LIỆU**](#_heading=h.gvqzizry5exv)10

[**CHƯƠNG 4. XÂY DỰNG MÔ HÌNH**](#_heading=h.362s7n2diopx)10

[1. Mô hình seq-2-seq](#_heading=h.8lwac1ummh4b) 10

[1.1. Kiến trúc LSTM](#_heading=h.h8nlfbslw43) 10

[1.2. Mô hình Seq2Seq](#_heading=h.6w6o7tw5gqxx) [12](#_heading=h.6w6o7tw5gqxx)

[2. Mô hình transformer](#_heading=h.vxwlvav2yt6c) 14

[2.1.Giới thiệu](#_heading=h.5tix98fw4qjx) 14

[2.2 Cơ chế Self attention:](#_heading=h.j62gy9886035) 15

[2.3 Cơ chế Masked Self Attention](#_heading=h.419n7ol3l8wd) 18

[2.4 Cơ chế Multi-Head Attention](#_heading=h.9wua64yamabi) 18

[2.5 Tổng quan kiến trúc transformer](#_heading=h.q1a7im8mb92m) 19

[2.5.1 Input embedding](#_heading=h.45up2sbaufyq) 20

[2.5.2 Positional Encoding](#_heading=h.qnkdati2tzfe) 21

[2.5.3 Lớp Normalization](#_heading=h.flpbokl4y9wq) 21

[2.5.4 Kết nối Residual](#_heading=h.1bj6u1t365wo) 21

[2.5.5 Feed-forward](#_heading=h.36w3rs9i30rd) 21

[**CHƯƠNG 5. THỬ NGHIỆM VÀ ĐÁNH GIÁ**](#_heading=h.y3bo5w4yds8o)21

[1. Phương pháp đánh giá](#_heading=h.pc38avftpbuy) 21

[2. Kết quả](#_heading=h.ufrx1mxi0aa8) 22

[**CHƯƠNG 6. TRIỂN KHAI BÀI TOÁN LÊN WEB**](#_heading=h.ht80274lbn1k)25

[1.Triển khai website:](#_heading=h.xn6eulbdy78v) 25

[2.Một số kết quả dịch](#_heading=h.t64qokf3gh75) 26

[**CHƯƠNG 7. KẾT LUẬN**](#_heading=h.tz2y12jpg7ie)27

[1.Kết luận](#_heading=h.352d1dwpx3pd) 27

[2.Hướng phát triển](#_heading=h.mevehsuw4gqf) 27

# **GIỚI THIỆU**

Dịch tự động hay còn gọi là dịch máy là một nhánh của xử lý ngôn ngữ tự nhiên. Như tên gọi, dịch tự động từ ngôn ngữ nguồn sáng một ngôn ngữ đích, không có sự can thiệp của con người trong quá trình dịch. Dịch máy là một bài toán sequence-to-sequence điển hình do đầu vào là một chuỗi và đầu ra cũng là chuỗi. Một vấn đề của bài toán là độ dài của chuỗi đầu vào và đầu ra biến đổi liên tục và không giống nhau.

Neural machine translation(NMT) là một trong số những phương pháp dịch máy. Nó sử dụng các mô hình neural network để học một mô hình thống kê cho quá trình dịch máy. Với phương pháp này, người ta chỉ cần huấn luyện một hệ thống duy nhất trên tập văn bản nguồn và văn bản đích (end-to-end system), không cần phải xây dựng một pipeline gồm các hệt thống chuyên biệt giống như SMT, không cần phải có nhiều kiến thức chuyên môn về ngôn ngữ, nhờ vậy mà có thể áp dụng cho các cặp ngôn ngữ khác nhau khá dễ dàng.

RNN, LSTM, GRU là các phương pháp tiếp cận hiện đại trong mô hình ngôn ngữ và dịch máy, từ đó khắc phục được những hạn chế của việc phụ thuộc xa trong mạng truyền thống. Việc áp dụng kết hợp cơ chế attention mang lại hiệu quả cao hơn. Cách tiếp cận seq2seq with attention là một trong những kỹ thuật kết hợp với LSTM. Vấn đề về dịch máy cũng rất được quan tâm tại Việt Nam. Tuy nhiên, cũng còn hạn chế khi nghiên cứu về Tiếng Việt. Khoảng những năm trở lại đây có một số nhóm nghiên cứu về dịch máy tiếng Việt nhưng chủ yếu tập trung vào hệ dịch Anh-Việt, Pháp-Việt. Hệ dịch mở Papago dịch khá tốt giữa tiếng Hàn và tiếng Việt. Trong phạm vi nghiên cứu của đề tài này chủ yếu trình bày về 2 mô hình seq2seq và transformer để ứng dụng dịch máy Hàn – Việt.

Bài báo cáo gồm có 5 nội dung chính:

1. Chương 1: Tổng quan về bài toán
2. Chương 2: Thu thập và xử lý dữ liệu
3. Chương 3: Trực quan hóa dữ liệu
4. Chương 4: Xây dựng mô hình
5. Chương 5: Thử nghiệm và đánh giá
6. Chương 6: Triển khai bài toán lên web

# **CHƯƠNG 1. TỔNG QUAN VỀ BÀI TOÁN**

## **1.Khái niệm dịch máy**

Dịch máy (machine translation) được hiểu là việc thực hiện dịch một ngôn ngữ này (gọi là ngôn ngữ nguồn) sang một hoặc nhiều ngôn ngữ khác (gọi là ngôn ngữ đích) một cách tự động, không có sự can thiệp của con người trong quá trình dịch. Hiện nay, trên thế giới có khoảng 5650 ngôn ngữ khác nhau, với số lượng ngôn ngữ lớn như vậy đã gây ra rất nhiều khó khăn trong việc trao đổi thông tin. Để có thể trao đổi thông tin phải cần đến một đội ngũ phiên dịch khổng lồ để dịch các văn bản, tài liệu, lời nói từ tiếng này sang tiếng khác. Vì vậy, con người đã nghĩ đến việc thiết kế một hệ thống tự động trong việc dịch.

Diagram

Description automatically generated

*Hình 1. Mô tả quá trình xử lý tài liệu dịch máy.*

Đầu vào của một hệ dịch máy là một văn bản được viết bằng một ngôn ngữ nguồn, quá trình dịch có thể chia thành hai giai đoạn. Đầu tiên, văn bản được phân tích thành các thành phần, sau đó được dịch thành văn bản ở dạng ngôn ngữ đích. Kết quả dịch có thể được con người hiệu chỉnh để trở thành bản dịch tốt. Như vậy trong một quá trình dịch, con người có thể tác động vào các bước xử lý với mục đích làm cho kết quả dịch tốt hơn.

## **2.Cách tiếp cận**

Có nhiều cách tiếp cận dịch máy, ở bài nghiên cứu này, nhóm chúng em tiếp cận dựa trên phương pháp dịch máy mạng nơ ron. Đây là một cách dịch tương tự như cách mà con người dịch. Tức là hệ thống sẽ thực hiện đọc toàn bộ câu nguồn, “hiểu” ý nghĩa của nó, sau đó dịch sang ngôn ngữ đích.

# **CHƯƠNG 2. THU THẬP VÀ XỬ LÝ DỮ LIỆU**

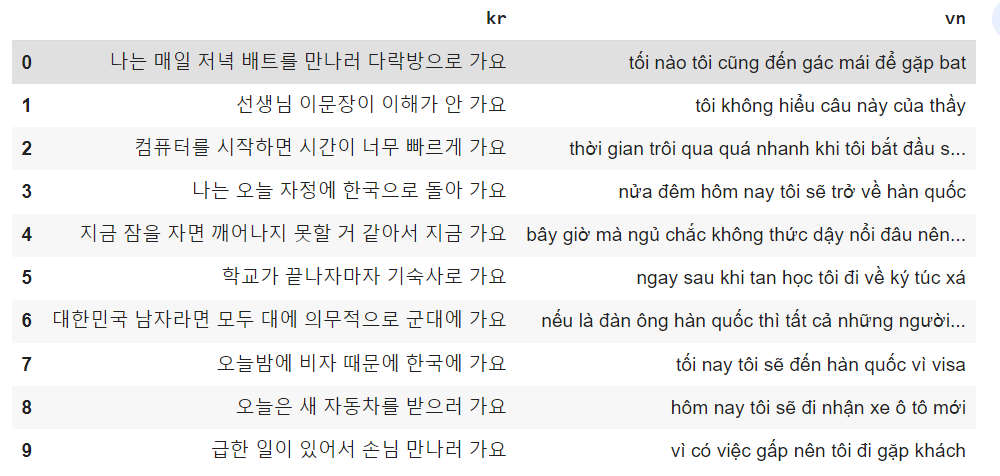
## **1.Thu thập dữ liệu**

Dữ liệu của bài toán được thu thập chủ yếu bằng tay. Qua các nguồn [báo song ngữ](http://saokhue.edu.vn/) Hàn -Việt, tập các file pdf luyện dịch hàn việt qua các tập [báo chí](https://thuvienpdf.com/luyen-dich-song-ngu-han-viet-qua-3000-tieu-de-bao-chi), và thu thập từ github. Bộ dữ liệu thô sơ thu thập được gần khoảng 150.000 trong suốt 2 tuần. Sau khi có bộ các câu tiếng Hàn thu thập trên mạng, chúng em chủ yếu dùng trang [papago](https://papago.naver.com/) để dịch sang tiếng Việt và thu thập về bằng kỹ thuật cào dữ liệu bằng selenium.

## **2.Xử lý dữ liệu**

### *2.1Tập hợp dữ liệu*

Dữ liệu được tập hợp từ các tập tin định dạng csv bao gồm 120000 cặp câu Korean-Vietnamese và được chuyển đổi sang cấu trúc dataframe.



*Hình 2. Minh họa tập dữ liệu Hàn - Việt.*

### *2.2 Làm sạch dữ liệu*

Quá trình làm sạch dữ liệu bao gồm:

* Xóa các dòng dữ liệu chứa dữ liệu trống (Nan).
* Xóa khoảng trắng, ký tự đặc biệt, dấu chấm câu trong câu.
* Chuyển toàn bộ dữ liệu sang chữ thường.
* Xóa các dữ liệu bị trùng

Sau quá trình làm sạch dữ liệu thu được 99670 cặp câu Korean-Vietnamese dùng cho quá trình dậy dựng mô hình.

### *2.3 Lưu dữ liệu*

Dữ liệu được lưu lại theo định dạng csv.

## **3. Chuẩn bị dữ liệu cho mô hình**

### *3.1 Thêm mã thông báo*

Thêm vào đầu và cuối câu của dữ liệu tiếng việt mã thông báo “*startseq”* và “*stopseq”.*

* startseq: Thông báo bắt đầu của chuỗi và trạng thái đầu tiên của vecto ngữ cảnh.
* stopseq: Thông báo kết thúc quá trình dự đoán.

### *3.2 Tách từ*

Đối với dữ liệu tiếng Hàn sử dụng thư viện *konlpy* để tách các từ trong một câu.

Đối với dữ liệu tiếng Việt dùng thư viện *underthesea* để tách các từ trong một câu.

### *3.3 Tạo từ điển và vectơ hóa*

Từ dữ liệu các từ đã được tách thực hiện tạo từ điển theo mô hình *bag of words*.

Mô hình *bag of words* tạo danh sách từ điển chứa tất các các từ riêng biệt trong bộ dữ liệu từ đó vecto hóa các câu dựa trên danh sách các từ trong từ điển.

### *3.4 Padding*

Mô hình nhận vào các vectơ từ có cùng hình dạng và kích thước. Tuy nhiên, các câu trong tập dữ liệu không cùng độ dài với nhau. Vì vậy, quá trình padding sẽ làm cho các vectơ từ có kích thước bằng nhau khi đưa vào mô hình.

Quá trình được thực hiện như sau:

* Từ các vectơ từ được tạo từ quá trình vectơ hóa tìm ra câu có độ dài lớn nhất.
* Thêm các số 0 vào cuối các vectơ từ để có cùng độ dài với vectơ từ có độ dài lớn nhất.

### *3.5 Chia dữ liệu train-test-validation*

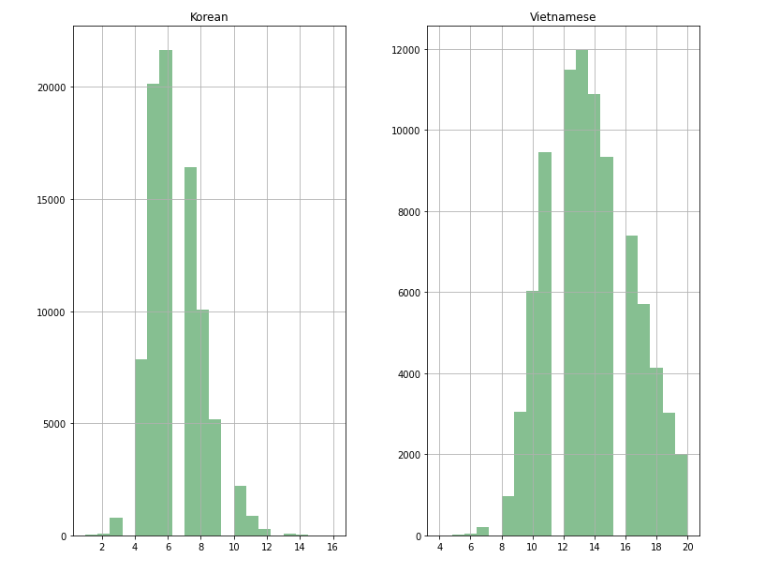
Dữ liệu được chia theo tỷ lệ train-test-validation là 76:12:12.

|  | Training set | Validation set | Testing set |
| --- | --- | --- | --- |
| Số lượng dữ liệu | 75750 | 11960 | 11960 |

*Bảng 1. Thống kê dữ liệu sử dụng trong mô hình.*

# **CHƯƠNG 3. TRỰC QUAN HÓA DỮ LIỆU**

Việc trực quan hóa được ứng dụng trong bài toán dịch máy nhằm thể hiện số lượng các câu có độ dài bằng bao nhiêu. Cụ thể là:



*Hình 3. Biểu đồ thể hiện độ dài của các câu trong tập dữ liệu Hàn - Việt*

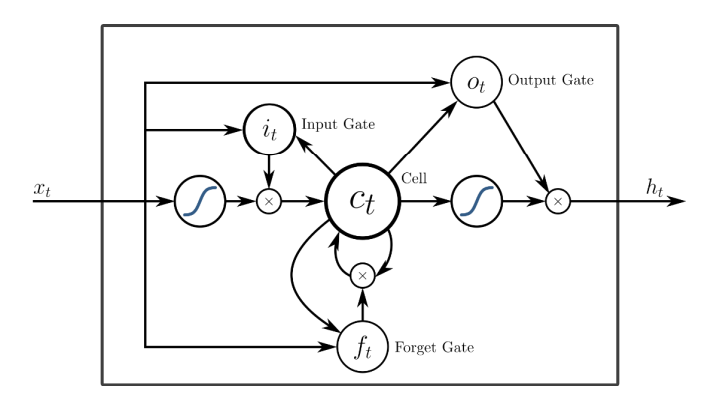
* Cột X thể hiện số từ có trong một câu
* Cột Y thể hiện số lượng câu có cùng số từ
* Ở biểu đồ trên, khi số lượng các từ trong câu đã được giới hạn trong phạm vi 20 từ thì biểu đồ đã thể hiện được các vấn đề như sau:
  + Đối với tiếng Hàn số từ trong câu phân bố nhiều trong đoạn từ 4 từ đến 10 từ, đặc biệt câu có 6 từ chiếm số lượng cao nhất
  + Đối với tiếng Việt số lượng từ trong câu phân bố nhiều trong đoạn từ 8 từ - 20 từ, đặc biệt câu có 13 từ có số lượng cao nhất.

# **CHƯƠNG 4. XÂY DỰNG MÔ HÌNH**

## **1. Mô hình seq-2-seq**

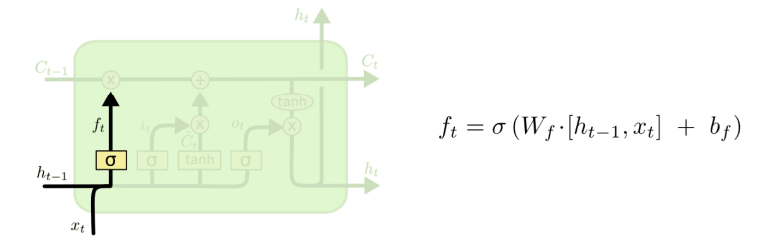
### *1.1. Kiến trúc LSTM*

* Long Short Term Memory networks - thường được gọi là: “LSTM”, là trường hợp đặc biệt của RNN, có khả năng học với sự phụ thuộc lâu dài của các nơ-ron. Mô hình này được giới thiệu bởi Hochreiter & Schmidhuber(1997), và được cải tiến lại bởi Ayako Mikami (2016). Mục tiêu chính của LSTM là quyết định thông tin nào được lưu lại và loại bỏ tại mỗi nơ-ron của RNN



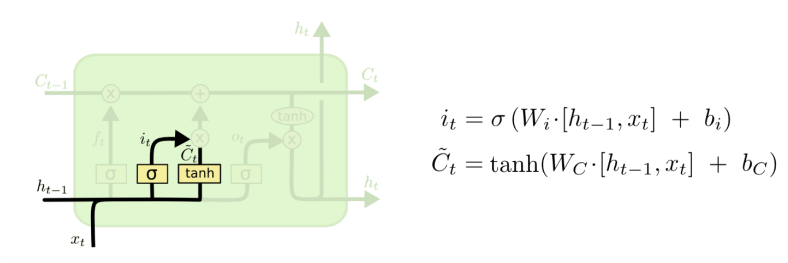
*Hình 4. Mô tả mô hình LSTM*

* Cổng quên: Cổng này quyết định xem thông tin nào trong bộ nhớ hiện tại được giữ và thông tin nào bị loại bỏ. Thông tin đầu vào được cho vào hàm Sigmoid. Đầu ra của hàm này đóng vai trò là mask để lọc thông tin từ trạng thái cell.



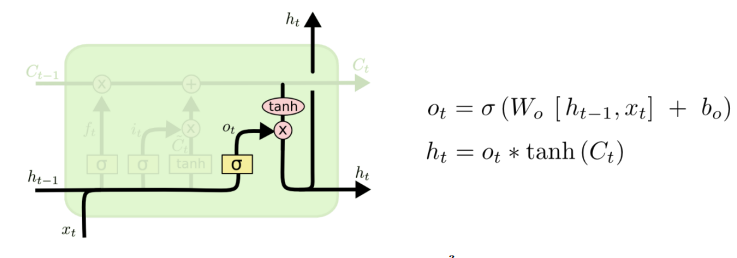
*Hình 5. Mô tả cổng quên trong mô hình LSTM*

* Cổng vào: Cổng này dùng để cập nhật bộ nhớ với các thông tin mới. Ở đây có xuất hiện 2 hàm Sigmoid và hàm tanh. Tác dụng của chúng cũng như trên. Output từ hàm Sigmoid sẽ có tác dụng lọc thông tin đã qua xử lý từ output hàm tanh.



*Hình 6. Mô tả cổng vào trong mô hình LSTM*

* Cổng ra: Cổng này quyết định output của từ hiện tại là gì. Nó được lấy thông tin từ 2 nguồn: trạng thái cell và input hiện tại. Trạng thái cell sau khi chỉnh sửa sẽ đi qua hàm tanh và input hiện tại thì được đi qua hàm Sigmoid. Kết hợp 2 kết quả trên để có được kết quả đầu ra. Kết quả đầu ra và cả trạng thái cell đều được đưa vào bước tiếp theo.



*Hình 7. Mô tả kết hợp 2 kết quả trên.*

### *1.2. Mô hình Seq2Seq*

* Đây là mô hình được đưa ra trong bài nghiên cứu Sequence to Sequence Learning with Neural Networks, mà theo đó một câu sẽ được dịch bằng cách đưa vào một bộ mã hóa và nối tiếp với một bộ giải mã để dịch ra một câu ở ngôn ngữ khác.
* Cụ thể, ngôn ngữ nguồn sẽ được mã hóa thành một vector và được đưa vào bộ mã hóa. Sau đó bộ giải mã sẽ lần lượt sinh từng từ trong chuỗi đầu ra dựa trên vector đầu vào và những từ được dự đoán trước đó tới khi gặp từ kết thúc câu. Trong mô hình Sequence to Sequence chúng ta có thể sử dụng những thành phần khác nhau cho bộ mã hóa và giải mã như RNN hoặc những cải tiến như LSTM và GRU để giải quyết vấn đề phụ thuộc xa và biểu diễn các mối quan hệ phụ thuộc vào ngữ cảnh của câu.



*Hình 8. Kiến trúc mô hình Seq2Seq*

* Các thành phần chính của mô hình Sequence-to-Sequence bao gồm:
* Bộ Encoder được sử dụng để ánh xạ chuỗi token trong ngôn ngữ nguồn đầu vào thành một vector có kích thước cố định. Tại mỗi bước mã hóa, Encoder sẽ nhận vector tương ứng với mỗi token trong chuỗi đầu vào để tạo ra vector trạng thái ẩn đại diện cho chuỗi đầu vào tại bước mã hóa cuối cùng
* Bộ Decoder sử dụng vector như khởi tạo cho trạng thái ẩn đầu tiên và tạo ra chuỗi các token ở ngôn ngữ đích tại mỗi bước giải mã. Do đó, hàm xác suất có điều kiện có thể được phân tích như sau:

****

* Trong vế phải của công thức trên, mỗi phân bố **𝑝(𝑦𝑗 |𝑠, … , 𝑦𝑗−1 )** mô tả xác suất xuất hiện của token với vector đại diện cho câu đầu vào và các token trong chuỗi đầu ra đứng trước nó. Phân bố này được biểu diễn bằng một hàm softmax trên tất cả token trong tập từ vựng ở ngôn ngữ đích.
* Công thức trên có thể được viết lại thành dạng như sau:



* Mỗi token có xác suất xuất hiện được tính như sau:

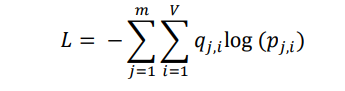


+Trong đó **g** là hàm dùng để biến đổi trạng thái ẩn của bộ giải mã tại bước giải mã tương ứng thành vector có kích thước bằng kích thước của tập từ vựng trong ngôn ngữ đích. Trạng thái ẩn được tính như sau:



-Trong đó f là hàm biểu diễn chung cho quá trình tính trạng thái ẩn tại bước hiện tại từ trạng thái ẩn đầu ra của bước trước bằng mạng RNN hoặc bằng những cải tiến khác như LSTM và GRU

* Hàm mất mát cần tối ưu hóa trong quá trình huấn luyện là một hàm có dạng tích của các hàm cross-entropy:

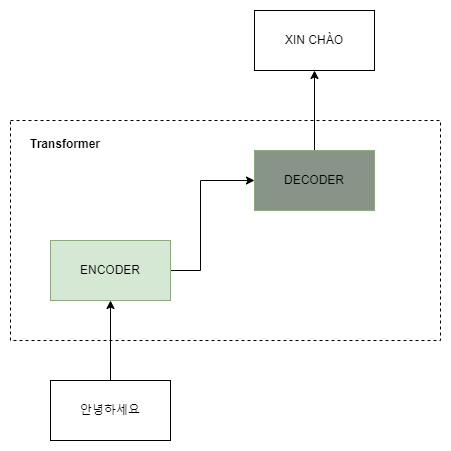


* Trong đó, là phần tử thứ i của vector one-hot có kích thước V tại bước giải mã thứ j. Vector biểu diễn cho token thứ j trong chuỗi token nhãn đầu ra từ tập huấn luyện. là phần tử thứ i của vector cũng có kích thước V với = 𝑠𝑜𝑓𝑡𝑚𝑎𝑥(𝑔()).
* Mô hình Seq2Seq cơ bản có nhược điểm là yêu cầu RNN decoder sử dụng toàn bộ thông tin mã hóa từ chuỗi đầu vào cho dù chuỗi đó dài hay ngắn. Thứ hai, RNN encoder cần phải mã hóa chuỗi đầu vào thành một vector duy nhất và có độ dài cố định. Ràng buộc này không thực sự hiệu quả vì trong thực tế việc sinh ra từ tại một bước thời gian trong chuỗi đầu ra có khi phụ thuộc nhiều hơn vào một số những thành phần nhất định trong chuỗi đầu vào. Ví dụ, khi dịch một câu từ tiếng nước này sang tiếng nước khác, chúng ta thường quan tâm nhiều hơn đến ngữ cảnh xung quanh từ hiện tại so với các từ khác trong câu.

## **2. Mô hình transformer**

### *2.1.Giới thiệu*

Giống như các mô hình dịch máy khác, kiến trúc tổng quan của mô hình transformer bao gồm 2 phần là encoder và decoder. Encoder dùng để học vector biểu diễn của câu với mong muốn rằng vector này mang thông tin hoàn hảo của câu đó. Decoder thực hiện chức năng chuyển vector biểu diễn kia thành ngôn ngữ đích. Trong ví dụ dưới đây, encoder của mô hình transformer nhận vào một câu tiếng hàn, encode thành một vector biểu diễn ngữ nghĩa của câu “안녕하세요”. Sau đó mô hình decoder nhận vector này và dịch nó thành câu tiếng việt “Xin chào”.

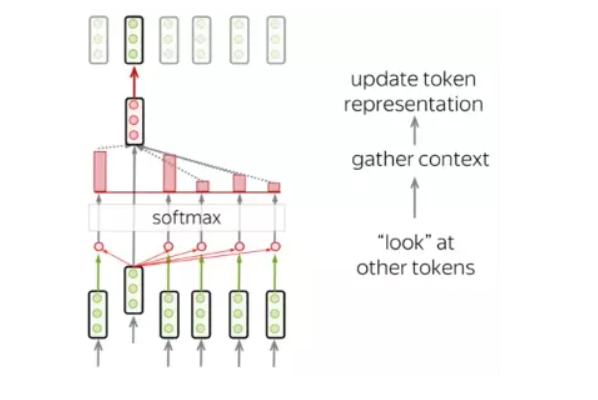


*Hình 9. Ví dụ minh họa cho mô hình transformer*

Một trong những ưu điểm của nó là khả năng xử lý song song cho các từ. Vì encoders của transformer là một dạng feedforward neural nets, bao gồm nhiều encoder layer khác, mỗi encoder này xử lý đồng thời các từ. Trong khi đó, LSTM thì các từ phải xử lý tuần tự. Ngoài ra, mô hình Transformer còn xử lý câu đầu vào theo 2 hướng mà không cần phải stack thêm một hình LSTM nữa như trong kiến trúc Bidirectional LSTM.

### *2.2 Cơ chế Self attention:*

Self-attention - “trái tim” của mô hình transformer. Đây là cơ chế giúp mô hình “hiểu” được sự liên quan giữa các từ trong một câu. Nó giống như cơ chế tìm kiếm. Với một từ cho trước, cơ chế self-attention sẽ cho phép mô hình tìm trong các từ còn lại để xác định từ nào liên quan để sau đó thông tin được mã hóa dựa trên tất cả các từ

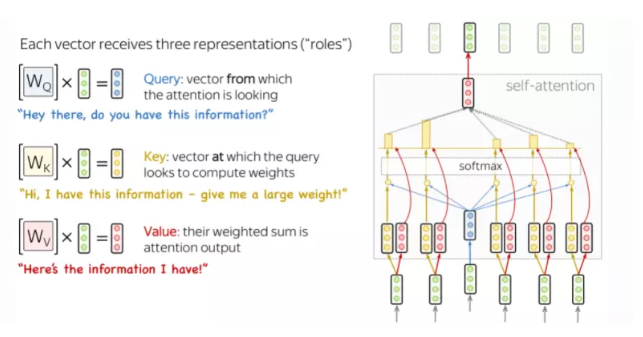


*Hình 10. Mô tả cơ chế self-attention*

Để xây dựng cơ chế này cần chú ý đến hoạt động của 3 vector biểu diễn cho mỗi từ:

* Query: dùng để chứa thông tin của từ được tìm kiếm, so sánh.
* Key: dùng để biểu diễn thông tin các từ được so sánh với từ cần tìm kiếm ở trên
* Value: biểu diễn nội dung, ý nghĩa của các từ.

Query được sử dụng khi một token "quan sát" những tokens còn lại, nó sẽ tìm kiếm thông tin xung quanh để hiểu được ngữ cảnh và mối quan hệ của nó với các tokens còn lại. Key sẽ phản hồi yêu cầu của Query và được sử dụng để tính trọng số attention. Cuối cùng, Value được sử dụng trọng số attention vừa rồi để tính ra vector đại diện (attention vector). Trong ảnh 3 ma trận và chính là các hệ số mà mô hình cần huấn luyện.



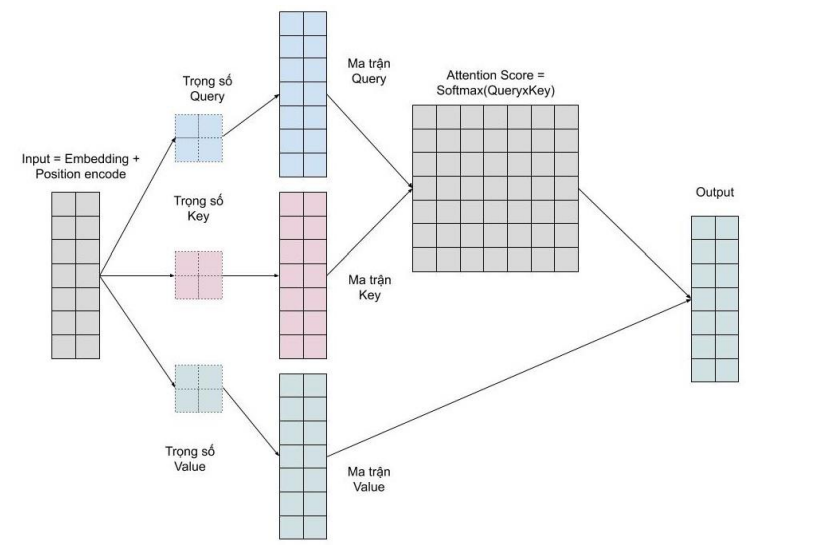
*Hình 11. Mô tả cơ chế attention và 3 vector biểu diễn.*

Vector attention cho một từ thể hiện tính tương quan giữa 3 vector này được tạo

ra bằng cách nhân tích vô hướng giữa chúng và sau đó được chuẩn hóa bằng hàm

softmax. Cụ thể quá trình tính toán như sau:

* Bước 1: Tính ma trận query, key, value bằng cách nhân input với các ma trận trọng số tương ứng
* Bước 2: Nhân hai ma trận query, key vừa tính được với nhau với ý nghĩa so sánh giữa câu query và key để học mối tương quan. Sau đó các giá trị sẽ được chuẩn hóa về khoảng [0-1] bằng hàm softmax với ý nghĩa 1 khi câu query giống với key ngược lại, 0 có nghĩa là không giống.
* Bước 3: Output sẽ được tính bằng cách nhân ma trận vừa được tạo ra ở bước 2 với ma trận value.



*Hình 12. Quá trình tính toán vector attention.*

Công thức tính attention vector như sau:

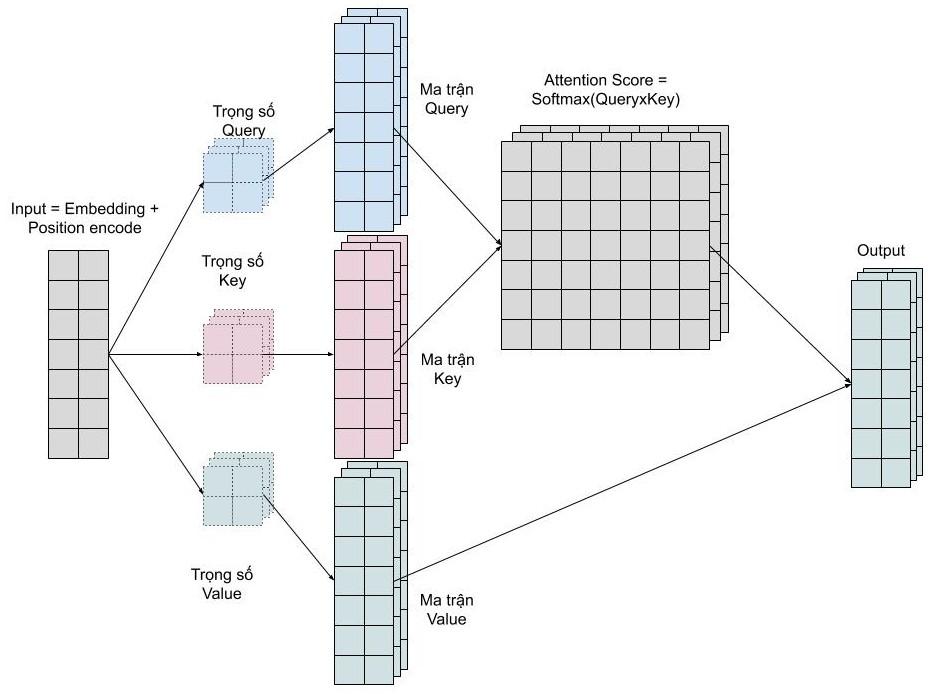
Với là số chiều của vector Key với mục đích tránh tràn luồng!

### *2.3 Cơ chế Masked Self Attention*

Đây là cơ chế được sử dụng cho decoder trong transformer, cụ thể nó thực hiện nhiệm vụ chỉ cho phép target token tại time-step hiện tại chỉ được phép dùng các tokens ở time-step trước đó. Về hoạt động nó cũng giống như đã giới thiệu ở trên, ngoại trừ việc nó không tính đến attention của những tokens trong tương lai.

### *2.4 Cơ chế Multi-Head Attention*

Chúng ta muốn mô hình có thể học nhiều kiểu mối quan hệ giữa các từ với nhau. Với mỗi self-attention, chúng ta học được một kiểu pattern, do đó để có thể mở rộng khả năng này, chúng ta đơn giản là thêm nhiều self-attention. Tức là chúng ta cần nhiều ma trận query, key, value mà thôi. Giờ đây ma trận trọng số key, query, value sẽ có thêm 1 chiều depth nữa.



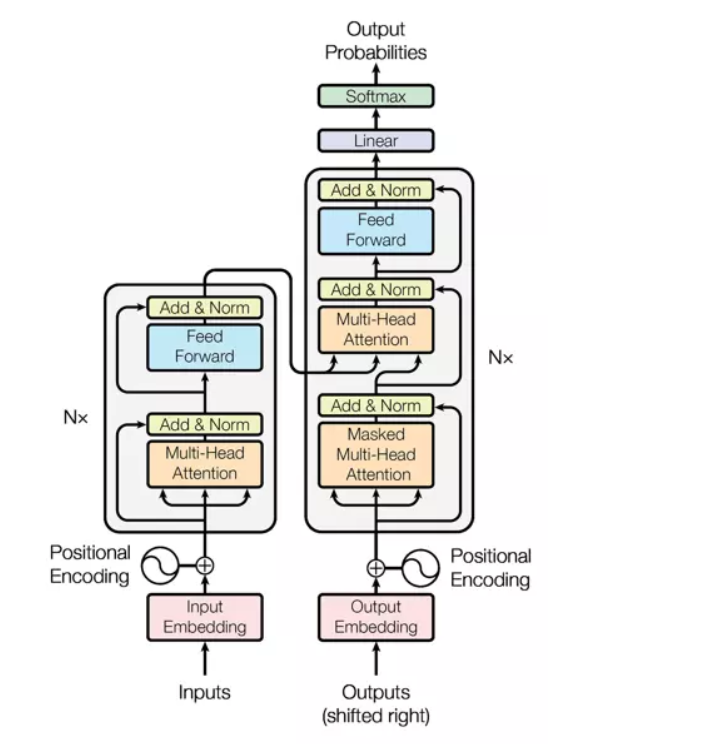
*Hình 13. Mô tả cơ chế Multi-head attention.*

Multi head attention cho phép mô hình chú ý đến đồng thời những pattern dễ quan sát được như sau.

* Chú ý đến từ kế trước của một từ
* Chú ý đến từ kế sau của một từ
* Chú ý đến những từ liên quan của một từ

### *2.5 Tổng quan kiến trúc transformer*

Đây là mô hình được các kỹ sư của Google giới thiệu năm 2017 trong bài báo Attention Is All You Need. Chi tiết mô hình được biểu diễn bởi hình vẽ bên dưới.



*Hình 14. Kiến trúc mô hình transformer*

Ở bên trái là encoder, thường có Nx=6 layers chồng lên nhau. Mỗi layer sẽ có lớp đầu tiên Multi-head attention và lớp thứ 2 là mạng Feed-forward kết nối đầy đủ. Ở decoder cũng tương tự như encoder tuy nhiên sẽ có thêm khối masked multi-head attention ở vị trí đầu tiên.

#### *2.5.1 Input embedding*

Các câu đầu vào sẽ được mã hóa thành các vector bằng việc sử dụng Word Embedding.

#### *2.5.2 Positional Encoding*

Input embedding giúp ta biểu diễn ngữ nghĩa của một từ, tuy nhiên cùng một từ ở vị trí khác nhau của câu lại mang ý nghĩa khác nhau. Mặc khác, transformer không có mạng hồi tiếp hay tích chập nên nó không biết được thứ tự của các token đầu vào. Vì vậy, Transformers có thêm một phần Positional Encoding để cho biết thêm thông tin về vị trí của một từ.

Như vậy bộ mã hóa sẽ nhận ma trận biểu diễn của các từ đã được cộng với vị trí thông qua positional encoding. Sau đó, ma trận này sẽ được xử lý bởi Multi head attention.

#### *2.5.3 Lớp Normalization*

Trong hình ảnh cấu trúc, có lớp "Add & Norm" thì từ Norm thể hiện cho lớp Normalization. Lớp này đơn giản là sẽ chuẩn hóa lại đầu ra của multi-head attention, mang lại hiệu quả cho việc nâng cao khả năng hội tụ.

#### *2.5.4 Kết nối Residual*

Bản chất rất đơn giản: thêm đầu vào của một khối vào đầu ra của nó. Với kết nối này giúp mạng có thể chồng được nhiều layers. Như trên hình, kết nối residual sẽ được sử dụng sau các khối FFN và khối attention. Như trên hình từ "Add" trong "Add & Norm" sẽ thể hiện cho kết nối residual.

#### *2.5.5 Feed-forward*

Đây là khối cơ bản, sau khi thực hiện tính toán ở khối attention ở mỗi lớp thì khối tiếp theo là FFN. Có thể hiểu là cơ chế attention giúp thu thập thông tin từ những tokens đầu vào thì FFN là khối xử lí những thông tin đó.

# **CHƯƠNG 5. THỬ NGHIỆM VÀ ĐÁNH GIÁ**

## ***1. Phương pháp đánh giá***

Bilingual Evaluation Understudy Score hay ngắn gọn là BLEU score là một thang điểm được dùng phổ biến trong đánh giá chất lượng dịch máy. BLEU được Kishore Papineni và cộng sự đề xuất lần đầu vào năm 2002 qua bài nghiên cứu “A Method for Automatic Evaluation of Machine Translation”.

BLEU được tính dựa trên số lượng n-grams giống nhau giữa câu dịch của mô hình (output) với các câu tham chiếu tương ứng có xét tới độ dài của câu. Số n-grams tối đa của BLEU là không giới hạn, nhưng vì xét về ý nghĩa, cụm từ quá dài thường không có nhiều ý nghĩa, và nghiên cứu cũng đã cho thấy là với 4-gram, điểm số BLEU trung bình cho khả năng dịch thuật của con người cũng đã giảm khá nhiều nên n-grams tối đa thường được sử dụng là 4-gram. Công thức để tính điểm đánh giá như sau:

Trong đó,

* là số lượng các n-grams trong phân đoạn j của bản dịch dùng để tham khảo.
* là số lượng các n-grams trong phân đoạn j của bản dịch bằng máy.
* là số lượng các n-grams trong phân đoạn j của bản dịch bằng máy.
* là số lượng các từ trong bản dịch bằng máy.

Giá trị score đánh giá mức độ tương ứng giữa hai bản dịch và nó được thực hiện trên từng phân đoạn, ở đây phân đoạn được hiểu là đơn vị tối thiểu trong các bản dịch,mthông thường mỗi phân đoạn là một câu hoặc một đoạn. Việc thống kê đồ trùng khớp của các n-grams dựa trên tập hợp các ngrams trên các phân đoạn, trước hết là nó được tính trên từng phân đoạn, sau đó tính lại giá trị này trên tất cả các phân đoạn.

## ***2. Kết quả***

Dưới đây là điểm số BLEU khi sử dụng hệ thống để dịch bộ dữ liệu kiểm thử từ tiếng Hàn sang tiếng Việt. Để đánh giá, chúng em có so sánh với mô hình seq2seq LSTM, RNN+attention và transformer.

***Bảng 2. Bảng điểm BLEU của hệ thống dịch máy Trung – Việt***

***đánh giá trên tập kiểm thử - testing set***

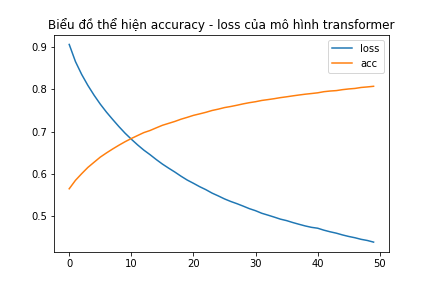
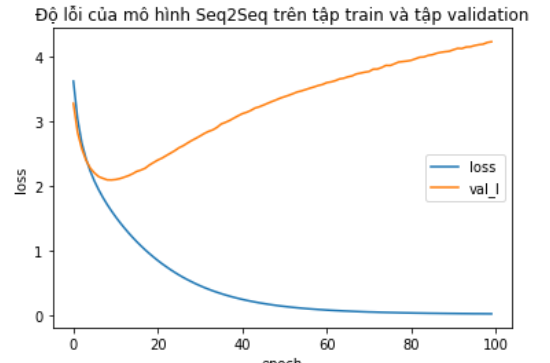
| Mô hình | Tổng tham số | BLEU SCORE |
| --- | --- | --- |
| seq2seq-1 | 34,362,621 | 0.2526 |
| seq2seq-2 | 34,362,621 | 0.3019 |
| Transformer-1 | 26,375,650 | 0.7853 |
| Transformer-2 | 7,895,576 | 0.3556 |
| Transformer-3 | 19,970,456 | 0.5960 |

Dưới đây là một số kết quả dịch ngôn ngữ Hàn - Việt sử dụng mô hình Transformer.

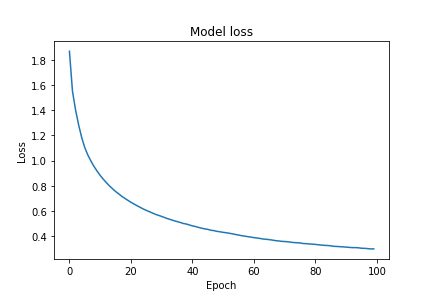
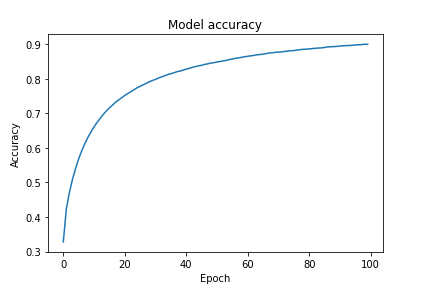
***Bảng 3. Một số kết quả dịch***

| Đầu vào | Đầu ra | Tham chiếu |
| --- | --- | --- |
| 일요일 오후에 약속 잡아 놓겠어요 | tôi sẽ hẹn vào chiều chủ nhật | tôi sẽ hẹn vào chiều chủ nhật |
| 입금확인서를 먼저 보내드리도록 하겠어요 | tôi sẽ gửi cho bạn trước | tôi sẽ gửi giấy xác nhận tiền gửi trước |
| 생각보다 비싸네요 | mình nghĩ vậy hơn mình nhiều lắm | mắc hơn mình nghĩ đó |
| 글쎄 남동생이 내 숙제를 다 지워 버린 것이에요 | xem nào em trai tôi đã loại bỏ tất cả bài tập của tôi | xem nào em trai tôi đã xóa hết bài tập của tôi |
| 기념품으로 많이 사 들고 갑니다 | mình đã mua rất nhiều quà tặng | mình đã mua rất nhiều quà lưu niệm |
| 미국 드라마에 대한 내용 들이 나와 있었어 | có rất nhiều nội dung liên quan đến phim truyền hình | có nội dung liên quan đến phim truyền hình mỹ |

Một số biểu đồ biểu thị đánh giá của mô hình đánh giá trên tập train và tập validation của 2 mô hình seq2seq-1 và transformer-2.



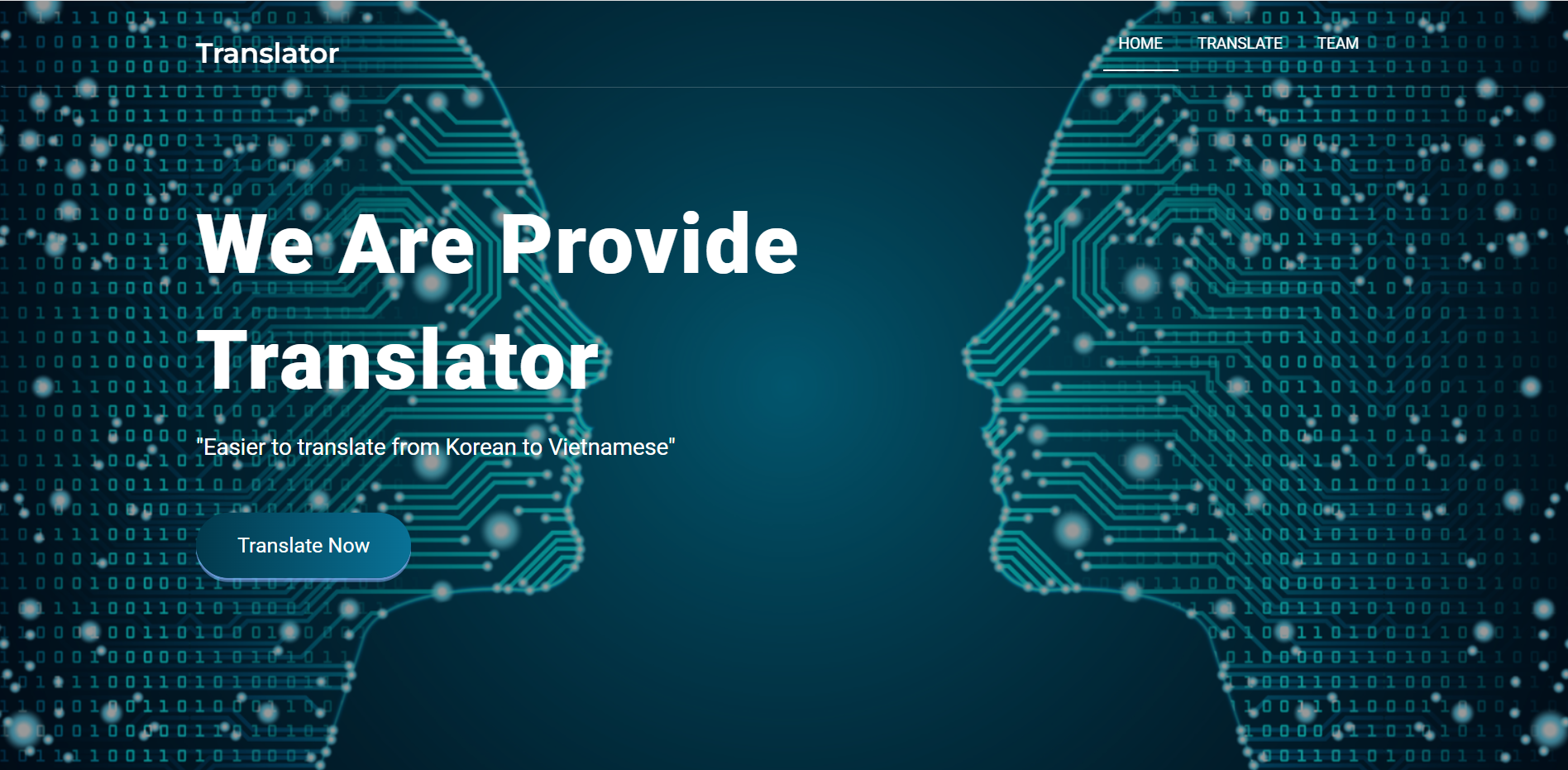
Biểu đồ đánh giá trong mô hình Transformer-1.



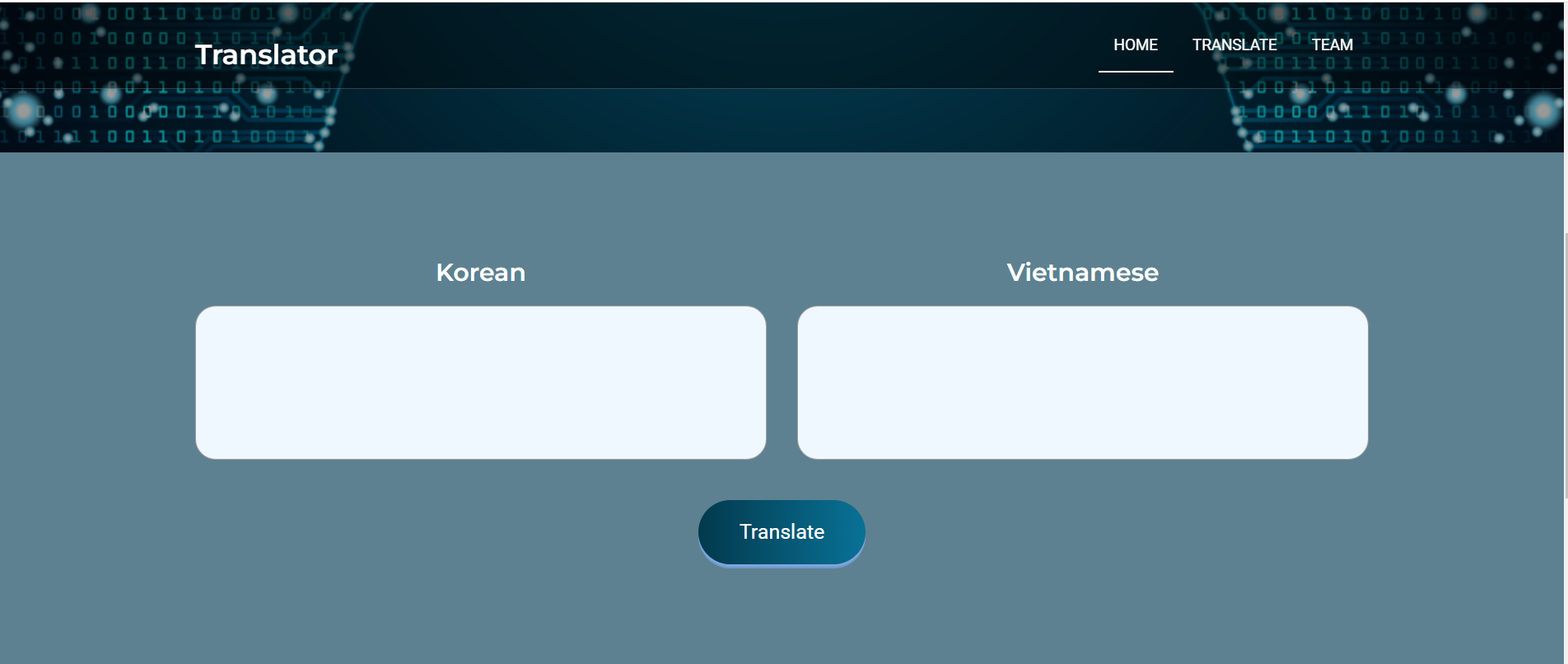
# **CHƯƠNG 6. TRIỂN KHAI BÀI TOÁN LÊN WEB**

## **1.Triển khai website:**

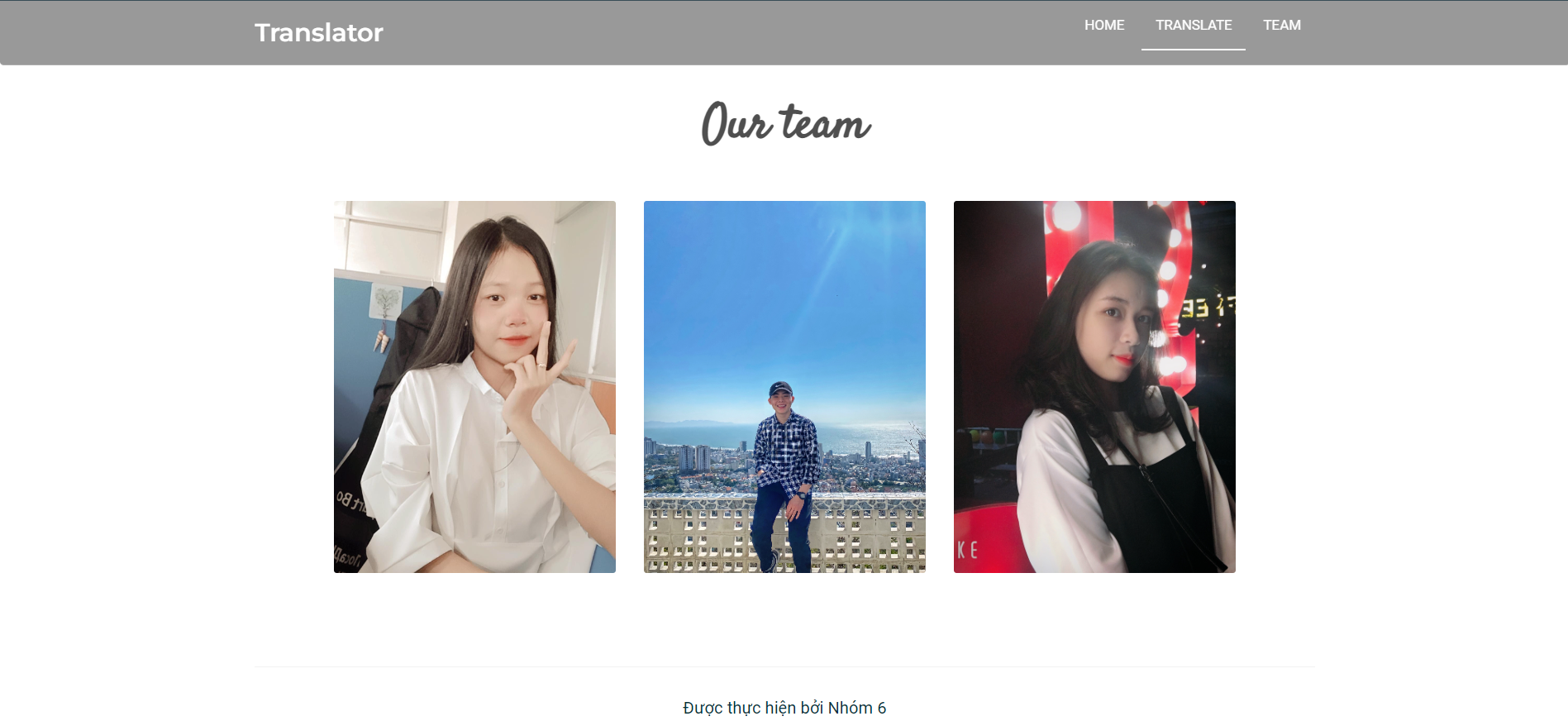
* Giao diện:
  + Sử dụng html, css để thiết kế giao diện
  + Giao diện gồm 3 phần chính:
    - Trang chủ



* + - Trang dịch



* + - Trang giới thiệu thành viên

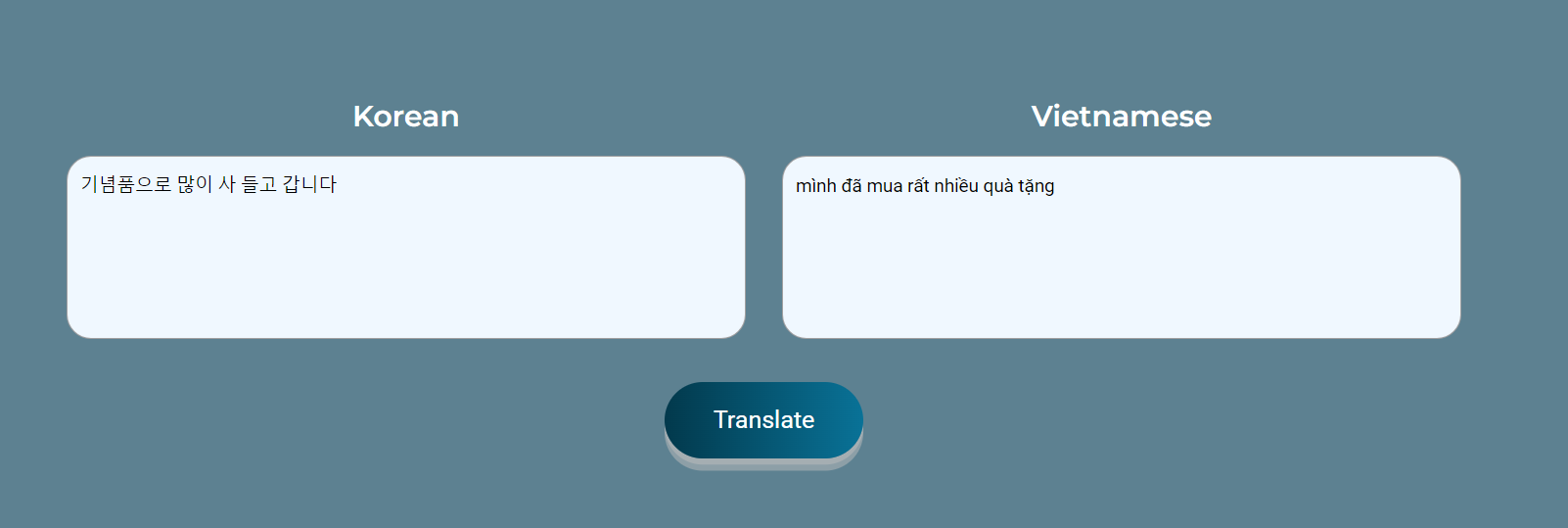


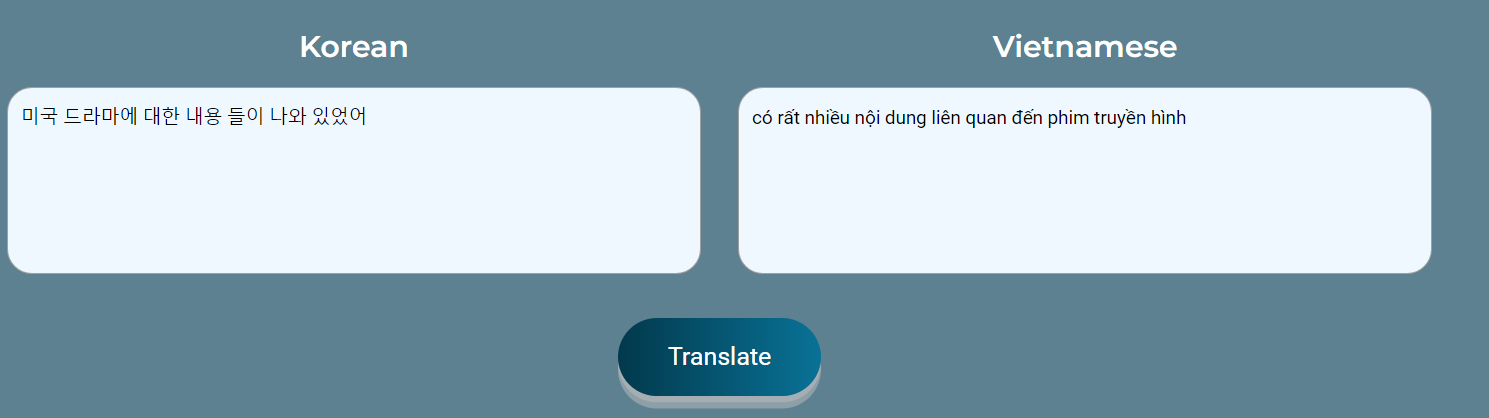
* Tích hợp model vào web
  + Model
  + Tập từ điển
* AWS
  + Tạo EC2 ( Phiên bản Ubuntu )
  + Truy cập vào server Ubuntu thông qua SSH
  + Cập nhật server
  + Tải source code của web lên server
  + Chạy project trên server
* Link web: [dịch máy hàn - việt](http://52.221.186.74/)

## **2.Một số kết quả dịch**

Một số kết quả dịch được thực hiện trên web với các trường hợp minh họa như trên bảng “*Một số kết quả dịch”*







# **CHƯƠNG 7. KẾT LUẬN**

## **1.Kết luận**

Sau một thời gian nghiên cứu và thực hiện nhóm đã hoàn thành đề tài xây dựng mô hình dịch máy từ tiếng Hàn sang tiếng Việt. Đề tài đã đạt được một số kết quả chính như sau:

* Tìm hiểu các kiến thức có liên quan đến mô hình dịch máy
  + Tổng quan về mô hình dịch máy.
  + Tìm hiểu các mô hình ứng dụng cho dịch máy Seq2Seq, Transformer.
* Tạo dữ liệu và làm sạch dữ liệu.
* Chuẩn bị dữ liệu cho quá trình xây dựng mô hình.
* Xây dựng mô hình Seq2Seq và Transformer.
* So sánh và đánh giá từ kết quả của các mô hình đã xây dựng.

## **2.Hướng phát triển**

Từ kết quả của đề tài, nhóm sẽ tiếp tục bổ sung dữ liệu vào hệ thống, tìm hiểu sâu hơn về đặc trưng của ngôn ngữ nhằm nâng cao chất lượng dữ liệu, đồng thời nhóm sẽ tập trung phát triển và tối ưu mô hình Transformer để nâng cao chất lượng dịch thuật của mô hình.