TRƯỜNG ĐẠI HỌC KỸ THUẬT – CÔNG NGHỆ CẦN THƠ

KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN HỌC PHẦN 3

NHẬN DIỆN CẢM XÚC VĂN BẢN TIẾNG VIỆT VỚI PHOBERT

GVHD: **Th.S Lê Anh Nhã Uyên**

GVPB:**.....**

Sinh viên thực hiện:

1: Từ Thái Bảo **MSSV**: 1900222

2: Lâm Thiện Nhân **MSSV**: 1900558

3: Nguyễn Duy Khánh **MSSV**: 1900540

Ngành: Khoa học máy tính - 2019

Cần Thơ, năm 2022

TRƯỜNG ĐẠI HỌC KỸ THUẬT – CÔNG NGHỆ CẦN THƠ

KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN HỌC PHẦN 3

NHẬN DIỆN CẢM XÚC VĂN BẢN TIẾNG VIỆT VỚI PHOBERT

GVHD: **Th.S Hà Lê Ngọc Dung**

GVPB:**.....**

Sinh viên thực hiện:

1: Từ Thái Bảo **MSSV**: 1900222

2: Lâm Thiện Nhân **MSSV**: 1900558

3: Nguyễn Duy Khánh **MSSV**: 1900540

Ngành: Khoa học máy tính - 2019

Cần Thơ, năm 2022

NHẬN XÉT, ĐÁNH GIÁ CỦA GIẢNG VIÊN HƯỚNG DẪN

Đồ án 3

Đề tài: ***NHẬN DIỆN CẢM XÚC VĂN BẢN TIẾNG VIỆT VỚI PHOBERT***

Sinh viên thực hiện:

1. Từ Thái Bảo **MSSV**: 1900222
2. Lâm Thiện Nhân **MSSV**: 1900558
3. Nguyễn Duy Khánh **MSSV**: 1900540

Ngành: Khoa học máy tính - 2019

Giảng viên hướng dẫn: Th.S Lê Anh Nhã Uyên

Nhận xét, đánh giá:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

……, ngày ... tháng ... năm 2022

Giảng viên hướng dẫn

NHẬN XÉT, ĐÁNH GIÁ CỦA GIẢNG VIÊN PHẢN BIỆN

Đồ án 3

Đề tài: ***NHẬN DIỆN CẢM XÚC VĂN BẢN TIẾNG VIỆT VỚI PHOBERT***

Sinh viên thực hiện:

1. Từ Thái Bảo — **MSSV**: 1900222
2. Lâm Thiện Nhân — **MSSV**: 1900558
3. Nguyễn Duy Khánh — **MSSV**: 1900540

Ngành: Khoa học máy tính - 2019

Giảng viên hướng dẫn: Th.S Lê Anh Nhã Uyên

Nhận xét, đánh giá:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

……, ngày ... tháng ... năm 2022

Giảng viên phản biện

LỜI CAM ĐOAN

Nhóm chúng em xin cam đoan rằng đồ án này là công trình nghiên cứu của nhóm thực hiện dựa trên những kiến thức đã được học, nghiên cứu và tìm hiểu một số đề tài và các phương án đi trước, không sao chép từ bất cứ công trình đã có trước đó. Mọi thứ được dựa trên sự cố gắng cũng như sự nỗ lực của bản thân.

Những phần có sử dụng tài liệu tham khảo có trong đồ án đã được liệt kê và nêu rõ ra tại phần tài liệu tham khảo.

Cần Thơ, ngày 15 tháng 12 năm 2022

Nhóm sinh viên thực hiện

Sinh viên thực hiện 1	Sinh viên thực hiện 2	Sinh viên thực hiện 3
		
Từ Thái Bảo	Lâm Thiện Nhân	Nguyễn Duy Khánh

LỜI CẢM ƠN

Được sự giúp đỡ nhiệt tình của quý Thầy, Cô bộ môn Khoa học máy tính, Khoa Công nghệ thông tin, Trường Đại học Kỹ thuật - Công nghệ Cần Thơ, những người đã dìu dắt chúng em tận tình, đã truyền đạt cho em những kiến thức và những bài học quý giá trong suốt thời gian chúng em theo học tại trường. Sau gần ba tháng nghiên cứu chúng em đã hoàn thành đề tài: ***"NHẬN DIỆN CẢM XÚC VĂN BẢN TIẾNG VIỆT VỚI PHOBERT"***.

Chúng em chân thành gửi lời cảm ơn đến tất cả các Thầy Cô trong khoa Công Nghệ Thông Tin đã hỗ trợ chúng em hoàn thành tốt đồ án lần này. Đặt biệt nhất là chúng em muốn dành lời cảm ơn đến Cô Lê Anh Nhã Uyên, người đã tận tâm hướng dẫn và trực tiếp giúp đỡ chúng em trong suốt quá trình nghiên cứu đồ án. Với sự chỉ bảo nhiệt tình của Cô, chúng em đã có định hướng tốt trong việc triển khai và thực hiện các yêu cầu trong quá trình làm đồ án.

Để có thể hoàn thành tốt được đề tài, ngoài này sự nỗ lực học hỏi của bản thân, còn có sự hướng dẫn tận tình từ quý Thầy, Cô và sự giúp đỡ tận tình từ gia đình và bạn bè đã tạo cho chúng em những điều kiện tốt nhất trong suốt quá trình học tập và nghiên cứu.

Xin chân thành cảm ơn!

TÓM TẮT ĐỒ ÁN

Đây là đồ án phục vụ cho việc nghiên cứu của sinh viên tìm hiểu về các mô hình ngôn ngữ BERT và PhoBERT. Với tham vọng tìm hiểu, học hỏi những điểm mới mẽ của các kỹ thuật này nên chúng em mạnh dạn đăng ký đề tài này mong rằng sẽ một phần nào đó giúp cho mọi người có thể nắm bắt và tìm hiểu được các mô hình ngôn ngữ BERT và PhoBERT. Ngoài ra đồ án còn xây dựng chương trình thực nghiệm sữ dụng PhoBert để nhận dạng cảm xúc của người dùng.

Bố cục của đề tài: ***NHẬN DIỆN CẢM XÚC VĂN BẢN TIẾNG VIỆT VỚI PHOBERT*** bao gồm 6 chương:

- Chương I: Tổng quan về Xử lý ngôn ngữ tự nhiên
- Chương II: Mô hình ngôn ngữ BERT
- Chương III: Mô hình ngôn ngữ PhoBERT
- Chương IV: Xây dựng chương trình nhận dạng cảm xúc văn bản Tiếng Việt với PhoBERT
- Chương V: Kết luận
- Chương VI: Tài liệu tham khảo

MỤC LỤC

DANH MỤC BẢNG BIỂU, HÌNH ẢNH

DANH MỤC TỪ VIẾT TẮT, THUẬT NGỮ

STT	Từ viết tắt	Tên đầy đủ	Dịch nghĩa
1	NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên
2	BERT	Bidirectional Encoder Representations from Transformers	Mô hình ngôn ngữ BERT
3	LBP	Local binary patterns	Mẫu nhị phân địa phương
4	POS	Part of speech	Gán nhãn từ loại
5	NER	Named Entity Recognition	Nhận dạng thực thể tên
6	NLI	Natural language inference	Suy luận ngôn ngữ tự nhiên
7	BPE	Byte Pair Encoding	Thuật toán nén dữ liệu
8	CFG	Context-free grammar	Văn phạm phi ngữ cảnh
9	CCG	Combinatory categorial grammar	Văn phạm danh mục kết nối
10	DG	Dependency grammar	Văn phạm phụ thuộc

LỜI NÓI ĐẦU

Trong bất kỳ xã hội nào con người luôn có nhu cầu được giao tiếp và thể hiện, hình thức được sử dụng phổ biến đó là diễn đạt bằng ngôn ngữ. Ngôn ngữ sử dụng từ ngữ hoặc dấu hiệu để diễn tả được thể hiện qua lời nói, chữ viết hoặc các hình ảnh. Với sự bùng nổ của Internet và các trang mạng xã hội, các trang web tài liệu, sách báo, các trang sản phẩm, email,.. một lượng lớn dữ liệu văn bản của ngôn ngữ được tạo ra mỗi ngày. Để giúp máy tính hiểu được những dữ liệu này là công việc quan trọng để hỗ trợ hoặc quyết định dựa trên ngôn ngữ.

Xử lý ngôn ngữ tự nhiên nghiên cứu sự tương tác bằng ngôn ngữ tự nhiên giữa máy tính và con người. Trong thực tế, việc sử dụng các kỹ thuật xử lý ngôn ngữ tự nhiên để xử lý và phân tích dữ liệu văn bản rất phổ biến, chẳng hạn như các mô hình ngôn ngữ trong hay các mô hình dịch máy. Để có thể xây dựng các phương pháp xử lý ngôn ngữ thì trước tiên chúng ta cần quan tâm đến việc biểu diễn ngôn ngữ tự nhiên như thế nào. Một số phương pháp biểu diễn ngôn ngữ đã được giới thiệu được sử dụng trong các nhiệm vụ xử lý ngôn ngữ tự nhiên như: sự xuất hiện và tần suất xuất hiện, mô hình ngôn ngữ, thông tin nhãn từ loại thông tin phân tích ngữ pháp biểu diễn vecter từ, nhúng ký tự, mạng ngữ nghĩa, mạng từ điển quan điểm,…Các phương pháp biểu diễn ngôn ngữ này giúp trích xuất các đặc trưng từ ngôn ngữ sử dụng cho các mô hình xử lý ngôn ngữ tự nhiên giúp nâng cao hiệu quả cho các phương pháp phân tích. Do đó, nghiên cứu về các phương pháp biểu diễn ngôn ngữ nhằm tìm ra các đặc trưng hữu ích cho bài toán NLP là nhiệm vụ quan trọng.

Gần đây, Google AI giới thiệu mô hình ngôn ngữ BERT được coi là một bước đột phá lớn trong học máy vì khả năng ứng dụng của nó vào nhiều bài toán xử lý ngôn ngữ tự nhiên khác nhau với kết quả rất tốt. Tiếp theo đó, PhoBERT ra đời nhằm xây dựng mô hình ngôn ngữ BERT riêng cho tiếng Việt với kết quả tốt nhất cho nhiều bài toán xử lý ngôn ngữ tự nhiên tiếng Việt. Với sự phát triển của các trang mạng xã hội và các trang đánh giá sản phẩm, dữ liệu bình luận khen chê của khách hàng đang gia tăng một cách nhanh chóng tạo thành kho dữ liệu đánh giá khổng lồ. Việc hiểu xem khách hàng đánh giá về một sản phẩm, dịch vụ hay vấn đề được quan tâm là tích cực hay tiêu cực là nhiệm vụ được các nhà nghiên cứu quan tâm trong những thập niên gần đây và đã có nhiều ứng dụng trong thực tế. Chính vì những lý do đó,nhóm em chọn đề tài " ***Nhận diện cảm xúc văn bản Tiếng Việt với PhoBert*** "nhằm tìm hiểu các phương pháp mới biểu diễn cho ngôn ngữ tiếng Việt và áp dụng nó cho bài toán phân loại bình luận tiếng Việt.

CHƯƠNG I: TỔNG QUAN VỀ XỬ LÝ NGÔN NGỮ TỰ NHIÊN

1.1. Ngôn ngữ tự nhiên

1.1.1. Tổng quan

Từ xưa đến nay, ngôn ngữ đã thể hiện được vai trò quan trọng của mình trong cuộc sống của con người nói riêng, cùng tất cả muôn loài sinh sống trên trái đất nói chung. Đâu ai có thể sinh ra mà chỉ sống có một mình thôi được, khi đó ngôn ngữ nghiễm nhiên trở thành một phương tiện giúp chúng ta đến gần với nhau hơn.

Ngôn ngữ được coi là một phần không thể thiếu trong cuộc sống của con người. Ngôn ngữ giúp con người giao tiếp, liên kết và kết nối với nhau. Ngôn ngữ tồn tại ở dưới nhiều trạng thái và rất đa dạng. Ngôn ngữ được tồn tại dưới nhiều trạng thái khác nhau như dưới dạng âm thanh dưới dạng văn bản Dù tồn tại ở bất cứ hình thức nào thì ngôn ngữ cũng được coi là sự liên kết quý giá, thể hiện được văn hóa và đặc trưng của dân tộc tạo ra nó. Việc hình thành từ tự nhiên đã khiến ngôn ngữ trở nên giá trị hơn bao giờ hết bởi nó không chịu sự ảnh hưởng của bất cứ yếu tố xung quanh. Trong cuộc sống con người ngày nay, việc tìm những ngôn ngữ tự nhiên để ứng dụng nó vào những lĩnh vực khoa học ngày càng phổ biến.

1.1.2. Khái niệm

Ngôn ngữ tự nhiên là một thành phần trong lĩnh vực ngôn ngữ học rộng lớn. Trong ngôn ngữ quốc tế, ngôn ngữ tự nhiên được viết là Natural Language. Ngôn ngữ tự nhiên được hiểu là bất cứ ngôn ngữ nào được phát sinh, được tạo ra mà không trải qua bất cứ một suy nghĩ nào trước đó trong não bộ của con người

Ngôn ngữ tự nhiên tồn tại dưới nhiều trạng thái trong cuộc sống của chúng ta. Đây được coi là một loại ngôn ngữ mà bất cứ một đứa trẻ nào cũng có thể tiếp thu và học tập thông qua ngôn ngữ nói để hình thành kiến thức cho bản thân. Việc không tuân thủ theo bất cứ một sự định hướng cũng như hướng dẫn chỉ định từ đầu đã tạo nên những nét riêng biệt khiến ngôn ngữ tự nhiên khác với những ngôn ngữ thông thường.

1.1.3. Tính chất của ngôn ngữ tự nhiên

Đa nghĩa

Một từ hoặc một cụm từ trong ngôn ngữ tự nhiên có thể có nhiều nghĩa khác nhau, tùy thuộc vào ngữ cảnh trong đó nó được sử dụng.

Tính đa nghĩa là một tính chất rất đáng quý của ngôn ngữ trong giao tiếp hàng ngày, trong văn học và nghệ thuật. Tuy nhiên tính chất này lại gây ra khá nhiều khó khăn cho việc sử dụng ngôn ngữ tự nhiên trong khoa học, kỹ thuật, luật pháp,

Giàu khả năng biểu đạt

Tất cả các ngôn ngữ tự nhiên đều rất giàu khả năng biểu đạt. Người ta có thể dùng ngôn ngữ tự nhiên trong rất nhiều lĩnh vực. Có thể dùng chúng để trò chuyện, trao đổi

thường ngày, có thể dùng chúng để làm thơ, viết văn, để bàn luận về thời sự, về chính trị, về luật pháp; có thể dùng chúng để nghiên cứu và trình bày các tư tưởng và công trình khoa học,… Ngoài ra, với ngôn ngữ tự nhiên, cùng một sự vật hoặc hiện tượng có thể được mô tả, được biểu đạt bằng các cách khác nhau, bằng các biểu thức ngôn ngữ khác nhau.

Đóng về ngữ nghĩa

Trong ngôn ngữ tự nhiên vừa có bộ phận từ và câu nói về các đối tượng bên ngoài ngôn ngữ, nói về thế giới bên ngoài ngôn ngữ, ví dụ, nói về thời tiết, về kinh tế, về các vật dụng, … và có cả những bộ phận từ và câu nói về các đối tượng của bản thân ngôn ngữ, ví dụ, nói về ngữ pháp, về cú pháp, về danh từ, động từ, câu, … Sự có mặt của cả hai thành phần như vậy trong ngôn ngữ được gọi là tính đóng về ngữ nghĩa của nó.

Có nhiều cấp độ ngôn ngữ

Trong cùng một đoạn văn hoặc một câu của ngôn ngữ tự nhiên, từ ngữ có thể thuộc về nhiều cấp độ khác nhau. Nếu không phân biệt các cấp độ ngôn ngữ khác nhau như vậy thì ta sẽ cho rằng đây là câu nói chứa đựng nghịch lý.

Một phần thông tin không được biểu đạt tường minh

Thông tin chứa đựng trong các câu, các đoạn văn trong ngôn ngữ tự nhiên có thể chỉ có một phần được biểu đạt dưới dạng tường minh, còn phần khác được ngầm hiểu. Để suy luận đúng đắn ta cần phải xác định được toàn bộ nội dung thông tin mà câu hoặc đoạn văn chứa, cả hiển ngôn và hàm ngôn.

Ngôn ngữ tự nhiên rất thuận tiện cho quá trình trao đổi trong cuộc sống hàng ngày. Nó cũng rất thuận lợi cho các hoạt động văn học nghệ thuật. Nếu dùng ngôn ngữ tự nhiên để nghiên cứu và trình bày các vấn đề khoa học kỹ thuật thì ta gặp phải nhiều khó khăn vì tính đa nghĩa của nó. Vì ngôn ngữ tự nhiên đóng về ngữ nghĩa nên nó có thể chứa các nghịch lý. Điều này khiến ta không thể dùng nó để xây dựng các lý thuyết khoa học chặt chẽ bởi lẽ khoa học không được phép chứa đựng các nghịch lý.

1.2. Xử lý ngôn ngữ tự nhiên

1.2.1. Khái niệm

Xử lý ngôn ngữ tự nhiên là một nhánh của Trí tuệ nhân tạo, tập trung vào việc nghiên cứu sự tương tác giữa máy tính và ngôn ngữ tự nhiên của con người, dưới dạng tiếng nói hoặc văn bản. Mục tiêu của lĩnh vực này là giúp máy tính hiểu và thực hiện hiệu quả những nhiệm vụ liên quan đến ngôn ngữ của con người như: tương tác giữa người và máy, cải thiện hiệu quả giao tiếp giữa con người với con người, hoặc đơn giản là nâng cao hiệu quả xử lý văn bản và lời nói.

Xử lý ngôn ngữ tự nhiên ra đời từ những năm 40 của thế kỷ 20, trải qua các giai đoạn phát triển với nhiều phương pháp và mô hình xử lý khác nhau. Có thể kể tới các phương pháp:

- Sử dụng Otomat và mô hình xác suất
- Các phương pháp dựa trên ký hiệu
- Các phương pháp ngẫu nhiên
- Các phương pháp sử dụng học máy truyền thống
- Học sâu

Xử lý ngôn ngữ tự nhiên có thể được chia ra thành hai nhánh lớn, không hoàn toàn độc lập, bao gồm xử lý tiếng nói và xử lý văn bản Xử lý tiếng nói tập trung nghiên cứu, phát triển các thuật toán, chương trình máy tính xử lý ngôn ngữ của con người ở dạng tiếng nói Các ứng dụng quan trọng của xử lý tiếng nói bao gồm nhận dạng tiếng nói và tổng hợp tiếng nói. Nếu như nhận dạng tiếng nói là chuyển ngôn ngữ từ dạng tiếng nói sang dạng văn bản thì ngược lại, tổng hợp tiếng nói chuyển ngôn ngữ từ dạng văn bản thành tiếng nói. Xử lý văn bản tập trung vào phân tích dữ liệu văn bản. Các ứng dụng quan trọng của xử lý văn bản bao gồm tìm kiếm và truy xuất thông tin, dịch máy, tóm tắt văn bản tự động, hay kiểm lỗi chính tả tự động. Xử lý văn bản đôi khi được chia tiếp thành hai nhánh nhỏ hơn bao gồm hiểu văn bản và sinh văn bản. Nếu như hiểu liên quan tới các bài toán phân tích văn bản thì sinh liên quan tới nhiệm vụ tạo ra văn bản mới như trong các ứng dụng về dịch máy hoặc tóm tắt văn bản tự động.

1.2.2. Các bước trong quá trình Xử lý ngôn ngữ tự nhiên

Phân tích hình vị

Là sự nhận biết, phân tích, và miêu tả cấu trúc của hình vị trong một ngôn ngữ cho trước và các đơn vị ngôn ngữ khác, như từ gốc, biên từ, phụ tố, từ loại,… Trong xử lý tiếng Việt, hai bài toán điển hình trong phần này là tách từ và gán nhãn từ loại.

Phân tích cú pháp

Là quy trình phân tích một chuỗi các biểu tượng, ở dạng ngôn ngữ tự nhiên hoặc ngôn ngữ máy tính, tuân theo văn phạm hình thức. Văn phạm hình thức thường dùng trong phân tích cú pháp của ngôn ngữ tự nhiên bao gồm Văn phạm phi ngữ cảnh (CFG), Văn phạm danh mục kết nối (CCG) và Văn phạm phụ thuộc (DG). Đầu vào của quá trình phân tích là một câu gồm một chuỗi từ và nhãn từ loại của chúng, và đầu ra là một cây phân tích thể hiện cấu trúc cú pháp của câu đó.

Phân tích ngữ nghĩa

Là quá trình liên hệ cấu trúc ngữ nghĩa, từ cấp độ cụm từ, mệnh đề, câu và đoạn đến cấp độ toàn bài viết, với ý nghĩa độc lập của chúng. Nói cách khác, việc này nhằm tìm ra ngữ nghĩa của đầu vào ngôn từ. Phân tích ngữ nghĩa bao gồm hai mức độ: Ngữ nghĩa từ vựng biểu hiện các ý nghĩa của những từ thành phần, và phân biệt nghĩa của từ; Ngữ nghĩa thành phần liên quan đến cách thức các từ liên kết để hình thành những nghĩa rộng hơn.

Phân tích diễn ngôn

Là phân tích văn bản có xét tới mối quan hệ giữa ngôn ngữ và ngữ cảnh sử dụng. Phân tích diễn ngôn, do đó, được thực hiện ở mức độ đoạn văn hoặc toàn bộ văn bản thay vì chỉ phân tích riêng ở mức câu.

1.2.3. Các ứng dụng của xử lý ngôn ngữ tự nhiên

Nhận dạng tiếng nói

Chuyển đổi ngôn ngữ từ dạng tiếng nói sang dạng văn bản, thường được ứng dụng trong các chương trình điều khiển qua giọng nói.

Truy xuất thông tin

Có nhiệm vụ tìm các tài liệu dưới dạng không có cấu trúc.đáp ứng nhu cầu về thông tin từ những nguồn tổng hợp lớn. Những công cụ này cho phép tiếp nhận một câu truy vấn dưới dạng ngôn ngữ tự nhiên làm đầu vào và cho ra một danh sách các tài liệu được sắp xếp theo mức độ phù hợp.

Trích chọn thông tin

Nhận diện một số loại thực thể được xác định trước, mối quan hệ giữa các thực thể và các sự kiện trong văn bản ngôn ngữ tự nhiên. Trích chọn thông tin trả về chính xác thông tin mà người dùng cần. Những thông tin này có thể là về con người, địa điểm, tổ chức, ngày tháng, hoặc thậm chí tên công ty, mẫu sản phẩm hay giá cả.

Trả lời câu hỏi

Có khả năng tự động trả lời câu hỏi của con người ở dạng ngôn ngữ tự nhiên bằng cách truy xuất thông tin từ một tập hợp tài liệu.

Tóm tắt văn bản tự động

Là bài toán thu gọn văn bản đầu vào để cho ra một bản tóm tắt ngắn gọn với những nội dung quan trọng nhất của văn bản gốc

Chatbot

Là việc chương trình máy tính có khả năng trò chuyện , hỏi đáp với con người qua hình thức hội thoại dưới dạng văn bản. Chatbot thường được sử dụng trong ứng dụng hỗ trợ khách hàng, giúp người dùng tìm kiếm thông tin sản phẩm, hoặc giải đáp thắc mắc.

Dịch máy

Là việc sử dụng máy tính để tự động hóa một phần hoặc toàn bộ quá trình dịch từ ngôn ngữ này sang ngôn ngữ khác.

Kiểm lỗi chính tả tự động

Là việc sử dụng máy tính để tự động phát hiện các lỗi chính tả trong văn bản (lỗi từ vựng, lỗi ngữ pháp, lỗi ngữ nghĩa) và đưa ra gợi ý cách chỉnh sửa lỗi.

CHƯƠNG II: MÔ HÌNH NGÔN NGỮ BERT

2.1. Tổng quan

BERT là viết tắt của Bidirectional Encoder Representations from Transformers là một mô hình ngôn ngữ được tạo ra bởi Google AI và được giới thiệu vào năm 2008. BERT được coi như là đột phá lớn trong Machine Learning vì khả năng ứng dụng của nó vào nhiều bài toán NLP khác nhau với kết quả rất tốt.

Các nhà nghiên cứu làm việc tại Google AI tái khẳng định, sự thiếu hụt dữ liệu huấn luyện là một trong những thách thức lớn nhất trong lĩnh vực xử lý ngôn ngữ tự nhiên. Đây là một lĩnh vực rộng lớn và đa dạng với nhiều nhiệm vụ riêng biệt, hầu hết các tập dữ liệu đều chỉ đặc thù cho từng nhiệm vụ. Để thực hiện được tốt những nhiệm vụ này ta cần những bộ dữ liệu lớn chứa hàng triệu thậm chí hàng tỷ ví dụ mẫu. Tuy nhiên, trong thực tế hầu hết các tập dữ liệu hiện giờ chỉ chứa vài nghìn hoặc vài trăm nghìn mẫu được đánh nhãn bằng tay bởi chuyên gia ngôn ngữ học. Sự thiếu hụt dữ liệu có nhãn chất lượng cao để huấn luyện mô hình gây cản trở lớn cho sự phát triển của NLP nói chung.

Để giải quyết thách thức này, các mô hình xử lý ngôn ngữ tự nhiên sử dụng một cơ chế tiền xử lý dữ liệu huấn luyện bằng việc transfer từ một mô hình chung được huấn luyện từ một lượng lớn các dữ liệu không được gán nhãn. Ví dụ một số mô hình đã được nghiên cứu trước đây để thực hiện nhiệm vụ này như Word2vec, Glove hay FastText.

Việc nghiên cứu các mô hình này sẽ giúp thu hẹp khoảng cách giữa các tập dữ liệu chuyên biệt cho huấn luyện bằng việc xây dựng mô hình tìm ra đại diện chung của ngôn ngữ sử dụng một số lượng lớn các văn bản chưa được gán nhãn lấy từ các trang web. Các mô hình được huấn luyện trước khi được tinh chỉnh lại trên các nhiệm vụ khác nhau với các bộ dữ liệu nhỏ như Question Answering, Sentiment Analysis,...sẽ dẫn đến sự cải thiện đáng kể về độ chính xác cho so với các mô hình được huấn luyện trước với các bộ dữ liệu này.

Tuy nhiên, các mô hình kể trên có những yếu điểm riêng của nó, đặc biệt là không thể hiện được sự đại diện theo ngữ cảnh cụ thể của từ trong từng lĩnh vực hay văn cảnh cụ thể. Tiếp nối sự thành công nhất định của các mô hình trước đó, Google đã công bố thêm 1 kỹ thuật mới được gọi là BERT.

2.2. Tại sao Bert lại ra đời

Một trong những thách thức lớn nhất của NLP là vấn đề dữ liệu. Trên internet có hàng tá dữ liệu, nhưng những dữ liệu đó không đồng nhất; mỗi phần của nó chỉ được dùng cho một mục đích riêng biệt, do đó khi giải quyết một bài toán cụ thể, ta cần trích ra một bộ dữ liệu thích hợp cho bài toán của mình, và kết quả là ta chỉ có một lượng rất ít dữ liệu.

Nhưng có một nghịch lý là các mô hình Deep Learning cần lượng dữ liệu rất lớn - lên tới hàng triệu - để có thể cho ra kết quả tốt. Do đó một vấn đề được đặt ra: làm thế nào để tận dụng được nguồn dữ liệu vô cùng lớn có sẵn để giải quyết bài toán của mình. Đó là tiền đề cho một kỹ thuật mới ra đời: Transfer Learning. Với Transfer Learning, các mô hình "chung" nhất với tập dữ liệu khổng lồ trên internet được xây dựng và có thể được "tinh chỉnh" (fine-tune) cho các bài toán cụ thể.

BERT là một trong những đại diện ưu tú nhất trong Transfer Learning cho xử lý ngôn ngữ tự nhiên, nó gây tiếng vang lớn không chỉ bởi kết quả mang lại trong nhiều bài toán khác nhau, mà còn bởi vì nó hoàn toàn miễn phí, tất cả chúng ta đều có thể sử dụng BERT cho bài toán của mình.

2.3. Mô hình BERT

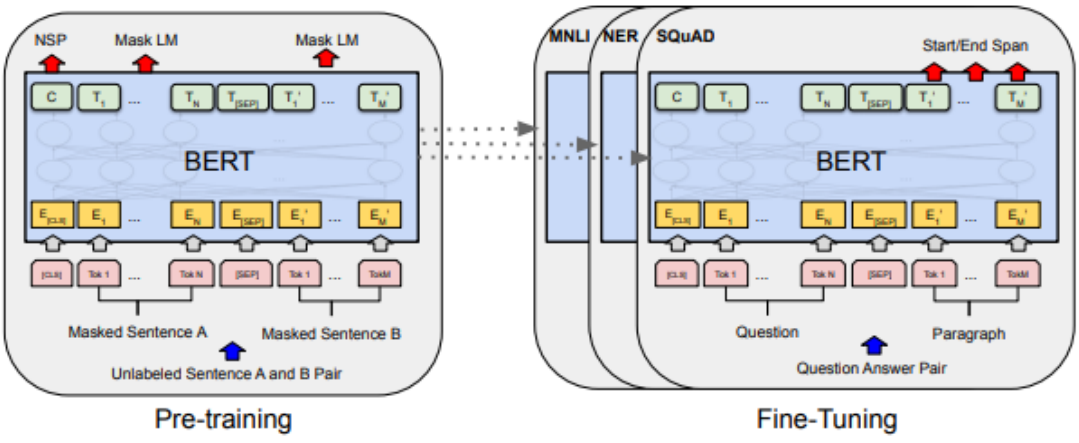
BERT là mô hình biểu diễn từ theo 2 chiều ứng dụng kỹ thuật Transformer. BERT được thiết kế để huấn luyện trước các từ nhúng. Điểm đặc biệt ở BERT đó là nó có thể điều hòa cân bằng bối cảnh theo cả 2 chiều trái và phải.

Cơ chế chú ý của Transformer sẽ truyền toàn bộ các từ trong câu văn đồng thời vào mô hình một lúc mà không cần quan tâm đến chiều của câu. Do đó Transformer được xem như là huấn luyện hai chiều mặc dù trên thực tế chính xác hơn chúng ta có thể nói rằng đó là huấn luyện không chiều.

Đặc điểm này cho phép mô hình học được bối cảnh của từ dựa trên toàn bộ các từ xung quanh nó bao gồm cả từ bên trái và từ bên phải.

2.3.1. Mô hình BERT tinh chỉnh (Fine-tuning model BERT)

Một điểm đặc biệt ở BERT mà các mô hình nhúng trước đây chưa từng có đó là kết quả huấn luyện có thể tinh chỉnh được. Chúng ta sẽ thêm vào kiến trúc mô hình một tầng đầu ra để tùy biến theo nhiệm vụ huấn luyện.



Hình 2. 1: Toàn bộ tiến trình pre-training và fine-tuning của BERT

Một kiến trúc tương tự được sử dụng cho cả mô hình huấn luyện trước và mô hình tinh chỉnh. Chúng ta sử dụng cùng một tham số huấn luyện trước để khởi tạo mô hình

cho các nhiệm vụ sau khác nhau. Trong suốt quá trình tinh chỉnh thì toàn bộ các tham số của các tầng học chuyển giao sẽ được điều chỉnh. Đối với các nhiệm vụ sử dụng đầu vào là một cặp chuỗi ví dụ như câu hỏi và trả lời thì ta sẽ thêm mã khởi tạo là [CLS] ở đầu câu, mã [SEP] ở giữa để ngăn cách 2 câu.

Tiến trình áp dụng tinh chỉnh sẽ như sau:

- Bước 1: Nhúng toàn bộ các mã của cặp câu bằng các vector nhúng từ mô hình huấn luyện trước. Các mã nhúng bao gồm cả 2 mã là [CLS] và [SEP] để đánh dấu vị trí bắt đầu của câu hỏi và vị trí ngăn cách giữa 2 câu. Hai mã này sẽ được dự báo ở đầu ra để xác định các phần mở rộng bắt đầu/kết thúc của câu đầu ra.
- Bước 2: Các vector nhúng sau đó sẽ được truyền vào kiến trúc chú ý nhiều đầu vào với nhiều mã khối (thường là 6, 12 hoặc 24 khối tùy theo kiến trúc BERT). Ta thu được một vector đầu ra ở encoder.
- Bước 3: Để dự báo phân phối xác suất cho từng vị trí từ ở decoder, ở mỗi bước thời gian chúng ta sẽ truyền vào decoder véc tơ đầu ra của encoder và véc tơ nhúng đầu vào của decoder để tính chú ý mã hóa và giải mã. Sau đó ánh xạ qua tầng tuyến tính và hàm softmax để thu được phân phối xác suất cho đầu ra tương ứng ở bước thời gian t.
- Bước 4: Trong kết quả trả ra ở đầu ra của transformer ta sẽ cố định kết quả của câu hỏi sao cho trùng với câu hỏi ở đầu vào. Các vị trí còn lại sẽ là thành phần mở rộng bắt đầu/kết thúc tương ứng với câu trả lời tìm được từ câu đầu vào.

Quá trình huấn luyện chúng ta sẽ tinh chỉnh lại toàn bộ các tham số của mô hình BERT đã loại bỏ tầng tuyến tính ở đỉnh và huấn luyện lại từ đầu các tham số của tầng tuyến tính mà chúng ta thêm vào kiến trúc mô hình BERT để tùy chỉnh lại phù hợp với bài toán.

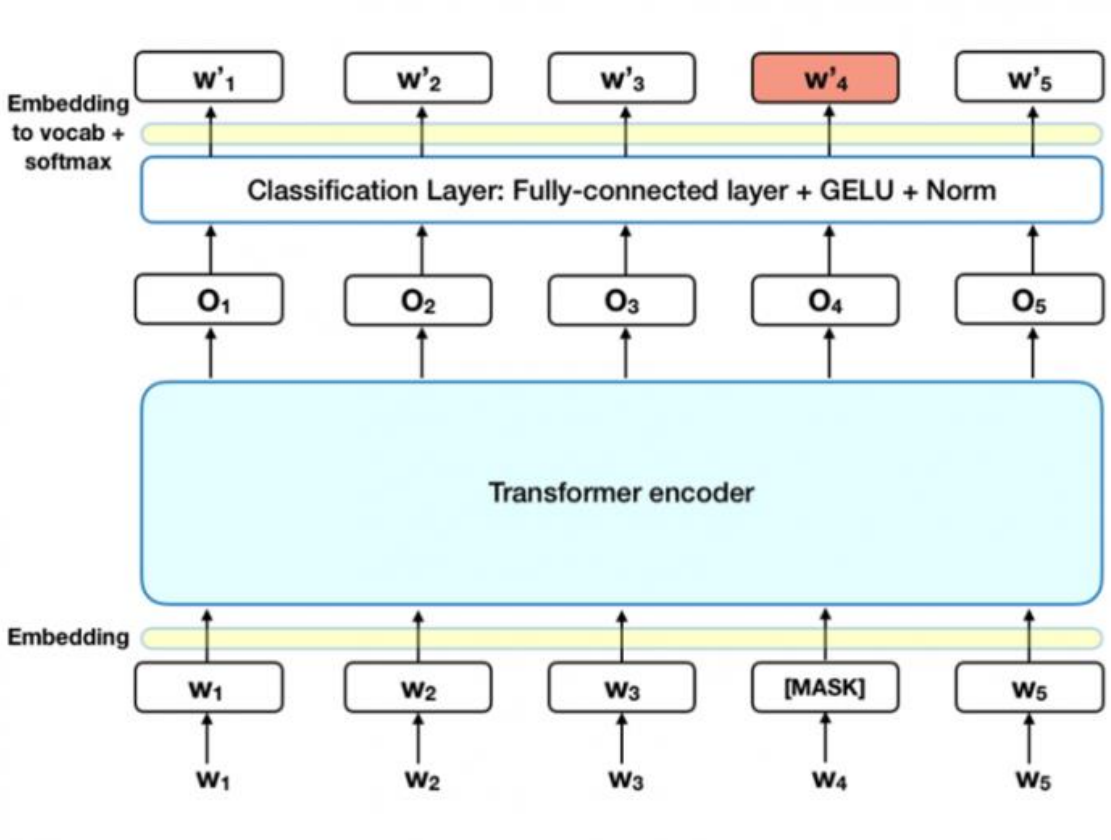
2.3.2. . Mô hình ngôn ngữ được đánh dấu (Masked Language Model)

Masked ML là một tác vụ cho phép "tinh chỉnh" fine-tuning lại các biểu diễn từ trên các bộ dữ liệu unsupervised-text bất kỳ. Chúng ta có thể áp dụng Masked ML cho những ngôn ngữ khác nhau để tạo ra biểu diễn embedding cho chúng. Các bộ dữ liệu của tiếng anh có kích thước lên tới vài vài trăm tới vài nghìn GB được huấn luyện trên BERT đã tạo ra những kết quả khá ấn tượng.

Trước khi cho các chuỗi từ vào BERT, 15% số từ trong mỗi chuỗi được thay thế bằng mã thông báo [MASK]. Mô hình sau đó cố gắng dự đoán giá trị ban đầu của các từ bị che, dựa trên ngữ cảnh được cung cấp bởi các từ khá không bị che ở trong chuỗi. Về mặt kỹ thuật, dự đoán của các từ đầu ra yêu cầu:

- Thêm một lớp phân loại ở đầu ra của bộ mã hóa.

– Nhân các vector đầu ra với ma trận nhúng, chuyển đổi chúng thành các chiều từ vựng.
– Tính xác suất của mỗi từ trong từ vựng với hàm softmax.



Hình 2. 2: Kiến trúc BERT cho tá vụ Masked ML

Theo đó:

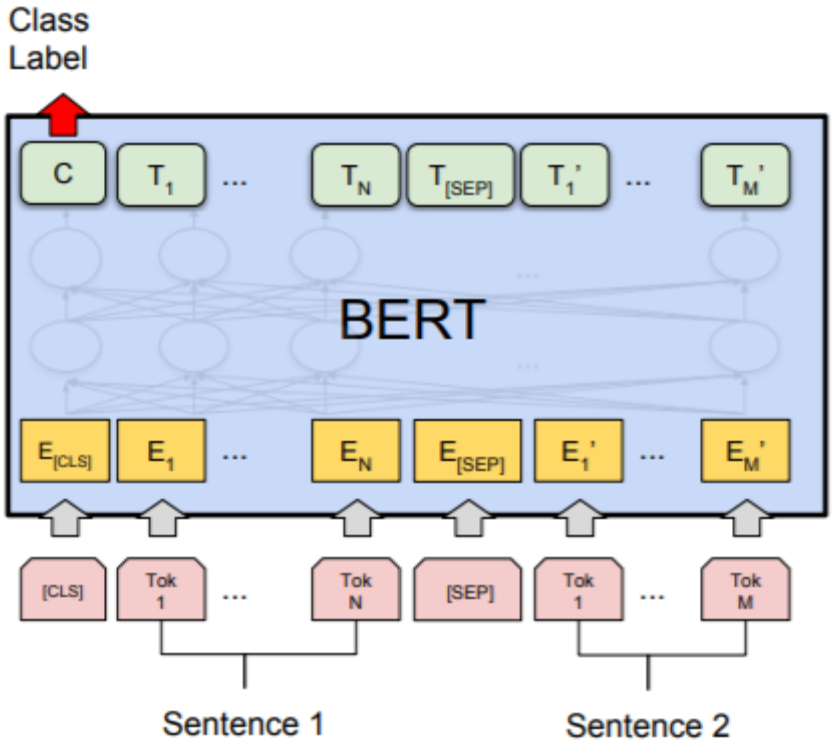
- Khoảng 15 % các mã của câu đầu váo được thay thế bởi [MASK] mã trước khi truyền vào mô hình đại diện cho những từ bị che dấu. Mô hình sẽ dựa trên các từ không được che dấu xung quanh [MASK] và đồng thời là bối cảnh của [MASK] để dự báo giá trị gốc của từ được che dấu.
- Bản chất của kiến trúc BERT vẫn là một mô hình seq2seq gồm 2 pha encoder giúp nhúng các từ đầu váo và decoder giúp tìm ra phân phối xác suất của các từ ở đầu ra. Kiến trúc Transfomer encoder được giữ lại trong nhiệm vụ Masked ML. Sau khi thực hiện self-attention và feed forward ta sẽ thu được các véc tơ nhúng ở đầu ra là O1,O2,…,O5
- Để tính toán phân phối xác suất cho từ đầu ra, chúng ta thêm một Fully connect layer ngay sau Transformer Encoder. Hàm softmax có tác dụng tính toán phân phối xác suất. Số lượng units của fully connected layer phải bằng với kích thước của từ điển.
- Ta thu được vector nhúng của mỗi một từ tại vị trí MASK sẽ là nhúng vector giảm chiều của vector Oi sau khi đi qua fully connected layer. Hàm loss function

của BERT sẽ bỏ qua mất mát từ những từ không bị che dấu và chỉ đưa vào mất mát của những từ bị che dấu. Do đó mô hình sẽ hội tụ lâu hơn nhưng đây là đặc tính bù trừ cho sự gia tăng ý thức về bối cảnh.

Hàm mất mát BERT chỉ xem xét dự đoán các giá trị bị che và bỏ qua dự đoán của các từ không bị che. Kết quả là, mô hình hội tụ chậm hơn các mô hình định hướng nhưng được bù đắp bởi nhận thức ngữ cảnh tăng lên của nó.

2.3.3. Next Sentence Prediction (NSP)

Đây là một bài toán phân loại học có giám sát với 2 nhãn. Đầu vào của mô hình là một cặp câu sao cho 50% câu thứ 2 được lựa chọn là câu tiếp theo của câu thứ nhất và 50% được lựa chọn một cách ngẫu nhiên từ bộ văn bản mà không có mối liên hệ gì với câu thứ nhất. Cần đánh dấu các vị trí đầu câu thứ nhất bằng token [CLS] và vị trí cuối các câu bằng token [SEP]. Các token này có tác dụng nhận biết các vị trí bắt đầu và kết thúc của từng câu thứ nhất và thứ hai.



Hình 2. 3: Mô hình đầu ra của NSP

Thông tin đầu vào được tiền xử lý trước khi đưa vào mô hình huấn luyện bao gồm:

- Ngữ nghĩa của từ: Thông qua các nhúng vector cho từng từ. Các vector được khởi tạo từ mô hình huấn luyện trước.
- Loại câu: Gồm hai vector là EA nếu từ thuộc câu thứ nhất và EB nếu từ thuộc câu thứ hai.

- Vị trí của từ trong câu là các vector E0,…,E10. Tương tự như nhúng vị trí trong transformer. Vector đầu vào sẽ bằng tổng của cả ba thành phần nhúng theo từ, câu và vị trí.

2.4. Các kiến trúc mô hình BERT

Hiện tại có nhiều phiên bản khác nhau của mô hình BERT. Các phiên bản đều dựa trên việc thay đổi kiến trúc của Transformer tập trung ở 3 tham số:

- L: số lượng các khối các tầng con trong transformer
- H: kích thước của VECTOR nhúng (hay còn gọi là hidden size)
- A: Số lượng từ đầu trong tầng nhiều từ đầu (multi-head layer), mỗi một từ đầu sẽ thực hiện một có chế tự chú ý (self-attention).

Tên gọi của 2 kiến trúc bao gồm:

- BERTBASE (L=12,H=768,A=12): Tổng tham số 110 triệu.
- BERTLARGE (L=24,H=1024,A=16): Tổng tham số 340 triệu.

Như vậy ở kiến trúc BERT Large chúng ta tăng gấp đôi số tầng, tăng kích thước ẩn của vector nhúng gấp 1.33 lần và tăng số lượng từ đầu trong multi-head layer gấp 1.33 lần.

CHƯƠNG III: MÔ HÌNH NGÔN NGỮ PHOBERT

3.1. Giới thiệu PhoBERT

Cho đến nay BERT vẫn được sử dụng cho nhiều bài toán NLP cho kết quả tốt với các phiên bản cải tiến, biến thể như RoBERTa, ALBERT, DistilBERT,... Tuy nhiên , huấn luyện mô hình BERT cho Tiếng Việt lại không hề đơn giản do đó rất khó để có thể áp dụng BERT cho các nhiệm vụ Tiếng Việt dù cho Google cũng có huấn luyện trước cho nhiều ngôn ngữ bao gồm cả tiếng Việt nhưng chưa cho kết quả thực hiện tốt nhất.

PhoBERT đã được ra đời là một mô hình BERT được huấn luyện trước cho tiếng Việt và đạt được nhiều kết quả tốt nhất cho nhiều nhiệm vụ trong xử lý ngôn ngữ tiếng Việt. Tác giả lấy tên Pho vì đây là món ăn phổ biến của Việt Nam.

PhoBERT dễ sử dụng, nó được xây dựng để sử dụng trong các thư viện như FAIRSeq của Facebook hay Transformers của Hugging Face nên giờ đây BERT lại càng phổ biến ngay cả với ngôn ngữ tiếng Việt hay tiếng Anh.

3.2. Cấu trúc của PhoBERT

Đây là một mô hình huấn luyện trước được huấn luyện cho chỉ huấn luyện dành riêng cho tiếng Việt. Tương tự như BERT, PhoBERT cũng có 2 phiên bản là:

- PhoBERTbase với 12 khối transformers
- PhoBERTlarge với 24 khối transformers.

PhoBERT được huấn luyện trên khoảng 20GB dữ liệu bao gồm khoảng 1GB ngữ liệu Wikipedia Tiếng Việt và 19GB còn lại lấy từ ngữ liệu tin tức tiếng Việt. Đây là một lượng dữ liệu đủ lớn để huấn luyện một mô hình như BERT. PhoBERT sử dụng RDRSegmenter của VnCoreNLP để tách từ cho dữ liệu đầu vào trước khi qua mã hóa BPE. PhoBERT chỉ sử dụng nhiệm vụ Mô hình ngôn ngữ đánh dấu để huấn luyện và không sử dụng nhiệm vụ dự đoán câu tiếp theo.

Các mô hình ngôn ngữ được huấn luyện trước, đặc biệt là BERT gần đây đã trở nên cực kỳ phổ biến và tạo ra các cải tiến đáng kể cho các nhiệm vụ NLP khác nhau. Sự thành công của BERT được huấn luyện và các biến thể của nó phần lớn đạt được kết quả tốt trong ngôn ngữ tiếng Anh.

Về mô hình tiếng Việt, có hai vấn đề cần quan tâm chính như sau:

- Kho tài liệu Wikipedia tiếng Việt là dữ liệu duy nhất được sử dụng để huấn luyện các mô hình ngôn ngữ đơn ngữ và đây cũng là tập dữ liệu tiếng Việt duy nhất được đưa vào làm dữ liệu tiền huấn luyện được sử dụng bởi tất cả các mô hình đa ngôn ngữ ngoại trừ XLM-R.
- Tất cả các mô hình ngôn ngữ dựa trên BERT đơn và đa ngôn ngữ được phát hành công khai đều không nhận thức được sự khác nhau giữa âm tiết tiếng

Việt và mã từ. Sự mơ hồ này đến từ thực tế là khoảng trắng cũng là được sử dụng để tách các âm tiết tạo thành từ khi viết bằng tiếng Việt

Để giải quyết hai mối quan tâm trên, các nhà nghiên cứu dựa trên mô hình đơn ngữ quy mô lớn đầu tiên BERTbase và BERTlarge tạo ra kho dữ liệu tên Tiếng Việt có tới 20GB cấp độ từ. Đánh giá các mô hình của mình trên bốn nhiệm vụ sau NLP cho tiếng Việt: gán nhãn từ loại POS, phân tích cú pháp phụ thuộc (Dependency Paser) và nhận dạng thực thể tên (NER), và nhiệm vụ hiểu ngôn ngữ của suy luận ngôn ngữ tự nhiên (NLI) có thể được xây dựng như một nhiệm vụ cấp độ âm tiết hoặc từ.

Kết quả thử nghiệm cho thấy rằng các mô hình của tác giả cho kết quả tốt nhất hiện tại (SOTA) trên tất cả các nhiệm vụ này. Những đóng góp đó như sau:

- Tạo ra các mô hình ngôn ngữ đơn ngữ quy mô lớn đầu tiên được huấn luyện huấn luyện trước dành riêng cho Tiếng Việt.
- Các mô hình đó giúp tạo ra các kết quả SOTA trên bốn nhiệm vụ cơ bản là gắn nhãn từ loại POS, phân tích cú pháp phụ thuộc DP, nhận dạng thực thể NER và suy luận ngôn ngữ tự nhiên NLI, do đó cho thấy hiệu quả của việc dựa trên BERT quy mô lớn các mô hình ngôn ngữ đơn ngữ cho tiếng Việt.
- Thực hiện bộ thử nghiệm đầu tiên để so sánh các mô hình ngôn ngữ đơn ngữ với mô hình đa ngôn ngữ tốt nhất gần đây XLM-R trong bốn nhiệm vụ cơ bản trên. Các thử nghiệm cho thấy rằng các mô hình hoạt động tốt hơn XLM-R về tất cả các nhiệm vụ này, do đó có thể thấy rằng các mô hình dành riêng cho ngôn ngữ cụ thể vẫn tốt hơn các mô hình đa ngôn ngữ.
- Đưa ra các mô hình dưới tên PhoBERT có thể được sử dụng với fairseq và transformer. Qua đó hi vọng PhoBERT có thể phục vụ như một cơ sở vững chắc cho NLP tiếng Việt trong tương lai để nghiên cứu và ứng dụng.

3.2.1. Thiết lập thử nghiệm

Hiệu suất của PhoBERT được đánh giá trên bốn nhiệm vụ NLP tiếng Việt: gán nhãn từ loại POS, Phân tích cú pháp phụ thuộc, nhận dạng thực thể và xử lý ngôn ngữ tự nhiên. Với bốn nhiệm vụ này dữ liệu được phân chia như sau:

Task	#training	#valid	#test
POS tagging†	27,000	870	2,120
Dep. parsing†	8,977	200	1,020
NER†	14,861	2,000	2,831
NLI‡	392,702	2,490	5,010

Hình 3. 1: Thống kê các bộ dữ liệu

Để cho nhiệm vụ POS, DP, và NER, thực hiện các bước tiền xử lý sử dụng bộ dữ liệu VnCoreNLP sử dụng các đánh giá tiêu chuẩn của VLSP với tập dữ liệu gán nhãn từ

loại 2013, ngân hàng câu phân tích phụ thuộc với gán nhãn từ loại của VnCoreNLP và VLSP 2016 Bộ dữ liệu NER.

Đối với NLI, sử dụng bộ kiểm tra và xác thực Tiếng Việt được xây dựng thủ công từ kho ngữ liệu NLI phiên bản 1.0 trong đó bộ huấn luyện tiếng Việt được phát hành dưới dạng phiên bản dịch bằng máy của bộ huấn luyện tiếng Anh tương ứng . Phân tích cú pháp Phụ thuộc và tập dữ liệu NER cung cấp từ vàng phân đoạn, đối với NLI, chúng tôi sử dụng RDRSegmenter để phân đoạn văn bản thành các từ trước khi áp dụng BPE để tạo ra các từ khóa con từ các mã thông báo từ.

3.2.2. Kết quả thực nghiệm

POS tagging (word-level)		**Dependency parsing** (word-level)	
Model	Acc.	Model	LAS / UAS
RDRPOSTagger (Nguyen et al., 2014a) [♣]	95.1	_	_
BiLSTM-CNN-CRF (Ma and Hovy, 2016) [♣]	95.4	VnCoreNLP-DEP (Vu et al., 2018) [★]	71.38 / 77.35
VnCoreNLP-POS (Nguyen et al., 2017) [♣]	95.9	jPTDP-v2 [★]	73.12 / 79.63
jPTDP-v2 (Nguyen and Verspoor, 2018) [★]	95.7	jointWPD [★]	73.90 / 80.12
jointWPD (Nguyen, 2019) [★]	96.0	Biaffine (Dozat and Manning, 2017) [★]	74.99 / 81.19
XLM-R_{base} (our result)	96.2	Biaffine w/ XLM-R_{base} (our result)	76.46 / 83.10
XLM-R_{large} (our result)	96.3	Biaffine w/ XLM-R_{large} (our result)	75.87 / 82.70
PhoBERT$_{base}$	96.7	Biaffine w/ PhoBERT$_{base}$	**78.77 / 85.22**
PhoBERT$_{large}$	**96.8**	Biaffine w/ PhoBERT$_{large}$	77.85 / 84.32

Hình 3. 2: : Điểm hiệu suất (tính bằng%) trong bộ dữ liệu đánh giá NER và NLI

Trong đó: [♦], [♠] và [■] biểu thị kết quả được báo cáo tương ứng. Kết quả Tiếng Việt NLI được huấn luyện cho XLM-R sẽ cao hơn khi tinh chỉnh sự kết hợp của tất cả 15 bộ dữ liệu huấn luyện từ ngữ liệu XNLI. Tuy nhiên, những kết quả đó có thể không thể so sánh được vì chúng ta chỉ sử dụng dữ liệu huấn luyện tiếng Việt đơn ngữ để tinh chỉnh. PhoBERT giúp tạo ra kết quả hiệu suất cao cho tất cả bốn nhiệm vụ ở trên.

Để gán nhãn từ loại, mô hình thần kinh chung WPD để gán nhãn từ loại và phân tích cú pháp phụ thuộc và mô hình dựa trên đặc trưng VnCoreNLP-POS là hai mô hình SOTA trước đó, có được độ chính xác vào khoảng 96,0%. PhoBERT cao hơn 0,8% độ chính xác hai mô hình này.

Đối với phân tích cú pháp Phụ thuộc, mức cao nhất trước đó điểm phân tích cú pháp LAS và UAS có được bởi Biaffine phân tích cú pháp lần lượt là 75,0% và 81,2%. PhoBERT giúp thúc đẩy trình phân tích cú pháp Biaffine với cải thiện tuyệt đối khoảng 4%, đạt được LAS ở mức 78,8% và UAS là 85,2%

Đối với NER, PhoBERTlarge tạo ra 1,1 điểm F1 cao hơn PhoBERTbase. Ngoài ra, PhoBERTbase cao hơn 2 điểm so với phương pháp SOTA trước đó và dựa trên mạng nơron mô hình VnCoreNLP-NER và BiLSTM-CNN-CRF được huấn luyện với bộ 15K dựa trên BERT nhúng từ ETNLP.

Đối với NLI, PhoBERT vượt trội hơn BERT đa ngôn ngữ và mô hình đa ngôn ngữ BERTbase với mục tiêu mô hình ngôn ngữ dịch mới XLMMLM + TLM với độ

chính xác cao. PhoBERT cũng hoạt động tốt hơn mô hình đa ngôn ngữ tốt nhất gần đây được huấn luyện trước XLM-R nhưng sử dụng ít tham số hơn XLM-R: 135M (PhoBERTbase) so với 250M (XLM-Rbase); 370 triệu (PhoBERTlarge) so với 560M (XLM-Rlarge).

3.2.3. Nhận xét

Kết quả cho thấy rằng PhoBERTlarge đạt được mức thấp hơn 0,9% điểm phân tích cú pháp phụ thuộc hơn PhoBERTbase. Một lý do có thể là lớp Transformer cuối cùng trong kiến trúc BERT có thể không phải là mã hóa thông tin tối ưu phong phú nhất về cấu trúc cú pháp. Trong tương lai cần nghiên cứu lớp Transformer của PhoBERT chứa thông tin cú pháp phong phú hơn bằng cách đánh giá hiệu suất phân tích cú pháp tiếng Việt từ mỗi lớp.

Việc sử dụng nhiều dữ liệu trước khi huấn luyện hơn có thể nâng cao chất lượng của ngôn ngữ được huấn luyện trước mô hình. Vì vậy, không có gì đáng ngạc nhiên rằng PhoBERT giúp tạo ra hiệu suất tốt hơn ETNLP trên NER và BERT đa ngôn ngữ và XLMMLM + TLM trên NLI.

Theo phương pháp tinh chỉnh đối với PhoBERT, chúng ta đã tinh chỉnh cẩn thận XLM-R cho các nhiệm vụ gán nhãn từ loại tiếng Việt, phân tích cú pháp Phụ thuộc và NER . PhoBERT cũng hoạt động tốt hơn XLM-R trên ba nhiệm vụ cấp từ này. Cần lưu ý rằng XLM-R sử dụng kho dữ liệu huấn luyện trước 2,5TB chứa 137GB văn bản tiếng Việt. PhoBERT thực hiện phân đoạn từ tiếng Việt để phân đoạn các câu ở cấp độ âm tiết thành các mã thông báo từ trước khi áp dụng BPE để phân đoạn các câu được phân đoạn từ thành các đơn vị từ khóa con, trong khi XLM-R trực tiếp áp dụng BPE cho các câu luyện trước tiếng Việt ở cấp độ âm tiết. Điều này xác nhận lại rằng các mô hình dành riêng cho ngôn ngữ cụ thể vẫn hoạt động tốt hơn đa ngôn.

3.2.4. Kết luận

Có thể thấy rằng PhoBERT –mô hình ngôn ngữ đơn ngữ quy mô lớn đầu tiên được huấn luyện đào tạo dành riêng cho Tiếng Việt - hoạt động tốt hơn so với sản phẩm tốt nhất gần đây mô hình đa ngôn ngữ XLM-R và giúp diễn giải SOTA thực hiện bốn nhiệm vụ NLP sau của Việt Nam là Gán nhãn từ loại, Sự phụ thuộc phân tích cú pháp, NER và NLI. Bằng cách phát hành công khai các mô hình PhoBERT, hy vọng rằng chúng có thể thúc đẩy các nghiên cứu và ứng dụng trong tương lai của NLP Việt Nam.

CHƯƠNG IV: NHẬN DẠNG CẢM XÚC VĂN BẢN TIẾNG VIỆT VỚI PHOBERT

4.1. Giới thiệu

Phân tích cảm xúc (Sentiment Analysis) hay khai phá quan điểm người dùng là lĩnh vực đã và đang thu hút được sự quan tâm của cộng đồng các nhà nghiên cứu cũng như các nhà phát triển ứng dụng trong lĩnh vực NLP. Cùng với sự phát triển của mạng máy tính toàn cầu và các thiết bị di động, người dùng đã tạo ra một lượng dữ liệu đánh giá khổng lồ trong quá trình họ tương tác trên các trang mạng xã hội, các trang diễn đàn, các trang đánh giá sản phẩm,… Do đó, việc khai thác các thông tin hữu ích từ dữ liệu đã được bình luận trên mạng sẽ giúp họ nắm được xu thế đang được đánh giá, bình luận hay thể hiện tình cảm về các sản phẩm, dịch vụ, sự kiện,… là khen hay chê và được thể hiện như thế nào.

Thể hiện cảm xúc là nhu cầu cơ bản của con người và chúng ta sử dụng ngôn ngữ không chỉ để truyền đạt sự thật, mà còn cả cảm xúc của chúng ta. Cảm xúc quyết định sự chất lượng cuộc sống của chúng tôi và chúng tôi tổ chức cuộc sống của mình để tối đa hóa trải nghiệm của những cảm xúc tích cực và giảm thiểu trải nghiệm của những cảm xúc tiêu cực

Phân tích cảm xúc là nhằm phát hiện ra thái độ mang tính lâu dài, màu sắc tình cảm, khuynh hướng niềm tin vào các đối tượng hay người nào đó. Các vấn đề xung quanh việc phân tích cảm xúc:

- Nguồn gốc của cảm xúc.
- Mục tiêu của cảm xúc.
- Các loại cảm xúc: thích, yêu, ghét, đánh giá, mong mỏi...
- Về mức độ cảm xúc: tích cực, tiêu cực, trung tính.
- Văn bản hàm chứa cảm xúc: một câu hoặc một đoạn văn bản.

Bài toán phân tích cảm xúc thuộc dạng bài toán phân tích ngữ nghĩa văn bản. Vì vậy, cần phải xây dựng một mô hình để hiểu được ý nghĩa của câu văn, đoạn văn để quyết định xem câu văn đó hoặc đoạn văn đó mang màu sắc cảm xúc chủ đạo nào.

Theo góc nhìn của máy học thì phân tích cảm xúc là bài toán phân lớp cảm xúc dựa trên ngôn ngữ tự nhiên. Đầu vào của bài toán là một câu hay một đoạn văn bản, còn đầu ra là các giá trị xác suất của N lớp cảm xúc mà ta cần xác định. Trong loại bài toán phân tích cảm xúc được phân thành các bài toán có độ khó khác nhau như sau:

- Đơn giản: Phân tích cảm xúc (thái độ) trong văn bản thành 2 lớp: tích cực (positive) và tiêu cực (negative).
- Phức tạp hơn: Xếp hạng cảm xúc (thái độ) trong văn bản từ 1 đến 5.

- Khó: Phát hiện mục tiêu, nguồn gốc của cảm xúc (thái độ) hoặc các loại cảm xúc (thái độ) phức tạp.

Hiện tại thì cộng đồng khoa học mới chỉ giải quyết tốt bài toán phân tích cảm xúc ở cấp độ đơn giản, tức là phân tích cảm xúc với 2 lớp cảm xúc tiêu cực và tích cực với độ chính xác hơn 85%.

Việc phân tích cảm xúc trong văn bản được ứng dụng trong hàng loạt các vấn đề như: Quản trị thương hiệu doanh nghiệp, thương hiệu sản phẩm, quản trị quan hệ khách hàng, khảo sát ý kiến xã hội học, phân tích trạng thái tâm lý con người...

Tuy nhiên, ngoài nét mặt, có thể sử dụng nhiều nguồn thông tin khác nhau để phân tích cảm xúc kể từ khi nhận dạng cảm xúc đã nổi lên như một quan trọng khu vực nghiên cứu. Và trong những năm gần đây, nhận dạng cảm xúc trong văn bản đã trở nên phổ biến hơn. Phổ biến do các ứng dụng tiềm năng to lớn của nó trong tiếp thị, an ninh, tâm lý học, tương tác giữa người và máy tính, trí tuệ nhân tạo,…

4.2. Dữ liệu

Nhận dạng cảm xúc là một cách tiếp cận cao hơn hoặc trường hợp đặc biệt của phân tích cảm xúc. Nhiệm vụ này, kết quả không được tạo ra theo hai cực: tích cực hoặc tiêu cực hoặc ở dạng xếp hạng (1-5) mà ở mức độ phân tích tình cảm chi tiết hơn, trong đó kết quả được mô tả bằng nhiều cách diễn đạt hơn như nỗi buồn , thích thú, tức giận, ghê tởm, sợ hãi và ngạc nhiên.

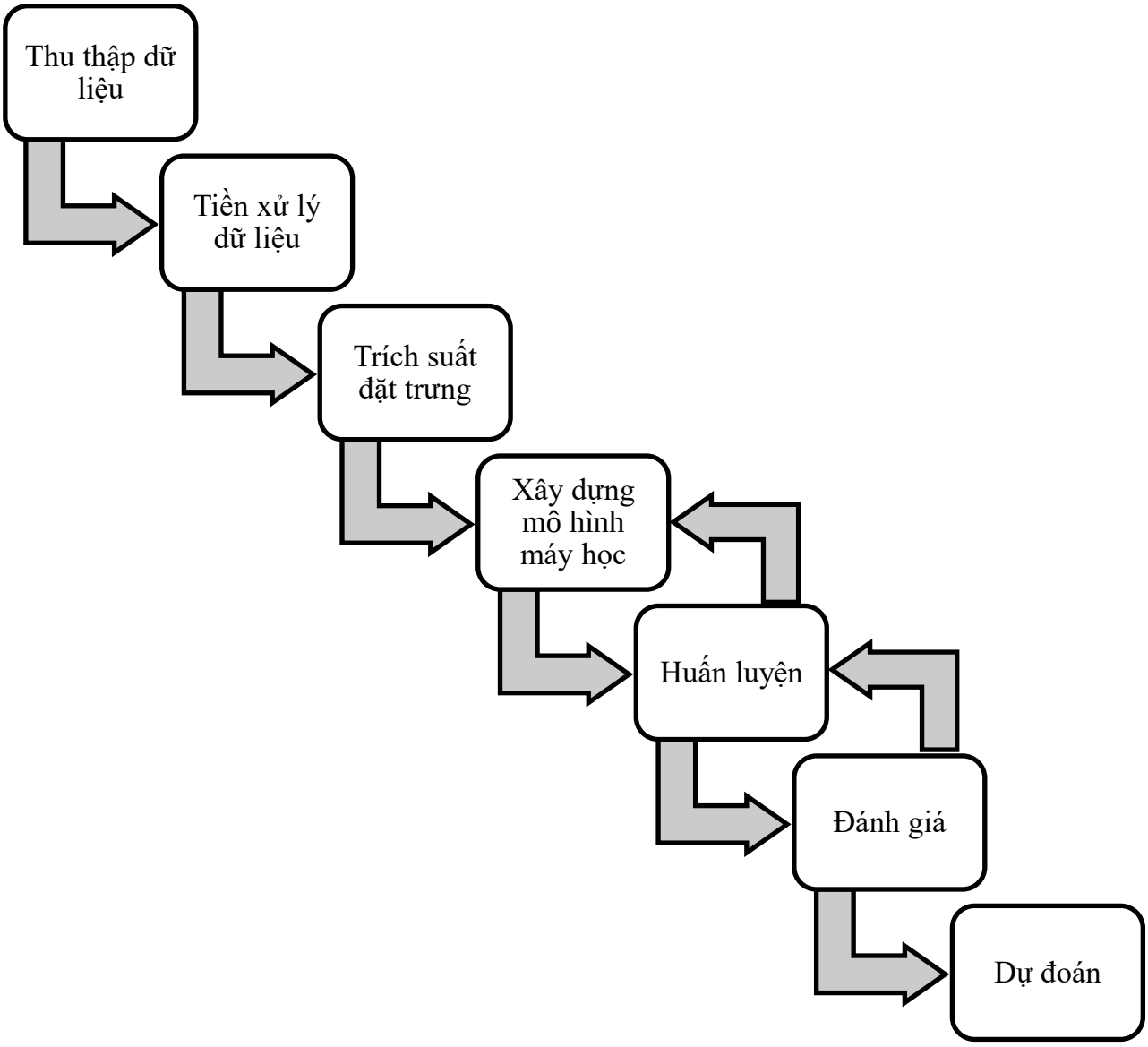
Bộ dữ liệu được thu thập ở Kaggle: Bộ dữ liệu gồm bình luận, điể đánh giá và nhãn. Các nhãn của bình luận là: POS, NEU, NEG tương đương 3 trạng thái cảm xúc: Tích cực, Trung tính và Tiêu cực.

https://www.kaggle.com/code/linhlpv/vietnamese-sentiment-analyst-base/data

Ví dụ bình luận tích cực: “sản phẩm đẹp quá”, “giao hàng hơi trễ 1 chút, nhưng sản phẩm tuyệt vời”

Ví dụ bình luận tiêu cực: “ quá thất vọng”, “sản phẩm quá đắt mà chất lượng bình thường”

4.3. Tiến trình học máy



4.4. Thực nghiệm

4.4.1. Thiết lập các thư viện về PhoBERT

Để load model PhoBERT sẽ cần cài đặt các packages sau đây:

- fairseq: Là project của facebook chuyên hỗ trợ các nghiên cứu và dự án liên quan đến model seq2seq.
- fastBPE: Là package hỗ trợ tokenize từ thành các từ phụ theo phương pháp mới nhất được áp dụng cho các pretrain model NLP hiện đại như BERT và các biến thể của BERT.
- vncorenlp: Là một package NLP trong Tiếng Việt, hỗ trợ tokenize và các tác vụ NLP khác.
- transformers: Là một project của huggingface hỗ trợ huấn luyện các model dựa trên kiến trúc transformer như BERT, GPT-2, RoBERTa, XLM,

DistilBert, XLNet, T5, CTRL,... phục vụ cho các tác vụ NLP trên cả nền tảng pytorch và tensorflow.

```
!pip3 install transformers
!pip install underthesea
!pip3 install fairseq
!pip3 install fastbpe
!pip3 install vncorenlp
!pip3 install fastBPE
```

Download các model pretrain từ PhoBERT

Ở đồ án này sử dụng pretrain model BERT base được huấn luyện từ package fair seq.

Sau khi download và giải nén pretrain file kiểm tra thấy bên trong folder sẽ bao gồm 3 files đó là:

* bpe.codes: Là BPE token mà mô hình để mã hóa văn bản sang index.
* dict.txt: Từ điển của bộ dữ liệu huấn luyện.
* model.pt: File lưu trữ của mô hình trên pytorch

```
!wget https://public.vinai.io/PhoBERT_base_fairseq.tar.gz
!tar -xzvf PhoBERT_base_fairseq.tar.gz
```

Tải model từ fairseq

```
from fairseq.models.roberta import RobertaModel
phoBERT = RobertaModel.from_pretrained('PhoBERT_base_fairseq', checkpoint_file='model.pt')
```

Tải pre-trained của PhoBERT.

```
!wget https://public.vinai.io/PhoBERT_base_transformers.tar.gz
!tar -xzvf PhoBERT_base_transformers.tar.gz
```

PhoBERT base transformers 4 file nhỏ bao gồm:

- `config.json` chứa config của model.
- `model.bin` lưu trữ pre-trained weight của model.
- `bpe.codes` và `dict.txt` chứ từ điển sẵn có của PhoBERT.

```
from fairseq.data.encoders.fastbpe import fastBPE
from fairseq import options
from fairseq.data import Dictionary
# Load the model in fairseq
from fairseq.models.roberta import RobertaModel

phoBERT_cls = RobertaModel.from_pretrained('PhoBERT_base_fairseq', checkpoint_file='model.pt')
phoBERT_cls.eval()  # disable dropout (or leave in train mode to finetune

# Load BPE
class BPE():
  bpe_codes = 'PhoBERT_base_fairseq/bpe.codes'

args = BPE()
phoBERT_cls.bpe = fastBPE(args) #Incorporate the BPE encoder into PhoBERT
```

Download package VnCoreNLP

```
# Download VnCoreNLP-1.1.1.jar & its word segmentation component (i.e. RDR Segmenter)
!mkdir -p vncorenlp/models/wordsegmenter
!wget https://raw.githubusercontent.com/vncorenlp/VnCoreNLP/master/VnCoreNLP-1.1.1.jar
!wget https://raw.githubusercontent.com/vncorenlp/VnCoreNLP/master/models/wordsegmenter/vi-vocab
!wget https://raw.githubusercontent.com/vncorenlp/VnCoreNLP/master/models/wordsegmenter/wordsegmenter.rdr
!mv VnCoreNLP-1.1.1.jar vncorenlp/
!mv vi-vocab vncorenlp/models/wordsegmenter/
!mv wordsegmenter.rdr vncorenlp/models/wordsegmenter/

from vncorenlp import VnCoreNLP
rdrsegmenter = VnCoreNLP("vncorenlp/VnCoreNLP-1.1.1.jar", annotators="wseg", max_heap_size='-Xmx500m')
```

4.4.2. Thu thập dữ liệu

Cài đặt các thư viện

```
import numpy as np
import pandas as pd
import torch
import regex
import re
from tqdm import tqdm
import torch

import nltk
nltk.download('punkt')
from nltk.tokenize import word_tokenize

from transformers import AutoModel, AutoTokenizer
from sklearn.model_selection import
train_test_split
from sklearn.preprocessing import LabelEncoder

from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

from sklearn import preprocessing
from sklearn.metrics import f1_score,
precision_score, recall_score
from sklearn.metrics import classification_report,
 confusion_matrix
```

Xây dựng hàm lấy dữ liệu

```
def get_data(path):
    df = pd.read_csv('/content/data - data.csv')
```

```
    df.columns = ['comment', 'label', 'rate', 'unname
d']
    # unused column
    df.drop(columns=['unnamed'], inplace=True)
    return df
df = get_data('/content/data - data.csv')
df.info()
```

4.4.3. Tiền xữ lý dữ liệu

Thực biện tách từ, loại bỏ các stopword.

```
def tokenize(column):
    tokens = nltk.word_tokenize(column)
    return [w for w in tokens if w.isalpha()]

stopword = set()
with open('/content/stopwords.txt', 'r', encoding='ut
f-8') as _fp:
    word = _fp.readlines()
stopword = [n.replace('\n', '') for n in word]

def remove_stopwords(tokenized_column):
    stops = stopword
    return [word for word in tokenized_column if
not word in stops]

def rejoin_words(tokenized_column):
    return ( " ".join(tokenized_column))

df['tokenized'] = df.apply(lambda x: tokenize(x['comm
ent']), axis=1)
df['stopwords_removed'] = df.apply(lambda x: remove_s
topwords(x['tokenized']), axis=1)
df['Sentence'] = df.apply(lambda x: rejoin_words(x['s
topwords_removed']), axis=1)
```

Xoá bỏ các cột thừa sao khi tiền xử lý dữ liệu

```
df.drop(columns=['comment'], inplace=True)
df.drop(columns=['tokenized'], inplace=True)
```

```
df.drop(columns=['stopwords_removed'], inplace=True)
df.drop(columns=['rate'], inplace=True)
```

Chia dữ liệu thành 3 tập Train, Test, Valid để huấn luyện mô hình.

```
X_train, X_test, y_train, y_test = train_test_split(df['Sentence'],df['label'], test_size = .15, shuffle = True, stratify=df['label'])

X_train, X_valid, y_train, y_valid = train_test_split(X_train, y_train, test_size = .2,
shuffle = True, stratify=y_train)
```

4.4.4. Trích suất đặt trưng

Tokenize các câu văn sang chuỗi index và padding câu văn về cũng một độ dài.

```
max_sequence_length = 256
def convert_lines(lines, vocab, bpe):
  '''
  lines: list các văn bản input
  vocab: từ điển dùng để encoding subwords
  bpe:
  '''
  # Khởi tạo ma trận output
  outputs = np.zeros((len(lines), max_sequence_length), dtype=np.int32) # -->
> shape (number_lines, max_seq_len)
  # Index của các token cls (đầu câu), eos (cuối câu), padding (padding token)
  cls_id = 0
  eos_id = 2
  pad_id = 1

  for idx, row in tqdm(enumerate(lines), total=len(lines)):
    # Mã hóa subwords theo byte pair encoding(bpe)
    subwords = bpe.encode('<s> '+ row +' </s>')
    input_ids = vocab.encode_line(subwords, append_eos=False, add_if_not_exist=False).long().tolist()
    # Truncate input nếu độ dài vượt quá max_seq_len
```

```
    if len(input_ids) > max_sequence_length:
      input_ids = input_ids[:max_sequence_length]
      input_ids[-1] = eos_id
    else:
      # Padding nếu độ dài câu chưa bằng max_seq_len
      input_ids = input_ids + [pad_id, ]*(max_sequenc
e_length - len(input_ids))

    outputs[idx,:] = np.array(input_ids)
  return outputs

# Load the dictionary
vocab = Dictionary()
vocab.add_from_file("/content/PhoBERT_base_transforme
rs/dict.txt")
```

Trích đặt trưng cho tập Train, Test, Valid.

```
X_train_pho = convert_lines(X_train,vocab, phoBERT_cl
s.bpe)
X_test_pho = convert_lines(X_test, vocab, phoBERT_cls
.bpe)
X_valid_pho = convert_lines(X_valid, vocab, phoBERT_c
ls.bpe)
```

Chuẩn hoá các nhãn về dạng số, và đổi kích thước nhãn

```
lb = LabelEncoder()
lb.fit(df['rate'])
y_train = lb.fit_transform(y_train)
y_test = lb.fit_transform(y_test)
y_train = y_train.reshape(-1,1)
y_test = y_test.reshape(-1,1)
print(lb.classes_)
print('Top 5 classes indices: ', y_train[:5])
```

4.4.5. Xây dựng mô hình và huấn luyện

Support vector machine – SVM

```
SVM = SVC(C = 1, kernel='rbf', gamma=1)
```

```
SVM.fit(X_train_pho, y_train)

y_pred_SVM = SVM.predict(X_test_pho)
y_valid_SVM = SVM.predict(X_valid_pho)

print("Validation accuracy: ", accuracy_score(y_pred_
SVM, y_test))
```

K-nearest neighbor - KNN

```
KNN = KNeighborsClassifier(n_neighbors = 100, p = 2)
KNN.fit(X_train_pho, y_train)

y_pred_KNN = KNN.predict(X_test_pho)

print("Validation accuracy: ", accuracy_score(y_pred_
KNN, y_test))
```

Random Forest

```
RF=RandomForestClassifier(n_estimators=100)
RF.fit(X_train_pho,y_train)

y_pred_RF = RF.predict(X_test_pho)

print("Validation accuracy: ", accuracy_score(y_pred_
RF, y_test))
```

4.4.6. Đánh giá

```
    def Eval(y_pred_md, y_test):
        print('F1 score:', f1_score(y_test, y_pred_md,
 average="macro"))
        print('Recall:', recall_score(y_test, y_pred_m
d, average="macro"))
        print('Precision:', precision_score(y_test, y_
pred_md, average="macro"))
        print('\n confussion matrix:\n', confusion_mat
rix(y_test, y_pred_md))
        a = classification_report(y_test, y_pred_md)
        return print(a)
```

```
print('Evaluate SVM: ')
Eval(y_test, y_pred_SVM)

print('Evaluate KNN: ')
Eval(y_test, y_pred_KNN)

print('Evaluate RF: ')
Eval(y_test, y_pred_RF)
```

CHƯƠNG V: KẾT LUẬN

5.1. Kết quả đạt được

Đồ án nghiên cứu đã trình bày được các mô hình ngôn ngữ BERT và PhoBERT.

Xây dựng được những mô hình máy học phân lớp.

Thực nghiệm đặc trưng PhoBert với các mô hình máy học phân lớp.

Xây dựng được một chương trình minh hoạ ứng dụng các mô hình máy học phân lớp để dự đoán trạng thái cảm xúc cho văn bản.

5.2. Hạn chế

Do thời gian thực hiện đồ án tương đối hạn chế nên không thể tránh được những thiếu sót nhất định.

Hệ thống nhận dạng chưa được tối ưu hóa về mặt giao diện, vì vậy chưa phù hợp để triển khai cho người dùng.

Dự liệu huấn luyện chưa đủ lớn để trích suất đặt trưng từ PhoBERT.

Độ chính xác của các mô hình chưa cao, do các nguyên nhân như tiền xữ lý dữ liệu kém dẫn đến dư thừa dữ liệu bị nhiễu. Kèm với lượng dữ liệu của các nhãn không cân bằng nên tần suất bị hỗn loạn dẫn đến độ chính sát chưa cao.

Áp dụng với những bộ dữ liệu gồm 2 nhãn độ chính sát lên đến 80%. Những với những bộ dữ liệu gồm nhiều nhãn độ chính sát sẽ khá thấp.

Chưa xây dựng được những mô hình Deep Learning để phân loại cảm xúc văn bản.

Chưa nghiên cứu sâu về đề tài, khả năng thu thập, trắc lọc thông tin còn kém.

Hạn chế về vốn hiểu biết Tiếng Anh, dẫn đến hạn chế về việc sử dụng các tài liệu nước ngoài.

5.3. Hướng phát triển

Tiếp tục nghiên cứu và tìm hiểu về các mô hình ngôn ngữ, các phương pháp trích chọn đặc trưng khác và phân tích ưu nhược điểm của chúng.

Xây dựng giao diện cơ bản để trực quan hoá mô hình để người dùng dễ dàng sử dụng.

Nghiên cứu và tìm hiểu một số thuật toán mạnh mẽ để nhận dạng được chính xác hơn. Các phương pháp tiền xử lý các trường hợp có thể xảy ra trong thực tế.

Nghiên cứu ứng dụng cho bài toán nhận dạng, phân loại cảm xúc. Các tập dữ liệu lớn hơn, các thuật toán khác của học máy cần được thử nghiệm và đánh giá để có cái nhìn toàn diện hơn về các kết quả đạt được.

CHƯƠNG VI: TÀI LIỆU THAM KHẢO

[1]. Bùi Văn Minh (2010). Kiểm duyệt bài viết và bình luận Tiếng Việt có nội dung không phù hợp trên mạng xã hội Facebook. Luận văn Thạc Sĩ, Trường Đại Học Công Nghệ, Đại Học Quốc Gia Hà Nội.

[2]. Đào Thị Lệ Thủy, Trịnh Văn Loan, Nguyễn Hồng Quang, Lê Xuân Thành (2017). Ảnh hưởng của đặc trưng phổ tín hiệu tiếng nói đến nhận dạng cảm xúc Tiếng Việt. Kỷ yếu Hội nghị Quốc gia lần thứ X về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR), Đà Nẵng

[3]. Ths. Nguyễn Thị Xuân Hương, Nguyễn Thành Long (2021). Tìm hiểu về mô hình ngôn ngữ PhoBERT cho bài toán phân loại quan điểm bình luận Tiếng Việt. Trường Đại Học Quản Lý và Công Nghệ Hải Phòng.

[4]. Hu Xu , Bing Liu , Lei Shu and Philip S. Yu, BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis.

[5]. RoBERTa: A Robustly Optimized BERT Pretraining Approach by Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov.

[6]. Nguyen Dat Quoc, Nguyen Anh. (2020). PhoBERT: Pre-trained language models for Vietnamese.