|  |  |
| --- | --- |
|  |  |
| **VƯƠNG THỊ THANH CẢNH** | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------** |
|  |
| **ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC** |
| **NGÀNH CÔNG NGHỆ THÔNG TIN** |
| **TÌM HIỂU MỘT SỐ MÔ HÌNH XỬ LÝ NGÔN NGỮ TỰ NHIÊN VỚI BERT VÀ ỨNG DỤNG TRONG NHẬN DIỆN CẢM XÚC VĂN BẢN** |
|  |
|  |
| **CBHD :TS. Đặng Trọng Hợp** |
| **CÔNG NGHỆ THÔNG TIN** | **Sinh viên : Vương Thị Thanh Cảnh** |
| **Mã số sinh viên: 2019606720** |
|  |
|  |
|  |
| **Hà Nội – Năm 2023** |
|  |

# LỜI CẢM ƠN

Đầu tiên, em xin cảm ơn thầy Đặng Trọng Hợp đã giúp đỡ và hướng dẫn tận tình cho em trong suốt thời gian em thực hiện đề tài đồ án tốt nghiệp này.

Em xin gửi lời cảm ơn đến Ban Giám Hiệu Trường Đại học Công nghiệp Hà Nội, các Ban, Ngành đã hỗ trợ hết mức tạo điều kiện tốt nhất để em có thể đăng ký đề tài đồ án tốt nghiệp.

Em xin cảm ơn đến các thầy, các cô Khoa Công nghệ thông tin, Trường Đại học Công nghiệp Hà Nội, đã giúp em có những kiến thức cực kì bổ ích trong vòng 4 năm vừa qua, giúp em có được nền tảng kiến thức vững chắc để em có thể thực hiện được đồ án.

Với những kiến thức còn thiếu sót, những kỹ năng còn hạn hẹp, em không thể tránh được những sai sót, em mong thầy cô có thể thông cảm và bỏ qua cho em.

Lời cuối cùng em xin chúc quý thầy cô sẽ luôn khỏe mạnh, hoàn thành tốt công việc của mình và gặt hái được nhiều thành công hơn trong tương lai. Em xin chúc nhà trường ngày càng phát triển và đạt được thêm nhiều thành tích cao hơn nữa.

Em xin chân thành cảm ơn!

Sinh viên

Vương Thị Thanh Cảnh

**MỤC LỤC**

[LỜI CẢM ƠN 2](#_Toc134495095)

[DANH MỤC CÁC THUẬT NGỮ, KÝ HIỆU VÀ CHỮ VIẾT TẮT 6](#_Toc134495096)

[DANH MỤC BẢNG BIỂU 7](#_Toc134495097)

[DANH MỤC HÌNH ẢNH 8](#_Toc134495098)

[MỞ ĐẦU 9](#_Toc134495099)

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 11](#_Toc134495100)

[1.1. Tên đề tài 11](#_Toc134495101)

[1.2. Lý do chọn đề tài 11](#_Toc134495102)

[1.3. Mục tiêu của đề tài 12](#_Toc134495103)

[1.4. Đối tượng và phạm vi 12](#_Toc134495104)

[1.5. Kết quả dự kiến đạt được 13](#_Toc134495105)

[1.6. Tổng kết chương 1 13](#_Toc134495106)

[CHƯƠNG 2: TÌM HIỂU VỀ MÔ HÌNH BERT 14](#_Toc134495107)

[2.1. Khái niệm 14](#_Toc134495108)

[2.2. Tại sao lại cần BERT? 15](#_Toc134495109)

[2.3. Một số khái niệm liên quan 16](#_Toc134495110)

[2.3.1. Nhiệm vụ phía sau (Downstream task) 16](#_Toc134495111)

[2.3.2. Điểm khái quát đánh giá mức độ hiểu ngôn ngữ (GLUE score benchmark) 16](#_Toc134495112)

[2.3.3. Phân tích cảm xúc (Sentiment Analysis) 16](#_Toc134495113)

[2.3.4. Hỏi đáp (Question and Answering) 17](#_Toc134495114)

[2.3.5. Suy luận ngôn ngữ (Natural Language Inference) 17](#_Toc134495115)

[2.3.6. Quan hệ văn bản (Textual Entailment) 17](#_Toc134495116)

[2.3.7. Ngữ cảnh (Contextual) 17](#_Toc134495117)

[2.3.8. Phương pháp Hiện đại nhất (SOTA) 18](#_Toc134495118)

[2.3.9. Mô hình Left – to – Right (LTR) 18](#_Toc134495119)

[2.3.10. Masked Language Model - Mô hình ngôn ngữ được đánh dấu (MLM) 18](#_Toc134495120)

[2.4. Ngữ cảnh (Contextual) và vai trò trong NLP 18](#_Toc134495121)

[2.5. Phương pháp TRANSFORMER 20](#_Toc134495122)

[2.5.1. Encoder 21](#_Toc134495123)

[2.5.2. Decoder 26](#_Toc134495124)

[2.6. Mô hình BERT 27](#_Toc134495125)

[2.6.1. Mô hình BERT tinh chỉnh (Fine-tuning model BERT) 28](#_Toc134495126)

[2.6.2. Cách huấn luyện BERT 28](#_Toc134495127)

[2.7. Các kiến trúc mô hình BERT 32](#_Toc134495128)

[2.8. Tổng kết chương 2 33](#_Toc134495129)

[CHƯƠNG 3: PHOBERT 34](#_Toc134495130)

[3.1. Sự ra đời của PhoBERT 34](#_Toc134495131)

[3.2. Cấu trúc của PhoBERT 34](#_Toc134495132)

[3.3. Ứng dụng của PhoBERT 36](#_Toc134495133)

[3.4. Kết luận 36](#_Toc134495134)

[3.5. Tổng kết chương 3 36](#_Toc134495135)

[CHƯƠNG 4: ỨNG DỤNG PHOBERT VÀO BÀI TOÁN NHẬN DIỆN CẢM XÚC VĂN BẢN 38](#_Toc134495136)

[4.1. Phát biểu bài toán 38](#_Toc134495137)

[4.2. Dữ liệu, công cụ và môi trường thực nghiệm 40](#_Toc134495138)

[4.2.1. Dữ liệu 40](#_Toc134495139)

[4.2.2. Công cụ và môi trường thực nghiệm 42](#_Toc134495140)

[4.3. Các bước thực hiện 44](#_Toc134495141)

[4.3.1. Cài đặt các thư viện cần thiết 45](#_Toc134495142)

[4.3.2. Import các thư viện cần thiết 45](#_Toc134495143)

[4.3.3. Load model PhoBERT 46](#_Toc134495144)

[4.3.4. Chuẩn hóa dữ liệu 46](#_Toc134495145)

[4.3.5. Load dữ liệu từ file train.csv để train model 47](#_Toc134495146)

[4.3.6. Tạo BERT features 48](#_Toc134495147)

[4.3.7. Train model SVM 49](#_Toc134495148)

[4.3.8. Test thử kết quả chạy model 49](#_Toc134495149)

[4.4. Tổng kết chương 4 52](#_Toc134495150)

[KẾT LUẬN 53](#_Toc134495151)

[TÀI LIỆU THAM KHẢO 54](#_Toc134495152)

# DANH MỤC CÁC THUẬT NGỮ, KÝ HIỆU VÀ CHỮ VIẾT TẮT

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Từ viết tắt** | **Đầy đủ** | **Ý nghĩa** |
| 1 | BERT | Bidirectional Encoder Representations from Transformers | Mô hình biểu diễn từ theo 2 chiều ứng dụng Transformers |
| 2 | NLP | Natural Language Processing | Xử lý ngôn ngữ tự nhiên |
| 3 | NSP | Next Sentence Prediction | Dữ báo câu tiếp theo |
| 4 | SOTA | State-Of-the-Art | Hiện đại nhất |
| 5 | MLM | Masked Language Model | Mô hình ngôn ngữ Masked |
| 6 | RNN | Recurrent Neural Network | Mạng neural hồi quy |
| 7 | LM | Language Model | Mô hình ngôn ngữ |
| 8 | ELMo | Embeddings from Language Model | Nhúng từ Mô hình ngôn ngữ |
| 9 | GLUE | General Language Understanding Evaluation | Điểm khái quát đánh giá mức độ hiểu ngôn ngữ |
| 10 | NLI | Natural Language Inference | Suy luận ngôn ngữ tự nhiên |
| 11 | NER | Name Entity Recognition | Nhận diện thực thể trong câu |
| 12 | SQuAD | Stanford Question Answering Dataset | Tác vụ hỏi – đáp |

# DANH MỤC BẢNG BIỂU

[Bảng 4.1. Kết quả chạy model PhoBERT đã huấn luyện trong nhận diện cảm xúc văn bản 53](#_Toc134399295)

# DANH MỤC HÌNH ẢNH

[Hình 2.1. Sơ đồ kiến trúc của Transformer 23](#_Toc134112841)

[Hình 2.2. Encoder 24](#_Toc134112842)

[Hình 2.3. Biểu diễn nhúng từ 25](#_Toc134112843)

[Hình 2.4. Ví dụ mã hóa vị trí từ nhúng 26](#_Toc134112844)

[Hình 2.5. Ví dụ về self - attention 27](#_Toc134112845)

[Hình 2.6. Multi-head Attention trong câu 27](#_Toc134112846)

[Hình 2.7. Residuals trong transformer 28](#_Toc134112847)

[Hình 2.8. Tương quan giữa Encoder và Decoder 29](#_Toc134112848)

[Hình 2.9. Toàn bộ tiến trình pre-training và fine-tuning của BERT 31](#_Toc134112849)

[Hình 2.10. Sơ đồ kiến trúc BERT cho nhiệm vụ ngôn ngữ mô hình được đánh dấu 33](#_Toc134112850)

[Hình 2.11. Các bước tạo đầu vào trong tác vụ NSP 35](#_Toc134112851)

[Hình 2.12. Mô hình lấy đầu ra trong NSP 35](#_Toc134112852)

[Hình 4.1. Sơ đồ phân tích cảm xúc văn bản 39](#_Toc134495057)

[Hình 4.2. Mô hình nhận diện cảm xúc văn bản với HaUI 40](#_Toc134495058)

[Hình 4.3. Cài đặt thư viện 45](#_Toc134495059)

[Hình 4.4. Import thư viện 46](#_Toc134495060)

[Hình 4.5. Load model PhoBERT 46](#_Toc134495061)

[Hình 4.6. Hàm chuẩn hóa câu 47](#_Toc134495062)

[Hình 4.7. Load danh sách các từ vô nghĩa 47](#_Toc134495063)

[Hình 4.8. Load dữ liệu trừ file train.csv 48](#_Toc134495064)

[Hình 4.9. Padding input 48](#_Toc134495065)

[Hình 4.10. Thêm attention\_mask 48](#_Toc134495066)

[Hình 4.11. Đưa vào model và lấy ra output 49](#_Toc134495067)

[Hình 4.12. Train model SVM 49](#_Toc134495068)

# MỞ 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 (ngôn ngữ tự nhiên của con người) 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 (Presence) và tần suất xuất hiện (Frequency), mô hình ngôn ngữ (n-gram), thông tin nhãn từ loại (Parts of Speech), thông tin phân tích ngữ pháp (Syntactic parsing), biểu diễn véc tơ từ (Word2Vec), nhúng ký tự (Character Embedding), mạng ngữ nghĩa (WordNet), mạng từ điển quan điểm (SentiWordNet), v.v. 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, em chọn đề tài “Tìm hiểu mô hình PhoBert cho bài toán phân loại quan điểm bình luận Tiếng Việt” nhằm tìm hiểu các phương pháp mới biểu diễn cho ngôn ngữ tiếng Việt và áp dụng nó cho bài toán nhận diện cảm xúc văn bản tiếng Việt. Đồ án thiết kế gồm 4 chương:

- Chương 1: Tổng quan về đề tài

- Chương 2: Tìm hiểu về mô hình BERT - trình bày về mô hình BERT và các khái niệm liên quan

- Chương 3: Mô hình PhoBERT - trình bày về các tìm hiểu cho mô hình PhoBERT

- Chương 4: Ứng dụng PhoBERT cho bài toán nhận diện cảm xúc văn bản tiếng Việt trong đó trình bày về bài toán, công cụ sử dụng và các cài đặt thử nghiệm, cuối cùng là phần đánh giá và kết luận.

# CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

## Tên đề tài

Tìm hiểu một số mô hình xử lý ngôn ngữ tự nhiên với BERT và ứng dụng trong nhận diện cảm xúc văn bản.

## Lý do chọn đề tài

Nhận diện 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,...

Chúng ta đang sống trong kỷ nguyên số, đặc biệt những năm gần đây nổi lên với mạng xã hội, với hàng triệu người dùng trên thế giới, với lượng thông tin nội dung được người dùng tạo ra hằng ngày cực kỳ lớn, với đa dạng các hình thức như dòng trạng thái, hình ảnh, video. Mạng xã hội có những đặc điểm là: thông tin do người dùng tạo ra, mang tính cá nhân cho nên chất lượng nội dung hay tính đúng đắn, xác thực là tương đối; một thông tin mới được tạo lại có sức lan tỏa nhanh đến đông đảo các người dùng khác, so với các kênh thông tin truyền thống như truyền hình, báo chí, diễn đàn,...

Điều này đặt ra cho các doanh nghiệp lớn giải quyết bài toán quản trị thương hiệu doanh nghiệp, quản trị thương hiệu sản phẩm trước các dư luận không tốt trên mạng xã hội rất khó khăn, cả về nguồn xuất phát thông tin, cả về khối lượng thông tin cần xử lý. Chưa kể việc các đối thủ cạnh tranh trên thương trường lợi dụng mạng xã hội để cố ý tạo các thông tin bất lợi cho nhau.

Giải pháp công nghệ hiện nay được gọi là “lắng nghe mạng xã hội”, tức là các doanh nghiệp CNTT mua các dữ liệu thời gian thực (real time) từ các công ty mạng xã hội về để xử lý các thông tin liên quan đến doanh nghiệp hay các sản phẩm mà doanh nghiệp đó kinh doanh, nhằm phát hiện và ngăn chặn sớm sự lan rộng các thông tin bất lợi trên mạng xã hội, có hình thức đính chính phản hồi đến các khách hàng của mình, đồng thời thương lượng, ngăn chặn tận gốc những người tạo ra các nội dung đó. Điều cốt yếu của giải pháp này chính là phân tích cảm xúc của các dòng trạng thái trên mạng xã hội nhằm lọc ra các thông tin bất lợi để xử lý.

Đối với tiếng Việt, mô hình PhoBERT được coi là một SOTA – Language Model đỉnh nhất để thực hiện các tác vụ xử lý ngôn ngữ tự nhiên. Nghiên cứu các mô hình xử lý ngôn ngữ trong nhận diện cảm xúc văn bản hiện nay là một đề tài hấp dẫn, có tính cấp thiết và có giá trị thực tiễn cao. Nhận thức được sự quan trọng và cũng như để có thể áp dụng được những kiến thức đã được học và tìm hiểu, em xin được áp dụng những kiến thức đã được học và em chọn đề tài đồ án “**Tìm hiểu một số mô hình xử lý ngôn ngữ tự nhiên với BERT và ứng dụng trong nhận diện cảm xúc văn bản**”, trong đó tập trung nghiên cứu sử dụng mô hình PhoBERT để nhận diện cảm xúc văn bản.

## Mục tiêu của đề tài

Đề tài: Tìm hiểu một số mô hình xử lý ngôn ngữ tự nhiên với BERT và ứng dụng trong nhận diện cảm xúc văn bảnđáp ứng được những mục tiêu:

Nghiên cứu mô hình BERT trong nhiệm vụ xử lý ngôn ngữ.

Nắm được các kiến thức cơ bản về Python, model PhoBERT và các thư viện cần thiết để train model,… cùng các kiến thức, công cụ liên quan.

Ứng dụng PhoBERT trong nhận diện cảm xúc văn bản với bộ dữ liệu đầu vào là các đánh giá về trường Đại học Công nghiệp Hà Nội đã được gán nhãn đại diện cho 3 sắc thái tích cực, tiêu cực và trung tính. Từ đó, phân loại được các câu bình luận về trường theo 3 nhãn trên.

## Đối tượng và phạm vi

Đối tượng nghiên cứu: Mô hình PhoBERT trong nhận diện cảm xúc văn bản tiếng Việt.

Phạm vi nghiên cứu: Dựa trên bộ dữ liệu đầu vào được huấn luyện sẵn, đã bao gồm gán nhãn phân loại tích cực, tiêu cực và trung tính. Mô hình hoàn thiện sẽ cho ra kết quả phân loại câu văn bản theo 3 nhãn như trên.

## Kết quả dự kiến đạt được

Nghiên cứu một cách tổng quan về mô hình Transformer, mô hình BERT, nghiên cứu chi tiết mô hình PhoBERTbase.

Nghiên cứu các công cụ phần mềm được sử dụng trong nhận diện cảm xúc văn bản. Từ đó, xây dựng được ứng dụng nhận diện cảm xúc văn bản tiếng Việt.

## Tổng kết chương 1

Trong chương này em đã trình bày những hướng cơ bản trong nghiên cứu để tìm hiểu về đề tài. Em tìm hiểu các nội dung về lý do chọn đề tài, mục tiêu của đề tài, đối tượng và phạm vi nghiên cứu từ đó tìm ra được hướng để tìm hiểu và hoàn thiện đề tài.

# CHƯƠNG 2: TÌM HIỂU VỀ MÔ HÌNH BERT

## Khái niệm

BERT (Bidirectional Encoder Representations from Transformers) là một mô hình ngôn ngữ hay mô hình biểu diễn ngôn ngữ (Language Model – LM) được tạo ra bởi Google AI và được giới thiệu vào năm 2018. BERT được coi như là đột phá lớn trong Machine Learning bởi vì khả năng ứng dụng của nó vào nhiều bài toán xử lý ngôn ngữ tự nhiên NLP (Natural Language Processing) khác nhau như: Question Answering, Natural Language Inference,... 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 con người (các 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 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 Bidirectional Encoder Representations from Transformers (BERT).

## Tại sao lại cần BERT?

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.

Ví dụ : Trong OpenAI GPT, các tác giả sử dụng đã kiến trúc left-to-right, nghĩa là các từ chỉ phụ thuộc vào các từ ở trước đó.

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 (model) “chung” nhất với tập dữ liệu khổng lồ trên internet (pre-training) được xây dựng và có thể được “tinh chỉnh” (fine-tune) cho các bài toán cụ thể.

Nhờ có kỹ thuật này mà kết quả cho các bài toán được cải thiện rõ rệt, không chỉ trong xử lý ngôn ngữ tự nhiên mà còn trong các lĩnh vực khác như Computer Vision,... 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.

## Một số khái niệm liên quan

### Nhiệm vụ phía sau (Downstream task)

Là những nhiệm vụ học hỏi được giám sát được cải thiện dựa trên những mô hình được huấn luyện trước.

Ví dụ: Chúng ta sử dụng lại các biểu diễn từ học được từ những mô hình được huấn luyện trước trên bộ văn bản lớn vào một nhiệm vụ phân tích cảm xúc huấn luyện trên bộ văn bản có kích thước nhỏ hơn. Áp dụng nhúng huấn luyện trước (pretrain-embedding) đã giúp cải thiện mô hình. Như vậy nhiệm vụ sử dụng nhúng huấn luyện trước được gọi là nhiệm vụ sau.

### Điểm khái quát đánh giá mức độ hiểu ngôn ngữ (GLUE score benchmark)

GLUE score benchmark là một tập hợp các chỉ số được xây dựng để đánh giá khái quát mức độ hiểu ngôn ngữ của các mô hình NLP.

Các đánh giá được thực hiện trên các bộ dữ liệu tiêu chuẩn được quy định tại các hiệp định về phát triển và thúc đẩy NLP. Mỗi bộ dữ liệu tương ứng với một loại tác NLP vụ như:

• Phân tích tình cảm (Sentiment Analysis)

• Hỏi đáp (Question and Answering)

• Suy luận ngôn ngữ tự nhiên (NLI - Natural Languague Inference)

• Dự báo câu tiếp theo (NSP - Next Sentence Prediction)

• Nhận diện thực thể trong câu (NER - Name Entity Recognition)

### Phân tích cảm xúc (Sentiment Analysis)

Phân loại cảm xúc văn bản thành 2 nhãn tích cực (positive) và tiêu cực (negative). Thường được sử dụng trong các hệ thống đánh giá bình luận của người dùng.

### Hỏi đáp (Question and Answering)

Là thuật toán hỏi và đáp. Đầu vào là một cặp câu (pair sequence) bao gồm: câu hỏi (question) có chức năng hỏi và đoạn văn bản (paragraph) chứa thông tin trả lời cho câu hỏi. Một bộ dữ liệu chuẩn nằm trong GLUE dataset được sử dụng để đánh giá nhiệm vụ hỏi và đáp là SQuAD - Stanford Question Answering Dataset.

### Suy luận ngôn ngữ (Natural Language Inference)

Là các nhiệm vụ suy luận ngôn ngữ đánh giá mối quan hệ giữa các cặp câu, cũng tương tự như Textual Entailment.

### Quan hệ văn bản (Textual Entailment)

Là nhiệm vụ đánh giá mối quan hệ định hướng giữa 2 văn bản. Nhãn đầu ra của các cặp câu được chia thành đối lập (contradiction), trung lập (neutral) hay có quan hệ đi kèm (textual entailment).

Ví dụ, chúng ta có các câu:

• A: Hôm nay trời mưa.

• B: Tôi mang ô tới trường.

• C: Hôm nay trời không mưa.

• D: Hôm nay là thứ 3.

Khi đó (A, B) có mối quan hệ đi kèm. Các cặp câu (A, C) có mối quan hệ đối lập và (A, D) là trung lập.

### Ngữ cảnh (Contextual)

Là ngữ cảnh của từ. Một từ được định nghĩa bởi một cách phát âm nhưng khi được đặt trong những câu khác nhau thì có thể mang ngữ nghĩa khác nhau. Ngữ cảnh có thể coi là môi trường xung quanh từ để góp phần định nghĩa từ.

Xét ví dụ:

- Câu A: Tôi đồng ý với ý kiến của anh.

- Câu B: Lão Hạc phải kiếm từng đồng để nuôi cậu Vàng.

thì từ “ đồng ” trong câu A và B có ý nghĩa khác nhau. Chúng ta biết điều này vì dựa vào ngữ cảnh của từ.

### Phương pháp Hiện đại nhất (SOTA)

Là viết tắt của state – of – the – art, để nói đến những phương pháp, kỹ thuật tốt nhất mang lại hiệu quả cao nhất từ trước đến nay. Mô hình biểu diễn mã hóa 2 chiều dựa trên biến đổi (BERT-Bidirectional Encoder Representation from Transformer) được xem là lớp mô hình SOTA trong nhiều nhiệm vụ của GLUE score benchmark.

### Mô hình Left – to – Right (LTR)

Là mô hình học bối cảnh theo một chiều duy nhất từ trái sang phải. Chẳng hạn như lớp các model RNN.

### 2.3.10. Masked Language Model - Mô hình ngôn ngữ được đánh dấu (MLM)

Là mô hình mà bối cảnh của từ được học từ cả 2 phía bên trái và bên phải cùng một lúc từ những bộ dữ liệu không có giám sát.

Dữ liệu vào sẽ được đánh dấu (tức thay bằng một mã đánh dấu token MASKED) một cách ngẫu nhiên với tỷ lệ thấp. Huấn luyện mô hình dự báo từ mã được đánh dấu dựa trên bối cảnh xung quanh là những từ không được đánh dấu nhằm tìm ra biểu diễn của từ.

## Ngữ cảnh (Contextual) và vai trò trong NLP

Bản chất của ngôn ngữ là âm thanh được phát ra để diễn giải dòng suy nghĩ của con người. Trong giao tiếp, các từ thường không đứng độc lập mà chúng sẽ đi kèm với các từ khác để liên kết mạch lạc thành một câu. Hiệu quả biểu thị nội dung và truyền đạt ý nghĩa sẽ lớn hơn so với từng từ đứng độc lập.

Ngữ cảnh trong câu có một sự ảnh hưởng rất lớn trong việc giải thích ý nghĩa của từ. Dựa trên đó, các thuật toán xử lý ngôn ngữ tự nhiên tốt nhất đều cố gắng đưa ngữ cảnh vào mô hình nhằm tạo ra sự đột phá và cải tiến. Trong đó mô hình BERT cũng sử dụng cách tiếp cận này.

Phân cấp mức độ phát triển của các phương pháp nhúng từ trong NLP có thể bao gồm các nhóm:

• **Không bối cảnh (Non-context):** Là các thuật toán không tồn tại bối cảnh trong biểu diễn từ. Đó là các thuật toán như “ Word2vec, Glove, Fasttext ”. Chúng ta chỉ có duy nhất một biểu diễn vector cho mỗi một từ mà không thay đổi theo bối cảnh.

Ví dụ :

- Câu A : Đơn vị tiền tệ của Việt Nam là “đồng”.

- Câu B : Vợ “đồng” ý với ý kiến của chồng là tăng thêm mỗi tháng 500k tiền tiêu vặt.

Thì từ “đồng” sẽ mang 2 ý nghĩa khác nhau nên phải có hai biểu diễn từ riêng biệt. Các thuật toán không có bối cảnh đã không đáp ứng được sự đa dạng về ngữ nghĩa của từ trong NLP.

• **Một chiều (Uni-directional):** Là các thuật toán đã bắt đầu xuất hiện bối cảnh của từ. Các phương pháp nhúng từ dựa trên RNN là những phương pháp nhúng từ một chiều. Các kết quả biểu diễn từ đã có bối cảnh nhưng chỉ được giải thích bởi một chiều từ trái qua phải hoặc từ phải qua trái.

Ví dụ:

- Câu C: Hôm nay tôi mang 200 tỷ “gửi” ở ngân hàng.

- Câu D: Hôm nay tôi mang 200 tỷ “gửi” ….

Như vậy vectơ biểu diễn của từ “gửi” được xác định thông qua các từ liền trước với nó. Nếu chỉ dựa vào các từ liền trước “Hôm nay tôi mang 200 tỷ” thì ta có thể nghĩ từ phù hợp ở vị trí hiện tại là “cho vay, mua, thanh toán,....”

Ví dụ đơn giản trên đã cho thấy các thuật toán biểu diễn từ có bối cảnh tuân theo theo một chiều sẽ gặp hạn chế lớn trong biểu diễn từ hơn so với biểu diễn 2 chiều.

ELMo là một ví dụ cho phương pháp một chiều. Mặc dù phương pháp ELMo có kiến trúc dựa trên một mạng BiLSTM xem xét bối cảnh theo hai chiều từ trái sang phải và từ phải sang trái nhưng những chiều này là độc lập nhau nên ta coi như đó là biểu diễn một chiều. Thuật toán ELMo đã cải tiến hơn so với WORD2VEC và FASTTEXT đó là tạo ra nghĩa của từ theo bối cảnh. Trong ví dụ về từ “đồng” thì ở mỗi câu A và B chúng ta sẽ có một biểu diễn từ khác biệt.

• **Hai chiều (Bi-directional):** Ngữ nghĩa của một từ không chỉ được biểu diễn bởi những từ liền trước mà còn được giải thích bởi toàn bộ các từ xung quanh. Luồng giải thích tuân theo đồng thời từ trái qua phải và từ phải qua trái cùng một lúc. Đại diện cho các phép biểu diễn từ này là những mô hình sử dụng kỹ thuật Transformer. Gần đây, những thuật toán NLP theo trường phái hai chiều như BERT, ULMT, OpenAI GPT đã đạt được những kết quả SOTA trên hầu hết các nhiệm vụ của GLUE benchmark.

## Phương pháp TRANSFORMER

Trước khi hiểu về BERT chúng ta cùng tìm hiểu về kỹ thuật transformer. Đây là một lớp mô hình SEQ2SEQ gồm 2 pha: mã hóa (Encoder) và giải mã (Decoder). Mô hình hoàn toàn không sử dụng các kiến trúc mạng hồi quy của RNN mà chỉ sử dụng các các tầng chú ý (attention layer) để nhúng các từ trong câu. Kiến trúc cụ thể của mô hình như sau:

Diagram

Description automatically generated

Hình 2.1. Sơ đồ kiến trúc của Transformer

### Encoder

Là tổng hợp xếp chồng lên nhau của n layers xác định (thường là 6) tương tự nhau. Mỗi layer bao gồm 2 layer con (sub-layer) trong nó. Sub-layer đầu tiên là multi-head self - attention (cơ chế tự chú ý nhiều đầu). Layer thứ 2 đơn thuần chỉ là các fully-connected feed forward network (mạng truyền thẳng kết nối đầy đủ). Kết thúc quá trình Encoder ta thu được một vector đầu ra nhúng cho mỗi từ.

Ngoài ra còn có skip connections (residual block – khối dư/thừa) và normalization layer (lớp chuẩn hóa). Trong đó multi-head attention là một layer mới tạo nên sự khác biệt giữa mô hình LSTM và mô hình Transformer.

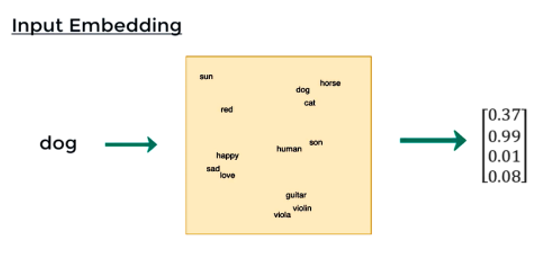
Diagram

Description automatically generated

Hình 2.2. Encoder

1. Input Embedding

Trong các mô hình học máy, học sâu (deep learning), dữ liệu đầu vào phải được mã hóa dưới dạng số thực hoặc phức, cũng như được biểu diễn bởi các cấu trúc toán học như vector, ma trận. Do vậy, cùng với sự phát triển của tiếp cận deep learning, các phương pháp học biểu diễn là một hướng nghiên cứu được quan tâm. Gần đây, một số mô hình học biểu diễn cho từ được đề xuất như GloVe, Fasttext, Word2Vec.



Hình 2.3. Biểu diễn nhúng từ

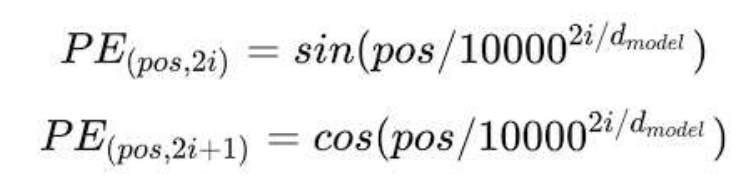
1. Positional Encoding

Word embeddings giúp biểu diễn ngữ nghĩa của một từ, tuy nhiên cùng một từ ở vị trí khác nhau của câu lại mang ý nghĩa khác nhau.

VD: AJ’s dog is cute => Position 2

AJ has a cute dog => Position 5

Do đó Transformers có thêm một phần Positional Encoding để đưa thêm thông tin về vị trí của một từ trong câu.

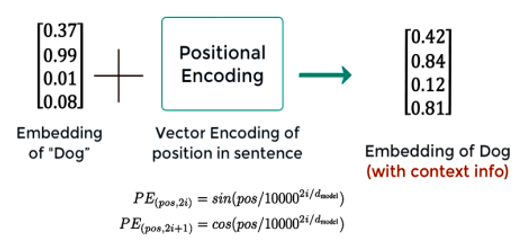


Với: pos là vị trí của từ trong câu

PE là giá trị phần tử thứ i trong embeddings

dmodel là độ dài của phần tử thứ i trong embeddings

Sau đó cộng PE vector và Embedding vector ta được kết quả là vector nhúng từ.



Hình 2.4. Ví dụ mã hóa vị trí từ nhúng

1. Self – Attention

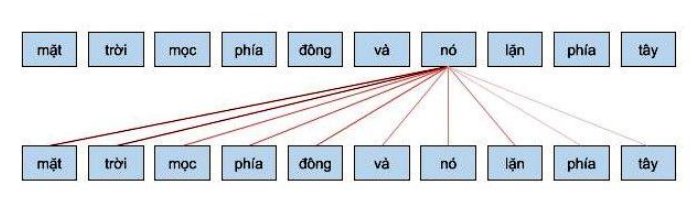
Self – Attention là cơ chế giúp Transformer “hiểu” được sự liên quan giữa các từ trong một câu.

Ví dụ:

|  |  |
| --- | --- |
|  | * Từ “it” trong câu dưới đây liên quan mật thiết đến từ “animal” * “it” là chính nó nên luôn liên quan mạnh * “street” là đối tượng khác có thể có liên quan nên là liên kết vừa vừa * “the” là mạo từ nên hầu như không có sự liên kết |

Self-Attention cho phép mô hình khi mã hóa một từ có thể sử dụng thông tin của những từ liên quan tới nó.

Ví dụ: khi từ “nó” được mã hóa, nó sẽ chú ý vào các từ liên quan như là “mặt trời”. Các từ còn lại có mức độ liên quan nhạt hơn hoặc gần như không liên quan.

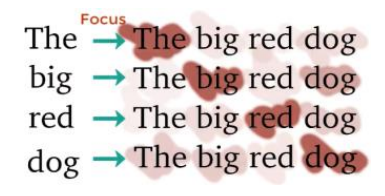
****

Hình 2.5. Ví dụ về self - attention

Cơ chế Self-Attention giống như cơ chế tìm kiếm. Với một từ cho trước, cơ chế này sẽ cho phép mô hình tìm kiếm trong các từ còn lại, từ nào “giống”, để sau đó thông tin sẽ được mã hóa dựa trên tất cả các từ trên.

1. Multi-head Attention

Vấn đề của self-attention là attention của một từ sẽ luôn “chú ý” vào chính nó vì “nó” phải liên quan đến “nó” nhiều nhất. Ví dụ như sau:



Hình 2.6. Multi-head Attention trong câu

Sự tương tác giữa các từ khác nhau trong câu được thực hiện bởi multi-head attention: thay vì sử dụng một self-attention (một head) bằng cách sử dụng nhiều attention khác nhau (multi-head), mỗi attention sẽ chú ý đến một khía cạnh ngữ nghĩa khác nhau trong câu.

Để mô hình có thể học nhiều kiểu mối quan hệ giữa các từ với nhau, mỗi self-attention được học một kiểu pattern. Do đó để có thể mở rộng khả năng này, đơn giản là thêm nhiều self-attention

Multi-head Attention cho phép mô hình chú ý đến đồng thời những pattern dễ quan sát được như sau:

− Chú ý đến từ kế trước của một từ;

− Chú ý đến từ kế sau của một từ;

− Chú ý đến những từ liên quan của một từ.

Ngoài ra còn có các pattern như cùng nói về kích thước, màu sắc, độ dài,…

1. Residuals

Trong mô hình tổng quát, mỗi sub-layer đều là một residual block (khối dư ra/ thừa ra). Việc bỏ qua các kết nối bị dư ra này trong Transformers cho phép thông tin đầu vào đi qua sub-layer trực tiếp. Thông tin này (x) được cộng với attention (z) của nó và thực hiện Layer Normalization (chuẩn hóa lớp).

Diagram

Description automatically generated

Hình 2.7. Residuals trong transformer

1. Feed Forward

Sau khi được chuẩn hóa, các vector z được đưa qua mạng Feed Forward (là các fully-connected feed forward networks) trước khi đẩy qua Decoder. Vì các vector này không phụ thuộc vào nhau nên có thể tận dụng được tính toán song song cho cả câu.

Trong kiến trúc của mô hình Transformer, residuals connection và normalization layer được sử dụng mọi nơi giúp cho mô hình huấn luyện nhanh hội tụ hơn và trách mất mát thông tin trong quá trình huấn luyện mô hình, ví dụ thông tin của vị trí các từ được mã hóa.

### Decoder

Decoder thực hiện chức năng giải mã vector của câu nguồn thành câu đích, do đó output của encoder sẽ là input của decoder. Kiến trúc của decoder giống với encoder, ngoại trừ có thêm một multi-head attention (2) nằm ở giữa dùng để học mối liên quan giữa từ đang được dịch với các từ được lấy ở câu nguồn và một masked multi-head attention (1).

Diagram

Description automatically generated

Hình 2.8. Tương quan giữa Encoder và Decoder

* Masked Multi-head Attention (1)

Công việc của Decoder là giải mã thông tin từ Encoder và sinh ra từng từ giải mã dựa trên những từ trước đó. Nếu sử dụng Multi-head Attention trên cả câu như ở Encoder, Decoder sẽ “thấy” luôn từ tiếp theo mà nó cần dịch. Để ngăn điều đó, khi Decoder dịch đến từ thứ i, phần sau của câu sẽ bị che lại (masked) và Decoder chỉ được phép “nhìn” thấy phần nó đã dịch trước đó.

* Multi-head Attention (2)

Trong Decoder còn có một Multi-head Attention khác có chức năng chú ý các từ ở mô hình Encoder do muốn so sánh sự tương quan giữa từ đang được dịch với các từ nguồn.

## Mô hình BERT

BERT là một model biểu diễn ngôn ngữ (Language Model - LM) được google giới thiệu vào năm 2018. BERT là viết tắt của cụm từ Bidirectional Encoder Representation from Transformer có nghĩa 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 (pre-train word embedding).

Điểm đặc biệt ở BERT đó là nó có thể điều hòa cân bằng bối cảnh theo cả 2 chiều trái và phải. Trước khi BERT ra đời thì các tác vụ như: phân loại cảm xúc văn bản (tốt hay xấu, tích cực hay tiêu cực), sinh văn bản, dịch máy,…. đều sử dụng kiến trúc RNN. Kiến trúc này có nhiều nhược điểm như train chậm, mất quan hệ giữa các từ xa nhau,... BERT kế thừa từ Transformer đã giải quyết được các nhược điểm này.

### 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.

Graphical user interface, diagram, application

Description automatically generated

Hình 2.9. Toàn bộ tiến trình pre-training và fine-tuning của BERT

### Cách huấn luyện BERT

BERT được huấn luyện đồng thời 2 nhiệm vụ gọi là Masked Language Model (MLM – để dự đoán từ thiếu trong câu) và Next Sentence Prediction (NSP – dự đoán câu tiếp theo câu hiện tại). Hai nhiệm vụ này được huấn luyện đồng thời và loss tổng sẽ là kết hợp loss của 2 nhiệm vụ và mô hình sẽ cố gắng minimize loss tổng này. Chi tiết 2 nhiệm vụ này như sau:

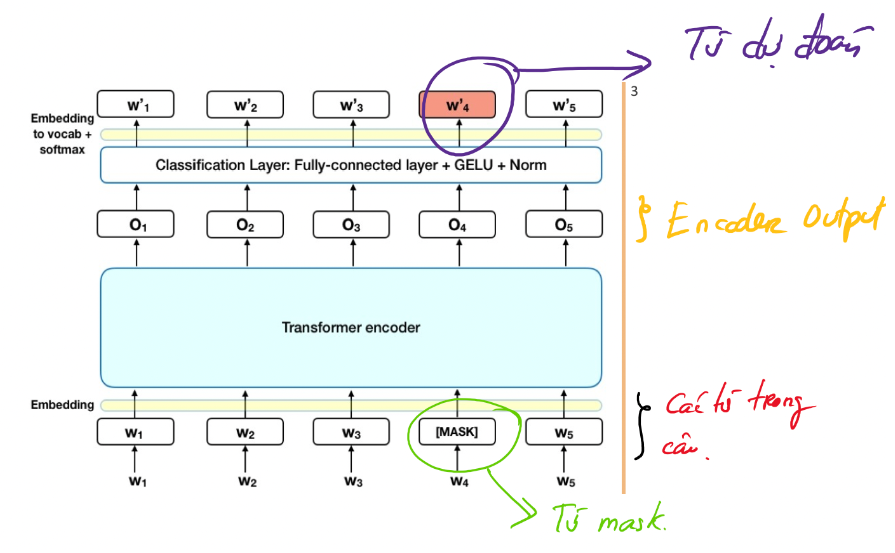
#### Mô hình ngô ngữ được đánh dấu (Masked Language Model)

Với nhiệm vụ này, ta huấn luyện sẽ thực hiện che đi tầm 15% số từ trong câu và đưa vào mô hình. Và ta sẽ huấn luyện để mô hình predict ra các từ bị che đó dựa vào các từ còn lại. Cụ thể là:

• Thêm một lớp classification lên trên Encoder đầu ra

• Đưa các vector trong Encoder ouput về vector bằng với vocab size, sau đó softmax để chọn ra từ tương ứng tại mỗi vị trí trong câu.

• Loss sẽ được tính tại vị trí masked và bỏ qua các vị trí khác (để đánh giá xem mô hình dự đoán từ mask đúng/sai).



Hình 2.10. Sơ đồ kiến trúc BERT cho nhiệm vụ ngôn ngữ mô hình được đánh dấu

Theo đó:

• Khoảng 15 % các mã của câu đầu váo được thay thế bởi [MASK] trước khi truyền vào mô hình đại diện cho những từ bị che dấu (masked). Mô hình sẽ dựa trên các từ không được che dấu (non-masked) xung quanh [MASK] và đồng thời là bối cảnh của [MASK] để dự báo giá trị gốc của từ được che dấu. Số lượng từ được che dấu được lựa chọn là một số ít (15%) để tỷ lệ bối cảnh chiếm nhiều hơn (85%).

• Bản chất của kiến trúc BERT vẫn là một mô hình seq2seq gồm 2 pha Encoder giúp nhúng các từ đầu vào 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 vector nhúng ở đầu ra là O1,O2,…,O5

• Để tính toán phân phối xác suất cho từ đầu ra, chúng ta thêm một Fully connected 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.

• Cuối cùng 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 như mô tả trên hình vẽ bên phải.

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. Việc lựa chọn ngẫu nhiên 15% số lượng các từ bị che dấu cũng tạo ra vô số các kịch bản đầu vào cho mô hình huấn luyện nên mô hình sẽ cần phải huấn luyện rất lâu mới học được toàn diện các khả năng.

#### Next Sentence Prediction (NSP)

Với nhiệm vụ này thì mô hình sẽ được feed cho một cặp câu và nhiệm vụ của nó là đầu ra ra giá trị 1 nếu câu thứ hai đúng là câu đi sau câu thứ nhất và 0 nếu không phải. Trong quá trinh huấn luyện, ta chọn 50% mẫu là Positive (đầu ra là 1) và 50% còn lại là Negative được ghép linh tinh (đầu ra là 0).

Cụ thể cách huấn luyện như sau:

• Bước 1: Ghép 2 câu vào nhau và thêm 1 số mã đặc biệt để phân tách các câu. Mã [CLS] thêm vào đầu cầu thứ nhất, mã [SEP] thêm vào cuối mỗi câu.

Ví dụ: ghép 2 câu “ Hôm nay em đi học ” và “ Học ở trường rất vui ” thì sẽ thành [CLS] Hôm nay em đi học [SEP] Học ở trường rất vui [SEP]

• Bước 2. Mỗi mã trong câu sẽ được cộng thêm một vector gọi là “Nhúng câu” (Sentence Embedding), thực ra là đánh dấu xem từ đó thuộc câu thứ nhất hay câu thứ hai.

Ví dụ: nếu thuộc câu thứ nhất thì cộng thêm 1 vector toàn số “0” có kích thước bằng “Từ nhúng”, và nếu thuộc câu thứ 2 thì cộng thêm một vector toàn số “1”.

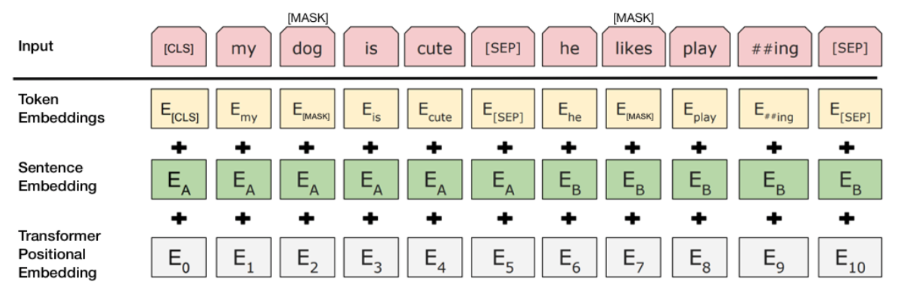
• Bước 3. Sau đó các từ trong câu đã ghép sẽ được thêm vector mã hóa vị trí (Positional Encoding) vào để đánh dấu vị trí từng từ trong câu đã ghép.

• Bước 4. Đưa chuỗi sau bước 3 vào mạng.

• Bước 5. Lấy encoder đầu ra tại vị trí mã [CLS] được biến đổi (transform) sang một vector có 2 phần tử [c1 c2].

• Bước 6. Tính softmax trên vector đó và đầu ra khả năng của 2 lớp: Đi sau và Không đi sau. Để thể hiện câu thứ hai là đi sau câu thứ nhất hay không, ta lấy argmax là được.

Các bước tạo đầu vào:



Hình 2.11. Các bước tạo đầu vào trong tác vụ NSP

Và đây là cách lấy đầu ra:

Diagram

Description automatically generated

Hình 2.12. Mô hình lấy đầu ra trong NSP

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

• Ngữ nghĩa của từ (token embeddings): Thông qua các nhúng vector cho từng từ. Các vector được khởi tạo từ mô hình huấn luyện trước.

Ngoài nhúng biểu diễn từ của các từ trong câu, mô hình còn nhúng thêm một số thông tin:

• Loại câu (segment embeddings): Gồm hai véc tơ là EA nếu từ thuộc câu thứ nhất và EB nếu từ thuộc câu thứ hai.

• Vị trí của từ trong câu (position embedding): là các vector E0,…,E10. Tương tự như nhúng vị trí (positional embedding) trong Transformer.

Vector đầu vào sẽ bằng tổng của cả ba thành phần nhúng theo từ, câu và vị trí.

## 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 đầu (head) trong tầng nhiều đầu (multi-head layer), mỗi một đầu sẽ thực hiện một cơ 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 BERTLarge chúng ta tăng gấp đôi số tầng, tăng kích thước của vector nhúng gấp 1.33 lần và tăng số lượng đầu trong multi-head layer gấp 1.33 lần.

## Tổng kết chương 2

Trong chương này, em đã trình bày các nghiên cứu của em về tổng quan mô hình BERT dựa trên kiến trúc Transformer và các khái niệm liên quan đến xử lý ngôn ngữ tự nhiên.

# CHƯƠNG 3: PHOBERT

## Sự ra đời của PhoBERT

BERT là một nghiên cứu mới mang đầy tính đột phá, một bước nhảy vọt thực sự của Google trong lĩnh vực xử lý ngôn ngữ tự nhiên. Sự ra đời của mô hình huấn luyện trước BERT đã mang lại những cải tiến đáng kể cho rất nhiều bài toán như Question Answering, Sentiment Analysis,...

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 (pre-trained multilingual) bao gồm cả tiếng Việt nhưng chưa cho kết quả thực hiện tốt nhất.

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,... BERT đã được áp sụng cho nhiều nhiệm vụ xử lý ngôn ngữ tự nhiên và trở lên áp đảo trong các nền tảng thi đấu như Kaggle, AIVIVN cũng như được trình bày tại nhiều hội nghị.

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.

## Cấu trúc của PhoBERT

PhoBERT là một mô hình dựa trên kiến trúc RoBERTa do 2 tác giả D. Q. Nguyen và A. T. Nguyen thuộc viện nghiên cứu VinAI Việt Nam huấn luyện, được giới thiệu vào tháng 03/2020. Đây là một mô hình huấn luyện trước được huấn luyện cho đơn ngôn ngữ (monolingual language), tức là chỉ huấn luyện dành riêng cho tiếng Việt. Tương tự như BERT, PhoBERT cũng có 2 phiên bản là PhoBERTBase với 12 khối transformers và PhoBERTLarge với 24 khối transformers.

PhoBERT được huấn luyện trên khoảng 20GB dữ liệu bao gồm khoảng 1GB ngữ liệu Wikipedia Tiếng Việt (Vietnamese Wikipedia corpus) 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 (MLM) để huấn luyện và không sử dụng nhiệm vụ dự đoán câu tiếp theo (NSP).

Thực nghiệm kết quả cho thấy PhoBERT luôn hoạt động tốt hơn so với mô hình đa ngôn ngữ được huấn luyện trước tốt nhất gần đây XLM-R và cải thiện tính năng hiện đại trong nhiều nhiệm vụ NLP dành riêng cho tiếng Việt bao gồm các nhiệm vụ như: phân tích giọng nói, phân tích cú pháp phụ thuộc, nhận dạng thực thể được đặt tên và suy luận ngôn ngữ tự nhiên.

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.

Đối với ngôn ngữ khác, người ta có thể huấn luyện lại bằng cách sử dụng mô hình kiến trúc BERT (Cui ,de Vries, Vu, Martin và các cộng sự ra mắt vào năm 2020) hoặc sử dụng các mô hình dựa trên BERT đa ngôn ngữ đã được huấn luyện trước (Devlin et al., 2019; Conneau và Lample, 2019; Conneau và cộng sự, 2020).

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 (Vu et al., 2019) 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 được sử dụng để tách các âm tiết tạo thành từ khi viết bằng tiếng Việt.

## Ứng dụng của PhoBERT

Trong Tiếng Việt thì chúng ta có thể ứng dụng PhoBERT trong một số nhiệm vụ như:

• Tìm từ đồng nghĩa, trái nghĩa, cùng nhóm dựa trên khoảng cách của từ trong không gian biểu diễn đa chiều.

• Xây dựng các vector nhúng cho các nhiệm vụ NLP như nhận diện cảm xúc, phân loại văn bản, nhận dạng thực thể, gắn nhãn từ loại, huấn luyện chatbot.

• Gợi ý từ khóa tìm kiếm trong các hệ thống search.

• Xây dựng các ứng dụng seq2seq như robot viết báo, tóm tắt văn bản, sinh câu ngẫu nhiên với ý nghĩa tương đồng.

## Kết luận

Có thế 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.

## Tổng kết chương 3

Trong chương này, em đã thực hiện tìm hiểu về mô hình PhoBERT: sự ra đời, kiến trúc và ứng dụng của PhoBERT trong các nhiệm vụ xử lý ngôn ngữ tự nhiên.

# CHƯƠNG 4: ỨNG DỤNG PHOBERT VÀO BÀI TOÁN NHẬN DIỆN CẢM XÚC VĂN BẢN

## 4.1. Phát biểu bài toán

Nhận diện cảm xúc văn bản (Sentiment Analysis) hay khai phá quan điểm (Opinion Mining) 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. 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,... Người dùng hay nhà sản xuất thường muốn biết sản phẩm hay dịch vụ mà họ quan tâm được đánh giá là tích cực hay tiêu cực để quyết định lựa chọn hay sản xuất nó. 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.

Nhận diện cảm xúc văn bản (Sentiment Analysis) 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, ta 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.

Phát biểu theo góc nhìn của máy học (Machine Learning) thì phân tích cảm xúc là bài toán phân lớp cảm xúc dựa trên văn bản 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 (điểm số) 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:

* Phân tích cảm xúc (thái độ) trong văn bản thành 2 lớp: tích cực (positive) và tiêu cực (negative).
* Xếp hạng cảm xúc (thái độ) trong văn bản từ 1 đến 5.
* 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%.

Vì vậy, bài toán phân tích cảm xúc trong Tiếng Việt trình bày trong đề tài này là kết quả của nghiên cứu nhận diện cảm xúc văn bản Tiếng Việt với 3 lớp cảm xúc là: tiêu cực (negative), tích cực (positive) và trung tính (neutral). Sơ đồ phân tích cảm xúc như sau:

Diagram

Description automatically generated

Hình 4.1. Sơ đồ phân tích cảm xúc văn bản

Đầu vào của mô hình xử lý Sentiment Analysis Vietnamese (SAV) là một đoạn văn Tiếng Việt, đầu ra là 3 giá trị xác suất mà đoạn văn đầu vào thuộc về lớp cảm xúc: tiêu cực (negative), trung tính (neutral) hay tích cực (positive).

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...

## 4.2. Dữ liệu, công cụ và môi trường thực nghiệm

### 4.2.1. Dữ liệu

Xét mô hình nhận diện cảm xúc văn bản đối với HaUI:

* **Mô hình tổng quát**

**Diagram

Description automatically generated**

Hình 4.2. Mô hình nhận diện cảm xúc văn bản với HaUI

* **Xây dựng bộ dữ liệu HaUI**

1. Thực hiện thu thập dữ liệu từ nhiều nguồn khác nhau được liệt kê dưới đây:

* Đánh giá của sinh viên, phụ huynh,… trên app MyHaUI.
* Fanpage HaUI, fanpage các khoa, HaUI Confessions.
* Ý kiến đánh giá, nhận xét trên hệ thống xem điểm trực tuyến của sinh viên.
* Các ý kiến nhận xét của doanh nghiệp trong ngày hội việc làm tổ chức hàng năm hoặc doanh nghiệp nơi sinh viên đang công tác.
* Các ý kiến đánh giá được thu thập từ các trang web review các trường Đại học

1. Tiền xử lý dữ liệu

Thực hiện các bước tiền xử lý thật kỹ để tăng hiệu suất cho mô hình được chính xác nhất. Dữ liệu thu thập từ các fangpage liên quan đến HaUI rất nhiều từ ngữ lóng, các kí hiệu, các từ viết tắt,… Vì vậy, ta cần tiến hành việc xử lý được mô tả dưới đây:

* Loại bỏ các HTML tag.
* Loại bỏ những ký tự lặp lại nhiều lần khi chúng không phải là alphanumeric. Ví dụ: “…” -> ”???”->”?”. Việc này cũng loại bỏ những khoảng trắng thừa và giúp cho bộ phận tách từ hoạt động hiệu quả hơn.
* Phân loại những các ký hiệu biểu hiện emoji
* Xóa những ký tự đánh dấu đầu dòng như: -, IV,…
* Chuyển tất cả chữ hoa về chữ thường,…

Đối với ngôn ngữ tiếng Việt thì việc tách từ trong tiền xử lý là không thể thiếu bởi vì từ vựng trong tiếng Việt có thể được tạo thành từ một từ ngữ hoặc nhiều từ ngữ khác nhau ví dụ như từ vựng “sinh viên” được tạo thành từ 2 từ “sinh” và “viên”, nhưng khi kết hợp 2 từ này lại thì nó lại mang ý nghĩa khác. Để giải quyết vấn đề này, chúng ta sử dụng thư viện vncorenlp hoặc underthesea để tách từ trong dữ liệu đầu vào.

1. Gán nhãn cảm xúc

Việc gán nhãn cảm xúc dựa trên quy tắc:

* **Tích cực**: Những câu văn thể hiện sự hài lòng khi khảo sát ở doanh nghiệp có sinh viên làm việc, sự hài lòng của người học, những lời ngợi khen, sự động viên tích cực nhất dành cho người học, các tổ chức đào tạo giáo dục về các nội dung như: giảng viên, nội dung môn học, cơ sở vật chất, chương trình đào tạo và một số khác. Ví dụ, câu văn “Thầy cô rất nhiệt tình với sinh viên” sẽ được gán nhãn cảm xúc tích cực.
* **Trung tính**: Những câu văn không hàm ý chứa bất kỳ cảm xúc nào, những câu văn không hoàn chỉnh, không rõ ràng về nghĩa, có ý nghĩa chung chung. Ví dụ, trong HaUI Confessions được ghi “Thực tế thì tùy vào ngành mà e chọn để định hướng nhé” được gán nhãn cảm xúc trung tính, câu văn này mang tính trao đổi nó như một cụm danh từ bình thường.
* **Tiêu cực**: Những câu văn thể hiện sự không hài lòng của người sử dụng lao động, của người học, các tổ chức đào tạo. Những lời phàn nàn, không đồng ý, những lời đề nghị cần thiết nhất đối với nhà trường như giảng viên, môn học, chương trình đào tạo, cơ sở vật chất và một số khác. Ví dụ, câu văn “Hệ thống wifi trường rất chậm” sẽ được gán nhãn cảm xúc tiêu cực.

Trường hợp khiến ta khó khăn trong câu thể hiện vừa mang quan điểm tiêu cực và tích cực, điều này làm khó khăn khi gán nhãn. Chúng ta thường gặp một số câu có sử dụng từ liên kết như: nhưng, tuy nhiên, mặc dù, dù, dù rằng, tuy rằng và một số khác. Trong trường hợp này, ta cần lựa chọn mệnh đề có tính phân cực mạnh hơn để gán nhãn. Ví dụ, câu văn “cần nâng cấp thư viện, phòng máy, phòng học, còn về môi trường xung quanh rất ổn” được gán nhãn tiêu cực, trong khi đó ta thấy “môi trường xung quanh rất ổn” mang cảm xúc tích cực.

### 4.2.2. Công cụ và môi trường thực nghiệm

**❖ Công cụ**

**Ngôn ngữ lập trình Python**

Python được Guido van Rossum phát triển vào cuối những năm tám mươi và đầu những năm chín mươi tại Viện nghiên cứu quốc gia về toán học và khoa học máy tính ở Hà Lan.

Python là ngôn ngữ lập trình hướng đối tượng, cấp cao, mạnh mẽ, được tạo ra bởi Guido van Rossum. Nó dễ dàng để tìm hiểu và đang nổi lên như một trong những ngôn ngữ lập trình nhập môn tốt nhất cho người lần đầu tiếp xúc với ngôn ngữ lập trình. Python hoàn toàn tạo kiểu động và sử dụng cơ chế cấp phát bộ nhớ tự động. Python có cấu trúc dữ liệu cấp cao mạnh mẽ và cách tiếp cận đơn giản nhưng hiệu quả đối với lập trình hướng đối tượng. Cú pháp lệnh của Python là điểm cộng vô cùng lớn vì sự rõ ràng, dễ hiểu và cách gõ linh động làm cho nó nhanh chóng trở thành một ngôn ngữ lý tưởng để viết script và phát triển ứng dụng trong nhiều lĩnh vực, ở hầu hết các nền tảng.

Tính năng chính của Python :

• Ngôn ngữ lập trình đơn giản, dễ học

• Miễn phí, mã nguồn mở

• Khả năng di chuyển

• Khả năng mở rộng và có thể nhúng

• Ngôn ngữ thông dịch cấp cao

• Thư viện tiêu chuẩn lớn để giải quyết những nhiệm vụ phổ biến

• Hướng đối tượng

**Thư viện mã nguồn mở Tensorflow**

Tensorflow là một thư viện mã nguồn mở cung cấp khả năng xử lí tính toán số học dựa trên biểu đồ mô tả sự thay đổi của dữ liệu, trong đó các node là các phép tính toán học còn các cạnh biểu thị luồng dữ liệu.

**Thư viện 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 xử lý ngôn ngữ tự nhiên trên cả nền tảng pytorch và tensorflow.

**Thư viện fastBPE:** Là package hỗ trợ tokenize từ (word) thành các từ phụ (subwords) theo phương pháp mới nhất được áp dụng cho các pretrain model xử lý ngôn ngữ tự nhiên hiện đại như BERT và các biến thể của BERT.

**Thư viện fairseq:** Là project của facebook chuyên hỗ trợ các nghiên cứu và dự án liên quan đến model seq2seq.

**Thư viện VnCoreNLP:** Là một package xử lý ngôn ngữ tự nhiên trong Tiếng Việt, hỗ trợ tokenize và các tác vụ xử lý ngôn ngữ khác.

**Underthesea:** Là một bộ dữ liệu và hướng dẫn mô-đun mã nguồn mở Python hỗ trợ nghiên cứu và phát triển trong xử lý ngôn ngữ tự nhiên tiếng Việt.

**PhoBERT đã được huấn luyện trước.**

**❖ Môi trường thực nghiệm:**

- Máy tính Chip: Intel(R) Core (TM) i5, Ram: 8.00 GB.

- Hệ điều hành Windows 10

- Công cụ lập trình: Python 3.11.2.

## 4.3. **Các bước thực hiện**

Trong đề tài này, em sử dụng PhoBERTbase để thực nghiệm trên bộ dữ liệu HaUI. Mục tiêu nhằm tạo ra một mô hình lấy một câu (giống như các câu trong tập dữ liệu), đầu ra cuối cùng của mô hình mang tính cảm xúc là tích cực, trung tính hay tiêu cực. Hình dưới đây là một ví dụ một câu văn được đưa vào tiền xử lý, tách từ, chuyển vào mô hình PhoBERT, hàm softmax dùng để phân loại kết quả mang tính cảm xúc.

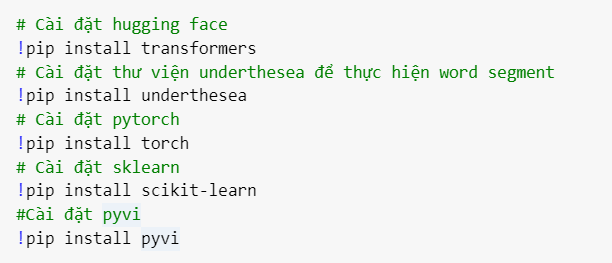
Text

Description automatically generated

Biểu diễn một câu văn vào mô hình phân tích quan điểm: Giả sử thực hiện dự đoán một câu “Nhà trường rất lắng nghe sinh viên” được mô tả ở hình trên:

* Bước 1: Sử dụng thư viện vncorenlp để thực hiện tách từ như sau: “nhà\_trường rất lắng\_nghe sinh\_viên”.
* Bước 2: Thêm mã thông báo [CLS] để đánh dấu đầu câu và [SEP] để đánh dấu cuối câu.
* Bước 3: Sử dụng thuật toán BPE (Byte Pair Encoding) để đưa câu đầu vào dưới dạng subword và ánh xạ các subword về dạng index trong từ điển.
* Bước 4: Chuyển vào mô hình PhoBERT Fine-tuning. Đầu ra là một vector đặc trưng, hàm Softmax để tính xác suất đầu ra, hàm Argmax để chọn giá trị lớn nhất chọn ra giá trị cuối cùng.

### Cài đặt các thư viện cần thiết



Hình 4.3. Cài đặt thư viện

### Import các thư viện cần thiết

* Thư viện tách từ
* Thư viện chia tách dữ liệu
* Thư viện BERT
* Thư viện train SVM

Graphical user interface, text, application, email

Description automatically generated

Hình 4.4. Import thư viện

### Load model PhoBERT

Sử dụng model PhoBERTbase

Text

Description automatically generated

Hình 4.5. Load model PhoBERT

### Chuẩn hóa dữ liệu

Dữ liệu thu thập từ trên mạng thường rất sạn. Sạn ở đây cụ thể là: từ viết tắt, dấu câu, sai chính tả, từ không dấu, từ lóng,…và chúng ta phải xử lý để chuẩn hoá dữ liệu thì model mới cho ra kết quả tốt được. Bao gồm:

* Chuẩn hóa câu: xóa dấu chấm, phẩy, hỏi,…
* Xóa các ký tự kéo dài: VD: đẹpppp => đẹp
* Chuyển thành chữ thường
* Chuẩn hóa tiếng Việt, xử lý emj/icons, chuẩn hóa tiếng Anh, thuật ngữ, từ lóng,…
* Load file dữ liệu là vietnamese\_stopwords.txt bao gồm danh sách các từ vô nghĩa trong tiếng Việt như à, ờ, á, ấy,... Lọc các từ vô nghĩa này giúp train model hiệu quả cao hơn và cải thiện tốc độ train model.

Text

Description automatically generated

Hình 4.6. Hàm chuẩn hóa câu

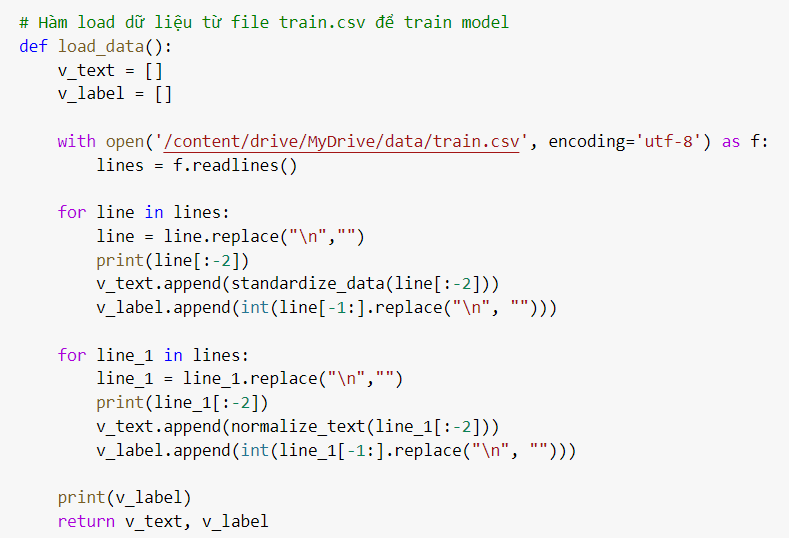
Graphical user interface, text, application

Description automatically generated

Hình 4.7. Load danh sách các từ vô nghĩa

### Load dữ liệu từ file train.csv để train model

* File train.csv là file dữ liệu đã được huấn luyện trước cho model. Đây là những dữ liệu đánh giá, nhận xét về HaUI đã được gán nhãn sẵn:
  + 0: tiêu cực
  + 1: tích cực
  + 2: trung tính
* Việc phân loại cho dữ liệu huấn luyện trước là cơ sở cho train model. Model dựa vào dữ liệu này để tự huấn luyện và tự đưa ra kết quả đánh giá về các câu văn bản người dùng đưa vào model kiểm tra.



Hình 4.8. Load dữ liệu trừ file train.csv

### Tạo BERT features

Padding và đưa vào model PhoBERT trích đặc trưng. Chúng ta phải padding để đảm bảo các input có cùng độ dài như nhau:

Text, Word

Description automatically generated

Hình 4.9. Padding input

Tuy nhiên, khi padding thế thì ta phải thêm một attention\_mask để model chỉ focus vào các từ trong câu và bỏ qua các từ được padding thêm:

A picture containing text, indoor, orange

Description automatically generated

Hình 4.10. Thêm attention\_mask

Cuối cùng là đưa nó vào model và lấy ra output:

Text

Description automatically generated

Hình 4.11. Đưa vào model và lấy ra output

### Train model SVM

Model SVM giúp phân loại các bình luận theo nhãn – dựa vào bộ dữ liệu đầu vào được lấy từ file train.csv

Text

Description automatically generated

Hình 4.12. Train model SVM

### Test thử kết quả chạy model

Quy định: Nhãn 0: tiêu cực

Nhãn 1: tích cực

Nhãn 2: trung tính

Bảng 4.1. Kết quả chạy model PhoBERT đã huấn luyện trong nhận diện cảm xúc văn bản

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Câu văn bản** | **Kết quả dự đoán** | **Kết quả mong đợi** |
| 1 | “Khuôn viên trường rộng và đẹp” | 1 | 1 |
| 2 | “gv tâm huyết, giảng dạy rất nhiệt tình” | 1 | 1 |
| 3 | “App rất xịn ạ, dùng rất oke” | 1 | 1 |
| 4 | “Ứng dụng hay, tiện cho sv” | 1 | 1 |
| 5 | “Good app” | 1 | 1 |
| 6 | “Nice” | 1 | 1 |
| 7 | “Very greatttttt” | 1 | 1 |
| 8 | “Haui vip pro” | 1 | 1 |
| 9 | “Tất cả CSVC hay chất lượng giảng dạy của trường đều tốt” | 1 | 1 |
| 10 | “Mặc dù em hơi bức xúc vì cách trường ra thông báo nhưng em vẫn đánh giá trg 5 sao ạ” | 1 | 1 |
| 11 | “Sv năng động, sáng tạo” | 1 | 1 |
| 12 | “Tuyệt vời, em yêu trường” | 1 | 1 |
| 13 | “là 1 Haui-er, em rất yêu trg em <3” | 1 | 1 |
| 14 | “Chất lượng tốt ! 👍🏻” | 1 | 1 |
| 15 | “Phần mềm quá ok 5 sao😍😍😍” | 1 | 1 |
| 16 | “Trường có csvc hiện đại, gv tốt” | 1 | 1 |
| 17 | “Giao diện đẹp. Mới update có nhiều tính năng mới” | 1 | 1 |
| 18 | “App hay, cập nhật phiên bản mới đi admin :3” | 1 | 1 |
| 19 | “😍😍😍😍” | 1 | 1 |
| 20 | “Trường đỉnh lắm ạ <3” | 1 | 1 |
| 21 | “Quá tuyệt vời ông mặt trời” | 1 | 1 |
| 22 | “Tạm thời vốt lại 5 sao z” | 1 | 1 |
| 23 | “Trường có 3 cơ sở khang trang, hiện đại. Môi trg học tập rất lý tg” | 1 | 1 |
| 24 | “Trong trường, khu vực xung quanh an ninh,an toàn,lành mạnh,” | 1 | 1 |
| 25 | “Mặc dù còn một vài lỗi nhưng nhìn chung app rất tiện lợi” | 1 | 1 |
| 26 | “Học phí quá cao” | 0 | 0 |
| 27 | “App tệ, nhiều lỗi” | 0 | 0 |
| 28 | “Kỳ vừa rồi gặp thầy cô hãm -> vote 1\*” | 0 | 0 |
| 29 | “Học phí ảo mà canada” | 0 | 0 |
| 30 | “Nhà trường tăng học phí quá thể đáng” | 0 | 0 |
| 31 | “Trường như cái quần què. Không thể yêu nổi trường” | 0 | 0 |
| 32 | “Đánh 1\* vì không có 0\*” | 0 | 0 |
| 33 | “chịu cái trg này🙃” | 0 | 0 |
| 34 | “Tiếng anh 4 điểm còn trượt bây h nâng lên 5 thì sống sao” | 0 | 0 |
| 35 | “trường phèn như jack )” | 0 | 0 |
| 36 | “Em xin trường .trường đừng ra quyết định như v” | 0 | 0 |
| 37 | “Trường lừa đảo moi tiền sv” | 0 | 0 |
| 38 | “Trường chơi ác quá” | 0 | 0 |
| 39 | “Trường đểu đừng học” | 0 | 0 |
| 40 | “Lý thuyết xuông, thực hành chỉ ở mức cơ bản” | 0 | 0 |
| 41 | “Thi xong được 1 tháng rồi mới đưa ra thông báo điểm qua môn từ 4 lên 5, quá chán kiểu làm ăn của trường” | 0 | 0 |
| 42 | “vô cùng bức xúcccccc” | 0 | 0 |
| 43 | “lag” | 0 | 0 |
| 44 | “Hãy miêu tả ngôi trường của em bằng 2 từ :Chối tỷ” | 0 | 0 |
| 45 | “Chỉ cho nộp tiền vào chứ k cho rút tiền ra :)))” | 0 | 0 |
| 46 | “App tệ vcl. Khảo sát cái qq” | 0 | 0 |
| 47 | “App lỗi, khảo sát xong cứ đơ, suốt ngày bắt cập nhật.” | 0 | 0 |
| 48 | “cần đến thì sập đell đc cái tích sự gì” | 0 | 0 |
| 49 | “App được, mà trường không được 🙅.” | 0 | 0 |
| 50 | “hfgue ewfegwe ffqqfwq” | 2 | 2 |
| 51 | “dựa vào khả năng của em mà chọn ngành” | 2 | 2 |
| 52 | “é tewwe reeeh” | 2 | 2 |
| 53 | “bạn Long k16 đọc đc cmt này vui lòng trả mình máy tính” | 0 | 2 |
| 54 | “Cho em hỏi là mật khẩu app này lấy ở đâu ạ” | 2 | 2 |
| 55 | “ko liên quan nhưng mà muốn xin inf bạn Hoa CNTT6 K13 ạ” | 2 | 2 |
| 56 | “Hello mọi người” | 2 | 2 |
| 57 | “Cho em hỏi làm thế nào để thay đổi yêu cầu phần dịch vụ 1 cửa” | 2 | 2 |
| 58 | “sfgegerhhiuhiohrtrjr” | 2 | 2 |
| 59 | “k biết nói gì” | 0 | 2 |
| 60 | “tôi ko có phàn nàn gì” | 2 | 2 |
| 61 | “app này bình thường” | 2 | 2 |
| 62 | “Trường ra công văn rất ...” | 2 | 2 |
| 63 | “App như ...” | 2 | 2 |
| 64 | “Em cần giấy xác nhận sv thì phải xin ở đâu ạ” | 2 | 2 |
| 65 | “{regr;.,l” | 2 | 2 |
| 66 | “một ma hình bản mượt mấy” | 2 | 2 |
| 67 | “trường l” | 2 | 2 |
| 68 | “tôi ko có ý kiến gì” | 2 | 2 |
| 69 | “không” | 2 | 2 |
| 70 | “ko” | 2 | 2 |
| 71 | “k” | 2 | 2 |
| 72 | “Thực tế thì tùy vào ngành mà e chọn để định hướng nhé” | 2 | 2 |
| 73 | “Xin chào mọi người ạ” | 2 | 2 |
| 74 | “Đợt mới tháng sau sẽ quay lại đánh giá” | 2 | 2 |
| 75 | “Bao giờ update sẽ đánh giá lại” | 2 | 2 |

**\* Nhận xét:** Qua bảng kết quả demo trên đây có thể thấy:

- Model PhoBERTbase được huấn luyện đã đạt được mục tiêu đặt ra của bài toán là nhận diện cảm xúc văn bản theo 3 nhãn tích cực, tiêu cực và trung tính.

- Model đạt được độ chính xác là trên 97%

- Model nhận diện được nhãn tích cực và tiêu cực là 100%; riêng nhãn trung tính, do tính chất câu còn mang sắc thái chung chung, đôi khi có chứa một vài từ mang hàm ý tích cực và tiêu cực gây nhiễu nên độ chính xác để nhận diện nhãn trung tính của model thấp hơn 2 nhãn trên.

## 4.4. Tổng kết chương 4

Trong chương này, em thực hiện huấn luyện model PhoBERT để nhận diện cảm xúc văn bản. Em đã huấn luyện được mô hình nhận diện cảm xúc văn bản theo 3 nhãn tích cực, tiêu cực và trung tính. Mô hình sau khi huấn luyện đã nhận diện được cảm xúc của câu đầu vào và đạt được độ chính xác trên 97%.

# KẾT LUẬN

* **Kết quả đạt được**

Báo cáo đã trình bày được cơ bản nghiên cứu về mô hình BERT và ứng dụng các mô hình được đào tạo dựa trên BERT trong các tác vụ xử lý ngôn ngữ tự nhiên NLP; tìm hiểu, cài đặt thực nghiệm và đánh giá các mô hình và thư viện được sử dụng trong xử lý ngôn ngữ tự nhiên. Từ đó huấn luyện được model giúp nhận diện cảm xúc văn bản tiếng Việt ứng dụng pre-train model PhoBERT. Nội dung chi tiết các phần mà báo cáo đã thực hiện được:

* Tìm hiểu về mô hình BERT với cơ sở là mô hình Transformer.
* Tìm hiểu về PhoBERT là mô hình được huấn luyện dành riêng cho tiếng Việt trong xử lý ngôn ngữ tự nhiên.
* Ứng dụng PhoBERT vào bài toàn nhận diện cảm xúc văn bản tiếng Việt, đưa ra kết quả là các nhãn tích cực, tiêu cực hay trung tính.
* **Hướng nghiên cứu trong tương lai**
* Cải thiện độ chính xác và tốc độ train model: Nghiên cứu để tăng tính nhất quán và đáng tin cậy trong việc nhận diện cảm xúc văn bản nhanh hơn và chính xác hơn.
* Train model nâng cao hơn: Xếp hạng cảm xúc (thái độ) trong văn bản từ 1 đến 5.
* Phát hiện mục tiêu, nguồn gốc của cảm xúc (thái độ) hoặc các loại cảm xúc (thái độ) phức tạp.

# TÀI LIỆU THAM KHẢO

1. Byte-Pair encoding (BPE) methods (Sennrich et al., 2016; Kudo and Richardson, 2018)
2. Deep Learning: Recurrent Neural Networks in Python: LSTM, GRU, and more RNN machine learning architectures in Python and Theano, 2016
3. Python Machine Learning By Example by Yuxi Liu, 2017
4. https://arxiv.org/abs/1810.04805
5. https://arxiv.org/pdf/1706.03762.pdf
6. https://github.com/VinAIResearch/PhoBERT#introduction
7. https://itzone.com.vn/vi/article/bert-mo-hinh-ngon-ngu-hien-dai-cho-xu-ly-ngon-ngu-tu-nhien-nlp/
8. https://miai.vn/2020/12/29/bert-series-chuong-3-thu-nhan-dien-cam-xuc-van-ban-tieng-viet-voi-phobert-cach-1/
9. https://phamdinhkhanh.github.io/2020/05/23/BERTModel.html
10. <https://phamdinhkhanh.github.io/2019/06/18/AttentionLayer.html>
11. <https://viblo.asia/>