TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÔNG NGHỆ PHẦN MỀM

TRƯƠNG PHẠM NHẬT TIẾN – NGUYỄN MINH TRÍ

XÂY DỰNG MÔ HÌNH DỊCH MÁY TỪ TIẾNG ANH SANG TIẾNG VIỆT

KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN CNTT

TP. HCM, NĂM 2020

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN KHOA CÔNG NGHỆ THÔNG TIN BỘ MÔN CÔNG NGHỆ PHẦN MỀM

TRƯƠNG PHẠM NHẬT TIẾN – 1612689
NGUYỄN MINH TRÍ – 1612726

XÂY DỰNG MÔ HÌNH DỊCH MÁY TỪ TIẾNG ANH SANG TIẾNG VIỆT

KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN CNTT

GIÁO VIÊN HƯỚNG DẪN

TS. NGÔ HUY BIÊN

KHÓA 2016 - 2020

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

……

……

……

……

……

……

……

……

……

……

……

TpHCM, ngày ….. tháng ….. năm …..

Giáo viên hướng dẫn

[Kí tên và ghi rõ họ tên]

NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

……..……

……..……

……..……

……..……

……..……

……..……

……..……

……..……

……..……

……..……

……..……

……..……

……..……

……..……

……..……

……..……

……..……

……..……

……..……

……..……

Khóa luận đáp ứng yêu cầu của Khóa luận cử nhân CNTT.

TpHCM, ngày ….. tháng …… năm ……

Giáo viên phản biện

[Ký tên và ghi rõ họ tên]

LỜI CẢM ƠN

Tri ân thầy - Tiến sĩ Ngô Huy Biên, người đã luôn trực tiếp hướng dẫn, định hướng cho hướng đi của luận văn, góp ý giúp đỡ nhiệt tình chúng em trong các vấn đề về kiến thức, nội dung, cách thức trình bày đồng thời luôn tạo điều kiện thoải mái nhất để chúng em có thể hoàn thành khóa luận, chúng em xin gửi đến thầy lời cảm ơn chân thành và sâu sắc nhất.

Chúng em xin gửi lời cảm ơn đến quý Thầy Cô trong khoa công nghệ thông tin của trường đại học Khoa Học Tự Nhiên đã tận tình giảng dạy nâng bước chúng em trong suốt gần 4 năm học vừa qua. Em xin chân thành cảm ơn Khoa Công Nghệ Thông Tin, trường Đại học Khoa Học Tự Nhiên, Đại học Quốc gia Tp. Hồ Chí Minh đã tạo điều kiện thuận lợi cho chúng em trong quá trình học tập và thực hiện đề tài tốt nghiệp. Đồng thời chúng em cũng không quên gửi những lời cảm ơn chân thành đến những người thân trong gia đình và bạn bè đã giúp đỡ chúng em trong quá trình thực hiện luận văn.

Do trình độ nghiên cứu và thời gian có hạn, chúng em đã cố gắng hết sức nhưng chắc chắn không tránh khỏi những thiếu sót và hạn chế. Rất mong nhận được sự góp ý và chỉ dẫn của quý Thầy Cô.

Cuối cùng, chúng em xin trân trọng cảm ơn và chúc sức khỏe quý Thầy Cô!

TpHCM, ngày . . . tháng . . . năm 2020

Sinh viên

ĐỀ CƯƠNG CHI TIẾT

Tên Đề Tài: Xây dựng mô hình dịch máy từ tiếng Anh sang tiếng Việt

Giáo viên hướng dẫn: TS. Ngô Huy Biên

Thời gian thực hiện: 20/11/2019 – 06/2020

Sinh viên thực hiện:

- Trương Phạm Nhật Tiến -1612689
- Nguyễn Minh Trí -1612726

Loại đề tài: Nghiên cứu và ứng dụng

Nội Dung Đề Tài:

1. Trình bày lý thuyết nền tảng và giải pháp để xử lý việc dịch một văn bản từ tiếng Anh sang tiếng Việt.
2. Xây dựng, thu thập dữ liệu và đào tạo mô hình để dịch một văn bản từ tiếng Anh sang tiếng Việt.
3. Xây dựng một trang web demo việc sử dụng mô hình để dịch một văn bản từ tiếng Anh sang tiếng Việt.

Kế Hoạch Thực Hiện:

Thời gian thực hiện	Công việc thực hiện	Người thực hiện
20/11/2019 – 25/11/2019	• Nhận đề tài • Xây dựng bản kế hoạch sơ bộ cho các công việc cần thực hiện	Tiến, Trí

26/11/2019 – 02/12/2019	• Tìm hiểu và phân tích các yêu cầu về kiến thức cho đề tài • Khảo sát và dùng thử các hệ thống cung cấp dịch vụ mẫu có sẵn trên thị trường	Tiến , Trí
03/12/2019 – 05/12/2019	• Thống nhất nội dung chính của ứng dụng demo việc sử dụng API	Tiến, Trí
06/12/2019 – 15/02/2019	• Tìm hiểu lý thuyết nền tảng trong máy học • Tìm hiểu lý thuyết nền tảng trong dịch máy	Tiến, Trí
20/12/2019 – 26/12/2019	• Executive summary • Project vision	Tiến, Trí
27/12/2019 – 02/01/2020	• Tạo Ec2 • Tạo trello	Tiến, Trí
03/01/2020 – 10/01/2020	• Viết release plan • Product backlog • Rise management	Tiến, Trí

11/01/2020 – 01/02/2020	• Tìm hiểu về các thư việc Scikit-Learn, Tensorflow, Keras	Tiến, Trí
02/02/2020 – 15/02/2020	• Tìm hiểu các model và kiến trúc, chạy thử các ví dụ để đánh giá	Tiến, Trí
16/02/2020 - 22/02/2020	• Chạy thử mô hình dịch máy từ tiếng Anh sang các ngôn ngữ khác	Tiến, Trí
23/02/2020 - 29/02/2020	• Thu thập dữ liệu ngôn ngữ • Viết chương 1 luận văn	Tiến, Trí
01/03/2020 - 06/03/2020	• Chỉnh sửa dữ liệu ngôn ngữ • Tìm hiểu và xây dựng mô hình dịch máy từ tiếng Anh sang tiếng Việt • Chỉnh sửa chương 1 luận văn	Tiến, Trí
07/03/2020 - 15/03/2020	• Huấn luyện mô hình • Viết chương 2 luận văn	Tiến, Trí
16/03/2020 - 21/03/2020	• Cải tiến mô hình	Tiến, Trí

	• Chỉnh sửa chương 2 luận vă	
22/03/2020 - 30/03/2020	• Viết chương 3 luận văn • Chỉnh sửa chương 3 luận vă	Tiến, Trí
01/04/2020 - 07/04/2020	• Xây dựng và triển khai hệ thống cung cấp dịch vụ web (API) • Viết chương 4 luận văn	Tiến, Trí
08/04/2020 - 15/04/2020	• Xây dựng ứng dụng demo việc sử dụng API trên nền tảng web • Chỉnh sửa chương 4 luận văn	Tiến, Trí
16/04/2020 - 21/04/2020	• Viết chương 5 luận văn • Chỉnh sửa chương 5 luận văn	Tiến, Trí
21/04/2020 - 30/04/2020	• Hoàn thành luận văn • Chỉnh sửa và cải thiện hiệu năng ứng dụng demo	Tiến, Trí

01/05/2020 - 30/05/2020	• Nâng cấp mô hình hoàn thiện hơn • Cải thiện hiệu năng hệ thống cung cấp dịch vụ web(API)	Tiến, Trí
03/06/2020 - 19/06/2020	• Hoàn chỉnh cuốn luận văn	Tiến, Trí
20/06/2020 - 30/06/2020	• Hoàn chỉnh slide trình bày • Hoàn chỉnh sản phẩm khoá luận	Tiến, Trí
01/07/2020 – 15/07/2020	• Cải tiến mô hình • Huấn luyện mô hình ngôn ngữ • Chỉnh sửa lại luận văn	Tiến, Trí
16/07/2020 – 30/07/2020	• Hoàn tất mô hình • Hoàn tất luận văn • Hoàn chỉnh sản phẩm khoá luận	Tiến, Trí

Xác nhận của GVHD	**Ngày……tháng……năm……** **SV Thực hiện**

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU LUẬN VĂN

1.1 GIỚI THIỆU ĐỀ TÀI

Trong cuộc sống hiện nay có rất nhiều thiết bị công nghệ được áp dụng rộng rãi và phục vụ cho nhiều mục đích khác nhau trong cuộc sống và hiện nay đang càng được quan tâm để mở rộng. Một trong những vấn đề đã và đang được phát triển đó chính là dịch máy. Trên thế giới có rất nhiều ngôn ngữ nói, viết khác nhau trên thế giới và sự khác biệt về ngôn ngữ là một trở ngại lớn trong hầu hết các mặt của đời sống. Sự khác biệt đó làm cho con người khó tiếp thu được các kiến thức đến từ các ngôn ngữ khác và nó cũng làm cho sự toàn cầu hoá bị tác động chậm lại.

Trên thế giới có khoảng bảy ngàn ngôn ngữ khác nhau, mỗi ngôn ngữ lại bị phân hoá theo từng vùng miền, quốc gia mà cách viết cũng khác nhau dẫn đến sự hình thành phương ngữ lẫn các biến thể ngôn ngữ dẫn đến các dịch vụ dịch máy vì thế cũng bị hạn chế đi rất nhiều. Ở các nước phát triển trên thế giới đã có nhiều dịch vụ dịch máy được phát triển. Tuy nhiên, các dịch vụ này phục vụ chủ yếu cho thị trường của họ nên những ngôn ngữ khác thì hạn chế hơn.

Vấn đề đặt ra khi sử dụng dịch vụ này, nhà phát triển ứng dụng chỉ quan tâm về độ hiệu quả của dịch vụ cho cặp ngôn ngữ đang sử dụng mang lại lợi ích trực tiếp cho họ.
Vì thế, con người mong muốn có thêm những công cụ hỗ trợ họ một phần giúp họ có thể tạm thời bỏ qua sự khác biệt ngôn ngữ khác nhau để hoàn thành một mục đích của họ. Từ đó các dịch vụ về ngôn ngữ được ra đời và phát triển một cách nhanh chóng. Trong những năm gần đây, dịch máy đóng vai trò như một công cụ hỗ trợ con người có thể bỏ qua rào cản ngôn ngữ để cập nhật thông tin từ nhiều nguồn khác nhau một cách dễ dàng. Nó đã giúp con người rất nhiều trong các công việc học tập và làm việc một cách hiệu quả hơn.

Nếu như ngày xưa để hiểu được một câu tiếng Anh bạn có thể cần phải tra trong từ điển dày cộm tốn thời gian và không hiệu quả. Thế nhưng ngày nay với các phần mềm dịch tiếng Anh sang tiếng Việt trực tuyến, bạn sẽ dễ dàng tra cứu ý nghĩa của các từ, câu, đoạn văn bản mình cần trong thời gian ngắn gần như là ngay lập tức.
Với sự phát triển kinh tế và toàn cầu hoá, các dịch vụ dịch cũng như các ứng dụng sử dụng dịch vụ này không chỉ hấp dẫn người sử dụng mà còn thu hút giới công nghệ trên thế giới. Một dịch vụ dịch máy thường sở hữu những đặc điểm nổi bật như:

- Sử dụng hoàn toàn miễn phí, kho từ phong phú.
- Phần mềm hỗ trợ dịch với độ chính xác cao.
- Sử dụng phần mềm một cách dễ dàng.
- Đôi khi không cần kết nối mạng, người dùng vẫn có thể sử dụng các chức năng phiên dịch từ của những ứng dụng này.
- Trợ giúp tự nhiên nhất cho người sử dụng.
- Tiềm năng kinh doanh lợi nhuận cũng rất to lớn.

Từ đó, dịch máy và các ứng dụng sử dụng dịch máy đã đem lại một lợi ích khổng lồ trong cuộc sống của con người.

1.2 LÝ DO LỰA CHỌN ĐỀ TÀI

Nhằm mục đích phát triển kiến thức của bản thân, nhóm sinh viên quyết định lựa chọn đề tài “Xây dựng mô hình dịch máy từ tiếng Anh sang tiếng Việt” để tìm hiểu thêm các kiến thức, có sản phẩm mang ý nghĩa thực tế và có tiềm năng trong tương lai. Đề tài cũng cung cấp cho nhóm sinh viên cơ hội học tập các kiến thức mới, lĩnh vực mới. Nhóm sinh viên cũng mong tạo ra một nền tảng để phát triển tiếng Việt sau này và giảm bớt sự phụ thuộc hiện nay vào các công ty nước ngoài.

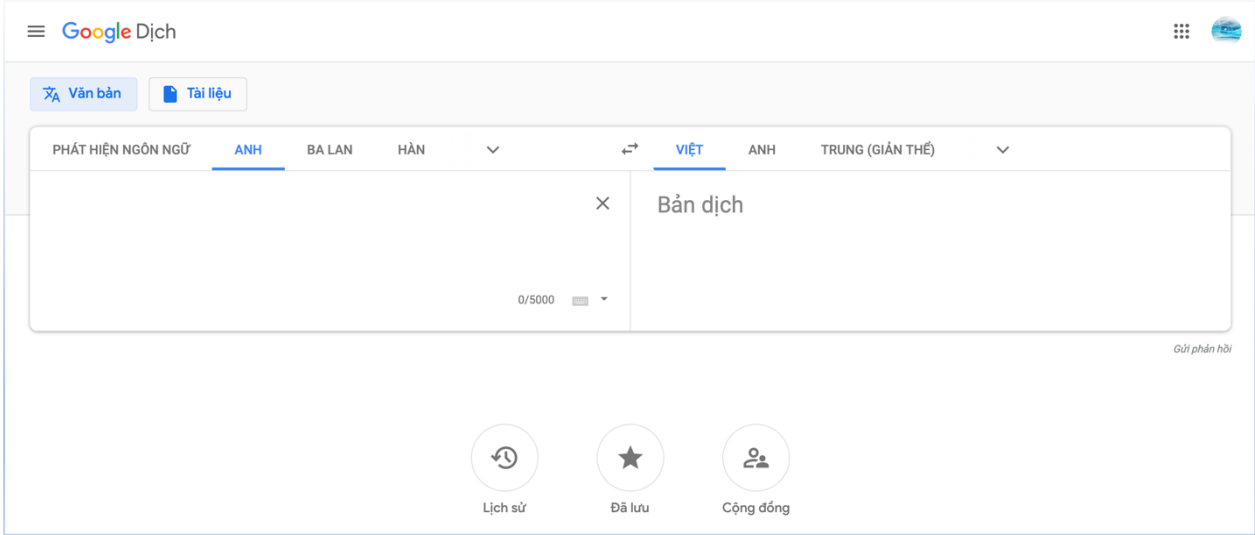
Ngoài ra việc lựa chọn đề tài này giúp nhóm sinh viên có thêm các kiến thức về các thư viện, các mã nguồn mở là những nguồn tài liệu quý giá đối với nhóm sinh viên. Với các

lợi ích kể trên và sau khi thực hiện luận, nhóm sinh viên sẽ có thêm hiểu biết về các quy trình phát triển dự án thực, không những thế nhóm sinh viên còn được học hỏi các kiến thức nền tảng và chuyên sâu liên quan đến dịch máy. Chính những điều đó sẽ là nền tảng giúp nhóm sinh viên phát triển hơn trong con đường học tập cũng như trong công việc của nhóm sinh viên.

Như đã trình bày ở trên, các ứng dụng sử dụng dịch vụ dịch máy ngày càng thu hút sự đầu tư và các nhà phát triển phần mềm lẫn người sử dụng phần mềm. Việt Nam đang đi trên con đường toàn cầu hoá nên thị trường Việt Nam là một thị trường đầy tiềm năng. Nhu cầu học tiếng Anh của các học sinh, sinh viên cũng như người dân là rất lớn. Trên thị trường hiện nay có khá nhiều các công cụ miễn phí cũng như thu phí nhưng những ứng dụng này cũng có một số ưu điểm cũng như khuyết điểm của nó.
Với các lý do trên, nhóm sinh viên quyết định chọn đề tài "Xây dựng mô hình dịch máy từ tiếng Anh sang tiếng Việt" để tạo do một dịch vụ miễn phí và độ chính xác tạm chấp nhận được để phục vụ cho cộng đồng.

Google Translate

Đây là một sản phẩm rất hữu ích của Google và là phần mềm dịch tiếng Anh sang tiếng Việt đầu tiên được rất nhiều người sử dụng cho các nhu cầu như là học tập.



Hình 1.1: Ứng dụng Google Translate

Ưu điểm:

- Google Translate có giao diện dễ nhìn, dễ sử dụng, đặc biệt có độ chính xác khá cao.
- Có thể sử dụng phầm mềm này trên máy tính, điện thoại hay cả máy tính bảng.
- Miễn phí và dịch được nhiều ngôn ngữ khác nhau trên thế giới.

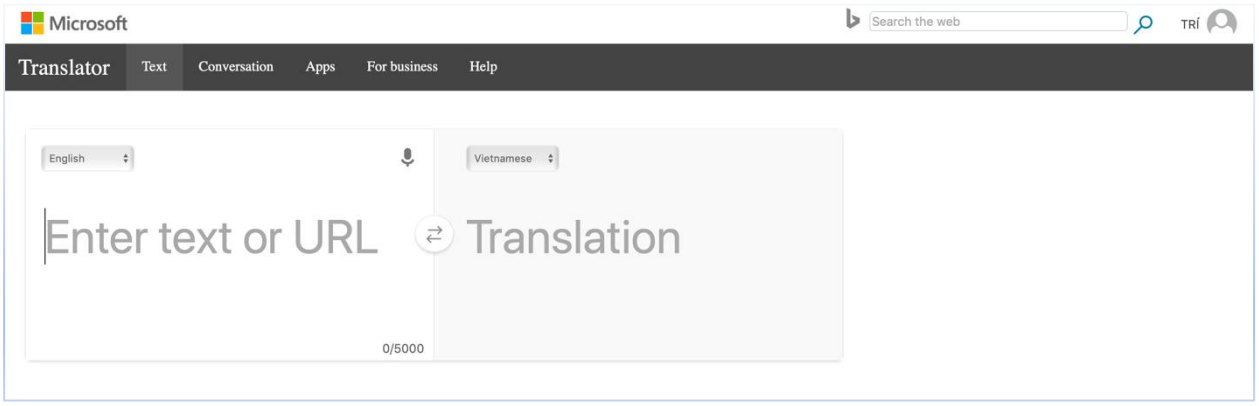
Nhược điểm:

- Để sử dụng phần mềm này cần phải có kết nối mạng cho thiết bị hoặc phải tải bộ dữ liệu để dùng khi không có mạng.
- Chỉ hiển thị giới hạn 5000 từ mỗi lần và cắt đoạn chưa hợp lí.

Link tham khảo: https://translate.google.com

Microsoft Translator

Nối tiếp Google, Microsoft cũng như cho ra đời phần mềm dịch trực tuyến của riêng mình.



Hình 1.2:Ứng dụng Microsoft Transtator

Ưu điểm:

- Giao diện dễ nhìn, độ chính xác cao.
- Có thể sử dụng trên các thiết bị khác nhau như: máy tính, điện thoại.
- Sử dụng miễn phí và dịch được nhiều ngôn ngữ..

Nhược điểm:

- Cần có kết nối mạng cho thiết bị.
- Chỉ giới hạn 5000 từ và không tự cắt đoạn văn bản để dịch tiếp.

- Giao diện không cân đối.

Link tham khảo: https://www.bing.com/translator

Cambridge Dictionary Translate

Phần mềm này được xuất bản bởi Đại học Cambridge.

Hình 1.3: Ứng dụng Cambride Dictionary Translate

Ưu điểm:

- Dịch được 23 ngôn ngữ.
- Có thể tự động phát hiện ngôn ngữ.
- Kết quả có độ chính xác cao.

Nhược điểm:

- Miễn phí tối đa 160 từ trong một lần dịch và 2000 từ trên một ngày.
- Cần phải qua một bước trung gian là ấn nút "Dịch" để thực hiện dịch.
- Cần có kết nối mạng có thiết bị.

Link tham khảo: https://dictionary.cambridge.org/vi/translate

1.3 HƯỚNG PHÁT TRIỂN CỦA LUẬN VĂN

Trong công cuộc phát triển dịch máy, có rất nhiều phương pháp đã được nghiên cứu và áp dụng đem lại thành công nhất định. Trong sự phát triển của dịch máy, có cách tiếp cận chủ yếu là dịch chuyển đổi, lịch liên ngữ và dịch dựa trên dữ liệu. Trong đó, dịch máy thống kê, một trong những phương pháp theo cách tiếp cận dựa trên dữ liệu, hiện

đang là một hướng pháp triển đầy tiềm năng, thu hút được nhiều sự quan tâm của các nhà nghiên cứu.

Ưu điểm vượt trội của phương pháp dịch máy thống kế là thay vì xây dựng các quy luật, từ điển được chuyển đổi bằng tay, nó tự động thiết lập các quy luật, từ điển dựa trên kết quả thống kê có được từ kho ngữ liệu. Chính vì thế nên dịch máy thống kê có tính linh hoạt cao có thể áp dụng được cho bất kì cho một cặp ngôn ngữ ngẫu nhiên. Dịch máy thống kê hiện nay có 3 hướng tiếp cận chính đó là: dịch máy thống kê theo đơn vị từ, dựa trên đơn vị cụm từ và dựa trên cú pháp.

Tuy nhiên các phươn pháp này vẫn có những hạn chế do sự thiếu hụt về thông tin ngôn ngữ. Mô hình dịch thống kê vẫn chưa giải quyết được một số vấn đề còn sai sót như trật tự từ, khả năng chọn cụm từ phù hợp.

Dịch máy thần kinh (Neural machine translation: NMT) là một cách tiếp cận dịch máy sử dụng mạng nơ-ron nhân tạo lớn để dự đoán chuỗi từ được dịch, bằng cách mô hình hóa toàn bộ các câu văn trong một mạng nơ-ron nhân tạo duy nhất.
Dịch máy thần kinh yêu cầu bộ nhớ ít hơn so với các mô hình dịch máy thống kê truyền thống (SMT). Hơn nữa, không giống như các hệ thống dịch thuật thông thường, tất cả các phần của mô hình dịch thuật nơ-ron được đào tạo cùng lúc với nhau (từ câu ngôn ngữ này sang câu ngôn ngữ khác) để tối đa hóa hiệu suất dịch thuật.
Luận văn sử dụng phương pháp dịch máy sử dụng phương pháp dịch máy thần kinh để tiếp cận hoàn toàn dựa trên ngữ liệu nên nó hoàn toàn độc lập với ngôn ngữ. Những tham số thống kê thu được từ việc huấn luyện trên ngữ liệu song ngữ sẽ được sử dụng cho các lần dịch sau.

Để hoàn thành luận văn, nhóm sinh viên tiến hành xây dựng một mô hình dịch máy từ tiếng Anh sang tiếng Việt và có ứng dụng thử nghiệm để đánh giá mô hình.

1.4 MỤC TIÊU CỦA LUẬN VĂN

Để hoàn thành tốt đề tài luận văn, bản luận văn và sản phẩm cuối cùng của nhóm sinh viên sẽ đảm bảo các mục tiêu sau đây:

- Trình bày lý thuyết nền tảng và giải pháp để xử lý việc dịch một văn bản từ tiếng Anh sang tiếng Việt.
- Xây dựng, thu thập dữ liệu và đào tạo mô hình để dịch một văn bản từ tiếng Anh sang tiếng Việt.
- Xây dựng trang web demo việc sử dụng mô hình để dịch một văn bản từ tiếng Anh sang tiếng Việt
- Viết 120 trang luận văn trình bày các nội dung liên quan theo đúng chuẩn nhà trường yêu cầu và có trích dẫn tài liệu liệu tham khảo một cách chi tiết và đầy đủ.

1.5 PHẠM VI ĐỀ TÀI

Sản phẩm được tạo ra hướng đến những người có mong muốn sử dụng các công cụ dịch.

- Sản phẩm chỉ cung cấp dịch từ tiếng Anh sang tiếng Việt.
- Sản phẩm yêu cầu kết nối internet để sử dụng.
- Sản phẩm luận văn sẽ được áp dụng và mở rộng các thư viện có sẵn cũng như các cặt thô để đáp ứng yêu cầu nhằm đặt được các mục tiêu đã đặt ra.

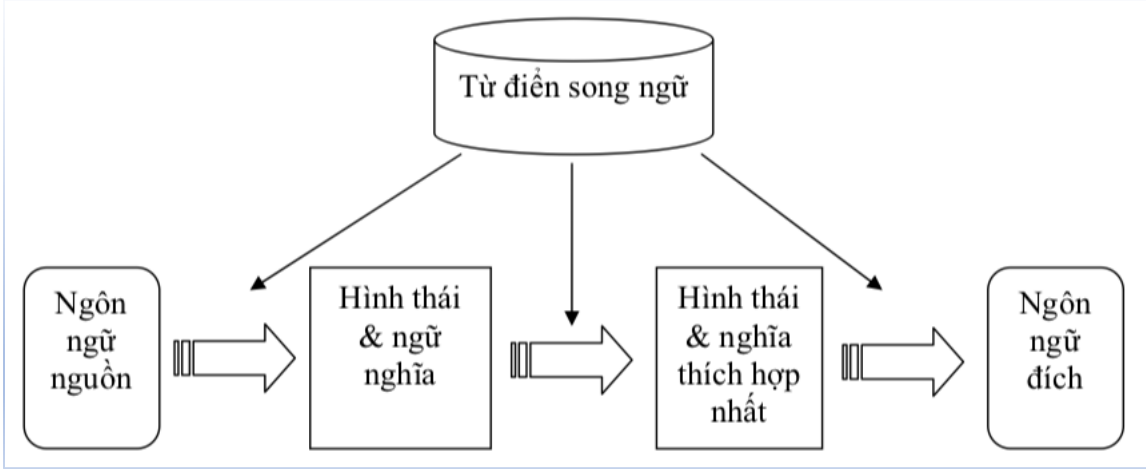
CHƯƠNG 2: LÝ THUYẾT NỀN TẢNG

2.1 LÝ THUYẾN NỀN TẢNG CỦA DỊCH MÁY:

Trong lịch sử phát triển của tác vụ dịch ngôn ngữ, đã có nhiều phương pháp được đưa ra và cũng mang lại những kết quả nhất định.

Dịch trực tiếp:

Tiếp cận dịch trực tiếp được áp dụng vào các chương trình dịch từ sớm nhất so với các hướng tiếp cận khác, đây là hướng tiếp cận được sử dụng và phát triển khá thành công trong hệ thống dịch Mark II (1964). Dịch trực tiếp là phương pháp phát triển cho cặp ngôn ngữ cụ thể, tiến trình dịch là một quá trình biến đổi từ ngôn ngữ nguồn sang ngôn ngữ đích dựa trên từ điển song ngữ và một số quy tắc từ vựng kết hợp với một số quy tắc xử lý ngữ pháp đơn giản. Sơ đồ hệ dịch trực tiếp được thể hiện ở mô hình dưới đây:



Hình 2a: Sơ đồ hệ dịch trực tiếp

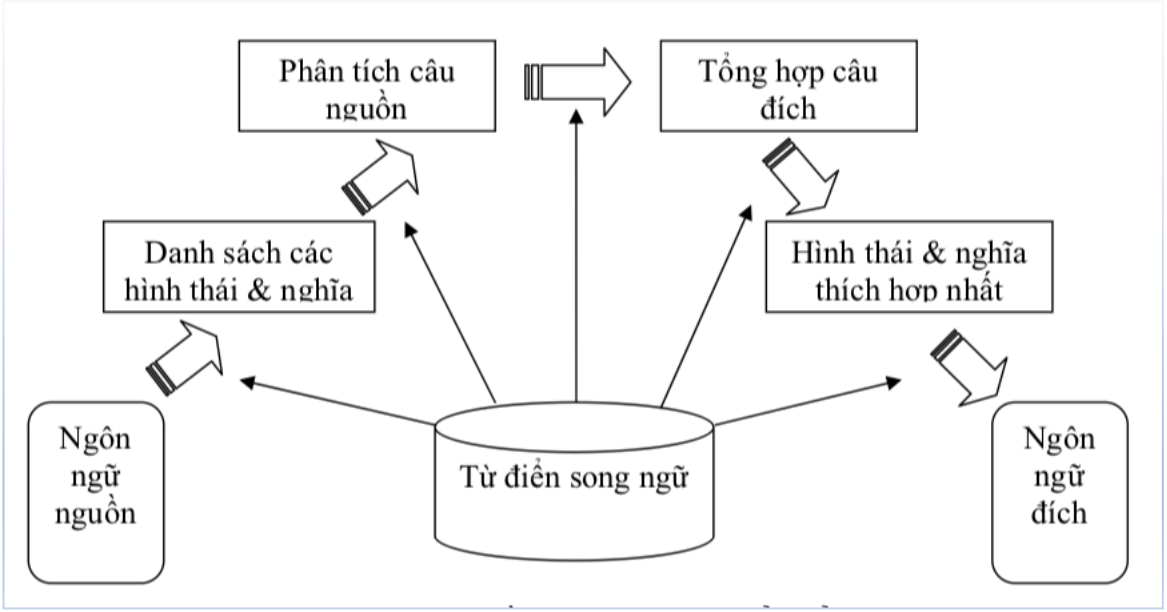
Các hệ dịch trực tiếp có ưu điểm là đơn giản và có tốc độ nhanh. Phương pháp rất thích hợp với việc dịch trong những lĩnh vực chuyên môn (không có nhiều nhập nhằng về ngữ nghĩa) và cho kết quả khá tốt khi áp dụng cho các cặp ngôn ngữ có nhiều điểm tương đồng về ngữ pháp và từ vựng (như tiếng Pháp và tiếng Anh,...). Với những cặp ngôn ngữ ít tương đồng hơn (tiếng Anh, Việt), hoặc với nguồn có không gian nghĩa mở (như các tác phẩm văn học), cách tiếp cận này tỏ ra thiếu hiệu quả.

Ví dụ ta cần dịch câu: “I am going to school” thì cách dịch trực tiếp sẽ phân tách câu thành các cụm từ “I am”, “go to”, “school” và thực hiện dịch bằng từ điển “I am” ứng với “Tôi”, “go to” ứng với “đang đi đến”, “school” ứng với “trường” và ghép lại thành câu “Tôi đang đi đến trường”.

Dịch chuyển đổi:

Dịch chuyển đổi cú pháp thực hiện phân tích cú pháp câu được nhập vào và sau đó áp dụng những luật ngôn ngữ và từ vựng (hay còn được gọi là những luật chuyển đổi) để ánh xạ thông tin văn phạm từ ngôn ngữ này sang ngôn ngữ khác. Do đó, không thể giải quyết các trường hợp nhập nhằng ngữ nghĩa của câu có cùng cấu trúc nhưng khác nghĩa nhau.

So với dịch trực tiếp, các hệ thống dịch chuyển đổi đi xa hơn các hệ dịch trực tiếp trong việc phân tích ngữ pháp (và ngữ nghĩa) của ngôn ngữ nguồn và ngôn ngữ đích. Đầu tiên, hệ dịch chuyển đổi tiến hành phân tích ngữ pháp trong ngôn ngữ nguồn, sau đó cố gắng chuyển đổi sang cấu trúc ngữ pháp tương đương ở ngôn ngữ đích và cuối cùng sinh câu đích từ cấu trúc ngữ pháp đã chuyển đổi. Sơ đồ hoạt động của hệ dịch chuyển đổi được thể hiện ở mô hình dưới đây:



Hình 2b: Sơ đồ hệ dịch chuyển

Ta có thể nhận thấy một vài đặc điểm của sơ đồ trên :

- Sự phụ thuộc nặng nề của quá trình dịch đối với ngôn ngữ nguồn. Cây cú pháp của ngôn ngữ nguồn quyết định cách thức biên dịch văn bản sang ngôn ngữ đích. Điều này dẫn đến sự suy biến của bước tổng hợp : ta không thấy có khối tổng hợp cú pháp của ngôn ngữ đích. Công đoạn phức tạp nhất chính là phân tích cú pháp. Kết quả là phải cần rất nhiều quy tắc dịch (cho những tình huống khác biệt giữa hai ngôn ngữ) kéo theo rất nhiều quy tắc phân tích văn phạm (có dạng tương tự nhau trên ngôn ngữ nguồn nhưng khác nhau về luật dịch sang ngôn ngữ đích)
- Dữ liệu chỉ sử dụng được cho dịch một chiều và cho một cặp ngôn ngữ. Để dịch ngược lại ta phải xây dựng lại toàn bộ hệ quy tắc và từ vựng.
  Ví dụ ta cần dịch câu "Old men like to drink tea in the afternoon". Dịch chuyển đổi đi sâu hơn vào phân tích cú pháp và ngữ nghĩa của câu. Danh từ sẽ là "Old men", "tea" - động từ là "like", "drink" – thời gian "the afternoon". Sau đó nó tổng hợp và đưa ra hình thái, ngữ nghĩa thích hợp nhất là "Những người đàn ông lớn tuổi thích uống trà vào buổi chiều" chẳng hạn.

Dịch máy thống kê:

Tiếp cận dịch máy dựa trên thống kê xuất hiện vào cuối những năm 1980, được đề xuất bởi trung tâm nghiên cứu IBM TJ Watson với hệ dịch máy Anh-Pháp Candide. Ý tưởng dịch máy bằng thống kê rất đơn giản và thuần túy toán học: Thay vì xây dựng các từ điển, các quy luật chuyển đổi bằng tay, hệ dịch này tự động xây dựng các từ điển, các quy luật dựa trên thống kê. Cách tiếp cận này không đòi hỏi sự phân tích sâu về ngôn ngữ, chúng thực hiện hoàn toàn tự động các quá trình phân tích, chuyển đổi, tạo câu dựa trên kết quả thống kê có được từ kho ngữ liệu (corpus).

Ý tưởng đằng sau dịch máy thống kê đến từ lý thuyết thông tin. Tài liệu được dịch theo phân bố xác suất $p(e|f)$ đó e là ngôn ngữ đích (ví dụ, Tiếng Việt) dịch từ f là ngôn ngữ nguồn (ví dụ, Tiếng Anh).

Các vấn đề của mô hình phân phối xác suất $p(e|f)$ đã được tiếp cận theo một số cách. Một cách tiếp cận trực quan là áp dụng định lý Bayes, đó là :

$$\mathrm{p(e|f)} = \frac{\mathrm{p(f|e)} * \mathrm{p(e)}}{\mathrm{p(f)}}$$

trong đó $p(f|e)$ là xác suất để chuỗi nguồn f là bản dịch của chuỗi đích e, xác suất này gọi là mô hình dịch, và $p(e)$ là xác suất chuỗi e thực sự xuất hiện trong ngôn ngữ đích, xác suất này gọi là mô hình ngôn ngữ. Phân tích này giúp tách các vấn đề thành hai bài toán con. Bản dịch tốt nhất được tìm bằng cách chọn ra bản có xác suất cao nhất:

$$\tilde{\mathrm{e}} = \underset{\mathrm{e} \in \mathrm{e}^*}{\mathrm{argmax}}\, \mathrm{p(e|f)} = \underset{\mathrm{e} \in \mathrm{e}^*}{\mathrm{argmax}}\, \mathrm{p(f|e)} * \mathrm{p(e)}$$

Để áp dụng phương pháp này một cách đầy đủ, cần thực hiện việc tìm kiếm trên tất cả các chuỗi e^* của ngôn ngữ đích. Khối lượng tìm kiếm này rất lớn, và nhiệm vụ thực hiện tìm kiếm hiệu quả là công việc của một bộ giải mã dịch máy, sử dụng nhiều kỹ thuật để hạn chế không gian tìm kiếm nhưng vẫn giữ chất lượng dịch thuật chấp nhận được.

Phương pháp dịch dựa trên thống kê đòi hỏi phải có một tập dữ liệu cực lớn các câu tương đương giữa ngôn ngữ nguồn và ngôn ngữ đích để có thể ra kết quả thống kê chính xác, đây là trở ngại lớn cho các đề án dịch theo đuổi phương pháp này vì việc xây dựng kho ngữ liệu lớn như vậy đòi hỏi công sức, chi phí rất lớn nên khó áp dụng cho ngôn ngữ tiếng Việt.

Ví dụ khi dịch một câu “I am going to school”, ta có P(f) không đổi với mỗi câu f nên ta chỉ cần quan tâm đến P(e) là xác xuất câu đích được tính bởi mô hình ngôn ngữ và P(f|e) được ước lượng dựa trên tài liệu song ngữ từ đó áp dụng công thức

$\text{argmax}(e \in e^*)\, p(f|e) * p(e)$ để có thể tính ra với câu e là "Tôi đang đi đến trường" có xác xuất cao nhất và chọn nó làm câu đầu ra.

Ở chương 1, nhóm sinh viên đã đề xuất sử dụng phương pháp dịch máy thần kinh để tiếp cận yêu cầu ít ngữ liệu và độc lập với ngôn ngữ. Trong luận văn, nhóm sinh viên thực hiện xây dựng mô hình dịch máy Sequence to Sequence sử dụng Attention Mechanism (cơ chế chú ý) và kết hợp với mô hình ngôn ngữ để phát triển mô hình. Sau đây nhóm sinh viên sẽ trình bày các kiến thức liên quan được ghi lại trong quá trình thực hiện luận văn để xây dựng một mô hình dịch máy.

2.1.1 Định nghĩa:

Phần này trình bày một số khái niệm cốt lõi về dịch máy, các mô hình ngôn ngữ và phương pháp học theo đặc trưng trong xử lý ngôn ngữ tự nhiên (Neural Language Processing).

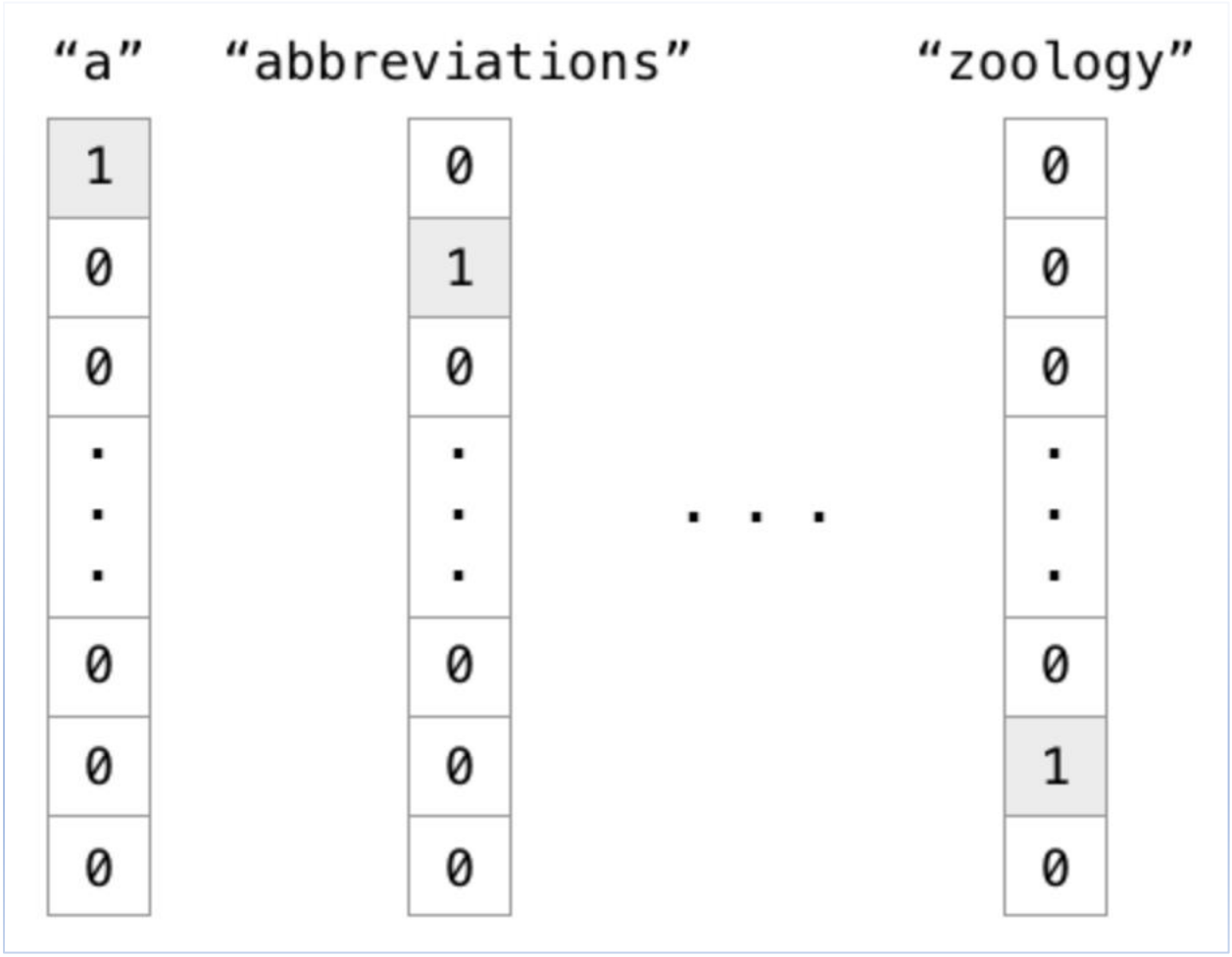
2.1.1.1 Định nghĩa dịch máy:

Dịch máy (machine translation) là một quá trình thay đổi văn bản từ ngôn ngữ này sang ngôn ngữ khác (gọi là ngôn ngữ đích) một cách tự động, không có sự can thiệp của con người trong quá trình dịch.

2.1.1.2 Word embeddings:

Xử lý đầu vào cho bài toán dịch máy là một bước rất quan trọng, các thuật toán, vì các kiến trúc Machine learning, Deep learning chúng chỉ có thể hiểu được đầu vào ở dạng là số nên cần chuyển đầu vào ở dạng văn bản sang dạng số để chúng có thể hiểu được. Nhưng nếu chỉ đơn giản biểu diễn từ bằng một con số có thể dẫn đến sai lệch mối quan hệ ngữ nghĩa giữa các từ. Ví dụ như nếu đánh dấu "mèo" là số 1 và "chó" là số 2, như vậy "mèo" + "mèo" = "chó".

Một kỹ thuật đơn giản được sử dụng để khác phục là One-hot vec-tơ, chúng chuyển các từ thành vec-tơ có số chiều bằng số từ của bộ từ vựng đầu vào, trong đó chỉ có duy nhất một phần tử bằng 1 (các phần tử khác bằng 0) tương ứng với vị trí từ đó trong bộ từ vựng. Tuy nhiên cách biểu diễn này là số chiều của vec-tơ lại rất lớn, ảnh hưởng đến quá trình xử lý và lưu trữ.



Hình 2.1: Hình mô tả cách mã hóa one-hot-vector

(Nguồn: Leonardo Barazza)

Một cách khác là xử dụng vec-tơ ngẫu nhiên, mỗi từ được biểu thị bằng một vec-tơ có giá trị các chiều là ngẫu nhiên, mỗi từ là một điểm trong không gian 3D, do đó làm giảm số chiều vec-tơ, tuy nhiên nó lại không biểu diễn quan hệ tương đồng giữa các từ.

Sử dụng Word embeddings được coi là cách tốt nhất để thể hiện các từ trong văn bản nó cũng gán mỗi từ với một vec-tơ nhưng các vec-tơ được tính toán để biểu diễn quan hệ tương đồng giữa các từ.

Word embeddings có 2 model nổi tiếng là Word2vec và Glove.

Ví dụ bảng sau đây mô tả tóm tắt cách biểu diễn từ của Word embedding:

	Man	Woman	King	Queen	Apple	Orange
Gender	-1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.7	0.69	0.03	-0.02
Food	0.04	0.01	0.02	0.01	0.95	0.97
Etc …						

Bảng 1: Mô tả biểu diễn word embedding

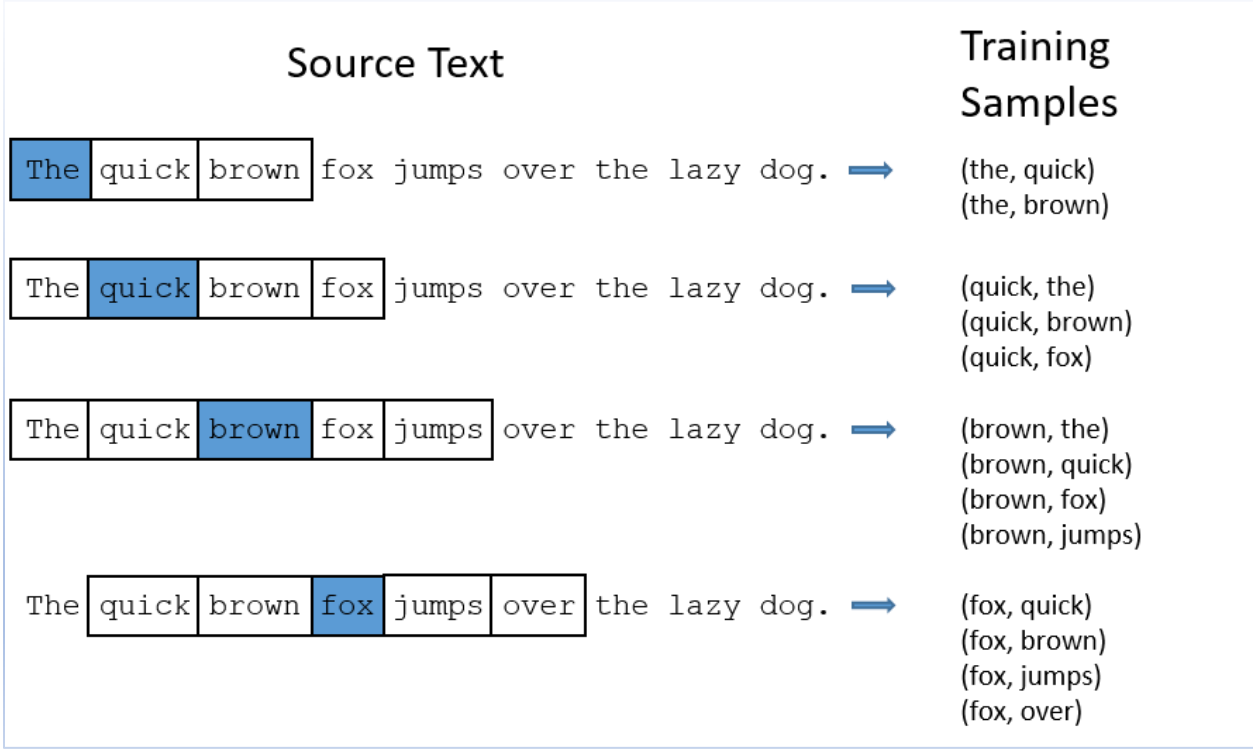
Từ bảng trên ta biểu diễn từ "Gender" thành một vec-tơ (-1, 1, -0.95, 0.97, 0.00, 0.01) thể hiện sự tương đồng với các từ. Các giá trị trong vec-tơ càng xa 0 thì sự phụ thuộc hay tương đồng càng mạnh. Như từ "Gender" tương đồng mạnh với từ "Man", "Woman", "King", "Queen" vì những từ này cũng chỉ rõ giới tính (gender). Ngược lại từ "Gender" không tương đồng với từ "Apple" và "Orange", giá trị trong vec-tơ ứng với hai từ này gần bằng 0 vì "Apple" và "Orange" là hai loại quả và không có liên quan gì đến "Gender".

2.1.1.2.1 Word2vec:

Word2vec là một model unsupervised learning nó dùng để thể hiện mỗi quan hệ giữa các từ, nó được kết hợp từ hai thuật toán Skip-gram và Continuous bag of words (CBOW).

2.1.1.2.2 Skip-gram:

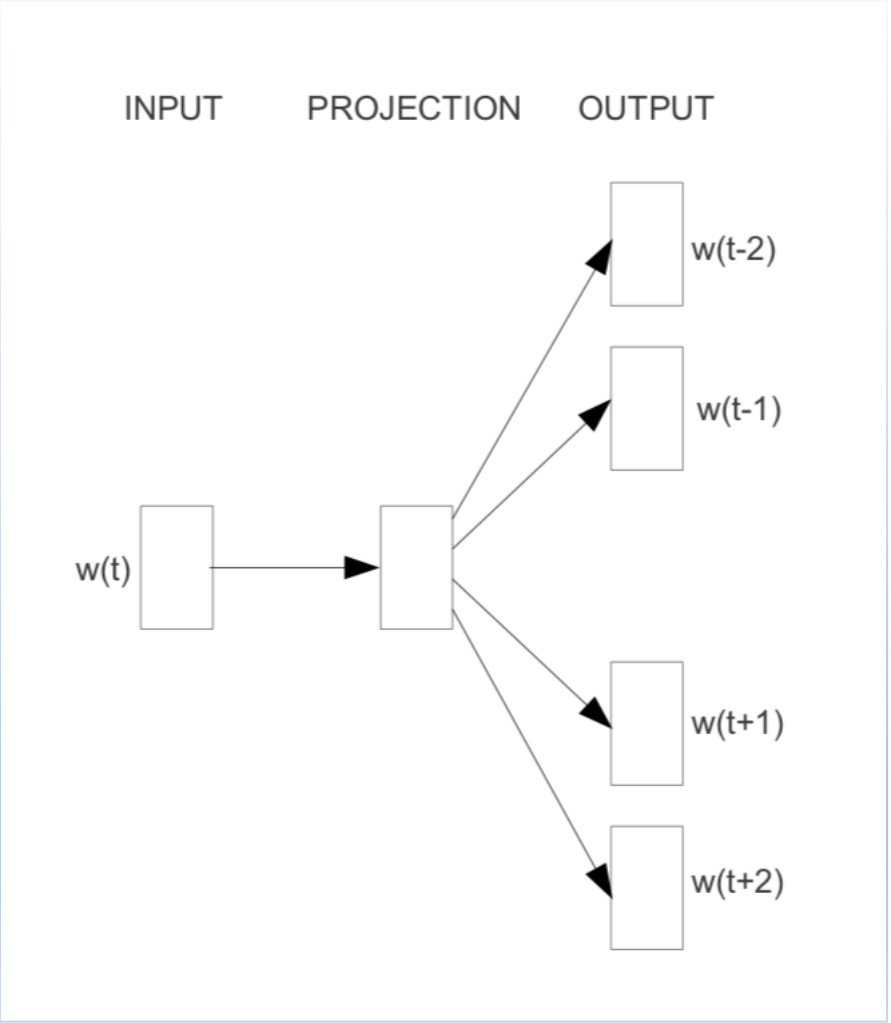
Ý tưởng chính của mô hình này là xác định các từ xung quanh từ mục tiêu trong một khoảng nhất định gọi là ‘window’.



Hình 2.2: Hình Mô Tả Trainning Với Window Bằng 2.

(Nguồn: Leonardo Barazza)

Đối với Skip-gram, đầu vào là từ đích, trong khi đầu ra là các từ xung quanh từ đích. Tất cả dữ liệu đầu vào và đầu ra có cùng kích thước được mã hóa bằng one-hot. Mạng chứa một lớp ẩn có kích thước bằng kích thước nhúng, nhỏ hơn vec-tơ đầu vào và đầu ra. Ở cuối lớp đầu ra, một hàm kích hoạt softmax được áp dụng sao cho mỗi phần tử của vec-tơ đầu ra mô tả khả năng một từ cụ thể sẽ xuất hiện trong ngữ cảnh.



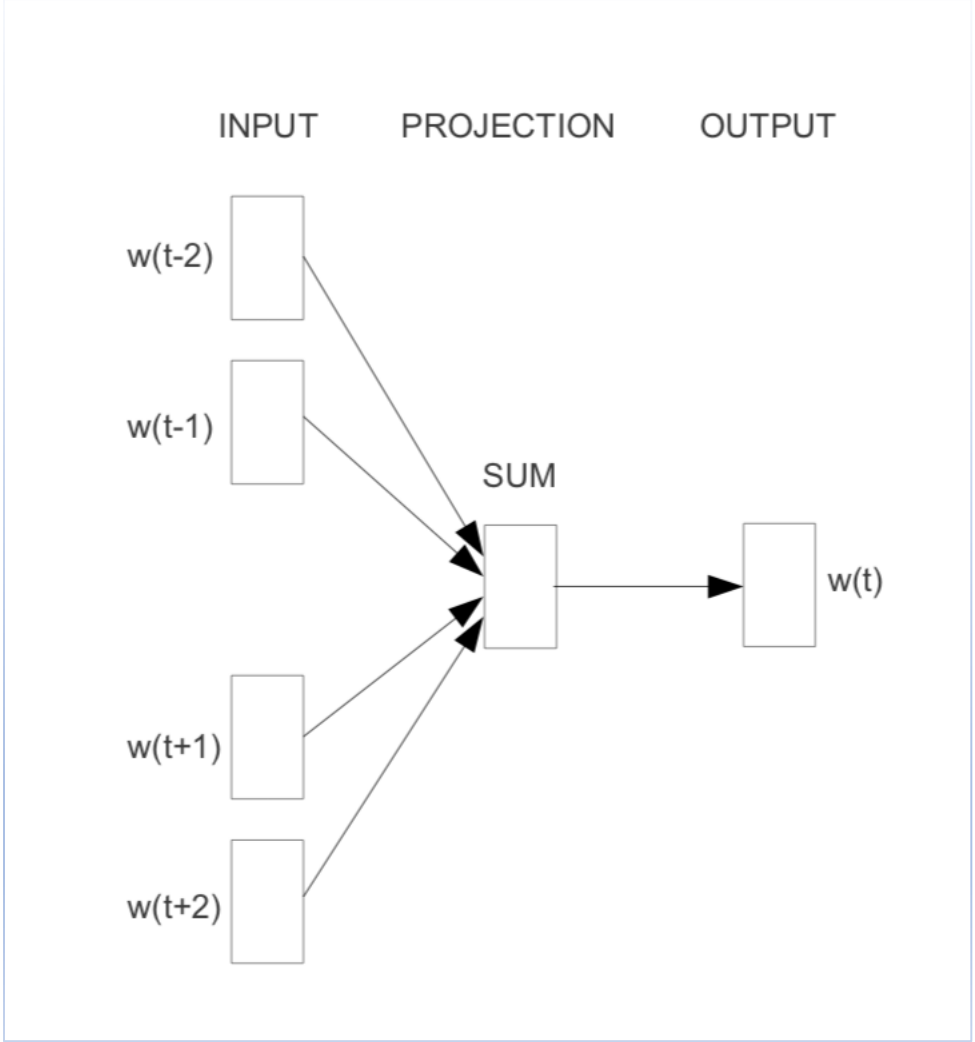
Hình 2.3: Hình mô tả cấu trúc mạng của Skip-gram (Nguồn : [11])

Với skip-gram, kích thước biểu diễn từ giảm từ kích thước bằng số từ trong bộ từ vựng xuống bằng chiều dài lớp ẩn. Hơn nữa các vec-tơ có ý nghĩa nhiều hơn về mặt mô tả mối quan hệ giữa các từ.

Ví dụ như ta có một dòng văn bản "I love you so much". Khi dùng một cửa sổ tìm kiếm có độ lớn là 3 ta thu được: {(i,you), love}, {(love,so), you},{(you,much), so}. Nhiệm vụ của skip-gram là khi cho 1 từ trung tâm ví dụ là “love” thì phải dự đoán được các từ xung quanh là “i”, “you”.

2.1.1.3 Continuous bag of words (CBOW)

Ngược lại với Skip-gram nó hoán đổi đầu vào và đầu ra, ý tưởng của thuật toán CBOW là đưa ra một bối cảnh và cho biết từ nào có khả năng xuất hiện nhiều nhất trong đó. Đối với CBOW, đầu vào là các từ xung quanh từ đích, trong khi đầu ra là từ đích.
Ví dụ với câu “I love you so much”. Nhiệm vụ của CBOW là khi cho các từ xung quanh như “i”,“love”, “so”, “much” thì phải dự đoán được từ “you” ở vị trí trung tâm.

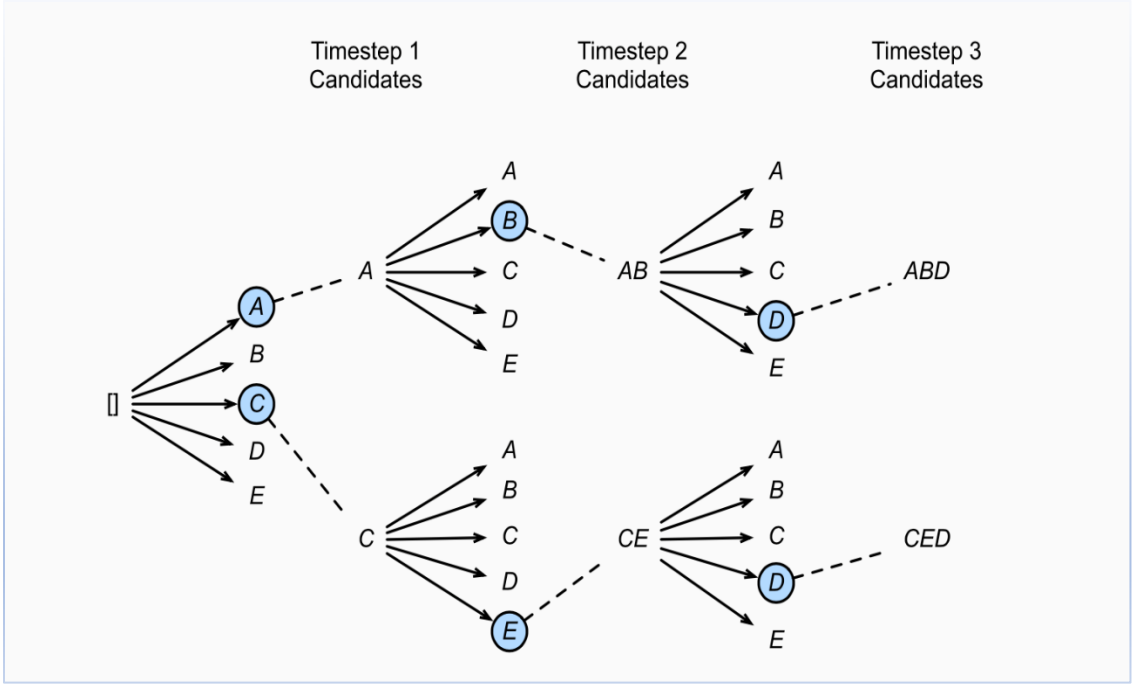


Hình 2.4: Mô Tả Mạng CBOW (Nguồn [11])

2.1.1.4 Thuật toán tìm kiếm tham lam và thuật toán tìm kiếm chùm tia (Greedy Search và Beam Search)

Trong dịch máy mô hình phải chọn câu văn phù hợp nhất thay vì để mô hình tạo ra từng từ một. Thông thường việc lựa chọn câu đầu ra được thực hiện bởi các thuật toán tìm kiếm: Greedy Search hoặc Beam Search.

Thuật toán Greedy Search chọn một ứng cử viên tốt nhất làm chuỗi đầu vào cho mỗi bước thời gian(ứng cử viên có xác suất cao nhất). Chọn chỉ một ứng cử viên tốt nhất có thể phù hợp với bước thời gian hiện tại, nhưng khi xây dựng câu đầy đủ, nó có thể là một lựa chọn tối ưu. Tuy nhiên thì xác suất cao nhất ở bước hiện tại chưa chắc sẽ cho ra xác xuất cao nhất ở bước tiếp theo, vậy nên thay vì chỉ giữ 1 kết quả có xác xuất cao nhất thuật toán sẽ giữ lại k kết quả có xác suất cao nhất và đó chính là Beam Search. Thuật toán Beam Search chọn nhiều lựa chọn thay thế cho chuỗi đầu vào tại mỗi bước thời gian dựa trên xác suất có điều kiện. Số lượng nhiều lựa chọn thay thế phụ thuộc vào một tham số gọi là Beam Width B. Ở mỗi bước thời gian, tìm kiếm chùm tia chọn B số lựa chọn thay thế tốt nhất với xác suất cao nhất là lựa chọn khả dĩ nhất cho bước thời gian (ví dụ với Beam Width B = 2 thì tại mỗi bước thời gian sẽ chọn ra 2 ứng cử viên có xác suất cao nhất làm đầu vào cho bước thời gian tiếp theo (t). Sau đó lại tiếp tục chọn 2 ứng cử viên ở bước thời gian tiếp theo (t+1) làm đầu vào ở bước thời gian t+2. Cứ như vậy cho đến cuối cùng ta sẽ thu được 3 kết quả và chọn ra kết quả từ đó). Thuật toán Beam Search nếu có Beam Width B = 1 thì nó trở thành thuật toán Greedy Search. Beam Width B = 10 thường được sử dụng và mang lại hiệu quả đủ tốt. Thuật toán tìm kiếm chùm tia được minh hoạ như hình 2.5.



Hình 2.5: Mô tả thuật toán tìm kiếm chùm tia (Beam search) hoạt động với beam-width = 2.

(Nguồn: https://d2l.ai/chapter_recurrent-modern/beam-search.html)

Ví dụ như hình 2.5, thuật toán sử dụng độ rộng là 2 nên ở bước đầu tiên thuật toán giữ lại hai từ có xác xuất cao nhất là ‘A’ và ‘C’. Đến bước thứ hai, thuật toán sẽ dự đoán từ tiếp theo với điều kiện từ đầu tiên là ‘A’ hoặc ‘C’ và chọn hai từ có xác xuất cao nhất là ‘B’ và ‘E’ tạo thành hai từ ghép “AB” và “CE”. Đến bước cuối cùng, thuật toán tính xác xuất hai từ có xác xuất cao nhất tiếp theo với điều kiện hai từ phía trước là “AB” hoặc “CE” và kết quả là “D”. Từ đó với độ rộng là 2 ta được hai câu với xác xuất cao nhất là “ABD” và “CED”.

Trong thực tế người ta thường dùng thuật toán tìm kiếm chùm tia với độ rộng là 10 cho bài bài toán dịch máy hoặc cao hơn với các bài toán khác, sử dụng độ rộng lớn cho kết quả càng cao nhưng sẽ tốn nhiều tài nguyên để tính toán hơn.

Ngoài ra, khi tìm ra được số câu ứng với độ rộng của thuật toán ta có thể lấy câu có xác xuất cao nhất hoặc kết hợp với mô hình ngôn ngữ để chọn ra câu có độ phù hợp nhất.

2.1.1.5 Bleu Score

BLEU là một thuật toán để đánh giá chất lượng văn bản đã được dịch bằng máy từ ngôn ngữ tự nhiên này sang ngôn ngữ tự nhiên khác. Chất lượng được coi là sự tương ứng giữa đầu ra của máy và của con người: "bản dịch máy càng gần với bản dịch chuyên nghiệp của con người thì càng tốt" - đây là ý tưởng trung tâm của BLEU.
Bilingual Evaluation Understudy Score hay ngắn gọn là BLEU score là một thang điểm được dùng phổ biến trong đánh giá Machine Translation. 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[1] giống nhau giữa câu dịch của mô hình (output) với các câu tham chiếu tương ứng (reference) có xét tới yếu tố 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.

BLEU được chia làm hai loại chính đó là: BLEU cá nhân và BLEU tích luỹ.
BLEU cá nhân được tính với từng n-gram riêng lẻ:

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n-gram \in C} Count_{clip}\ (n-gram)}{\sum_{C' \in \{Candidates\}} \sum_{n-gram' \in C'} Count(n-gram')}$$

BLEU tích luỹ được tính bằng cách tổng hợp các BLEU cá nhân riêng lẻ lại với nhau:

$$BLEU = \ BP * \ \exp\left(\sum_{n=1}^{N} w_n log p_n\right)$$

Trong đó:

- BP là brevity penalty dùng để phạt các câu có độ dài ngắn (vì khi các câu có độ dài ngắn thì điểm sẽ dễ cao hơn các câu dài).

$$BP = \begin{cases} 1 & if\ c > r \\ e^{(1-r/c)} & if\ c \leq r \end{cases}$$

Với c là độ dài câu được dịch từ hệ thống và r là độ dài câu tham chiếu.

- W_n là trọng số ứng với các loại BLEU cá nhân.
- P_n là điểm BLEU cá nhân.
- Log ở đây ứng với cơ số e.

Ví dụ:

Ta có câu tham chiếu và câu được dịch từ hệ thống dịch máy như sau:

- REF "The cat is on the mat" – câu tham chiếu.
- MT: "The cat the cat on the mat " – câu được dịch từ hệ thống.

Với BLEU-1 (1-gram) ta tính như sau:

	Count (MT)	Count-Clip (REF)
The	3	2
Cat	2	1
On	1	1
mat	1	1

BLEU-1 = 5/7.

Với BLEU-2 (2-gram):

	Count (MT)	Count-Clip (REF)
The cat	2	1
Cat the	1	0
Cat on	1	0
On the	1	1

The mat	1	1

BLEU-2 = 3/6

Với BLEU-3 (3-gram):

	Count (MT)	Count-Clip (REF)
The cat the	1	0
Cat the cat	1	0
The cat on	1	0
Cat on the	1	0
On the mat	1	1

BLEU-3 = 1/5

Với BLEU-4 (4-gram):

	Count (MT)	Count (REF)
The cat the cat	1	0
Cat the cat on	1	0
The cat on the	1	0
Cat on the mat	1	0

BLEU-4 = 0/4 = 0.

Phía trên là cách tính điểm BLEU cá nhân, sau đây nhóm sinh viên sinh trình bày tính điểm BLEU tích luỹ 3-gram và 4-gram.

Ta đã có $p_1 = 5/7$, $p_2 = 3/6$, $p_3 = 1/5$, $p_4 = 0$.

Điểm BLEU tích luỹ 3-gram (ta chỉ sử dụng BLEU-1, BLEU-2, BLEU-3):

- Vì độ dài của câu MT > độ dài câu REF nên ta có BP = 1.
- Wn = 1/N = 1/3 (vì tại đây ta sử dụng N = 3).

Từ đó, ta có điểm BLEU = $1 * e^{\frac{1}{3}(ln\frac{5}{7}+ln\frac{3}{6}+ln\frac{1}{5})}$ = 0.467.

Điểm BLEU tích luỹ 4-gram (ta sử dụng BLEU-1, BLEU-2, BLEU-3, BLEU-4):

- Vì độ dài của câu MT > độ dài câu REF nên ta có BP = 1.
- Wn = 1/N = 1/4 (vì tại đây ta sử dụng N = 4).

Từ đó, ta có điểm BLEU = $1 * e^{\frac{1}{4}(ln\frac{5}{7}+ln\frac{3}{6}+ln\frac{1}{5}+ln0)}$ = 0 (vì ln0 không xác định, nên ta gán cho nó bằng một số rất nhỏ để có thể tính toán).

2.1.2 Lý thuyết nền tảng mạng nơ-ron (Neural Network)

Mạng nơ-ron là một tập hợp các mô hình toán học được được xây dựng dựa trên tập hợp các nút, được kết nối với các hàm kích hoạt phi tuyến tính cùng các tham số có khả năng học. Mạng nơ-ron hiện là mô hình phổ biến nhất được sử dụng cho các ứng dụng máy học trong một loạt các lĩnh vực như thị giác máy tính, nhận dạng giọng nói, xử lý ngôn ngữ tự nhiên,…

2.1.2.1 Mô tả tế bào nơ-ron:

Một tế nào (nút) của mạng nơ-ron là một hàm của tập các trọng số tương ứng với các giá trị đầu vào (inputs) $\{x_0, .., x_N\}$.
Trong đó:

- w_i: trọng số của đầu vào x_i
- a: hàm kích hoạt (activation function)
- b: độ sai lệch (bias)

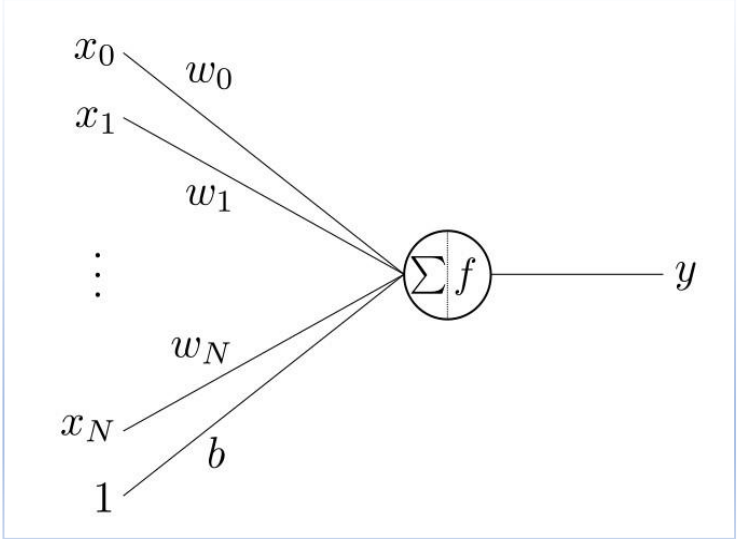
Ta sẽ sử dụng kí hiệu ma trận để làm đơn giản cách thể hiện, trong đó mỗi tế bào nơ-ron bao gồm một vec-tơ đầu vào $x = \{x_0, .., x_N\}$, một vec-tơ trọng số $w = \{w_0, .., w_N\}$ và một vec-tơ sai lệch b, khi đó đầu ra là:

$$y = a(w^T x + b)$$

Nếu hàm kích hoạt a là một biến thể của hàm Heaviside,

$$a(x) = \begin{cases} 1, & x \geq 0 \\ 0\ hoặc\ -1, & x < 0 \end{cases}$$

thì tế bào nơ-ron này được gọi là một perceptron, một bộ phân loại nhị phân đơn giản, là một trong những phương pháp học kết nối sớm nhất được phát minh bởi Rosenblatt.



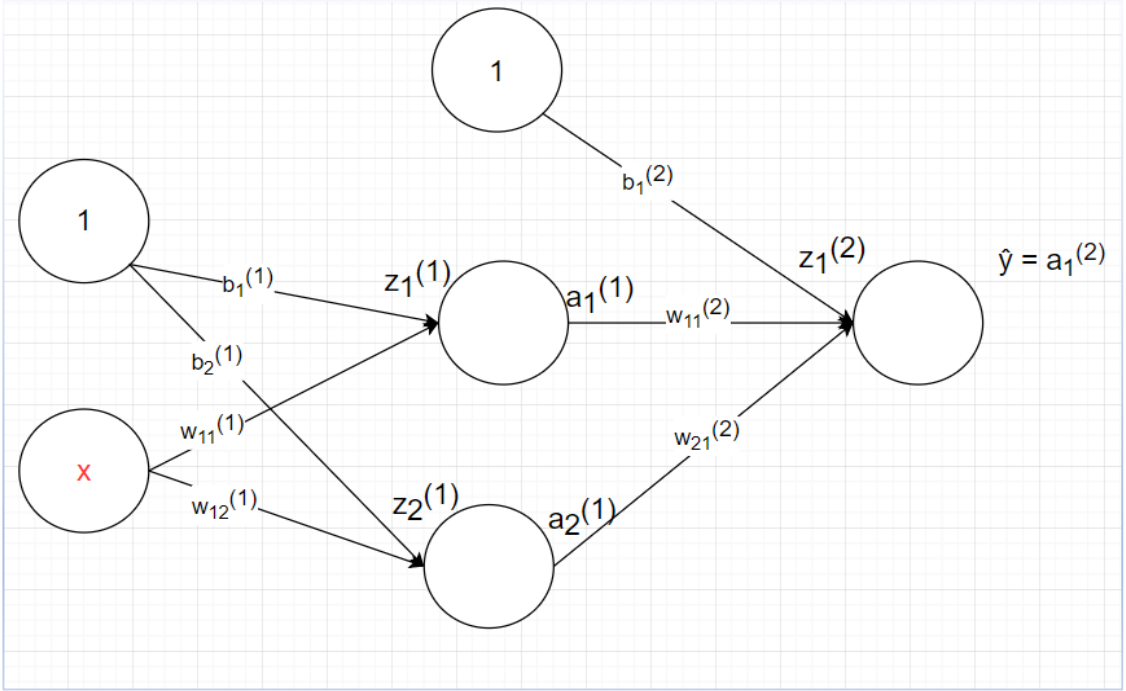
Hình 2.6: Minh họa kiến trúc điển hình của một tế bào mạng nơ-ron (Nguồn: [1])

2.1.2.2 Hàm kích hoạt (Activation function)

Hàm kích hoạt là phần rất quan trọng trong mạng nơ-ron, đặc biệt là mạng nơ-ron nhiều lớp ẩn. Nếu không có hàm kích hoạt phi tuyến tính, cho dù mạng nơ-ron có nhiều lớp ẩn đến cỡ nào thì cũng chỉ có sức mạnh đại diện cho phân loại tuyến tính, điều này tương đương với một mạng mà không có lớp ẩn nào. Vì bản chất tổng hợp các hàm tuyến tính là một hàm tuyến tính. Do đó, hàm kích hoạt a là một hàm phi tuyến tính được áp dụng cho đầu ra tại mỗi nút và dữ liệu đầu vào cho tầng tiếp theo, cho phép mạng nơ-ron nhiều lớp ẩn học các hàm phi tuyến phức tạp.

Ví dụ: Sử dụng hàm kích hoạt trong mạng nơ-ron là hàm tuyến tính thì mạng nơ-ron sẽ như thế nào?

Giả sử hàm kích hoạt tuyến tính dạng y = f(x) = 2*x + 3 và mạng nơ-ron như sau:



Hình 2.7: Mô hình mạng nơ-ron 1-2-1

Ta có:

$$z_1^{(1)} = w_{11}^{(1)} * x + b_1^{(1)} => a_1^{(1)} = f(z_1^{(1)}) = 2 * z_1^{(1)} + 3$$
$$= 2 * (w_{11}^{(1)} * x + b_1^{(1)} + 3)$$

Tương tự, $a_2^{(1)} = 2 * (w_{12}^{(1)} * x + b_2^{(1)} + 3$

Do đó có,

$$\hat{y} = a_1^{(2)} = f(z_1^{(2)}) = 2 * z_1^{(2)} + 3 = 2 * (a_1^{(1)} * w_{11}^{(2)} + a_2^{(1)} * w_{21}^{(2)} + b_1^{(2)}) + 3$$
$$= 2 * (b_1^{(2)} + (2 * (b_1^{(1)} + w_{11}^{(1)} * x) + 3)$$
$$* w_{11}^{(2)} + (2 * (b_2^{(1)} + w_{12}^{(1)} * x) + 3) * w_{21}^{(2)}) + 3$$
$$= x * (4 * w_{11}^{(1)} * w_{11}^{(2)} + 4 * w_{12}^{(1)} * w_{21}^{(2)}) +$$
$$(2 * (b_1^{(2)} + (2 * (b_1^{(1)} + 3) * w_{11}^{(2)}) + (2 * b_2^{(1)} + 3) * w_{21}^{(2)}) + 3)$$

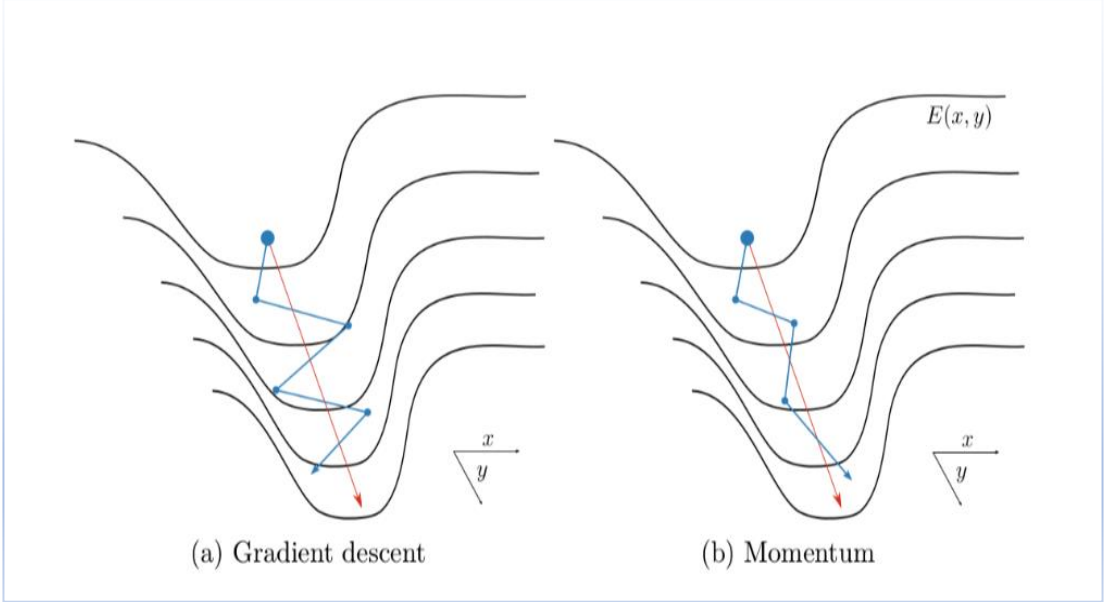
Tóm lại $\hat{y} = a * x + b$, vậy mạng nơ-ron không sử dụng hàm kích hoạt phi tuyến tính thì chỉ là mô hình hồi quy đơn giản, nó không thể học được các phụ thuộc phức tạp hơn.

2.1.2.3 Lan truyền ngược (Back propagation) và học với lan truyền ngược

Luận Văn "Xây Dựng Hệ Thống Nhận Dạng Âm Thanh Tiếng Việt" đã trình bày về lan truyền ngược và học với lan truyền ngược nên nhóm sinh viên xin phép không trình bày lại nội dung này.

2.1.2.4 Phương pháp giảm độ dốc với động lượng (Momentum)

Luận Văn "Xây Dựng Hệ Thống Nhận Dạng Âm Thanh Tiếng Việt" đã trình bày về phương pháp giảm độ dốc với động lượng. Sau đây nhóm sinh viên sẽ thảo luận thêm tác dụng của phương pháp này.



Hình 2.8: Minh hoạ vấn đề bề mặt độ cong bệnh lý và csach tiếp cận của độ giảm dốc và độ giảm dốc với động lượng (Nguồn: [1])

Phương pháp này có thể giúp cho sự hội tụ nhanh hơn. Hãy tưởng tượng một hàm mục tiêu có hình dạng như một hẻm núi dài và hẹp dần dần dốc về phía mức tối thiểu. Nói rằng chúng tôi muốn giảm thiểu chức năng này bằng cách sử dụng độ dốc giảm dần. Nếu chúng ta bắt đầu từ một số điểm trên tường hẻm núi, độ dốc âm sẽ chỉ theo hướng đi xuống dốc nhất, tức là chủ yếu hướng về tầng hẻm núi. Điều này là do các bức tường hẻm núi dốc hơn nhiều so với độ dốc dần dần của hẻm núi về phía mức tối thiểu. Nếu

tốc độ học tập (tức là kích thước bước) nhỏ, chúng ta có thể xuống tầng hẻm núi, sau đó theo nó về mức tối thiểu, nhưng tốc độ sẽ chậm. Chúng ta có thể tăng tỷ lệ học tập, nhưng điều này sẽ không thay đổi hướng của các bước. Trong trường hợp này, chúng ta sẽ vượt qua tầng hẻm núi và kết thúc ở bức tường đối diện. Sau đó, chúng ta sẽ lặp lại mô hình này, dao động từ tường này sang tường khác trong khi tiến triển chậm về mức tối thiểu. Động lượng có thể giúp đỡ trong tình huống này.

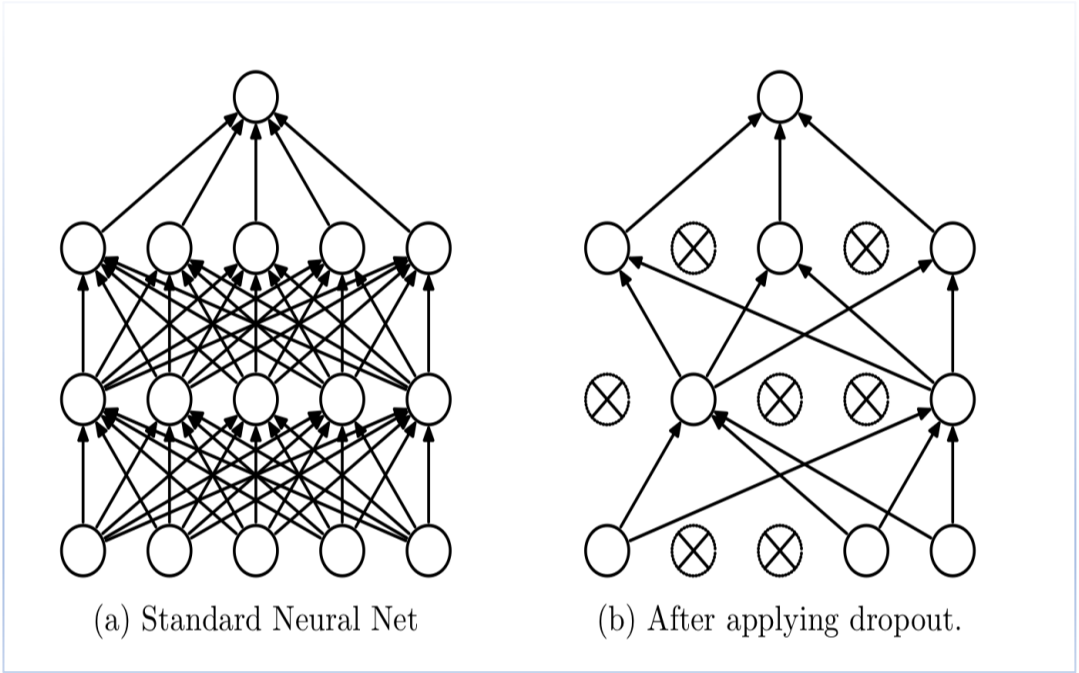
Động lượng đơn giản có nghĩa là một số phần của bản cập nhật trước được thêm vào bản cập nhật hiện tại, do đó các bản cập nhật lặp lại trong một hợp chất hướng cụ thể; chúng ta xây dựng đà di chuyển ngày càng nhanh hơn theo hướng đó. Trong trường hợp của hẻm núi, chúng ta sẽ xây dựng động lượng theo hướng tối thiểu, vì tất cả các bản cập nhật đều có một thành phần theo hướng đó. Ngược lại, di chuyển qua lại trên các bức tường hẻm núi liên quan đến hướng đảo ngược liên tục, do đó động lượng sẽ giúp làm giảm các dao động theo các hướng đó. Phương pháp giảm độ dốc theo động lượng giúp chúng ta tiết kiệm thời gian tối ưu và tránh được các cực tiểu cục bộ để tìm được điểm cực tiểu toàn cục

2.1.3 Phương pháp cắt giảm (Dropout)

Geoffrey E. Hinton, Srivastava, et al. (2012a) và Srivastava et al. (2014) đã giới thiệu phương pháp cắt giảm (dropout), một phương pháp ngăn chặn sự quá khớp (overfiting) trong các mạng lớn khi đào tạo. Ý tưởng chính là: Trong quá trình huấn luyện ta loại bỏ một tập các nút nơ-ron, được lấy mẫu ngẫu nhiên từ mỗi lớp với xác suất cố định p. Dropout cũng là một kỹ thuật đơn giản và cực kỳ hiệu quả. Trong quá trình huấn luyện, nhiều đơn vị ẩn bị tắt ngẫu nhiên và mô hình được huấn luyện trên các bộ tham số còn lại.

Với mỗi cách tắt các đơn vị, ta có một mô hình khác nhau. Với nhiều tổ hợp đơn vị bị tắt khác nhau, ta thu được nhiều mô hình. Việc kết hợp ở cuối cùng được coi như sự kết hợp của nhiều mô hình.

Ví dụ như hình 2.7 – Mô hình mạng nơ-ron 1-2-1 bao gồm 1 lớp đầu vào, 2 lớp ẩn và 1 lớp đầu ra. Ví dụ như lớp ẩn 1, ta dùng dropout với p = 0.6, nên chỉ giữ lại 2 trên 5 node cho mỗi lần cập nhật. Nguyên lí hoạt động như hình 2.9 phía dưới.



Hình 2.9: Mạng nơ-ron sử dụng và không sử dụng dropout.

2.1.4 Các kiến trúc mạng nơ-ron hồi quy

Một danh sách nghiên cứu đầy đủ của mọi kiến trúc học sâu DNN là điều không khả thi và nằm ngoài phạm vi của luận văn này, tuy nhiên ở đây, nhóm sinh viên đã nỗ lực khái quát về những kiến trúc nơ-ron hồi quy nổi bật trong những năm gần đây đồng thời truyền cảm hứng cho việc thiết lập kiến trúc hệ thống dịch trong các chương sau của nhóm.

2.1.4.1 Mạng nơ-ron hồi quy (RNN – Recurrent Neural Network)

Luận văn "Xây Dựng Hệ Thống Nhận Dạng Âm Thanh Tiếng Việt" đã trình bày lý thuyết nền tảng của mạng nơ-ron hồi quy như dưới đây.

Mạng nơ-ron hồi quy (RNN - Recurrent Neural Network) là mạng thần kinh lan truyền tới (Feedforward Neural Networks) [3] được tăng cường bằng cách bao gồm các cạnh mà kéo dài các bước thời gian liền kề, đưa ra khái niệm về thời gian cho mô hình. Giống như mạng lan truyền tới, mạng RNN không có chu trình giữa các cạnh thông thường. Tuy nhiên, các cạnh mà kết nối các bước thời gian liền kề, được gọi là cạnh hồi quy, có thể hình thành chu trình có độ dài bằng một, tự kết nối từ nột nút (node) tới chính nó theo thời gian. Tại thời điểm t, nút với cạnh hồi quy nhận đầu vào từ điểm dữ liệu hiện tại $x^{(t)}$ và từ các giá trị nút ẩn $h^{(t-1)}$ trong trạng thái trước đó của mạng. Đầu ra $\hat{y}^{(t)}$ cho mỗi thời điểm t được tính bằng các giá trị nút ẩn $h^{(t)}$ tại thời điểm t. Đầu vào $x^{(t-1)}$ tại thời điểm t-1 có thể ảnh hưởng đến giá trị đầu ra $\hat{y}^{(t)}$ tại thời điểm t và sau đó, bằng các kết nối hồi quy.

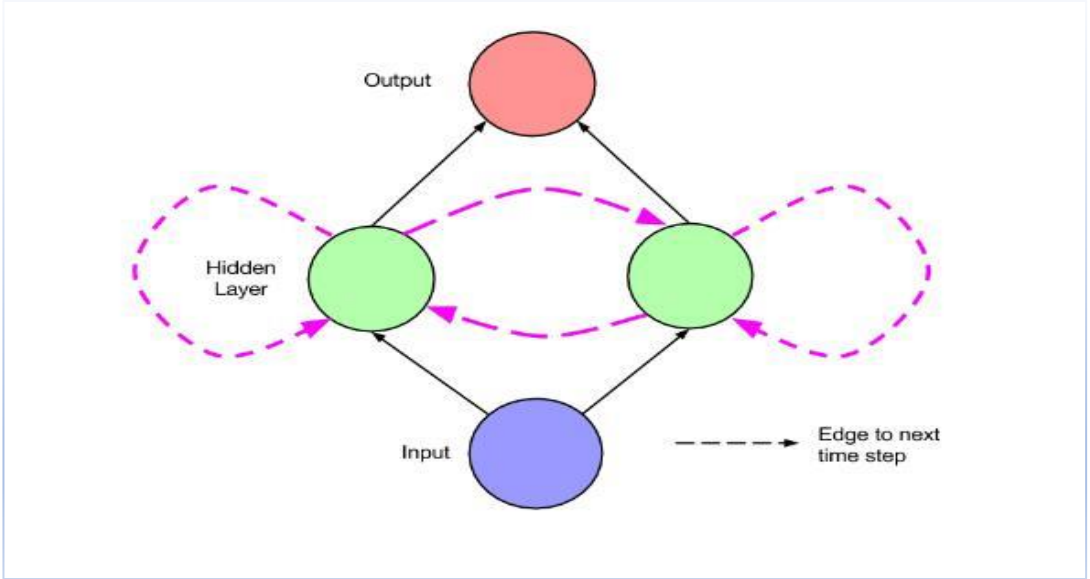
Các tính toán cần thiết tại mỗi bước thời gian trên đường là truyền tới của mạng RNN đơn giản (hình 2.9), được thể hiện như sau:

$$h^{(t)} = \sigma\left(W^{hx} + W^{hh}h^{(t-1)} + b_h\right)$$

$$\hat{y}^{(t)} = softmax(W^{yh}h^{(t)} + b_y)$$

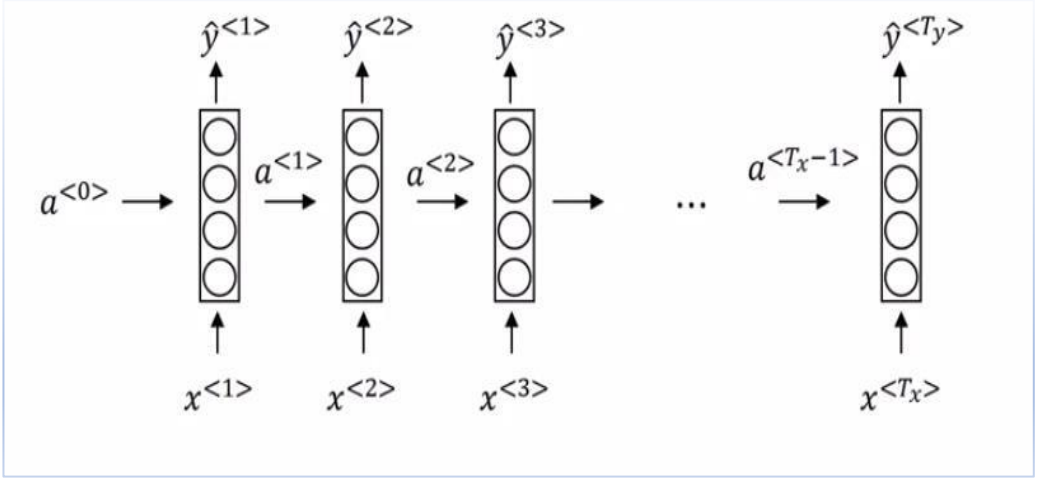
Trong đó:

- W^{hx} là ma trận trọng số thông thường giữa đầu vào và lớp ẩn.
- W^{hh} là ma trận trọng số hồi quy giữa lớp ảnh và chính nó ở các bước thời gian liền kề.
- Các vec-tơ b_h và b_y là các tham số sai lệ.



Hình 2.10: Mạng RNN đơn giản (Nguồn: [3])

Một biểu diễn dễ hiểu cho hình 2.10 được thể hiện ở hình 2.11, trong đó các bước thời gian được mở ra. Với hình ảnh này, mạng được hiểu không phải là chu trình, mà là một mạng học sâu với mỗi lớp tương ứng với mỗi bước thời gian, được chia sẻ trọng số qua các lớp.



Hình 2.11: Minh họa mạng RNN được mở ra từ hình 2.10
(Nguồn: Coursera Sequence Models)

Bây giờ lấy ví dụ, xét bài toán: Xác định chữ nào là một phần của tên người trong một câu.

Đọc một câu từ trái sang phải, đánh số mỗi từ trong câu là x_i ($1 \le i \le N$). Tiến trình nạp từ thứ nhất x_1 vào một lớp ẩn mạng nơ-ron, lớp này sẽ dự đoán kết quả $\hat{y}^{<1>}$ xem x_1 có phải là một phần của tên người hay không. Xét từ thứ hai x_2 thay vì chỉ dự đoán $\hat{y}^{<2>}$ từ x_2, thì lớp này nhận thêm giá trị kích hoạt $a^{<1>}$ từ bước 1. Các bước sau được thực hiện tương tự, cho đến khi kết thúc câu. Thông thường ở bước đầu tiên cũng được truyền thêm vào giá trị kích hoạt với $a^{<0>}$ (là một vec-tơ 0). Hình 2.11 minh hoạ mạng RNN lan truyền tới với giá trị $a^{<t>}$, $\hat{y}^{<t>}$ được tính theo công thức như sau:

$$a^{<t>} = g(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)$$
$$\hat{y}^{<t>} = g(W_{ya}a^{<t>} + b_y)$$

RNN duyệt dữ liệu từ trái sang phải và tham số dùng cho mỗi bước là được chia sẻ. Vì vậy, khi dự đoán kết quả $\hat{y}^{<3>}$, RNN không chỉ sử dụng đầu vào x_3 mà còn sử dụng thông tin từ x_1 và x_2.

Xét hai trường hợp đầu vào sau:
Trường hợp một:
(1) Anh ấy nói "*Teddy Roosevelt là một tổng thống tuyệt vời*".
(2) Anh ấy nói "*Teddy là loại gấu bông được mua nhiều nhất ở cửa hàng này*".
Trường hợp hai:
Thời thơ ấu, tôi thường nghe bố tôi nhắc tới một người anh hùng, ... bố tôi đang nhắc tới Phan Đình Giót.
Ở trường hợp một, "*Teddy*" là một phần tên người trong câu (1), còn câu (2) thì không phải. Như vậy, một điểm yếu của mạng RNN là chỉ dùng thông tin từ các bước phía trước trong chuỗi để thực hiện dự đoán kết quả, thông tin dùng để dự đoán này là không

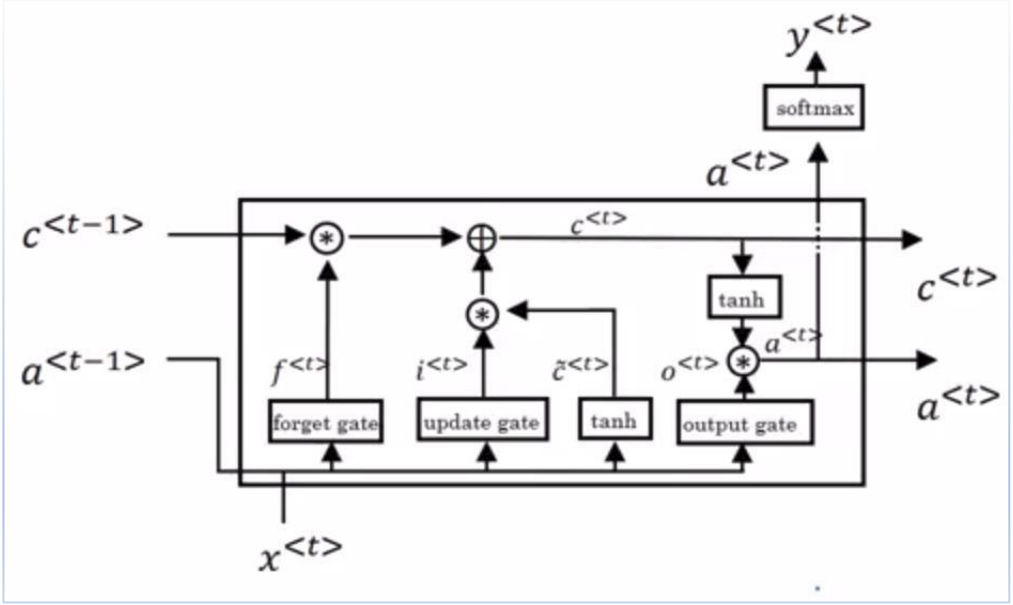
đủ. Để khắc phục nhược điểm này, mạng nơ-ron hồi quy hai chiều (BiRNN - Bidirectional recurrent neural network) ra đời.

Còn ở trường hợp hai, cụm từ "*Phan Đình Giót*" có được dự đoán là tên của người hay không phụ thuộc thông tin được mang đến từ cụm "*người anh hùng*". Nhưng mạng RNN bị hạn chế trong duy trì phụ thuộc tầm xa, nghĩa là nếu chuỗi đầu vào đủ dài, mạng đủ sâu, thì RNN khó khăn trong việc mang thông tin từ các bước trước tới các bước sau, thậm chí RNN có thể bỏ qua thông tin quan trọng đến từ những bước đầu tiên. Đây là vấn đề biến mất độ dốc (vanishing gradient).

2.1.4.2 Mạng bộ nhớ dài ngắn (Long Short Term Memory - LSTM)

Nội dụng về mạng bộ nhớ dài ngắn (LSTM) cũng đã được trình bày lí thuyết nền tảng trong luận văn “Xây Dựng Hệ Thống Nhận Dạng Âm Thanh Tiếng Việt”. Sau đây nhóm sinh viên sẽ trình bày thêm ví dụ về mạng LSTM này.

Mạng bộ nhớ dài-ngắn (Long Short-Term Memory - LSTM) được mô tả như hình dưới đây.



Hình 2.12: Mình họa một đơn vị LSTM

(Nguồn: Coursera Sequence Models)

Ví dụ ta có câu: “The cat, which already ate…, was full”.

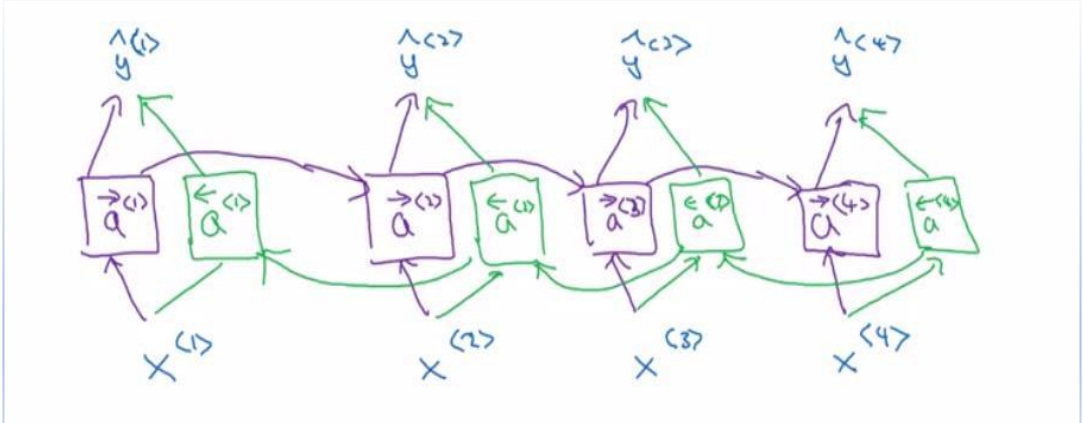
Để có thể dự đoán đúng từ “was” thì bạn có thể cần phải nhớ từ “cat” đang biểu diễn ở số ít để có thể chia động từ đúng. Tuy nhiên với RNN cơ bản với việc câu dài thì khi để dự đoán đúng được từ “was” thì thông tin được lưu trữ có thể không dự đoán được từ “was” này. LSTM sinh ra để giảm sự mất độ dốc và giải quyết sự thiếu thông tin này.

Khi ta đọc câu này từ trái sang phải, LSTM sẽ có một biến là c (ô nhớ). Thực tế, nó sẽ cung cấp cho chúng ta một bộ nhớ để lưu lại những thứ cần nhớ - “cat” là số ít, khi đi sâu vào câu hơn nó vẫn hoạt động khi xem xét chủ ngữ của câu là số ít hay số nhiều. Khi đó $c^{<2-was>} = 1$ để biểu diễn chủ ngữ là số ít. Khi qua mỗi bước $c^{<3>}$, $c^{<4>}$, … cho đến khi ta muốn dự đoán “was” ta cần thông tin c vẫn còn bằng 1 để hiểu chủ ngữ là số ít. LSTM sẽ cung cấp cho ta cổng input và cổng forget. Khi duyệt đến từ “cat” cổng input $\Gamma_i = 1$ để cập nhật $c^{<2-was>} = 1$ để lưu chủ ngữ là số ít. Khi duyệt đến $c^{<3>}, c^{<4>}$, ... Cổng forget $\Gamma_f = 1$ để giữ lại thông tin $c^{<t>} = 1$ được tính bởi $c^{<t>} = \Gamma_i * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>}$, cho nên thông tin chủ ngữ là số ít vẫn được giữ lại cho đến khi dự đoán từ tại vị trí từ “was” mô hình sẽ sử dụng thông tin $c = 1$ để dự đoán từ đó chính xác là từ “was” bởi vì chủ ngữ là số ít. Tuy nhiên LSTM vẫn chưa giải quyết được vấn đề khi thông tin đến từ bước thời gian phía sau của câu.

2.1.4.3 Mạng nơ-ron hồi quy hai chiều (Bidirectional Recurrent Neural Network – BiRNN)

Cùng với mạng LSTM, một trong những kiến trúc RNN được sử dụng nhiều nhất là mạng nơ-ron hồi quy hai chiều (BiRNN - Bidirectional recurrent neural network) với LSTM, khắc phục nhược điểm chỉ nhận thông tin từ các bước thời gian phía trước để dự đoán kết quả bước hiện tại.

Luận văn “Xây Dựng Hệ Thống Nhận Dạng Âm Thanh Tiếng Việt” đã trình bày lý thuyết nền tảng của mạng nơ-ron hồi quy hai chiều. Sau đây nhóm sinh viên sẽ đưa ra ví dụ cho mạng nơ-ron hồi quy hai chiều này.



Hình 2.13: Minh họa mạng BiRNN (Nguồn: Coursera Sequence Models)

Ví dụ trong tác vụ nhận dạng xem một từ có phải là một phần tên người hay không, ta có hai câu như sau:

- He said: “Teddy bears are on sale”
- He said: “Teddy Roosevelt was a great President”

Nếu ta áp dụng RNN bình thường vào tác vụ này, khi tính tới $\hat{y}^{<3>}$ ứng với từ “Teddy” mô hình sẽ chỉ nhận được thông tin chỉ bao gồm ba từ trước đó là “He”, “said” và “Teddy”. Vì vậy để quyết định xem $\hat{y}^{<3>}$ nên bằng 1 (là một phần tên người) hay bằng 0 (không phải là một phần tên người) hay không thì ta cần biết nhiều thông tin hơn vì chỉ ba từ đầu tiên “He said: Teddy” không cho ta biết họ đang nói về gấu Teddy hay nói về cựu tổng thống Hoa Kỳ Teddy Roosevelt. Vì vậy sự ra đời của mạng hồi quy hai chiều để giải quyết vấn đề này.

Khi áp dụng Bi-RNN, khi tính tới $\hat{y}^{<3>}$ ứng với từ “Teddy” mô hình sẽ nhận được thông tin từ “He said: Teddy” ở chiều tới và “bears are on sale” hay “Roosevelt was a great President” ở chiều lùi. Do đó mô hình sẽ quyết định được kết quả $\hat{y}^{<3>}$ bằng 1 ứng với “Roosevelt was a great President” và bằng 0 ứng với “bears are on sale”.

2.2 MÔ HÌNH DỊCH MÁY:

Các mô hình dịch máy hiện nay thường được chia làm ba thành phần chính: Nhúng từ (word embedding), bộ mã hoá (encoder) và bộ giải mã (decoder). Tuy nhiên tuỳ cách tiếp cận mà giải pháp cho mỗi mô hình dịch máy sẽ được tuỳ chỉnh như thêm các lớp tích chập hay thêm cơ chế chú ý để đưa ra một hệ thống phù hợp. Sau đây, nhóm sinh viên sẽ trình bày về mô hình dịch máy điển hình là: mô hình dịch máy dựa trên mô hình nơ-ron hồi quy (RNN) kết hợp với từ nhúng (sử dụng word2vec), mạng nơ-ron hồi quy hai chiều (Bi-RNN) tại bộ mã hoá (encoder) và cơ chế chú ý (Attention).

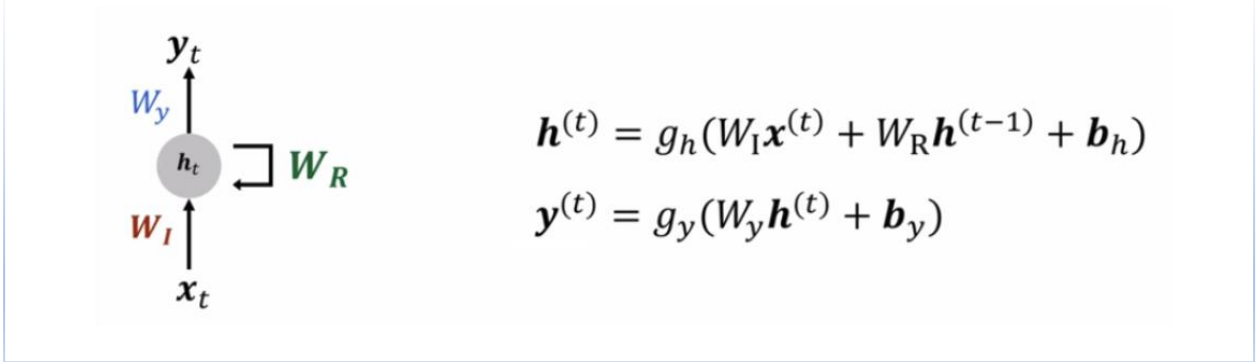
2.2.1 Giới thiệu và đặt vấn đề

Mạng nơ-ron là mô hình học mạnh mẽ, đã đạt được những kết quả vượt bậc trong nhiều tác vụ học máy. Những tiến bộ gần đây về thuật toán và phần cứng đã giúp con người có thể huấn luyện mạng nơ-ron hỗ trợ cho các nhiệm vụ mà trước đây đòi hỏi trình độ chuyên môn đáng kể từ con người. So với các cách tiếp cận truyền thống, mạng nơ-ron đòi hỏi ít sự nỗ lực hơn từ con người và mang lại kết quả cao hơn. Để đạt được kết quả khả quan này là nhờ sự phong phú đa dạng của dữ liệu ngày nay.

Mạng nơ-ron truy hồi (Recurrent Neural Network, viết tắt là RNN) được phát minh bởi John Hopfield năm 1982 [5]. Trong khoảng 5-6 năm gần đây, RNN được ứng dụng rộng rãi trong ngành NLP và thu được những thành tựu lớn. Mạng RNN mô hình hóa được bản chất của dữ liệu trong NLP (có đặc tính chuỗi và các thành phần như từ, cụm từ trong dữ liệu phụ thuộc lẫn nhau). Có thể nói việc áp dụng mạng RNN là một bước đột phá trong ngành NLP.

Trong mô hình mạng nơ-ron thông thường (Feed forward network), chúng ta coi input data là các dữ liệu độc lập, không có mối liên hệ với nhau. Tuy nhiên, trong ngôn ngữ tự nhiên thì mối liên hệ giữa các từ và ngữ cảnh đóng một vai trò quan trọng, quyết định ý nghĩa của câu văn. Do đó việc áp dụng môt hình mạng nơ-ron thông thường vào các bài toán xử lí ngôn ngữ tự nhiên thường không đạt kết quả mong muốn.

Để khắc phục nhược điểm này, chúng ta sử dụng mô hình RNN (Recurrent Neural Network). RNN coi dữ liệu đầu vào là một chuỗi (sequence) liên tục, nối tiếp nhau theo thứ tự thời gian. Ví dụ như một đoạn text có thể được coi là một chuỗi các từ vựng(words) hoặc là một chuỗi các ký tự (character). Tại thời điểm t, với dữ liệu đầu vào x_t ta có kết quả output là y_t. Tuy nhiên, khác với mạng Feed forward network, y_t lại được sử dụng là input để tính kết quả output cho thời điểm (t+1). Điều này cho phép RNN có thể lưu trữ và truyền thông tin đến thời điểm tiếp theo. Mô hình hoạt động của RNN có thể được mô tả trong hình dưới đây (thông thường hàm activation function g_h được sử dụng là tanh còn g_y có thể là sigmoid hoặc softmax function tùy thuộc vào từng bài toán cụ thể).



Hình 2.14: Mạng Recurrent Neural Network (Nguồn: [3])

Đặt vấn đề: Tại sao không sử dụng mô hình nơ-ron hồi quy cơ bản như trên để xây dựng mô hình dịch máy.

Chúng ta có thể hiểu một cách đơn giản rằng RNN là một mô hình mạng nơ-ron có bộ nhớ (memory) để lưu trữ thông tin của phần xử lý trước đó. Về mặt lý thuyết thì mạng nơ-ron hồi quy có thể xử lý và lưu trữ thông tin của một chuỗi dữ liệu với độ dài bất kỳ. Tuy nhiên trong thực tế thì nó chỉ tỏ ra hiệu quả với chuỗi dữ liệu có độ dài không quá lớn (short-term memory hay còn gọi là long-term dependency problem). Nguyên nhân của vấn đề này là do vấn đề mất độ dốc trong quá trình huấn luyện (vanishing gradient problem, độ dốc (gradient) được sử dụng để cập nhật giá trị của ma trận trọng số

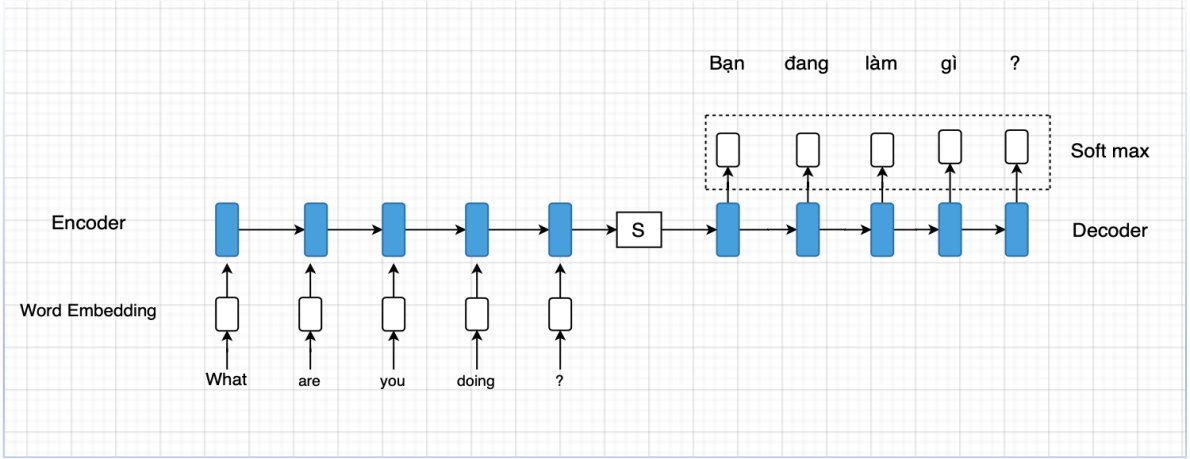
(weight matrix) trong mạng nơ-ron hồi quy và nó có giá trị nhỏ dần theo từng layer khi thực hiện back propagation. Khi gradient trở nên rất nhỏ (có giá trị gần bằng 0) thì giá trị của ma trận trọng số sẽ không được cập nhật thêm và do đó mạng nơ-ron sẽ dừng việc học tại lớp này. Đây cũng chính là lý do khiến cho mạng nơ-ron hồi quy không thể lưu trữ thông tin của các bước thời gian (timesteps) đầu tiên trong một chuỗi dữ liệu có độ dài lớn.

Với những hạn chế của mô hình trên, mô hình sequen to sequence sử dụng kiến trúc bộ nhớ dài ngắn (Long-Short Term Memory) đã được trình bày ở trên với cơ chế chú ý (Attention Mechanism) giúp giải quyết vấn đề trên và đưa ra một mô hình mạnh mẽ để thực hiện tác vụ dịch máy.

2.2.2 Mô hình dịch máy Sequence to Sequence với cơ chế chú ý (Attention Mechanism)

Sequence to Sequence Model (Seq2seq) là một mô hình Deep Learning với mục đích tạo ra một output sequence từ một input sequence mà độ dài của 2 sequences này có thể khác nhau. Seq2seq được giới thiệu bởi nhóm nghiên cứu của Google vào năm 2014 trong bài báo *Sequence to Sequence with Neural Networks* [6]. Mặc dù mục đích ban đầu của Model này là để áp dụng trong Machine Translation, tuy nhiên hiện nay Seq2seq cũng được áp dụng nhiều trong các hệ thống khác như Speech recognition, Text summarization, Image captioning,….

Seq2seq gồm 2 phần chính là Encoder và Decoder. Cả hai thành phần này đều được hình thành từ các mạng Neural Networks, trong đó Encoder có nhiệm vụ chuyển đổi dữ liệu đầu vào (input sequence) thành một representation với lower dimension còn Decoder có nhiệm vụ tạo ra output sequence từ representation của input sequence được tạo ra ở phần Encoder.



Hình 2.15: Sequence to Sequence Model in Machine Translation

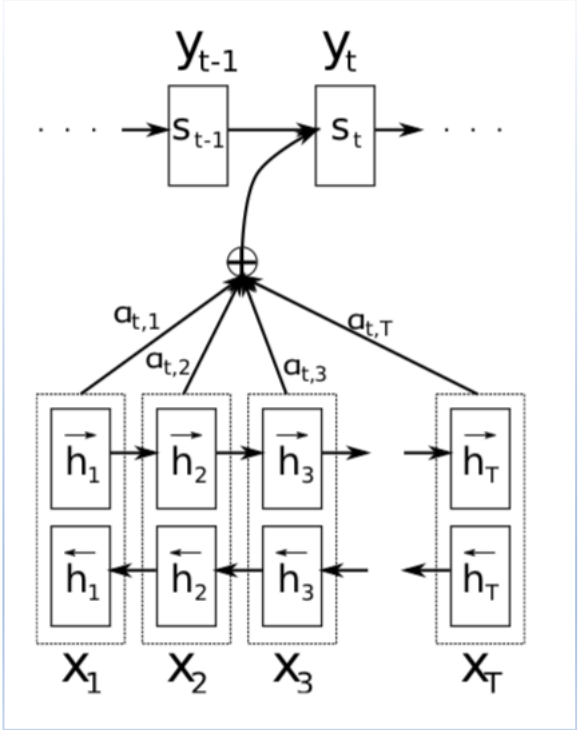
Trong luận văn, từ dữ liệu đầu vào (input sequence) là một câu dưới dạng văn bản, chúng ta sử dụng Embedding Layer để chuyển các từ này sang dạng Word Embedding rồi sử dụng Bi-directional LSTM để tạo ra một đại diện (representation) của câu đầu vào (trong hình 2.15 là S).

Decoder được tạo thành từ RNN với LSTM và sử dụng đầu ra của Encoder làm dữ liệu đầu vào để tạo ra một câu đầu ra (output sequence). Trong dịch máy chúng ta phải chọn câu văn phù hợp nhất thay vì để RNN cell tạo ra từng từ một. Thông thường việc lựa chọn output sequence được thực hiện bởi thuật toán tìm kiếm chùm tia (Beam Search). Tuy nhiên, trong thực tế việc sử dụng một vec-tơ đại diện (vector representation) thường không thể lưu trữ được toàn bộ thông tin của câu đầu vào (input sequence). Do đó, có một số phương pháp giúp tăng độ chính xác cho hệ thống này như: sử dụng nhiều lớp hơn và sử dụng mạng nơ-ron hồi quy hai chiều.
Tuy nhiên, phương pháp được sử dụng nhiều nhất và làm tăng đáng kể độ chính xác của các hệ thống là sử dụng cơ chế chú ý (Attention Mechanism). Phương pháp này được giới thiệu vào năm 2014 trong bài báo *Neural Machine Translation by Jointly Learning to Align and Translate* [7].

Nguyên tắc hoạt động chung của cơ chế chú ý (Attention Mechanism) là tại mỗi bước giải mã (Decoding Step), bộ giải mã (decoder) sẽ chỉ tập chung vào phần liên quan

trong câu đầu vào thay vì toàn bộ câu đầu vào. Mức độ tập chung này được thiết lập bởi ma trận chú ý (Attention weights) như mô tả trong hình 2.16.

Như vậy, tại mỗi bước giải mã, bộ giải mã nhận ba đầu vào là: Hidden state của bước giải mã trước, đầu ra của bước phía trước và vec-tơ chú ý (Attention vec-tơ). Vec-tơ chú ý chứa ma trận chú ý (Attention weight) của từng từ trong câu đầu vào. Từ nào chứa nhiều thông tin cần thiết cho việc giải mã thì sẽ có giá trị trọng số lớn hơn và tổng các trọng số của tất cả các từ trong câu đầu vào phải bằng 1. Giá trị của ma trận chú ý này được học thông qua quá trình huấn luyện.



Hình 2.16: Mô hình sequence to sequence với cơ chế chú ý ([Nguồn: [7])

Trong bài toán dịch máy, chúng ta thường hay sử dụng mô hình như ở hình 2.15. Bộ mã hoá phải nén tất cả thông tin của một câu lại thành một vec-tơ biểu diễn duy nhất, chứa toàn bộ thông tin cần thiết để bộ giải mã có thể dịch thành câu đích. Vấn đề nằm ở chỗ, những câu dài sẽ không được dịch chính xác vì thông tin không được lưu trữ đủ trong một vec-tơ biểu diễn duy nhất.

Ví dụ như ta cần dịch câu "mặt trời bé nhỏ" thành câu "little sun", khi ta cần mô hình dự đoán từ "little". Thay vì sử dụng bộ mã hoá nguyên câu "mặt trời bé nhỏ" thành một vec-tơ biểu diễn cuối cùng h_T như hình 2.16 rồi dùng bộ giải mã để dịch câu. Cơ chế chú ý sẽ tính tổng hợp trọng số các vec-tơ biểu diễn các từ "mặt", "trời", "bé", "nhỏ" tương ứng với các vec-tơ h_1, h_2, h_3, h_4 để thực hiện dự đoán từ "little". Khi đó, lúc dự đoán từ "little" các trọng số tại h_3, h_4 (tương ứng với "bé", "nhỏ") sẽ cao hơn nhiều so với h_1, h_2 (tương ứng với "mặt", "trời").

2.2.3 Mô hình ngôn ngữ (Language model)

Mô hình ngôn ngữ là một mô hình dùng để dự đoán xác suất của một chuỗi các từ. Trong dịch máy khi ta chuyển một chuỗi các từ từ ngôn ngữ nguồn sang ngôn ngữ đích, ta dùng mô hình kết hợp với thuật toán beam-search (thuật toán tìm kiếm chùm tia) để dự đoán mà không kết hợp mô hình ngôn ngữ, ta sẽ thu được một chuỗi đích có xác suất cao nhất. Điều này có thể bỏ qua các bản dịch khác có xác suất thấp hơn nhưng lại có thể phù hợp hơn với ngôn ngữ đích.

Ví dụ kết quả của dự đoán một câu tiếng Anh khi dùng beam-search với beam-with bằng 2, ta sẽ thu được hai câu như sau:
Câu một: "thứ hai hôm nay là"
Câu hai:"hôm nay là thứ hai"
Giả sử mô hình dự đoán câu "thứ hai hôm nay là" là câu có xác xuất cao hơn, thì ta đã bỏ qua câu "hôm nay là thứ hai", mặc dù xác suất khi dự đoán bằng mô hình có thể thấp hơn nhưng nó lại là câu phù hợp hơn với ngôn ngữ đích (trong trường hợp này là tiếng Việt).

Vì vậy khi dùng mô hình để dự đoán ta sẽ lấy hết tất cả các câu kết quả trong chùm tia beam-search đưa vào mô hình ngôn ngữ, sau đó ta sẽ thu được câu phù hợp nhất, tuy rằng câu đó có xác suất thấp hơn nhưng nó lại có xác suất cao hơn khi được dự đoán bằng mô hình ngôn ngữ.

Nhóm xin viên xin trình bày mô hình ngôn ngữ thống kê, cụ thể là mô hình ngôn ngữ N-gram. Trong đó xác suất của một câu sẽ được xác định theo công thức sau:

$$P(X_1 \dots X_n) = \prod_{k=1}^{n} P(X_k | X_1^{k-1})$$

(Nguồn: [10])

Nghĩa là xác suất của câu sẽ được xác định bằng tích của xác suất có điều kiện của các từ trong câu, tùy thuộc vào mô hình n-gram với n bằng bao nhiêu thì điều kiện là dựa trên (n-1) từ trước từ đang xét.

Công thức tính xác suất của một từ có điều kiện theo các từ trước đó là:

$$P(w_n | w_{n-1}) = P(\frac{C(w_{n-1} w_n)}{(w_{n-1})}$$

(Nguồn: [10])

Ví dụ đối với mô hình bigram (đây là mô hình N-gram với N =2) để xác định xác suất của câu: "hôm nay là thứ hai" với bộ dữ liệu đào tạo có các câu:

<s> hôm nay là thứ hai </s>

<s> thứ hai hôm nay là </s>

<s> hôm qua tôi đến trường </s>

Với <s> và </s> lần lượt là kí tự bắt đầu và kí tự kết thúc được thêm vào để phù hợp với bối cảnh của bigram. Khi đó ta sẽ xác định xác suất của câu đầu vào như sau:

$$P(\text{"hôm nay là thứ hai"}) = P(\text{hôm} | < s >) * P(\text{nay} | \text{hôm}) * P(\text{là} | \text{nay}) * P(\text{thứ} | \text{là}) * P(\text{hai} | \text{thứ}) * P(</s> | \text{hai})$$

Với $P(\text{hôm} | < s >) = \frac{C(<s> \text{hôm})}{C(<s>)} = \frac{2}{3}$ $P(\text{nay} | \text{hôm}) = \frac{C(\text{hôm nay})}{C(\text{hôm})} = \frac{2}{3}$

$P(\text{là} | \text{nay}) = \frac{C(\text{nay là})}{C(\text{nay})} = 1$ $P(\text{hai} | \text{thứ}) = \frac{C(\text{thứ hai})}{C(\text{thứ})} = 1$

$P(</s> | \text{hai}) = \frac{C(\text{hai}</s>)}{C(\text{hai})} = \frac{1}{3}$

Vậy $P(\text{hôm nay là thứ hai}) = \frac{2}{3} * \frac{2}{3} * 1 * 1 * \frac{1}{3} = 0.4444$

Đôi lúc trong thực tế sẽ có những câu xác suất rất nhỏ gần như bằng không, nên thực tế sẽ lấy exp log của xác suất

$$p_1 * p_2 * \ldots * p_n = \exp(\log p_1 + \log p_2 + \cdots + \log p_n)$$

CHƯƠNG 3: GIẢI PHÁP ĐỀ TÀI

3.1 TỔNG QUAN GIẢI PHÁP KIẾN TRÚC MÔ HÌNH

Để xây dựng lên mô hình dịch máy, nhóm sinh viên đã phát triển mô hình dựa trên kiến trúc ban đầu là Sequence to Sequence với LSTMs có ý tưởng từ bài báo *Sequence to Sequence Learning with Neural Networks* [6] do nhóm tác giả đến từ google được ông bố vào năm 2014 tại Silicon Valley AI Lab đã trình bày ý tưởng cụ thể để xây dựng một mô hình mạng nơ-ron hồi quy tối ưu với hướng đi mới so với các hệ thống dịch máy truyền thống.

Kiến trúc của mô hình trên gặp phải một hạn chế lớn đó là không thể lưu trữ thông tin của một chuỗi dữ liệu có độ dài lớn. Do đó theo ý tưởng của bài báo để khắc phục nhược điểm này cần phải sử dụng LSTM với cơ chế các cổng nhằm bổ sung thông tin hoặc loại bỏ những thông tin không cần thiết giúp tăng khả năng lưu trữ thông tin trong các câu dài.

Xét về mặt lý thuyết, khi sử dụng LSTM hoặc GRU có thể giúp lữu trữ thông tin một chuỗi dữ có độ dài lớn. Tuy nhiên trong thực nghiệm và thực tế việc sử dụng một vec-tơ để lưu trữ thông tin thường không thể lưu trữ toàn bộ thông tin của câu đầu vào được. Do đó dẫn đến nhóm sinh viên lựa chọn thêm các bổ sung như Bi-directional LSTM (LSTM hai chiều) ở bộ mã hoá và kết hợp với cơ chế chú ý (Attention Mechanism) từ bài báo *Neural Machine Translation by Jointly Learning to Align and Translate* [7] được thực hiện vào năm 2015.

Cơ chế chú ý cho phép mô hình có thể chú ý vào từng phần trong câu. Từ đó, thông tin không cần phải nén vào một vec-tơ biểu diễn duy nhất để lưu trữ thông tin của câu. Ngoài ra cơ chế này còn cho ta biết những từ nào trong câu đầu vào quyết định đến kết quả hiện tại.

Kiến trúc mô hình như trên được sử dụng rộng rãi trong lĩnh vực dịch máy. Với mục đích học tập và nghiên cứu, nhóm sinh viên đã quyết định áp dụng CNN vào để tính toán vec-tơ cho các câu.

Ví dụ như cho câu "Blog AI yêu thích của tôi", CNN sẽ tính toán vec-tơ đại diện cho từ và cụm từ: "Blog AI", "yêu thích", "Blog AI yêu thích". Khi đã có những vec-tơ trên ta có thể tính toán được vec-tơ đại diện cho câu "Blog AI yêu thích của tôi". Mục đích của nhóm sinh viên là để CNN có thể khái quát thông tin cho câu tốt hơn, có thể lưu trữ được nhiều thông tin hơn.

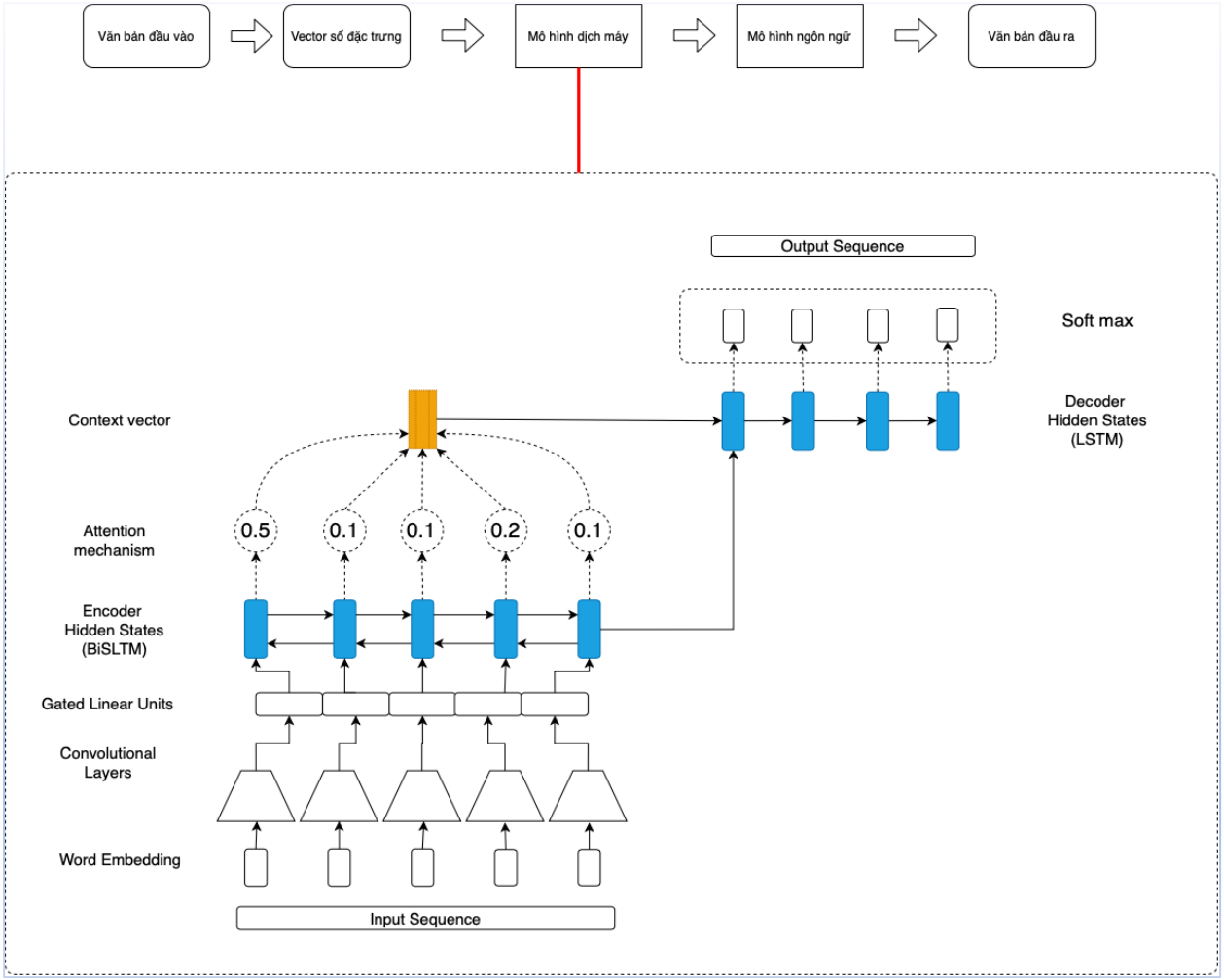
Tuy nhiên để tối ưu cho việc sử dụng CNN nhóm sinh viên áp dụng thêm một lớp đơn vị tuyến tính Gated Linear Units (GLU) được giới thiệu tại bài báo *Language Modeling with Gated Convolutional Networks* [9] cho việc sử lí các nhúng từ (word embedding) trước khi cho vào mạng nơ-ron hồi quy.

Lúc này đầu vào của lớp LSTM hai chiều sẽ là một vec-tơ gồm từ nhúng và vị trí của từ đó thay vì chỉ một vec-tơ từ nhúng như mô hình sequence-to-sequence truyền thống. Mục đích khi áp dụng lớp CNN và GLU là để phân loại và loại bỏ các thông tin gây nhiễu trong chuỗi đầu vào. Mỗi đầu ra của lớp CNN sẽ là đầu vào cho lớp GLU. Tuy nhiên, lớp GLU sẽ tách đầu vào thành hai phần một trong số đó sẽ qua một hàm sigmoid để lọc các thông tin liên quán đến các phần tử trong đầu vào. Qua đó, lớp tuyến tính GLU giúp kiểm soát được các thông tin đầu vào.

Tại bước giải mã (Decoder) nhóm sinh viên sử dụng thuật toán tìm kiếm chùm tia nhằm mục đích có thể tìm được một tập hợp chuỗi có xác suất cao nhất có thể là câu đầu ra. Tuy nhiên theo mô hình thông thường thì cuối cùng ta vẫn sẽ lấy chuỗi có xác suất cao nhất trong beam-width mà bỏ qua các kết quả còn lại dù có khả năng câu có xác suất cao nhất không phải là đầu ra lí tưởng nhất.

Để tối ưu cho thuật toán tìm kiếm chùm tia với nhiệm vụ tìm ra câu phù hợp nhất, nhóm sinh viên đề xuất sử dụng mô hình ngôn ngữ (Language Model) có thể kiểm tra được ngữ pháp, ngữ cảnh của câu đầu ra so với ngôn ngữ tiếng Việt để có thể tìm ra được chuỗi đầu ra đúng nhất.

Với những kiến trúc sẵn có và được đề xuất, nhóm sinh viên thể hiện kiến trúc của mô hình tổng quát cho tác vụ dịch máy như hình 3.1.



Hình 3.1: Tổng quan kiến trúc mô hình dịch máy

3.2 GIẢI PHÁP BIỂU DIỄN TỪ

3.2.1 Tổng quan về giải pháp

Để có thể sử dụng các mô hình Deep Learning (học sâu) phục vụ cho việc dịch máy, chúng ta cần biểu diễn các từ thành các số vì các mô hình chỉ làm việc với dữ liệu số. Vì thế dựa trên các kết quả tìm kiếm và thực nghiệm của các nhà khoa học, nhóm sinh viên đề xuất sử dụng Word Embedding (nhúng từ) dùng để biểu diễn các từ thành các vec-tơ số thực. Mô hình mà nhóm chọn là Word2vec với mục đích biểu diễn các từ tiếng Anh và tiếng Việt thành các vec-tơ số thực n chiều bằng nhau (mỗi chiều là một giá trị số thực) để phục vụ cho quá trình huấn luyện.

Word2vec là một mô hình học không giám sát (model unsupervised learning) nó dùng để thể hiện mỗi quan hệ giữa các từ, nó được kết hợp từ hai thuật toán Skip-gram và Continuous bag of words (CBOW). Ở đây nhóm sinh viên đề xuất sử dụng mô hình skip-gram cho biểu diễn từ. Với skip-gram, kích thước biểu diễn từ giảm từ kích thước bằng số từ trong bộ từ vựng xuống bằng chiều dài lớp ẩn. Hơn nữa các vec-tơ có ý nghĩa nhiều hơn về mặt mô tả mối quan hệ giữa các từ. Trong luận văn, nhóm sinh viên sử dụng mô hình Word2Vec Continuous Skipgram được pháp triển bởi nhóm công nghệ ngôn ngữ đại đại học Oslo. Nhóm sinh viên sử dụng mô hình Word2Vec với số chiều là 100.

Link chứa mô hình Word2Vec: http://vectors.nlpl.eu/repository/

3.2.2 Chi tiết giải pháp

Nhóm sinh viên sử dụng đầu vào là tập dữ liệu được chia làm 2 tập tin chính chia làm 2 ngôn ngữ tiếng Anh và tiếng Việt.

Với bộ dữ liệu “IWSLT’15 English-Vietnamese data” với khoảng 100.000 câu song ngữ English-Vietnamese và bộ dữ liệu khoảng 600 ngàn câu được thu thập từ TED và bộ dữ

liệu 20 ngàn câu do nhóm sinh viên được lấy từ các trang báo và sách song ngữ , với dữ liệu thô chia làm hai tập tin tiếng English-Vietnamese như sau:

```
It &apos;s a huge amount of stuff . It &apos;s equal to the weight of methane .
And because it &apos;s so much stuff , it &apos;s really important for the atmospheric system .
Because it &apos;s important to the atmospheric system , we go to all lengths to study this thing .
We blow it up and look at the pieces .
This is the EUPHORE Smog Chamber in Spain .
Atmospheric explosions , or full combustion , takes about 15,000 times longer than what happens in your car .
But still , we look at the pieces .
We run enormous models on supercomputers ; this is what I happen to do .
Our models have hundreds of thousands of grid boxes calculating hundreds of variables each , on minute timescales .
And it takes weeks to perform our integrations .
And we perform dozens of integrations in order to understand what &apos;s happening .
We also fly all over the world looking for this thing .
I recently joined a field campaign in Malaysia . There are others .
```

```
Đó là một lượng khí thải khổng lồ , bằng tổng trọng lượng của mêtan .
Chính vì lượng khí thải rất lớn , nó có ý nghĩa quan trọng với hệ thống khí quyển .
Chính vì nó có ý nghĩa quan trọng với hệ thống khí quyển , giá nào chúng tôi cũng theo đuổi nghiên cứu này đến cùng .
Chúng tôi cho nó nổ và xem xét từng mảnh nhỏ .
Đây là Phòng nghiên cứu khói bụi EUPHORE ở Tây Ban Nha .
Nổ trong không khí hay cháy hoàn toàn diễn ra chậm hơn 15,000 lần so với những phản ứng trong động cơ xe .
Dù vậy , chúng tôi vẫn xem xét từng mảnh nhỏ .
Chúng tôi chạy những mô hình khổng lồ trên siêu máy tính ; đây là công việc của tôi .
Mô hình của chúng tôi gồm hàng trăm ngàn thùng xếp chồng tính toán với hàng trăm biến số trong thời gian cực ngắn .
Mà vẫn cần hàng tuần mới thực hiện xong các phép tích phân .
Chúng tôi cần làm hàng tá phép tính như thế để hiểu được những gì đang xảy ra .
Chúng tôi còn bay khắp thế giới để tìm phân tử này .
Gần đây tôi tham gia một cuộc khảo sát thực địa ở Malaysia . Còn nhiều chuyến khác nữa .
```

❖ Bước 1

Bước đầu tiên chúng ta thực hiện xử lí các câu dữ liệu như: xoá dấu “?”, “.”, xoá dấu khoảng trắng thừa và một số thứ khác.

Nhóm sinh viên thực hiện loại bỏ các câu có độ dài hơn 100.

❖ Bước 2

Bước thứ hai, ta thực hiện tách từ để tạo từ điển của từng ngôn ngữ theo tập dữ liệu mà ta sử dụng.

```
The first 15 words:
 ['the', 'science', 'behind', 'a', 'climate', 'headline', 'i', '&', 'apos', ';', 'd', 'like', 'to', 'talk'

The first 15 words:
 ['khoa', 'học', 'đằng', 'sau', 'một', 'tiêu', 'đề', 'về', 'khí', 'hậu', 'tôi', 'muốn', 'cho', 'các', 'bạn
```

❖ Bước 3

Tiếp theo nhóm sinh viên thực hiện tạo các từ điển word2int và int2word cho cả hai ngôn ngữ English-Vietnamese và ta được kết quả như sau:

English:

```
The word2index:
 {'<pad>': 0, '<unk>': 1, '<s>': 2, '</s>': 3, '.': 4, ',': 5, 'the': 6, ';': 7, '&': 8, 'and': 9, 'apos': 10, 'to': 11, 'of': 12,

The int2word:
 {0: '<pad>', 1: '<unk>', 2: '<s>', 3: '</s>', 4: '.', 5: ',', 6: 'the', 7: ';', 8: '&', 9: 'and', 10: 'apos', 11: 'to', 12: 'of',
```

Vietnamese:

```
The word2index:
 {'<pad>': 0, '<unk>': 1, '<s>': 2, '</s>': 3, '.': 4, ',': 5, 'tôi': 6, 'là': 7, 'và': 8, 'có': 9, 'một': 10

The int2word:
 {0: '<pad>', 1: '<unk>', 2: '<s>', 3: '</s>', 4: '.', 5: ',', 6: 'tôi', 7: 'là', 8: 'và', 9: 'có', 10: 'một'
```

❖ Bước 4

Tiếp theo, nhóm sinh viên thực hiện chuyển từng câu song ngữ sang từng vec-tơ với từng từ ứng với vị trí của từ trong từ điển.

Với câu đầu vào tiếng Anh ta thực hiện thêm (padding) với những câu có độ dài bé hơn 100.

Với câu đầu vào tiếng Việt ta sẽ thực hiện thêm (padding) trong quá trình huấn luyện vì ta sẽ sử dụng độ dài thật của câu song ngữ để huấn luyện nhanh hơn. Và ta được kết quả như sau:

```
English:  [6, 310, 573, 13, 749, 4626, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
Vietnamese:  [326, 75, 1083, 116, 10, 372, 117, 41, 411, 743]
```

❖ Bước 5

Tại đây ta thực hiện lấy nhúng từ (word embedding) tất cả các từ có trong từ điển English-Vietnamese.

English:

```
The first 5 word embedding for English:
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0.]
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0.]
[-4.81609667e-01 -1.13084684e+00  1.36512550e-01  9.60756876e-01
 -1.88665400e-03 -5.57107444e-01  1.86223564e-01  1.11050531e+00
 -1.45235121e+00  8.51740666e-01 -3.01429590e-01  6.98918766e-01
  1.91483079e+00  1.37560757e-01 -2.39496092e+00  8.18544131e-01
  2.94883292e-01 -2.29742509e-01  1.63064879e+00  2.27782670e+00
  2.52556118e-01  2.32475884e-01  3.00857270e-01 -4.59514171e-01
 -1.08016205e+00 -3.71282781e-01 -1.36559598e+00 -9.10259979e-01
  7.26468332e-01  9.63630600e-02  5.44439876e-01 -1.45772109e+00
  1.19695592e+00  1.96201156e+00  1.41218486e+00  9.90601780e-01
 -4.11389629e-01  2.15227659e+00 -1.17872550e-01 -8.50457036e-01
  9.46037772e-01 -9.31007546e-01 -1.31065769e+00 -6.75534062e-02
 -7.61294201e-01 -1.06508788e+00 -1.55671767e+00  2.45023667e-01
 -8.48295748e-01 -1.36905856e+00 -1.00542131e+00  6.51051976e-01
 -4.50359761e-01 -2.82689957e+00 -6.57419200e-01 -1.80564905e-01
 -8.97636047e-01 -7.27110951e-01 -7.48422020e-01  9.82069266e-01
 -1.57249489e+00 -6.22635630e-01  2.89696859e-01 -1.32908870e+00
  8.24747705e-01 -1.20338627e+00 -5.49458917e-01 -2.72685047e-01
 -1.96749162e-03 -2.12043456e+00  1.46996242e+00  1.20262489e+00
  6.82162929e-01  3.67140082e-01 -1.81595542e+00  4.89857799e-01
  1.44245119e+00  6.41830432e-01 -2.83413216e+00  1.44971137e-01
 -1.28437888e+00 -1.30330778e+00  1.30202119e-02 -1.90893174e+00
 -1.16707235e+00  8.03624170e-01 -1.17234431e+00 -1.16265515e+00
  2.02894562e+00 -1.38666005e-01 -5.97084920e-01  2.90895875e-01
  9.08091787e-01  6.69844261e-01  2.80325531e-01  9.00879731e-01
  9.11844102e-01  6.76626996e-01 -8.58046664e-01  9.13045394e-01]
```

Vietnamese:

```
The first 5 word embedding Vietnamese:
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0.]
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0.]
[-0.01044086  0.90513951 -0.43956137 -0.69416605  0.16608956  0.90446738
 -0.28968965  1.27596079  0.68143317 -0.00693139  0.98952497 -0.64811462
  0.07302196 -1.38816016 -0.75803859 -0.11468189  0.17046202 -0.84500204
 -0.83069909  0.77976883 -0.6798142  -0.4080631   0.56602816  0.4034904
  0.57286388  1.32884753 -0.01333768 -0.04100347 -1.31925041 -0.01983607
  1.95748183 -0.18242306 -0.80583398  0.24760619 -0.89519989  1.57125641
  1.31849534 -0.0532795   0.8136146  -0.46576611 -1.26875609  1.15089051
 -0.52684527 -0.66380096 -0.65822303 -0.91495945 -1.00100071 -0.33117292
 -0.019482   -0.06668272  0.74621559 -1.6794455   1.38093937 -0.26780666
 -0.04849546  0.22112061 -1.48352859 -0.19885498  0.50529932  0.16309623
 -0.4171322  -0.65571055  0.75096135  0.32351025 -0.25188311  0.93658109
 -0.03215914 -1.01653399  1.40352285  0.67429556  0.04346623 -0.43866205
  2.20316459  1.55618194  0.33737555  1.00814376 -1.97777662 -0.63767324
 -2.0372233  -1.7913456   1.61275751  1.08898689  1.61160879 -0.56429882
 -1.52948648  0.64205088  1.44741794  0.25214407 -0.41775636 -0.1603769
 -0.49127496  0.81636376 -0.14409392  0.96754059  1.11953453 -0.74658347
 -1.49451565  0.45239827 -0.26922032  0.90311802]
```

Sau đó quá trình chuyển từ câu song ngữ (đã được chuyển thành vec-tơ số ứng với vị trí từng từ trong từ điển) thành vec-tơ nhúng từ sẽ được thực hiện trực tiếp trong mô hình.

3.3 GIẢI PHÁP XÂY DỰNG MÔ HÌNH DỊCH MÁY

3.3.1 Tổng quan về giải pháp

Dựa trên các đánh gia thực tế và điều kiện phần cứng lẫn lượng dữ liệu (data) cho phép, nhóm sinh viên lựa chọn phương pháp học sâu (deep learning) để xây dựng mô hình mạng nơ-ron hồi quy (Recurrent neural netword) trong mô hình dịch máy (machine neural translation). Mô hình được đào tạo từ đầu đến cuối từ những câu đã được biểu diễn dưới các nhúng từ (word embedding) để tạo ra các chuỗi đầu vào bộ mã hoá (encoder) và bộ giải mã (decoder). Do đó với lượng dữ liệu đủ lớn và khả năng tính

toán, mô hình có thể tự học một cách chính xác để thực hiện việc dịch một câu từ tiếng Anh sang tiếng Việt.

3.3.2 Mô hình mạng nơ-ron hồi quy và khung huấn luyện

Cốt lõi của quá trình đào tạo một mô hình RNN là để nhận vào một văn bản tiếng Anh và tạo ra một văn bản tiếng Việt tương ứng. Để dễ hình dung ta có ví dụ một tập huấn luyện $X = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), ...\}$ với x là một vec-tơ các nhúng từ (word embedding) tương ứng với câu tiếng Anh đầu vào và y là một nhãn tức là một vec-tơ các nhúng từ (word embedding) tương tứng với câu tiếng Việt ở đầu ra. Mỗi câu tiếng Anh $x^{(i)}$ là một chuỗi thời gian có độ dài $T^{(i)}$, trong đó mỗi đoạn thời gian nhất định là một vec-tơ nhúng từ (word embedding) $x_t^{(i)}$với t = 1,2,…, $T^{(i)}$. Mục tiêu của RNN là chuyển đổi đầu vào x thành một chuỗi xác suất ký tự cho nhãn y, với $\hat{y}_t = P(w_t|x)$, trong đó w_t thuộc các từ trong từ điển tiếng Việt và một vài ký tự đặc biệt khác.

Mô hình RNN được nhóm sinh viên chọn sử dụng là mô hình hồi quy với 3 thành phần chính. Thành phần đầu tiên là lớp nhúng từ (embedding), lớp (layer) này có nhiệm vụ chuyển các đầu vào của bộ mã hoá (encoder) và bộ giải mã (decoder) từ dạng int sang dạng nhúng từ (word embedding) để phục vụ cho công việc tính toán phía sau.
Chúng tôi xây dựng lớp Conv bằng cách gán lớp CNN kết hợp với GLU vào giữa lớp Word Embedding và lớp LSTM hai chiều. Lúc này đầu vào của lớp LSTM hai chiều sẽ là một véc-tơ gồm từ nhúng và vị trí của từ đó thay vì chỉ một véc-tơ từ nhúng như mô hình sequence-to-sequence truyền thống. Mục đích khi áp dụng lớp CNN và GLU là để phân loại và loại bỏ các thông tin gây nhiễu trong chuỗi đầu vào. Trong lớp CNN mỗi lớp sẽ là một lớp CNN một chiều với n bộ lọc và mỗi bộ lọc có kích thước là k. Tuy nhiên, so với các mạng RNN thì mạng CNN chỉ tạo biểu diễn cho chuỗi đầu vào kích thước cố định. Vì vậy, chúng tôi đã giải quyết vấn đề này bằng cách xếp chồng các lớp CNN lên nhau. Qua đó giúp chúng tôi kiểm soát chính xác độ dài tối đa của chuỗi đầu vào.

Mỗi đầu ra của lớp CNN sẽ là đầu vào cho lớp GLU. Tuy nhiên, lớp GLU sẽ tách đầu vào thành hai phần một trong số đó sẽ qua một hàm sigmoid để lọc các thông tin liên quán đến các phần tử trong đầu vào. Qua đó, lớp GLU giúp kiểm soát được các thông tin đầu vào. Tương tự như LSTM, các cổng trong lớp GLU σ(B) nhân từng phần tử của ma trận A và kiểm soát được các thông tin truyền trong hệ thống với công thức như sau:

$$h_c = A \otimes \sigma(B)$$

Tuy nhiên, h_c có kích thước bằng một nữa kích thước của chuỗi đầu vào. Vì vậy, chúng tôi đã thêm các véc-tơ không đầu vào ở mỗi lớp chập có kích thước bằng với chiều dài ban đầu của chuỗi đầu vào. Sau đó, h_c sẽ là đầu vào cho lớp LSTM hai chiều để tạo ra các trạng thái ẩn cho bộ mã hóa.

Thành phần thứ hai là bộ mã hoá (encoder), với bộ mã hoá chúng ta sử dụng Multi layer Bi-directional LSTM với số lượng layer và số lượng hidden units của LSTM cell được thiết lập trong param. Ngoài ra nhóm sinh viên còn sử dụng DropoutWrapper để thiết lập giá trị Drop Out cho các LSTM cell để tránh hiện tượng quá khớp (over-fitting) với dữ liệu huấn luyện.

Thành phần thứ ba là bộ giải mã (decoder). Đối với bộ giải mã, nhóm sinh viên chia thành hai trường hợp riêng biệt là huấn luyện mô hình(training) và dự đoán (inference). Trong quá trình huấn luyện nhóm sinh viên sử dụng TrainingHelper còn khi dự đoán, nhóm sinh viên sử dụng BasicDecoder với BeamSearchDecoder kết hợp với mô hình ngôn ngữ để có thể đưa ra câu đầu ra phù hợp nhất.

❖ Huấn luyện: nhóm sinh viên sử dụng BahdanauAttention và TrainingHelper để huấn luyện mô hình. Nhóm sinh viên còn sử dụng AdamOptimizer để cập nhật tham số cho mô hình và còn sử dụng Gradient Clipping để tránh mô hình bị bùng nổ độ dốc (exploding gradients).

❖ Dự đoán: sau khi huấn luyện xong mô hình và sử dụng mô hình này để dự đoán kết quả. Tuy nhiên do chúng ta không biết kết quả thực tế như trong quá trình huấn luyện, nên ta cần sử dụng các thuật toàn tìm kiếm để cho ra kết quả phù hợp nhất và nhóm sinh viên chọn sử dụng thuật toán tìm kiếm chùm tia (Beam Search) với beam-width = 10 và kết hợp với mô hình ngôn ngữ cho tiếng Việt để kiểm tra ngữ pháp và ngữ cảnh các câu để tối ưu cho thuận toán tìm kiếm chùm tia cho ra kết quả có khả năng đúng nhất.

3.4 GIẢI PHÁP XÂY DỰNG MÁY CHỦ

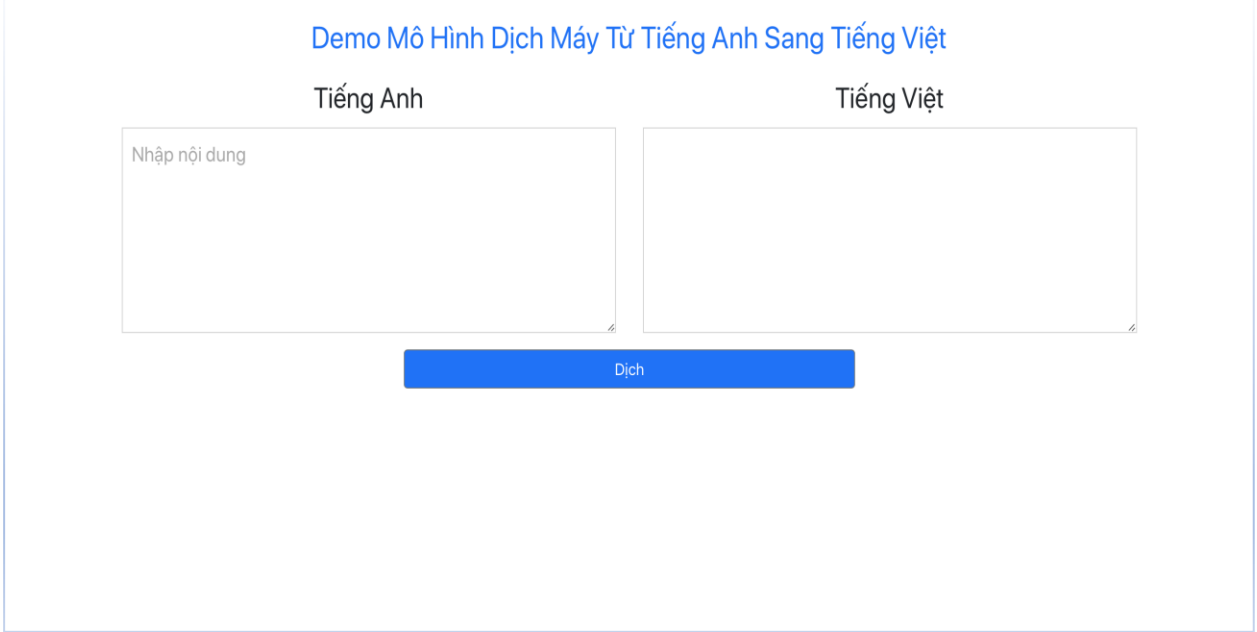
Máy chủ (server) được nhóm sinh viên chọn Amazon EC2 làm máy chủ với mục đích tạo ra một cầu nối giữa mô hình đã được huấn luyện (model) và phía ứng dụng sản phẩm (client) – được xây dựng với ReactJs. Vì vậy trong giới hạn của khoá luận, máy chủ chỉ cung cấp duy nhất một giao diện lập trình (API) với chức năng chuyển đổi từ một văn bản (text) tiếng Anh thành một văn bản (text) tiếng Việt tương ứng.

3.5 GIẢI PHÁP XÂY DỰNG ỨNG DỤNG

Để ứng dụng hoá hệ thống dịch máy từ tiếng Anh sang tiếng Việt, nhóm sinh viên quyết định xây dựng web để ứng dụng kết quả của hệ thống vào một tình huống cụ thể có thể ứng dụng và thương mại hoá tốt.
Ứng dụng web do nhóm sinh viên xây dựng có chức năng chính là chuyển đổi văn bản tiếng Anh do người dụng nhập vào và đưa ra văn bản tiếng Việt tương ứng.

3.5.1 Thiết kế giao diện ứng dụng

Giao diện ứng dụng chỉ có một màn hình với chức năng chính là chuyển đổi một văn bản tiếng Anh thành một văn bản tiếng Việt tương ứng.



Hình 3.2 Màn hình chính của ứng dụng

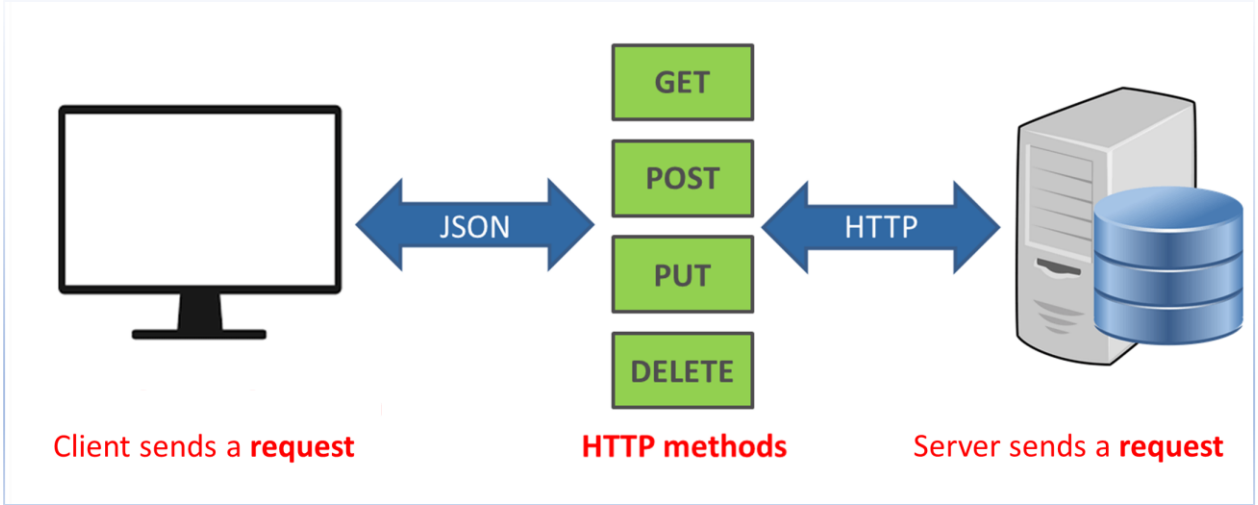
Để sử dụng, người dùng nhập văn bản tiếng Anh vào ô tiếng Anh tương ứng và nhập vào nút dịch. Kết quả sẽ được hiển thị tại ô tiếng Việt.

3.5.2 Thiết kế kiến trúc ứng dụng

Ứng dụng web minh hoạ thực tế cho mô hình dịch máy từ tiếng Anh sang tiếng Việt mà nhóm sinh viên đã xây dựng, các tác vụ không phức tạp nên nhóm sinh viên đề xuất sử dụng kiến trúc client-server cho ứng dụng của nhóm.

Kiến trúc client-server là một mô hình nổi tiếng trong mạng máy tính, được áp dụng rất rộng rãi và là mô hình của mọi trang web, ứng dụng di động hiện có. Ý tưởng của mô hình này là máy con – Client (đóng vài trò là máy khách) gửi một yêu cầu (request) để máy chủ – Server (đóng vai trò người cung ứng dịch vụ), máy chủ sẽ xử lý và trả kết quả về cho máy khách.

Kiến trúc client-server mà nhóm sinh viên sử dụng được minh hoạ như hình 3.3.



Hình 3.3: Kiến trúc client-server

[Nguồn: https://en.wikipedia.org/wiki/Client–server_model]

3.6 TỔNG KẾT

Thông qua chương 3, sinh viên đã làm rõ được các giải pháp cụ thể cho từng phần trong hệ thống dịch máy từ tiếng Anh sang tiếng Việt, hướng xây dựng máy chủ và cả ứng dụng trên nền tảng web.

Nhóm sinh viên đã trình bày một hệ thống dịch máy từ tiếng Anh sang tiếng Việt dựa trên việc học sâu (deep learning) từ đầu đến cuối có khả năng vượt trội và hiện đại trong hiện đại. Nhóm sinh viên tin rằng phương pháp này sẽ tiếp tục được cải thiện với các mô hình mới hơn, đơn giản hoặc phức tạp hơn khi tận dụng được sức mạnh tính toán phần cứng và kích thước dữ liệu được tăng thêm trong tương lai.

Chương 4 nhóm sinh viên sẽ trình bày về các thư viện, công cụ và những khó khăn cụ thể nếu có cho các giải pháp đã trình bày ở chương này.

CHƯƠNG 4: CÀI ĐẶT VÀ TRIỂN KHAI

4.1 GIỚI THIỆU VỀ PYTHON VÀ THƯ VIỆN TENSORFLOW

4.1.1 Python

Mã nguồn xây dựng mô hình huấn luyện của đề tài được phát triển dựa trên Python. Python là một ngôn ngữ lập trình thông dụng được sử dụng rất phổ biến trong lĩnh vực khoa học máy tính nhờ những ưu điểm sau:

❖ Đa nền tảng

Python có thể chạy trên nhiều hệ điều hành như Windows, MacOS, Linux/Unix và một số hệ điều hành khác trên máy tính. Ngoài ra, Python còn có cả những phiên bản chạy được trên .NET, máy ảo Java. Tất cả chỉ với cùng một mã nguồn cho một công việc.

❖ Đơn giản

Python có cú pháp rất đơn giản, rõ ràng. Cú pháp của Python dễ viết và dễ đọc hơn rất nhiều khi so sánh với những ngôn ngữ lập trình khác như Java, C/C++, C#, JavaScript, ... Điều này cũng giúp cho nhà phát triển tập trung vào việc phát triển giải pháp thay vì cú pháp.

❖ Mã nguồn mở

Python là một dự án mã nguồn mở nên nhà phát triển có thể thoải mái sử dụng cho các mục đích cá nhân và vì vậy nên cộng đồng phát triển Python thường xuyên đưa ra những bản cập nhật mới nhằm tăng trải nghiệm cũng như tối ưu hoá Python.

❖ Nhiều thư viện hỗ trợ

Python có một khối lượng lớn các thư viện tiêu chuẩn giúp cho công việc của nhà phát triển trở nên dễ dàng hơn rất nhiều, đặc biệt là các thư viện xử lý toán học của Python cực kỳ đa dạng và mạnh mẽ.

4.1.2 Tensorflow

Thư viện Tensorflow được sử dụng trong việc tính toán các biểu đồ và các dữ liệu dưới dạng số hoá trong sản phẩm khoá luận. Là một thư viện mã nguồn mở hỗ trợ mạnh mẽ các phép toán học để tính toán trong máy học. Để xây dựng một mô hình huấn luyện cho đề tài, nhóm sử dụng các giao diện lập trình cấp thấp (low level APIs) mà Tensorflow cung cấp:

❖ Tensor

Đây là một sự khái quát hóa các vectơ và ma trận cho các kích thước có khả năng cao hơn. Là cấu trúc dữ liệu đại diện cho tất cả các loại dữ liệu trong Tensorflow. Một tensor sẽ có 3 thuộc tính cơ bản nhất bao gồm:

- Số bậc (rank): giúp phân loại dữ liệu của tensor. (Scalar, Vector, Matrix, N-Tensor)
- Số chiều (shape): giúp xác định mức độ tương hợp giữa các tensor khi thực hiện tính toán.
- Kiểu dữ liệu (type): kiểu dữ liệu cho toàn bộ các thành phần (elements) trong tensor.

❖ Graph

Đây là một loại đồ thị với các đỉnh (node) là đại diện cho biến đầu vào hoặc một phép tính toán và các cạnh (edge) là đại diện cho dữ liệu truyền bên trong đồ thị tức dữ liệu đầu vào và đầu ra của các phép tính tại một đỉnh. Và trong tensorflow, tất cả thành phần bên trong một đồ thị đều ở dạng tensor. Cách xử lý tính toán theo hướng đồ thị này có thể giúp tensorflow tận dụng được khả năng tính toán song song bằng việc chia tách các phép toán độc lập và khả năng phân tán khi chia nhỏ công việc xử lý cho nhiều CPU, GPU khác nhau.

❖ Session

Đây là một phiên xử lý được định nghĩa trong thư viện tensorflow. Một đối tượng phiên (session) cung cấp quyền truy cập vào các thiết bị trong máy cục bộ và các thiết bị từ xa bằng cách sử dụng thời gian chạy phân tán. Nó cũng lưu trữ thông tin về đồ thị (graph) để có thể chạy cùng một tính toán hiệu quả nhiều lần. Nếu không có phiên (session), mọi tính toán trong đồ thị (graph) sẽ gần như không được triển khai.

4.2 DỮ LIỆU HUẤN LUYỆN MÔ HÌNH

Để huấn luyện một hệ thống dịch máy từ tiếng Anh sang tiếng Việt dựa trên mô hình nhóm sinh viên phát triển thì bộ dữ liệu nhóm sử dụng là "IWSLT'15 English-Vietnamese data" bao gồm khoảng 133.000 câu do đại học Stanford phát triển và khoảng 600.000 câu được sưu tầm trên TED và 20.000 câu do nhóm sinh viên thu thập ở các trang báo, sách song ngữ:

❖ Văn bản tiếng Anh

Để huấn luyện hệ thống dịch máy từ tiếng Anh sang tiếng Việt đủ tốt thì lượng dữ liệu văn bản dùng để huấn luyện cũng phải đủ nhiều và đủ tốt. Nhóm sinh viên đã thu thập được khoảng 800.000 câu song ngữ để tiến hành huấn luyện.

❖ Văn bản tiếng Việt

Là bản dịch tương ứng với nội dung của câu tiếng Anh. Dữ liệu tiếng Việt với khoảng 800.000 câu.

Trong quá trình thu thập dữ liệu, nhóm sinh viên đã gặp rất nhiều vấn đề về chất lượng dữ liệu như: các tập dữ liệu song ngữ English – Vietnamese có khá nhiều với các dự án, tuy nhiên các dự án này lại không công khai dữ liệu nên nhóm sinh viên phải thu thập khắp nơi. Đối với những mẫu có mức độ sai lệch nhỏ nhóm cố gắng tinh chỉnh sao cho phù hợp nhất. Những mẫu bị sai lệch nhiều hoặc chất lượng quá thấp buộc nhóm sinh viên phải bỏ. Việc này một phần sẽ giảm bớt tình trạng gây nhiễu cho mô hình trong quá

trình huấn luyện. Điều này dẫn đến thời gian huấn luyện mô hình còn khoảng 20 giờ cho khoảng 800.000 câu song ngữ. Trong đó, dữ liệu được chia nhỏ thành 3 bộ train, dev, test với kích thước như sau:

- ❖ Bộ train: chứa khoảng 800.000 câu song ngữ English-Vietnamese.
- ❖ Bộ dev: chứa khoảng 1500 câu song ngữ English-Vietnamese.
- ❖ Bộ test: chứa khoảng 1200 câu song ngữ English-Vietnamese.

4.3 CÀI ĐẶT

4.3.1 Giới thiệu

Mã nguồn được nhóm sinh viên phát triển dựa trên tham khảo các bài báo như: *Sequence to Sequence Learning with Neural Networks* [6] do nhóm tác giả đến từ google được ông bố vào năm 2014 tại Silicon Valley AI Lab đã trình bày ý tưởng cụ thể để xây dựng một mô hình mạng nơ-ron hồi quy tối ưu với hướng đi mới so với các hệ thống dịch máy truyền thống kết hợp cùng với cơ chế chú ý (Attention mechanism) từ bài báo *Neural Machine Translation by Jointly Learning to Align and Translate* [7] được thực hiện vào năm 2014.

Từ đó nhóm sinh viên tự phát triển mô hình dịch máy cho tác vụ dịch tiếng Anh sang tiếng Việt để phục vụ cho luận văn. Mô hình được pháp triển trên Python3 và thư viện Tensorflow là chính.

4.3.2 Cài đặt

Đầu tiên ta cần tải về mã nguồn mô hình dịch máy của nhóm sinh viên phát triển từ github (https://github.com/nmtri1912/Model).
Phần hướng dẫn cài đặt của nhóm sinh viên yêu cầu bắt buộc về những thư viện cũng như công cụ mà nhóm sinh viên có khuyến nghị trên liên kết github phía trên để có thể chạy mô hình của nhóm:

- ❖ Python 3.6

- ❖ Tensorflow 1.x
- ❖ Hệ điều hành MacOS hoặc Linux, Ubuntu

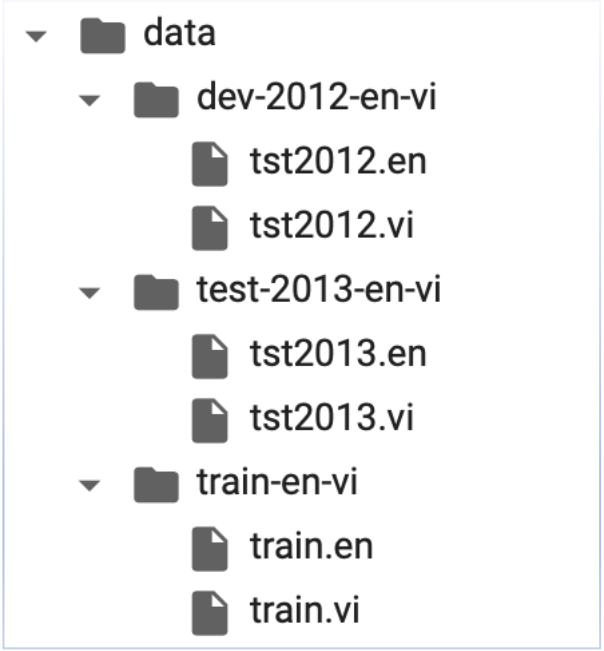
Để huấn luyện mô hình dịch máy cho tác vụ dịch ta cần cài đặt theo hướng dẫn của nhóm sinh viên để tránh gặp lỗi không đáng có.

4.4 HUẤN LUYỆN MÔ HÌNH

Để huấn luyện mô hình với bộ dữ liệu câu Anh-Việt, ta cần chuyển thành các số để mô hình có thể huấn luyện và phương pháp đó là sử dụng từ nhúng (word embedding). Nhóm sinh viên sử dụng pre-trained model Word2vec tại http://vectors.nlpl.eu/repository/ cho tiếng Anh và tiếng Việt có thông số như sau:

- ❖ English CoNLL17 corpus (ID = 40): được xây dựng dựa trên thuật toán "Word2Vec continuons Skipgram" với 100 chiều và hơn 4.000.000 từ.
- ❖ Vietnamese CoNLL17 corpus (ID = 74): được xây dựng dựa trên thuật toán "Word2Vec continouns Skipgram" với 100 chiều và khoảng 3.800.000 từ.

Để có thể huấn luyện mô hình dịch máy nhóm sinh viên cần một lượng lớn dữ liệu các câu song ngữ. Nhóm sinh viên đã sưu tầm các nguồn dữ liệu có sẵn như bộ dữ liệu chuẩn IWSLT 2015 và thu thập thêm dữ liệu từ các trang báo, sách song ngữ có được khoảng 20.000 câu và thu được tổng cộng khoảng 800 ngàn câu song ngữ Anh-Việt. Bộ dữ liệu được chia thành ba bộ train-dev-test được tổ chức như sau:



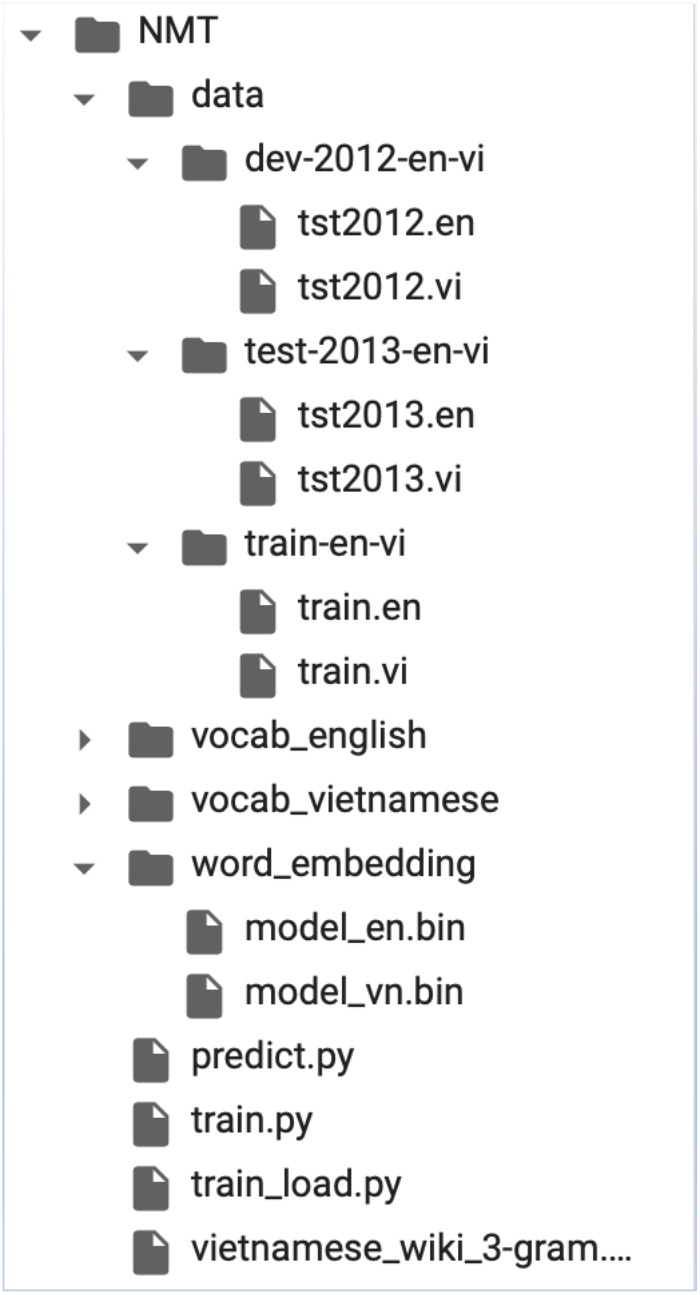
Trong đó:

Thư mục train-en-vi chứa 2 tập tin , tập tin train.en chứa các câu tiếng Anh được tổ chức thành từng dòng khác nhau. Tập tin train.vi chứa các câu tiếng Việt tương ứng và cũng được tổ chức thành các dòng như đã trình bày ở chương 3.

Thư mục test-2013-en-vi: chứa 2 tập tin song ngữ với 1268 dòng dùng để đánh giá mô hình có hiệu quả hay không.

Thư mục dev-2012-en-vi: chứa 2 tập tin song ngữ dùng để đánh giá mô hình lúc huấn luyện để điều chỉnh.

Để phục vụ cho công việc huấn luyện cũng như dự đoán một câu tiếng Anh sang tiếng Việt chúng ta cũng cần chuẩn bị các mô hình biểu diễn từ như mô hình word_embedding cũng như mô hình ngôn ngữ để kiểm tra ngữ nghĩa của câu đầu ra. Cấu trúc thư mục được tổ chức như sau.



Trong đó:

Thư mục data: chứa dữ liệu huấn luyện và đánh giá.

Thư mục vocab_english, vocab_vietnamese: chứa từ điển của ngôn ngữ tiếng Anh và tiếng Việt sẽ được tạo ra trong quá trình xử lí dữ liệu.

Thư mục word_embedding: chứa mô hình nhúng từ của ngôn ngữ tiếng Anh (model_en.bin) và tiếng Việt (model_vn.bin).

Để mô hình có thể sử dụng mô hình để đưa ra các dự đoán dịch chính xác, ta cần tìm ra các tham số phù hợp cho mô hình, nhóm sinh viên tham khảo và tiến hành công việc huấn luyện mô hình (điều chỉnh các siêu tham số - hyperparameters) để tìm ra những giá trị tối ưu để mô hình dự đoán chính xác nhất.
Để huấn luyện mô hình chúng ta cần chạy câu lệnh:

```
python3 train.py
  --language_src data/train-en-vi/train.en
  --language_targ data/train-en-vi/train.vi
  --vocab_src vocab_english/
  --vocab_targ vocab_vietnamese/
  --word_emb_src word_embedding/model_en.bin
  --word_emb_targ word_embedding/model_vn.bin
  --num_layer 2
  --num_hiddens 512
  --learning_rate 0.001
  --keep_prob 0.85
  --beam_width 10
  --batch_size 128
  --checkpoint NMT.ckpt
```

Quá trình huấn luyện được nhóm triển khai trên Google Colab có cấu hình GPU 25GB. Sau một thời gian huấn luyện với bộ dữ liệu "IWSLT'15 English-Vietnamese data" cho việc đánh giá mô hình và 800.000 câu phục vụ cho mô hình triển khai demo với 10 epoch và ghi nhận kết quả, nhóm sinh viên đã tổng hợp một số siêu tham số có ảnh hưởng đến kết quả dự đoán của mô hình như:

- num_layers: số lớp của mô hình
- num_hidden: độ rộng của một lớp khi khởi tạo trong mô hình
- batch_size: số lượng mẫu được sử dụng khi thực hiện bước tối ưu cho mô hình

4.4.1 Điều chỉnh num_layer

- Chọn num_layers = 1

Tên tham số	Giá trị	Ghi chú
num_layers	1	
num_hidden	256	Giá trị mặc định
batch_size	64	Giá trị mặc định
BLEU	**26.80**	

- Chọn num_layers = 2

Tên tham số	Giá trị	Ghi chú
num_layers	2	
num_hidden	256	Giá trị mặc định
batch_size	64	Giá trị mặc định
BLEU	**28.60**	

Kết luận: Chọn num_layers = 2 cho các lần huấn luyện sau

4.4.2 Điều chỉnh num_hidden

- Chọn num_hidden = 256

Tên tham số	Giá trị	Ghi chú
num_layers	2	
num_hidden	256	
batch_size	64	Giá trị mặc định
BLEU	**28.60**	

- Chọn num_hidden = 512

Tên tham số	Giá trị	Ghi chú
num_layers	2	

num_hidden	512	
batch_size	64	Giá trị mặc định
BLEU	**29.42**	

Kết luận: Chọn num_hidden = 512 cho các lần huấn luyện sau.

4.4.3 Điều chỉnh batch_size

- Chọn batch_size = 64

Tên tham số	Giá trị	Ghi chú
num_layers	2	
num_hidden	512	
batch_size	64	
BLEU	**29.42**	

- Chọn batch_size = 128

Tên tham số	Giá trị	Ghi chú
num_layers	2	
num_hidden	512	
batch_size	64	
BLEU	**29.63**	

Kết luận chọn batch_size = 128 cho lần huấn luyện sau.

Như vậy mô hình của nhóm sinh viên sử dụng có các tham số như sau:

- Num_layers = 2
- Num_hidden = 512
- Batch_size = 128
- Beam_width = 10
- Keep_prob = 0.85

4.5 ĐÓNG GÓI MÔ HÌNH

Nhóm sinh viên lưu các tham số mô hình học được sau mỗi lần chạy xong 1 epoch (một lần duyệt qua toàn tập huấn luyện) với định dạng **NMT.ckpt.** Tập tin này có thể hiểu là các tham số được chọn lọc trong quá trình huấn luyện.

Để sử dụng tập tin này để thực hiện tác vụ dịch máy, chúng ta cần định nghĩa lại một số thư cần thiết như: từ điển word2int, int2word, từ nhúng (word embedding), xử lí đầu vào và mô hình.

Để có thể sử dụng mô hình để dịch một tập tin tiếng Anh sang tiếng Việt ta chạy câu lệnh sau:

```
python3 predict.py
  --language_src data/test-2013-en-vi/tst2013.en
  --language_targ data/test-2013-en-vi/tst2013.vi
  --vocab_src vocab_english/
  --vocab_targ vocab_vietnamese/
  --word_emb_src word_embedding/model_en.bin
  --word_emb_targ word_embedding/model_vn.bin
  --num_layer 2
  --num_hiddens 512
  --learning_rate 0.001
  --keep_prob 0.85
  --beam_width 10
  --batch_size 128
  --checkpoint NMT.ckpt
```

4.6 XÂY DỰNG MÁY CHỦ (SERVER)

Flask Framework, AWS EC2 (hoặc AWS Elastic Beanstalk), Google Cloud Platform và Heroku là bốn nền tảng được nhóm sinh viên chọn để xây dựng hệ thống máy chủ nhằm đóng vai trò làm cầu nối giữa ứng dụng và mô hình dịch máy. Với các yếu tô như tốc độ

triển khai nhanh gọn, sự tiện ích và tính thông dụng nên việc chọn hai nền tảng này để xây dựng máy chủ là quyết định phù hợp với nhu cầu đặt ra của nhóm sinh viên.
Hệ thống máy chủ trong giới hạn luận văn này sẽ cung cấp ra bên ngoài duy nhất một giao diện lập trình ứng dụng (Application Programming Interface-API) để chuyển đổi văn bản tiếng Anh (dạng text) nhận được và trả về dữ liệu văn bản tiếng Việt tương ứng. Trong khi đó Web ứng dụng để sử dụng API mà hệ thống cung cấp, nhóm sinh viên sử dụng Reactjs Framework để xây dựng giao diện và được triển khai trực tiếp lên Heroku.

4.7 MỘT SỐ VẤN ĐỀ PHÁT SINH VÀ GIẢI PHÁP

- Thiếu nguồn dữ liệu: nhóm sinh viên thu tập thêm dữ liệu mới bằng cách tìm thêm một số nguồn dữ liệu được đóng góp và tự tạo dữ liệu bằng các trang báo, sách song ngữ.
- Thiếu tài nguyên để huấn luyện mô hình (GPU): nhóm sinh viên sử dụng dịch vụ google colab của google.
- Khi deploy lên server gặp phải trường hợp front-end gọi API được 2 đến 3 lần thì server bị tắt: nguyên nhân do máy chủ ở EC2 bị giới hạn nên nhóm sinh viên chuyển sang sài máy chủ tại Google Could Platform.
- Khi huấn luyện trên google colab thì gặp phải vấn đề giới hạn của google colab: train tối đa khoảng 10-12 tiếng sẽ bị mất kết nối, hoặc có sự cố phát sinh thì sẽ mất kết quả huấn luyện trước đó: nhóm sinh viên đặt checkpoint để lưu lại kết quả sau mỗi epoch. Khi mất kết nối thì ta chỉ cần tải lại checkpoint để huấn luyện tiếp mà không phải huấn luyện lại từ đầu.

4.8 TỔNG KẾT

Trong chương 4, nhóm sinh viên đã trình bày về cách thức cài đặt và triển khai cho các thành phần bao gồm hệ thống dịch máy, trang web chạy thử API của hệ thống. Nội dung chi tiết cho một số phần cài đặt được nhóm sinh viên trình bày chi tiết ở phần phụ lục, chương 5 sẽ là các tổng kết về quá trình thược hiện luận văn của nhóm sinh viên.

CHƯƠNG 5 : TỔNG KẾT VÀ ĐÁNH GIÁ

5.1 KIẾN THỨC ĐẠT ĐƯỢC

Trong thời gian thực hiện khoá luận tốt nghiệp, nhóm sinh viên đã được cung cấp thêm nhiều điều mới, kiến thức mới:

- Có những kiến thức tổng quan về dịch máy và các kỹ thuật trong mạng nơ-ron để dịch máy.
- Có được các kiến thức và kinh nghiệm trong việc thiết kế, xây dựng, triển khai và kiểm thử khi thực hiện phát triển một hệ thống cung cấp dịch vụ.
- Biết được quá trình phát triển của dịch máy.
- Học hỏi các quy trình phát triển dự án phần mềm như Kanban, Waterfall, Scrum và có cơ hội được áp dụng vào luận văn.
- Nâng cao khả năng làm việc nhóm và giao tiếp của các thành viên trở nên tốt hơn.
- Khả năng tìm kiếm, đọc tài liệu và sách báo nâng cao. Hình thành được các thói quen như trích dẫn tài liệu. Khả năng tổng hợp các kiến thức từ nhiều nguồn khác nhau được nâng cao.
- Nâng cao được khả năng lên kế hoạch và đảm bảo hoàn thành trong khoảng thời gian cho trước.
- Hình thành kĩ năng tự chủ, tự tập và tinh thần trách nhiệm công việc.
- Học hỏi được cách trình bày, viết tài liệu một cách hợp lí và đẹp mắt.
- Có thêm kinh nghiệm đọc hiểu, hiệu chỉnh từ mã nguồn đã được phát triển. Khả năng chỉnh sửa và khắc phục khi gặp lỗi được nâng cao.

5.2 KẾT QUẢ MÔ HÌNH HUẤN LUYỆN

Kết quả được đánh giá khi sử dụng thư viện nltk phiên bản 3.2.5 và so sánh với các mô hình ứng với các bài báo tại

https://paperswithcode.com/sota/machine-translation-on-iwslt2015-english-1

Mô hình	Bộ dữ liệu	BLEU score (test)
CNN + GLU + Bi-LSTM Encoder + LSTM Decoder + Language Model (nhóm sinh viên phát triển)	IWSLT English-Vietnamese	29.63
Transformer + BPE + dropout	IWSLT English-Vietnamese	33.27
Transformer + BPE + Fix Norm + ScaleNorm	IWSLT English-Vietnamese	32.8
Transformer + LayerNorm-simple	IWSLT English-Vietnamese	31.4

Những mô hình được so sánh sử dụng kiến trúc mô hình mới đó là Transformer. Mô hình này đã được minh chứng có hiệu quả tốt hơn trong tác vụ dịch máy.
Bảng sau đây thể hiện kết quả so sánh dịch 20 câu tiếng Anh sang tiếng Việt bằng các công cụ gồm: Luận Văn, Google Translate (GT), Bing Microsoft Translator (BMT) và Cambridge Translator (CT).

Công Cụ Dịch	Đầu vào	Đầu ra
Luận văn	what is your name?	tên bạn là gì?
GT	what is your name?	tên của bạn là gì?
BMT	what is your name?	bạn tên là gì?
CT	what is your name?	bạn tên là gì?
Luận văn	it’s a question that has lasted for thousands of years since people became aware	nó đã tồn tại từ hàng ngàn năm qua kể từ khi mọi người nhận ra

GT	it’s a question that has lasted for thousands of years since people became aware	đó là một câu hỏi đã tồn tại hàng ngàn năm kể từ khi mọi người biết đến
BMT	it’s a question that has lasted for thousands of years since people became aware	đó là một câu hỏi đã kéo dài cho hàng ngàn năm kể từ người dân đã trở thành nhận thức
CT	it’s a question that has lasted for thousands of years since people became aware	đó là một câu hỏi đã kéo dài cho hàng ngàn năm kể từ người dân đã trở thành nhận thức
Luận văn	to sum up, health is the most important, invaluable asset	tóm lại , sức khỏe là tài sản vô cùng quan trọng và vô giá .
GT	to sum up, health is the most important, invaluable asset	tóm lại, sức khỏe là tài sản quan trọng nhất, vô giá
BMT	to sum up, health is the most important, invaluable asset	tóm lại, sức khỏe là tài sản vô cùng quan trọng nhất
CT	to sum up, health is the most important, invaluable asset	tóm lại sức khỏe là tài sản vô cùng quan trọng nhất
Luận văn	are you okay?	bạn ổn chứ?
GT	are you okay?	bạn có ổn không?
BMT	are you okay?	bạn có ổn không?
CT	are you okay?	bạn có ổn không?
Luận văn	then, using the correct marketing mix, marketing groups make decisions that lead to customer’s satisfaction	sau đó , sử dụng kết nối tiếp thị , các nhóm tiếp thị đưa ra quyết định dẫn đến sự thỏa mãn khách hàng
GT	then, using the correct marketing mix, marketing	sau đó, bằng cách sử dụng hỗn hợp tiếp thị chính xác, các nhóm

	groups make decisions that lead to customer’s satisfaction	tiếp thị đưa ra quyết định dẫn đến sự hài lòng của khách hàng
BMT	then, using the correct marketing mix, marketing groups make decisions that lead to customer’s satisfaction	sau đó, sử dụng kết hợp tiếp thị chính xác, các nhóm tiếp thị đưa ra quyết định dẫn đến sự hài lòng của khách hàng
CT	then, using the correct marketing mix, marketing groups make decisions that lead to customer’s satisfaction	sau đó , sử dụng kết hợp tiếp thị chính xác, các nhóm tiếp thị đưa ra quyết định dẫn đến sự hài long của khách hàng
Luận văn	facebook is one of the most popular and biggest social media globally that most of the people worldwide use and access everyday	facebook là một trong những phương tiện xã hội phổ biến và lớn nhất mà hầu hết mọi người trên toàn cầu sử dụng và truy cập mỗi ngày
GT	facebook is one of the most popular and biggest social media globally that most of the people worldwide use and access everyday	facebook là một trong những phương tiện truyền thông xã hội phổ biến nhất và lớn nhất trên toàn cầu mà hầu hết mọi người trên toàn thế giới sử dụng và truy cập hàng ngày.
BMT	facebook is one of the most popular and biggest social media globally that most of the people worldwide use and access everyday	facebook là một trong những phổ biến nhất và lớn nhất xã hội phương tiện truyền thông trên toàn cầu mà hầu hết người dân trên toàn thế giới sử dụng và truy cập hàng ngày

CT	facebook is one of the most popular and biggest social media globally that most of the people worldwide use and access everyday	facebook là một trong những phổ biến nhất và lớn nhất xã hội phương tiện truyền thông trên toàn cầu mà hầu hết người dân trên toàn thế giới sử dụng và truy cập hàng ngày
Luận văn	we all know that natural disasters happen all over the world	chúng ta đều biết rằng những thảm họa tự nhiên xảy ra trên toàn thế giới
GT	we all know that natural disasters happen all over the world	Chúng ta đều biết rằng thiên tai xảy ra trên toàn thế giới
BMT	we all know that natural disasters happen all over the world	Chúng ta đều biết rằng Thiên tai xảy ra trên toàn thế giới
CT	we all know that natural disasters happen all over the world	chúng ta đều biết rằng thiên tai xảy ra trên toàn thế giới
Luận văn	how old are you?	bạn bao nhiêu tuổi?
GT	how old are you?	bạn bao nhiêu tuổi?
BMT	how old are you?	bạn bao nhiêu tuổi?
CT	how old are you?	bạn bao nhiêu tuổi?
Luận văn	what are your hobbies?	sở thích của bạn là gì ?
GT	what are your hobbies?	sở thích của bạn là gì ?
BMT	what are your hobbies?	sở thích của bạn là gì?
CT	what are your hobbies?	sở thích của bạn là gì ?
Luận văn	what’s up?	có chuyện gì?
GT	what’s up?	Có chuyện gì vậy?

BMT	what’s up?	sao rồi?
CT	what’s up?	sao rồi?
Luận văn	today I am here to present to you about deep learning	hôm nay tôi đến đây để trình bày với các bạn về việc học sâu
GT	today I am here to present to you about deep learning	hôm nay tôi ở đây để trình bày với bạn về học tập sâu
BMT	today I am here to present to you about deep learning	hôm nay tôi ở đây để trình bày cho bạn về Deep Learning
CT	today I am here to present to you about deep learning	hôm nay tôi ở đây để trình bày cho bạn về deep learning
Luận văn	thank you for your listening	cảm ơn vì đã lắng nghe
GT	thank you for your listening	cảm ơn các bạn đã lắng nghe
BMT	thank you for your listening	cảm ơn bạn đã lắng nghe của bạn
CT	thank you for your listening	cảm ơn you đã lắng nghe của you
Luận văn	what do you like doing in your free time?	bạn thích gì trong thời gian rảnh rỗi ?
GT	what do you like doing in your free time?	bạn thích làm gì trong thời gian rảnh?
BMT	what do you like doing in your free time?	những gì bạn thích làm trong thời gian rảnh của bạn?
CT	what do you like doing in your free time?	những gì bạn thích làm trong thời gian rãnh của bạn?
Luận văn	I can’t say for sure.	tôi không thể nói chắc chắn
GT	I can’t say for sure.	tôi không thể nói chắc chắn.
BMT	I can’t say for sure.	tôi có thể nói chắc chắn.
CT	I can’t say for sure.	tôi không thể nói chắc chắn

Luận văn	police officers have a lot of authority in society and they are very important people for enforcing the law and ensuring the safety.	cảnh sát có nhiều quyền lực trong xã hội và họ là những người rất quan trọng để thực thi luật pháp và đảm bảo an toàn .
GT	police officers have a lot of authority in society and they are very important people for enforcing the law and ensuring the safety.	các sĩ quan cảnh sát có rất nhiều thẩm quyền trong xã hội và họ là những người rất quan trọng để thực thi luật pháp và đảm bảo sự an toàn.
BMT	police officers have a lot of authority in society and they are very important people for enforcing the law and ensuring the safety.	sĩ quan cảnh sát có rất nhiều thẩm quyền trong xã hội và họ là những người rất quan trọng để thực thi pháp luật và đảm bảo sự an toàn.
CT	police officers have a lot of authority in society and they are very important people for enforcing the law and ensuring the safety.	sĩ quan cảnh sát có rất nhiều thẩm quyền trong xã hội và họ là những người rất quan trọng để thực thi pháp luật và đảm bảo sự an toàn.
Luận văn	as we stood waiting for a taxi outside of the airport	khi đang chờ taxi bên ngoài sân bay
GT	as we stood waiting for a taxi outside of the airport	khi chúng tôi đứng đợi taxi bên ngoài sân bay
BMT	as we stood waiting for a taxi outside of the airport	khi chúng tôi đứng chờ một chiếc taxi bên ngoài sân bay
CT	as we stood waiting for a taxi outside of the airport	khi chúng tôi đứng chờ đợi một chiếc taxi bên ngoài sân bay

Luận văn	it seems like every parent wants their kid to grow up to be a doctor	có vẻ như mọi người cha mẹ đều muốn con mình trở thành bác sĩ
GT	it seems like every parent wants their kid to grow up to be a doctor	có vẻ như mọi bậc cha mẹ đều muốn trở thành một bác sĩ
BMT	it seems like every parent wants their kid to grow up to be a doctor	Nó có vẻ như mọi phụ huynh muốn kid của họ để lớn lên được một bác sĩ
CT	it seems like every parent wants their kid to grow up to be a doctor	Nó có vẻ như mọi phụ huynh muốn kid của họ để lớn lên được một bác sĩ
Luận văn	I quickly realised why Da Lat was often referred to as a small version of the Netherlands.	tôi nhanh chóng tìm ra lý do tại sao da đỏ thường được gọi là phiên bản nhỏ của hà lan .
GT	I quickly realised why Da Lat was often referred to as a small version of the Netherlands.	Tôi nhanh chóng nhận ra tại sao Đà Lạt thường được gọi là một phiên bản nhỏ của Hà Lan.
BMT	I quickly realised why Da Lat was often referred to as a small version of the Netherlands.	Tôi nhanh chóng nhận ra lý do tại sao Đà Lạt thường được gọi là một phiên bản nhỏ của Hà Lan.
CT	I quickly realised why Da Lat was often referred to as a small version of the Netherlands.	Tôi nhanh chóng nhận ra lý do tại sao Đà Lạt thường được gọi là một phiên bản nhỏ của Hà Lan.
Luận văn	many years ago, when I was a little child, I admired adventures.	nhiều năm trước , khi tôi còn là một đứa trẻ , tôi yêu những cuộc phiêu lưu

GT	many years ago, when I was a little child, I admired adventures.	nhiều năm trước, khi tôi còn nhỏ, tôi rất ngưỡng mộ những cuộc phiêu lưu
BMT	many years ago, when I was a little child, I admired adventures.	nhiều năm trước, khi tôi đã là một đứa trẻ nhỏ, tôi ngưỡng mộ cuộc phiêu lưu
CT	many years ago, when I was a little child, I admired adventures.	nhiều năm trước, khi tôi đã là một đứa trẻ nhỏ, tôi ngưỡng mộ cuộc phiêu lưu
Luận văn	do you have someone who is great, spends time with you, cares for you, and is an important person?	bạn có một người tuyệt vời , dành thời gian cho bạn , quan tâm với bạn , và là một người quan trọng ?
GT	do you have someone who is great, spends time with you, cares for you, and is an important person?	bạn có một người tuyệt vời, dành thời gian cho bạn, quan tâm đến bạn và là một người quan trọng?
BMT	do you have someone who is great, spends time with you, cares for you, and is an important person?	bạn có một người là tuyệt vời, dành thời gian với bạn, quan tâm cho bạn, và là một người trọng đại?
CT	do you have someone who is great, spends time with you, cares for you, and is an important person?	Bạn có một người là tuyệt vời, dành thời gian với bạn, quan tâm cho bạn, và là một người trọng đại?

5.3 KẾT QUẢ HỆ THỐNG

5.3.1 Môi trường phát triển

Hệ điều hành: Ubutu, MacOS và Windows

Công cụ phát triển phần mềm: Visual Code, Jupyter notebook và Google Colab

Công cụ kiểm thử API: Postman

Các thư viện/nền tảng được sử dụng:

Tên thư viện / nền tảng	Tóm tắt chức năng
Python	Python là một ngôn ngữ lập trình thông dịch được sử dụng rất phổ biến trong lĩnh vực khoa học máy tính nhờ những ưu điểm như: Đa nền tảng, đơn giản, mã nguồn mở, nhiều thư viện hỗ trợ.
Flask	Thư viện Tensorflow được sử dụng trong việc tính toán các biểu đồ và các dữ liệu dưới dạng số hoá trong sản phẩm khoá luận. Là một thư viện mã nguồn mở hỗ trợ mạnh mẽ các phép toán học để tính toán trong máy học.
Matplotlib	Nó là một thư viện vẽ đồ thị rất mạnh mẽ, giúp ta có cái nhìn trực quan hơn về dữ liệu.
Numpy	NumPy là một thư viện cho ngôn ngữ lập trình Python, thêm hỗ trợ cho các mảng và ma trận lớn, đa chiều, cùng với một tập hợp lớn các hàm toán học

	cấp cao để hoạt động trên các mảng này.
Pandas	Pandas dùng để thao tác và phân tích dữ liệu. Cụ thể, nó cung cấp các cấu trúc dữ liệu và các thao tác để thao tác các bảng số và chuỗi thời gian.
Scipy	Scipy chứa các mô-đun để tối ưu hóa, đại số tuyến tính, tích hợp, nội suy, các chức năng đặc biệt, FFT, xử lý tín hiệu và hình ảnh, bộ giải ODE và các nhiệm vụ phổ biến khác trong khoa học và kỹ thuật.
Scikit-learn	Scikit-learn được thiết kế dựa trên nền Numpy và Scipy. Scikit-learn chứa hầu hết các thuật toán machine learning hiện đại nhất, đi kèm với tài liệu và luôn được cập nhật.
Tensorflow	Thư viện Tensorfow là thư viện mã nguồn mở dùng cho tính toán số học sử dụng đồ thị luồng dữ liệu.
Keras	Keras là một thư viện mạng nơ-ron mã nguồn mở được viết bằng Python. Được thiết kế để cho phép thử nghiệm nhanh với các mạng thần kinh sâu, nó

	tập trung vào việc thân thiện với người dùng, mô-đun và mở rộng.

5.3.2 Môi trường triển khai

Nền tảng đám mây Amazon EC2
Nền tảng đám mây Google Cloud (dự phòng)

5.3.3 Chức năng đã cài đặt

Cung cấp API nhận vào văn bản tiếng Anh dạng text và trả về đối tượng text chứa văn bản tiếng Việt tương ứng.

5.4 KẾT QUẢ ỨNG DỤNG WEB

5.4.1 Môi trường phát triển

- Hệ điều hành: Ubutu, MacOS và Windows
- Công cụ phát triển phần mềm: Visual Code
- Công cụ kiểm thử API: Postman
- Trình duyệt kiểm thử web: chrome, safari, microsoft edge
- Các thư viện/nền tảng được sử dụng:
  - React Framework: xây dựng giao diện
  - Nodejs: nền tảng để chạy React

5.4.2 Môi trường triển khai

Server: heroku
Thiết bị: Thiết bị cần có trình duyệt web như: safari, chrome, firefox, …
Hệ điều hành: MacOS, Ubuntu, Linux

5.4.3 Chức năng đã cài đặt

Cung cấp dịch vụ nhận vào văn bản tiếng Anh dạng text và trả về đối tượng text chứa văn bản tiếng Việt tương ứng.

5.5 SO SÁNH KẾT QUẢ VỚI CÁC MỤC TIÊU ĐẶT RA

Mục tiêu ban đầu	Nhận xét mức độ hoàn thành
Trình bày lý do xây dựng mô hình dịch máy	Đã trình bày các lý do ở chương 1 của luận văn
Trình bày lý thuyết nền tảng và giải pháp để xử lý việc dịch một văn bản từ tiếng Anh sang tiếng Việt.	Đã trình bày ở chương 2
Xây dựng, thu thập dữ liệu và đào tạo mô hình để dịch một văn bản từ tiếng Anh sang tiếng Việt.	Đã được trình bày ở chương 3 và 4
Xây dựng một trang web demo việc sử dụng mô hình để dịch một văn bản từ tiếng Anh sang tiếng Việt.	Đã được trình bày ở chương 3 và 4
Viết 120 trang luận văn theo đúng chuẩn yêu cầu và trích dẫn các tài liệu tham khảo đầy đủ.	Luận văn được viết tương đối đầy đủ và chính xác.

5.6 ĐỊNH HƯỚNG PHÁT TRIỂN VÀ NGHIÊN CỨU TRONG TƯƠNG LAI

- Cải thiện lại mã nguồn để dịch được chính xác và hợp lý hơn.
- Nghiên cứu kĩ hơn về lý thuyết nền tảng, từ đó có các bước cải thiện và thực hiện chức năng một cách đúng đắn.
- Chỉnh sửa các tài liệu nghiên cứu, hướng dẫn như luận văn, hướng dẫn sử dụng để giúp người dùng mau chóng nắm bắt được vấn đề.
- Hoàn thiện chức năng dịch văn bản, sửa một số lỗi còn tồn tại. Hoặc đổi phương pháp xây dựng mô hình để chuẩn xác hơn.
- Cải thiện tốc độ xử lý các tác vụ của ứng dụng, giúp ứng dụng chạy mượt mà và tạo trải nghiệm tốt hơn cho người dùng.
- Trình bày, chỉnh sửa mã nguồn theo khuôn mẫu để dễ dàng bảo trì và chỉnh sửa trong tương lai.
- Thu thập thêm dữ liệu để mô hình huấn luyện và dịch chính xác hơn.

LỜI KẾT

Luận văn "Xây dựng mô hình dịch máy từ tiếng Anh sang tiếng Việt", hệ thống cung cấp dịch vụ và ứng dụng được xây dựng là sản phẩm kết tinh của một quá trình học tập, làm việc và nghiên cứu nghiêm túc của nhóm sinh viên. Tuy hệ thống còn nhiều hạn chế về hệ thống lẫn khả năng xử lí của nó, song sản phẩm hệ thống cung cấp dịch vụ dịch máy từ tiếng Anh sang tiếng Việt đã đem lại cho nhóm sinh viên những kiến thức và kinh nghiệm quý báu cũng như cách để triển khai các dự án thực tế trong tương lai. Các hệ thống học sâu nói chung và dịch máy nói riêng hiện đang là những lĩnh vực nổi trội trên thế giới và nó đem lại lợi ích tuyệt vời trong cuộc sống. Những trải nghiệm trong luận văn là những kinh nghiệm quy báu cho nhóm sinh viên để có những kiến thức và tiếp tục nghiên cứu và phát triển sự nghiệp của bản thân.

TÀI LIỆU THAM KHẢO

[1] Yani Andrew Ioannou. “Structural Priors in Deep Neural Networks” September 2017.

[2] Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." *arXiv preprint arXiv:1502.03167* (2015).

[3] Zachary C. Lipton, John Berkowitz, Charles Elkan. “*A Critical Review of Recurrent Neural Networks for Sequence Learning*” June 5th, 2015, pp.10-11.

[4] Alex Graves, Navdeep Jaitly. “*Towards End-to-End Speech Recognition with Recurrent Neural Networks*”, pp.3-4.

[5] https://en.wikipedia.org/wiki/Hopfield_network

[6] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." *Advances in neural information processing systems*. 2014.

[7] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." *arXiv preprint arXiv:1409.0473* (2014).

[8] Luong, Minh-Thang, Hieu Pham, and Christopher D. Manning. "Effective approaches to attention-based neural machine translation." *arXiv preprint arXiv:1508.04025* (2015).

[9] DAUPHIN, Yann N., et al. Language modeling with gated convolutional networks. In: *International conference on machine learning*. 2017. p. 933-941.

[10] *Speech and Language Processing*. Daniel Jurafsky & James H. Martin. Copyright c 2019. All rights reserved. Draft of October 2, 2019..

[11] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).