

ỦY BAN NHÂN DÂN TỈNH BÌNH DƯƠNG
TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT

TRẦN THỊ NGỌC DUNG

**PHÁT TRIỂN MỘT ỨNG DỤNG QUẢN LÝ BẤT
ĐỘNG SẢN THÔNG MINH Ở BÌNH DƯƠNG**

CHUYÊN NGÀNH: HỆ THỐNG THÔNG TIN

MÃ SỐ: 8480104

LUẬN VĂN THẠC SĨ

BÌNH DƯƠNG – 2022

ỦY BAN NHÂN DÂN TỈNH BÌNH DƯƠNG

TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT

TRẦN THỊ NGỌC DUNG

**PHÁT TRIỂN MỘT ỨNG DỤNG QUẢN LÝ BẤT
ĐỘNG SẢN THÔNG MINH Ở BÌNH DƯƠNG**

CHUYÊN NGÀNH: HỆ THỐNG THÔNG TIN

MÃ SỐ: 8480104

LUẬN VĂN THẠC SĨ

BÌNH DƯƠNG – 2022

Lời cam đoan

Tên tôi là: Trần Thị Ngọc Dung Sinh ngày: 30/04/1994

Học viên lớp cao học CH18HT01 - Trường Đại học Thủ Dầu Một

Xin cam đoan: Đề tài “**Phát triển một ứng dụng quản lý bất động sản thông minh ở Bình Dương**” do Thầy PGS. TS. Quản Thành Thơ hướng dẫn là công trình nghiên cứu của riêng tôi. Tất cả tài liệu tham khảo đều có nguồn gốc, trích dẫn rõ ràng.

Tác giả xin cam đoan tất cả những nội dung trong luận văn đúng như nội dung trong đề cương và yêu cầu của thầy giáo hướng dẫn. Nếu sai tôi hoàn toàn chịu trách nhiệm trước hội đồng khoa học.

Bình Dương, 22 tháng 07 năm 2022

Tác giả luận văn

Trần Thị Ngọc Dung

Lời cảm ơn

Sau một thời gian nghiên cứu và làm việc nghiêm túc, được sự động viên, giúp đỡ và hướng dẫn tận tình của Thầy hướng dẫn PGS. TS. Quản Thành Thơ, luận văn Cao học “**Phát triển một ứng dụng quản lý bất động sản thông minh ở Bình Dương**” đã hoàn thành.

Tôi xin bày tỏ lòng biết ơn sâu sắc đến:

Thầy hướng dẫn **PGS. TS. Quản Thành Thơ** đã tận tình chỉ dẫn, giúp đỡ tôi hoàn thành luận văn này. Đồng thời tôi gửi lời cảm ơn đến các thầy, cô đã giảng dạy truyền đạt kiến thức quý báu cho tôi trong suốt thời gian học tập và nghiên cứu.

Tôi chân thành cảm ơn bạn bè, đồng nghiệp và gia đình đã động viên, khích lệ, tạo điều kiện giúp đỡ tôi trong suốt quá trình học tập, thực hiện và hoàn thành luận văn này.

Tóm tắt luận văn

Tên đề tài: Phát triển một ứng dụng quản lý bất động sản thông minh ở Bình Dương

Ngành: Hệ Thống Thông Tin.

Họ và tên học viên: Trần Thị Ngọc Dung.

Người hướng dẫn khoa học: PGS. TS. Quản Thành Thơ. Cơ sở đào tạo: Trường Đại học Thủ Dầu Một.

Tóm tắt nội dung: Nhận dạng thực thể có tên (NER - Named Entity Recognition) là một thành phần chính trong hệ thống xử lý ngôn ngữ tự nhiên (NLP - Natural language processing) để trả lời câu hỏi, truy xuất thông tin, trích xuất quan hệ, v.v... Vai trò chính của tác vụ này là nhận dạng các cụm từ trong văn bản và phân loại chúng vào trong các nhóm đã được định nghĩa trước như tên người, tổ chức, địa điểm, thời gian, loại sản phẩm, nhãn hiệu, v.v...

Trong luận văn này, mô hình tiền huấn luyện PhoBERT được áp dụng để giải quyết bài toán nhận dạng thực thể có tên (Named Entity Recognition) với tập dữ liệu liên quan đến ngành bất động sản.

Kết quả thu được của Luận văn là mô hình PhoBERT được xây dựng và kiểm thử trên cùng tập dữ liệu để so sánh độ chính xác với mô hình gốc và áp dụng vào một ứng dụng quản lý bất động sản ở Bình Dương.

MỤC LỤC

Lời cam đoan.....	ii
Lời cảm ơn	iii
Tóm tắt luận văn.....	iv
MỤC LỤC.....	v
Danh mục chữ viết tắt	viii
Danh mục hình ảnh	ix
MỞ ĐẦU.....	1
1. Lý do chọn đề tài.....	1
2. Mục tiêu nghiên cứu.....	2
3. Tổng quan nghiên cứu của đề tài	3
4. Đối tượng, phạm vi nghiên cứu.....	3
5. Phương pháp nghiên cứu.....	3
6. Đóng góp của đề tài.....	4
7. Cấu trúc của đề tài.....	4
Chương 1. CƠ SỞ LÝ THUYẾT CÓ LIÊN QUAN ĐẾN ĐỀ TÀI.....	5
1.1. Nhận dạng thực thể có tên	5
1.2. Nhúng từ	6
1.2.1. Phép nhúng từ là gì?.....	6
1.2.2. Công dụng của phép nhúng từ.....	7
1.2.3. Nhúng từ không ngữ cảnh	7
1.2.4. Nhúng từ có ngữ cảnh một chiều.....	7
1.2.5. Nhúng từ có ngữ cảnh hai chiều	8
1.3. Conditional Random Field.....	8
1.4. Transformer.....	10
1.4.1. Tổng quan về kiến trúc Transformer	11
1.4.2. Cơ chế self-attention.....	13

1.4.3.	Multi-head attention	19
1.4.4.	Biểu diễn thứ tự trong chuỗi với Positional Encoding	22
1.5.	BERT	24
1.5.1.	BERT là gì	24
1.5.2.	Sự ra đời của BERT	25
1.5.3.	Nền tảng của BERT	25
1.6.	PhoBERT	28
1.7.	Cách gán nhãn thực thể có tên	29
1.8.	Chỉ số đánh giá hệ thống.....	32
Chương 2.	CÁC CÔNG TRÌNH LIÊN QUAN.....	35
2.1.	Phương pháp tiếp cận dựa trên quy tắc (rule-based approach).....	36
2.2.	Phương pháp mạng neural học sâu	36
2.2.1.	Kiến trúc mạng nơ-ron học sâu cấp độ từ cho bài toán NER.. Error!	
	Bookmark not defined.	
2.2.2.	Kiến trúc mạng nơ-ron học sâu cấp độ ký tự cho bài toán NER	
	Error! Bookmark not defined.	
2.3.	Phương pháp BERT fine-tune	37
Chương 3.	MÔ TẢ HỆ THỐNG	38
Chương 4.	PHƯƠNG PHÁP NGHIÊN CỨU	40
4.1.	Sử dụng PhoBERT để huấn luyện	40
4.2.	Minh họa sử dụng thực tế	41
Chương 5.	CÁC CÔNG NGHỆ SỬ DỤNG.....	44
5.1.	Ngôn ngữ lập trình	44
5.1.1.	Python	44
5.1.2.	Javascript, HTML & CSS.....	45
5.2.	Thư viện - Framework	46
5.2.1.	Scrapy	46

5.2.2.	Django	46
5.2.3.	Nodejs	47
5.2.4.	Reactjs	47
5.2.5.	Tensorflow	47
5.2.6.	Pytorch.....	47
5.2.7.	VnCoreNLP	48
5.3.	Database	48
5.4.	Công cụ	48
5.4.1.	Docker	48
5.4.2.	Postman	48
Chương 6.	HIỆN THỰC HỆ THỐNG	50
6.1.	Hệ thống cào dữ liệu (Data Crawler).....	50
6.2.	Named Entity Recognition Service.....	52
6.3.	Hệ thống Django backend.....	54
6.4.	Hệ thống webapp frontend.....	55
6.5.	Kết quả trả về.....	57
Chương 7.	KIỂM THỬ VÀ ĐÁNH GIÁ	60
7.1.	Mô tả tập dữ liệu	60
7.2.	Kết quả thí nghiệm Mô hình PhoBERT.....	62
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN		67
1.	Các công việc đạt được	67
2.	Các hạn chế	67
3.	Bước phát triển.....	67
TÀI LIỆU THAM KHẢO.....		68

Danh mục chữ viết tắt

Ký Hiệu	Tên Tiếng Anh
AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers
CRF	Conditional Random Field
IE	Information Extraction
MLM	Masked Language Model
NER	Named Entity Recognition
NLP	Natural Language Processing
NSP	Next Sentence Prediction
RNN	Recurrent Neural Network

Danh mục bảng biểu

Bảng 1.1 Nhận các thực thể theo cấu trúc BIO.....	32
Bảng 7.1 Bảng các thực thể có tên cần xác định.....	61
Bảng 7.2 Bảng chỉ số và kết quả của các thực thể có tên	66

Danh mục hình ảnh

Hình 1.1 Mẫu tin rao bán bất động sản ở Bình Dương	1
Hình 1.1 Minh họa về nhận dạng thực thể có tên	5
Hình 1.2 Minh họa về phép nhúng từ	6
Hình 1.3 Kiến trúc mô hình Transformer	10
Hình 1.4 Kiến trúc đơn giản mô hình transformer cho bài toán dịch máy	11
Hình 1.5 Kiến trúc encoder - decoder bên trong mô hình Transformer cho bài toán dịch máy	11
Hình 1.6 Các ngăn xếp encoder - decoder bên trong mô hình Transformer.....	12
Hình 1.7 Hai lớp con bên trong một encoder của mô hình transformer	12
Hình 1.8 Ba lớp con bên trong một decoder của mô hình Transformer	13
Hình 1.9 Mô tả cơ chế self-attention giữa từ “it” và các từ khác trong câu tại layer thứ 5 của ngăn xếp encoder.....	14
Hình 1.10 Mô tả trừu tượng ba vector truy vấn (query), khóa (key) và giá trị (value) tạo ra từ các vector đầu vào.....	15
Hình 1.11 Tính điểm mối liên hệ giữa từ hiện tại và các từ khác trong câu.....	15
Hình 1.12 Cập nhật điểm bằng cách chia cho căn bậc 2 của số chiều của vector khóa và qua hàm softmax để chuẩn hóa	16
Hình 1.13 Các bước tính toán self-attention hoàn chỉnh	17
Hình 1.14 Tính toán ma trận truy vấn Q, khóa K và giá trị V từ ma trận đầu vào X	18
Hình 1.15 Tính toán ma trận attention từ các ma trận truy vấn Q, khóa K và giá trị V	19
Hình 1.16 Scaled Dot - Product Attention và Multi-head attention	19

Hình 1.17 Minh họa Multi-head attention dựa trên các câu hỏi khác nhau thì chú ý vào các từ khác nhau trong một câu.....	20
Hình 1.18 Minh họa hai đầu head #0 và #1 khi thực hiện Multi-head attention .	20
Hình 1.19 Tám đầu ra sau khi tính toán Multi-head attention	21
Hình 1.20 Nối tám ma trận Z_i và nhân với ma trận trọng số W^O để tạo 1 ma trận đầu ra duy nhất cho bước tính toán Multi-head attention	21
Hình 1.21 Toàn bộ quá trình tính toán Multi-head attention	22
Hình 1.22 Vector mã hóa vị trí (positional encoding) trong mô hình transformer	23
Hình 1.23 Ví dụ tính toán ma trận vị trí với $d = 4$ và $n = 100$	24
Hình 1.24 Minh họa token embeddings, segment embeddings và position embeddings trong mô hình BERT	26
Hình 1.25 Minh họa quá trình huấn luyện BERT cho bài toán MLM.....	27
Hình 1.26 Minh họa quá trình huấn luyện BERT cho bài toán Next Sentence Prediction	28
Hình 1.27 Cách tính Precision và Recall	33
Hình 2.1 Phương pháp BERT fine-tune.....	37
Hình 3.1 Kiến trúc tổng quát của hệ thống bài toán	38
Hình 4.1 Toàn bộ tiến trình pre-training và fine-tuning của BERT.....	40
Hình 4.2 Mẫu dữ liệu đầu vào.....	41
Hình 4.3 Sử dụng mô hình PhoBERT để tiên đoán	43
Hình 5.1 Các thư viện Machine Learning nổi tiếng có hỗ trợ Python.....	45
Hình 5.2 Xây dựng giao diện chatbot bằng ngôn ngữ HTML / CSS / JS.....	46
Hình 5.3 Sử dụng Postman để hỗ trợ kiểm thử kết quả từ Django API.....	49
Hình 6.1 Giao diện web browser của hệ thống cào dữ liệu từ trang web An Cư	51

Hình 6.2 Giao diện web browser của hệ thống cào dữ liệu từ trang web đăng bán nhà đất	51
Hình 6.3 Gán nhãn cho dữ liệu đã được cào về	52
Hình 6.4 Quá trình training model	53
Hình 6.5 Sử dụng postman để giả lập gửi API request tới NER service	54
Hình 6.6 Giao diện web hiển thị thông tin các mẫu tin được cào về	56
Hình 6.7 Thông tin chi tiết một mẫu tin cào về được gán nhãn.....	56
Hình 6.8 Thông tin được thống kê qua biểu đồ	57
Hình 6.9 Database lưu trữ dữ liệu sau khi crawler về.....	58
Hình 6.10 Dữ liệu sau khi được crawler về	59
Hình 7.1 Đồ thị training và validation loss theo epoch cho mô hình PhoBERT .	62
Hình 7.2 Đồ thị training và validation accuracy theo epoch cho mô hình PhoBERT	63
Hình 7.3 Đồ thị training và validation f1-score theo epoch cho mô hình PhoBERT	63

MỞ ĐẦU

1. Lý do chọn đề tài

Ngày nay, Trí Tuệ Nhân Tạo (AI - Artificial Intelligence) là một lĩnh vực phát triển rất mạnh với nhiều ứng dụng thực tế và các chủ đề nghiên cứu rất tích cực. Con người hiện nay tạo nên các phần mềm thông minh để tự động hóa công việc, nhận diện được âm thanh hay hình ảnh, chẩn đoán y học và có thể hỗ trợ nghiên cứu khoa học cơ bản.

AI giúp cho con người xử lý trong những lĩnh vực có dữ liệu nhiều và phức tạp. Một trong những lĩnh vực đó là bất động sản, đặc biệt là ở Bình Dương - nơi hiện đang là một trong những thị trường bất động sản rất được thu hút những năm gần đây.

Chúng ta xem xét ví dụ sau:

BÁN LÔ ĐẤT KHU DÂN CƯ HÒA LỢI, P.HÒA PHÚ, TP.THỦ DẦU MỘT, BÌNH DƯƠNG

7 tỷ 500 triệu

📍 Đường D18, Phường Hòa Phú, Thành phố Thủ Dầu Một, Bình Dương

Bán lô đất Khu dân cư Hòa Lợi, Phường Hòa Phú, TP.Thủ Dầu Một, tỉnh Bình Dương.
Diện tích: 8,25 x 30 = 247,1m2 full thổ cư
Nằm ngay bùng binh, gần chợ Hoà Phú (Hoà Lợi cũ), dân cư sầm uất, rất thuận tiện kinh doanh
Giá 7,5 tỷ
Liên hệ: 0909494841

Bấm gọi ngay: 0909494841 [Ấn số](#)

Thông tin cơ bản

💰 Giá: 7 tỷ 500 triệu	📄 Loại hình đất: Đất thổ cư
📐 Diện tích đất: 247 m²	📑 Giấy tờ pháp lý: Sổ đỏ

Hình 1.1 Mẫu tin rao bán bất động sản ở Bình Dương

Trong Hình 1.1 là một mẫu tin rao bán đất ở Bình Dương. Hiện nay, có hàng ngàn thông tin rao vặt như vậy, khiến cho con người gặp khó khăn trong việc tìm kiếm thông tin chính xác và phù hợp với nhu cầu của mình.

Do đó, cần có một hệ thống hỗ trợ tìm kiếm nhanh chóng. Một hệ thống

như vậy sẽ cần trích xuất các thông tin quan trọng như: loại bất động sản, giá, diện tích, vị trí, tiện ích xung quanh, thông tin pháp lý, thông tin liên hệ, vv... để có thể hiểu được ý nghĩa của mẫu rao vặt và từ đó, đáp ứng được việc giúp con người tìm thấy các thông tin liên quan đến bất động sản ở Bình Dương phù hợp với nhu cầu một cách nhanh và đầy đủ nhất.

Trong học thuật, bài toán trích xuất các thông tin như trên gọi là bài toán Xác Định Thực Thể có tên (NER - Named Entity Recognition). Trong đó, loại bất động sản, giá, diện tích, vị trí, tiện ích xung quanh, thông tin pháp lý, thông tin liên hệ, vv... được gọi là các thực thể có tên. Đây là một bài toán nổi tiếng trong lĩnh vực xử lý ngôn ngữ tự nhiên và cũng là mục tiêu của đề tài này.

Như vậy, với đề tài: **“Phát triển một ứng dụng quản lý bất động sản thông minh ở Bình Dương”** cho luận văn tốt nghiệp cao học của mình, tôi sẽ xây dựng một hệ thống hỗ trợ tìm kiếm thông tin với bài toán xác định thực thể có tên.

2. Mục tiêu nghiên cứu

Để xây dựng được hệ thống tự động rút trích các thông tin về bất động sản thì chúng ta cần xây dựng các thành phần cốt lõi sau đây:

- **Bước 1:** Thành phần thu thập thông tin bất động sản từ các trang web rao bán bất động sản hoặc các công ty bất động sản ở Bình Dương (Data Crawler)
- **Bước 2:** Thành phần trích xuất thông tin về bất động sản như giá trị, diện tích, địa điểm từ dữ liệu thu thập được ở bước 1 (Named Entity Recognition)
- **Bước 3:** Thành phần chuẩn hóa dữ liệu phục vụ việc lưu trữ các thông tin đã trích xuất được (Data Normalization)
- **Bước 4:** Thành phần lưu trữ thông tin thô và thông tin đã được trích xuất cũng như hỗ trợ truy vấn dữ liệu (Database)
- **Bước 5:** Thành phần nhận dạng thực thể có tên từ các yêu cầu tìm kiếm thông tin của người dùng (Named Entity Recognition)
- **Bước 6:** Thành phần giao diện tương tác với người dùng (UI) cho phép người dùng xem và truy xuất thông tin đã tìm kiếm ở bước 5.

Trong phạm vi nghiên cứu của đề tài này, tôi tập trung nghiên cứu thành phần nhận dạng thực thể có tên từ các yêu cầu của người dùng cũng như trích xuất thông tin về bất động sản Bình Dương như giá trị, diện tích, địa điểm, vv.. từ dữ liệu thu thập được từ các trang web rao bán hoặc cho thuê bất động sản.

3. Tổng quan nghiên cứu của đề tài

Bất động sản là một vấn đề rất được nhiều người và xã hội đặt biệt quan tâm, việc tìm kiếm một bất động sản nhằm phục vụ cho nhu cầu của đa số mọi người cực kỳ cao như để an cư, kinh doanh, hay buôn bán là khá nhiều.

Để giúp đỡ người dân trong vấn đề liên quan tới việc tìm kiếm bất động sản, cần xây dựng một ứng dụng thông minh trong lĩnh vực bất động sản là cần thiết, nhằm mục đích giải quyết một số vấn đề như: tiếp cận nguồn thông tin dễ dàng hơn, nắm bắt được đúng giá trị thực của bất động sản.

Theo tình hình thực tế và phân tích của một số trang mạng lớn như vnexpress, baobinhduong, batdongsan... nhu cầu bất động sản ở Bình Dương thời điểm hiện tại và trong tương lai có xu hướng tăng cao. Điều này đòi hỏi cần có một bài toán giúp người có nhu cầu tiếp cận thông tin một cách dễ dàng, một ứng dụng thông minh về bất động sản ở Bình Dương là cần thiết.

Xuất phát từ những vấn đề trên, tôi chọn đề tài nghiên cứu **“Phát triển một ứng dụng quản lý bất động sản thông minh ở Bình Dương”** để làm đề tài nghiên cứu luận văn thạc sỹ của mình.

4. Đối tượng, phạm vi nghiên cứu

Nhận dạng thực thể có tên là tác vụ cơ bản trong lĩnh vực xử lý ngôn ngữ tự nhiên và hiện nay đã có rất nhiều công trình nghiên cứu về vấn đề này. Trong luận văn này, tôi sẽ tập trung vào việc ứng dụng mô hình học máy PhoBERT trong bài toán nhận dạng thực thể có tên. Cuối cùng, mô hình PhoBERT được xây dựng, huấn luyện và kiểm thử trên cùng tập dữ liệu để ứng dụng vào hệ thống truy xuất, tìm kiếm tự động.

5. Phương pháp nghiên cứu

Trong đề tài này, tôi nghiên cứu lý thuyết về bài toán nhận dạng thực thể có tên, nghiên cứu mô hình BERT, PhoBERT và ứng dụng mô hình PhoBERT vào bài toán nhận dạng thực thể có tên.

6. Đóng góp của đề tài

- Giúp người dân có được một ứng dụng có đầy đủ thông tin về bất động sản ở Bình Dương. Dễ dàng truy cập thông tin từ nhiều nguồn khác nhau.
- Cập nhật chính xác giá trị thị trường bất động sản.
- Cung cấp dữ liệu một cách dễ dàng từ nhiều nguồn khác nhau.

7. Cấu trúc của đề tài


Nội dung luận văn được chia thành các phần như sau:

- **Chương 1:** Cơ sở lý thuyết có liên quan đến đề tài
- **Chương 2:** Các công trình liên quan
- **Chương 3:** Mô tả hệ thống
- **Chương 4:** Phương pháp nghiên cứu
- **Chương 5:** Các công nghệ sử dụng
- **Chương 6:** Hiện thực hệ thống
- **Chương 7:** Kiểm thử và đánh giá

Chương 1. CƠ SỞ LÝ THUYẾT CÓ LIÊN QUAN ĐẾN ĐỀ TÀI

1.1. Nhận dạng thực thể có tên

Nhận dạng thực thể có tên (NER - Named Entity Recognition) là một thành phần chính trong hệ thống xử lý ngôn ngữ tự nhiên (NLP - Natural language processing) để trả lời câu hỏi, truy xuất thông tin, trích xuất quan hệ, v.v... Vai trò chính của tác vụ này là nhận dạng các cụm từ trong văn bản và phân loại chúng vào trong các nhóm đã được định nghĩa trước như tên người, tổ chức, địa điểm, thời gian, loại sản phẩm, nhãn hiệu, v.v... Từ kết quả của tác vụ nhận dạng thực thể có tên, ta có thể xử lý cho nhiều bài toán phức tạp hơn như Chatbot, Question Answering, Search,...



Căn nhượng transaction căn nhà real_estate_type mới 100%. Căn nhà real_estate_type ở vị trí trung tâm Thủ_Dầu_Một city, xung quanh tiện ích đầy đủ. Khu vực rất nhiều biệt_thự surrounding sang trọng, yên tĩnh. Rất phù hợp với kinh_doanh_dịch_vụ_làm_đẹp usage tại nhà real_estate_type. Bác sĩ làm phòng khám usage tại nhà real_estate_type. Kinh doanh loại hình online usage tại nhà real_estate_type. Gia đình công chức, gia đình có trẻ nhỏ hoặc người già. Gia đình đông người... Nhà real_estate_type 1 trệt 1 lầu floor. Diện tích đất real_estate_type 175m2 area (Rộng 4.5 x dài 40 m area). Thô_cư 100 m2 area. Diện tích sàn xây dựng theo giấy phép được cấp là 139 m2 area. Vị trí căn_nhà real_estate_type tọa lạc tại khu phố 4, phường Chánh_Nghĩa ward, Thủ_Dầu_Một district, Bình_Dương city. Cách chợ_Thủ_Dầu_Một surrounding khoảng 800m. Cách đường Bùi_Quốc_Khánh street đi vào khoảng 100m. Cách ngã 3 Lò_Chén street giao nhau với <unk>Thăng_tám street khoảng 100m. Nhà real_estate_type có 3 phòng ngủ bed_room, 1 phòng thờ, 2 nhà vệ sinh bath_room. Diện tích sân trước rộng 100m2 area (để được 4 chiếc xe hơi). Nhà real_estate_type có camera an ninh gần xung quanh nhà real_estate_type. Nước máy năng lượng nóng lạnh. Nhà real_estate_type hướng Tây_Bắc direction. Phong cách thiết kế hiện đại, thoáng mát. Phòng ngủ bed_room lót sàn gỗ và ốp gỗ xung quanh nhà real_estate_type. Căn nhượng transaction giá 3,5 tỷ price. (Thương lượng trực tiếp). Hoa hồng môi giới 2%. Nhà real_estate_type chính chủ. ĐT: 0947.465.999 phone Mr_Minh author.

Hình 1.1 Minh họa về nhận dạng thực thể có tên

Nhận dạng thực thể có tên không chỉ hoạt động như một công cụ độc lập để trích xuất thông tin (IE - Information Extraction), mà còn đóng một vai trò thiết yếu trong nhiều loại xử lý ngôn ngữ tự nhiên như là các ứng dụng như hiểu văn bản - text understanding (Zhang, et al., 2019) và (Cheng & Erk, 2020), truy xuất thông tin (Guo, Xu, Cheng, & Li, 2009), tóm tắt văn bản tự động (Aone, 1999), trả lời câu hỏi (Moll, Zaanen, & Smith, 2006), dịch máy (Babych & Hartley, 2003) và cấu trúc cơ sở kiến thức (Etzioni, et al., 2005), v.v...

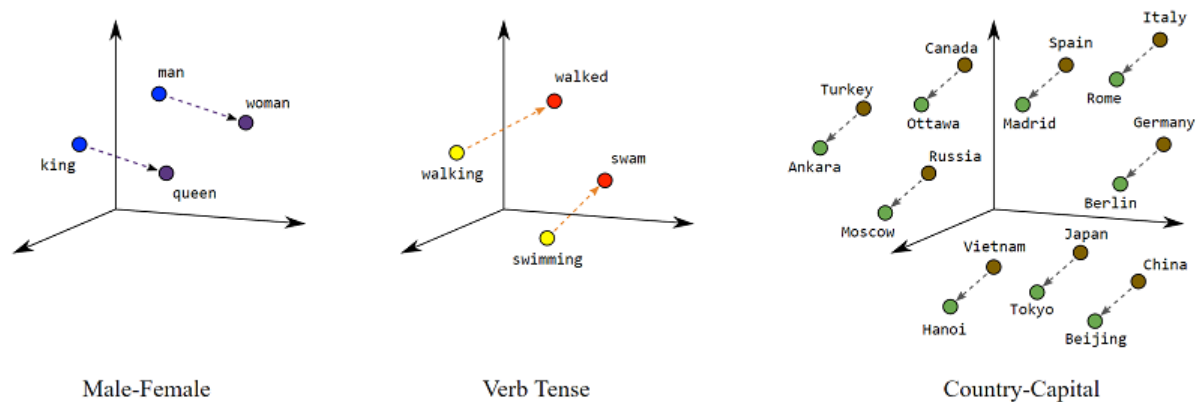
Từ năm 1995, hội thảo quốc tế chuyên đề Hiểu thông điệp (Message Understanding Conference - MUC) lần thứ 6 đã bắt đầu tổ chức đánh giá các hệ thống NER cho tiếng Anh. Tại hội thảo CoNLL năm 2002 và 2003, các hệ thống NER cho tiếng Hà Lan, Tây Ban Nha, Đức và Anh cũng được đánh giá. Trong các

tác vụ đánh giá này, người ta xét 4 loại thực thể có tên: tên người, tên tổ chức, tên địa danh và các tên khác. Gần đây, vẫn tiếp tục có các cuộc thi về NER được tổ chức, ví dụ GermEval 2014 cho tiếng Đức.

Đối với tiếng Việt, cũng có vài cuộc thi như VLSP 2016, VLSP 2019 nhằm đưa ra được một đánh giá khách quan về chất lượng các công cụ NER, khuyến khích phát triển các hệ thống trích rút thực thể có tên đạt độ chính xác cao. Điều này đã chỉ ra vai trò thiết yếu của NER trong nhiều bài toán xử lý ngôn ngữ tự nhiên.

1.2. Nhúng từ

1.2.1. Phép nhúng từ là gì?



Hình 1.2 Minh họa về phép nhúng từ

Phép nhúng từ (word embedding) là phương pháp ánh xạ (map) những từ ngữ vào các vector hoặc số thực, còn được gọi là phương pháp mô hình hóa ngôn ngữ/dữ liệu. Một phép nhúng từ tốt sẽ mang lại nhiều lợi ích cho việc tính toán lẫn minh họa dữ liệu.

Ví dụ, chúng ta muốn tìm thủ đô của Nga khi biết thủ đô của Việt Nam là Hà Nội thì sẽ làm như thế nào? Với một phép nhúng từ đủ tốt, vector Việt Nam - Hà Nội sẽ (gần như) song song với vector Nga - [thủ đô của Nga]. Do đó, trước tiên ta sẽ tính *vector* tịnh tiến từ Việt Nam đến Nga, tạm gọi là *vector* v_1 , sau đó

tìm ảnh của điểm tọa độ Hà Nội thông qua phép tịnh tiến theo *vector* v_1 . Cuối cùng, ta tra trong cơ sở dữ liệu từ ngữ ánh xạ với tọa độ vừa tìm được, kết luận Moscow là thủ đô của Nga.

1.2.2. Công dụng của phép nhúng từ

Các tác vụ xử lý ngôn ngữ tự nhiên thường có đầu vào là các câu chữ, song máy tính lại chỉ có thể tính toán dựa trên số nên chúng ta cần phải tìm cách chuyển đổi từ câu chữ sang các *vector*, ma trận. Phép nhúng từ là một trong những cách thực hiện điều đó. Hiện tại, các phép nhúng từ mới xuất hiện đều có sử dụng mạng neural (neural network) như Word2vec, GloVe, BERT, XLNet,... Bên cạnh đó cũng có những phương pháp nhúng từ khác dựa trên thống kê như BoW (Bag of Words), TF-IDF (Term Frequency, Inverse Document Frequency).

Các phương pháp dựa trên thống kê như BoW, TF-IDF có tác dụng khá tốt đối với các tập dữ liệu (dataset) kém phong phú. Trong khi những phương pháp sử dụng mạng neural sẽ có tác dụng tốt hơn hẳn khi xử lý các tác vụ phức tạp, có tập dữ liệu khổng lồ.

1.2.3. Nhúng từ không ngữ cảnh

Mỗi từ trong từ điển hay kho văn bản sẽ có một vector đại diện cho nó. Và trong bất cứ câu nào, đoạn văn nào thì từ đó vẫn chỉ được biểu diễn bởi duy nhất 1 vector.

Ví dụ ta có 2 câu sau:

- Con đường này thật là rộng!
- Chúng ta nên pha thêm đường vào ly café.

Rõ ràng từ “đường” ở hai câu trên mang nghĩa khác nhau, nhưng với phương pháp nhúng từ không ngữ cảnh thì cả hai từ này đều ánh xạ ra chung một vector nhúng từ.

1.2.4. Nhúng từ có ngữ cảnh một chiều

Chúng ta sử dụng kiến trúc mạng Recurrent Neural Network (RNN) để có

thể tạo ra mối quan hệ thứ tự giữa các từ trong câu, từ đó tạo ra vector nhúng từ có ngữ cảnh. Tuy nhiên việc này chỉ thực hiện được theo một chiều từ trái sang phải hoặc từ phải sang trái mà thôi. Một số mạng phức tạp hơn như BiLSTM có thể chạy dọc theo câu theo hai hướng ngược nhau, nhưng hai hướng này lại độc lập chả liên quan gì nhau nên có thể xem là một chiều mà thôi.

Ví dụ ta có câu sau:

- Hôm nay Nam đưa bạn gái đi chơi
- Giờ ta ẩn đi từ “bạn gái” và câu trên trở thành:
- Hôm nay Nam đưa [mask] đi chơi

Yêu cầu bài toán: dự đoán từ đã được ẩn đi (mask). Vì mô hình chỉ được huấn luyện một chiều, cho nên nó sẽ dự đoán [mask] từ các từ trước đó là “Hôm nay Nam đưa” và có thể ra kết quả [mask] là “tiền”, “hàng”, ...

1.2.5. Nhúng từ có ngữ cảnh hai chiều

BERT là viết tắt của Bidirectional Encoder Representations from Transformers. Ngay trong cái tên của BERT chúng ta đã thấy ngay chữ Bidirectional (2 chiều). Cụ thể là một từ trong câu sẽ được biểu diễn một cách có liên quan đến cả từ trước lẫn từ sau, hay nói cách khác là liên quan đến tất cả các từ còn lại trong câu. Do đó khi ta che 1 từ trong câu đi, ví dụ như từ “bạn gái” bên trên thì lập tức model có thể dự đoán ra khá chính xác vì dựa vào cả đoạn “Hôm nay Nam đưa” và “đi chơi”.

1.3. Conditional Random Field

Các mô hình phân loại truyền thống giả định rằng các mục dữ liệu là độc lập. Tuy nhiên, dữ liệu trong thế giới thực thường xen kẽ và có cấu trúc phức tạp. Giả sử chúng ta muốn phân loại các trang web thành các danh mục khác nhau, ví dụ: trang chủ của sinh viên và giảng viên. Danh mục của một trang web thường liên quan đến các danh mục của các trang được liên kết với nó. Thay vì phân loại

các trang một cách độc lập, chúng ta nên mô hình hóa chúng cùng nhau để kết hợp các dấu hiệu ngữ cảnh như vậy.

Trường ngẫu nhiên có điều kiện (CRF - Conditional Random Field) là một cách tiếp cận có điều kiện để phân loại dữ liệu có cấu trúc, do (Lafferty, McCallum, & Pereira, 2001) đề xuất. Trong khi các mô hình phân loại truyền thống dự đoán nhãn cho một mẫu đơn lẻ mà không xem xét các mẫu "lân cận", CRF có thể tính đến ngữ cảnh. Để làm như vậy, các dự đoán được mô hình hóa dưới dạng mô hình đồ họa, mô hình này thể hiện sự hiện diện của các phụ thuộc giữa các dự đoán. Loại đồ thị nào được sử dụng tùy thuộc vào ứng dụng. Ví dụ: trong xử lý ngôn ngữ tự nhiên, CRF "chuỗi tuyến tính" (linear chain) là phổ biến, mà mỗi dự đoán chỉ phụ thuộc vào các hàng xóm trực tiếp của nó. Trong xử lý hình ảnh, biểu đồ thường kết nối các vị trí với các vị trí lân cận hoặc tương tự để củng cố rằng các vùng ấy nhận được các dự đoán tương tự.

CRF là một mô hình xác suất cho các bài toán dự đoán có cấu trúc và đã được áp dụng rất thành công trong rất nhiều lĩnh vực như thị giác máy tính, xử lý ngôn ngữ tự nhiên, sinh-tin học, ... Trong mô hình CRF, các nút chứa dữ liệu đầu vào và các nút chứa dữ liệu đầu ra được kết nối trực tiếp với nhau.

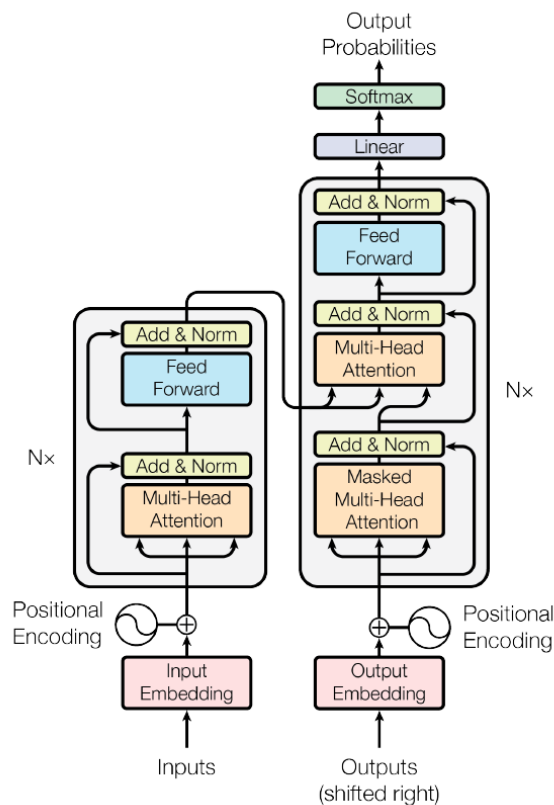
CRF có thể được sử dụng để gán nhãn tên riêng với đầu vào là các đặc trưng của một từ được rút trích bằng tay như:

- Chữ cái đầu có được viết hoa hay không?
- Viết hoa toàn bộ?
- Là số?
- Từ đứng trước là từ hoa?
- Từ đang xét
- Từ đứng trước và từ đang xét
- ...

1.4. Transformer

Mô hình transformer được giới thiệu bởi (Vaswani, et al., 2017) vào năm 2017 trong bài báo "Attention Is All You Need". Mô hình transformer được phát triển để giải quyết các vấn đề truyền tuần tự (sequence transduction) hoặc dịch máy (machine translation). Nghĩa là tất cả các bài toán liên quan đến việc biến đổi chuỗi đầu vào thành chuỗi đầu ra đều có thể được giải quyết bằng mô hình transformer, ví dụ như nhận dạng giọng nói (speech recognition), chuyển đổi văn bản thành giọng nói (text - to - speech transformation), v.v...

Kiến trúc tổng quát của mô hình transformer được thể hiện Hình 1.3. Tôi sẽ đi từng bước để giải thích cụ thể từng phần của kiến trúc tổng quát này ở phần sau của luận văn.



Hình 1.3 Kiến trúc mô hình Transformer

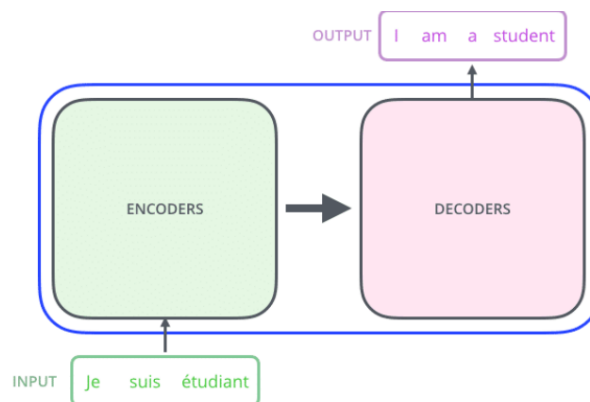
1.4.1. Tổng quan về kiến trúc Transformer

Đầu tiên, chúng ta hãy nhìn mô hình như một cái hộp đen. Trong một ứng dụng dịch máy, nó sẽ nhận vào một câu trong một ngôn ngữ và sinh ra bản dịch của nó trong một ngôn ngữ khác như mô tả ở Hình 1.4.



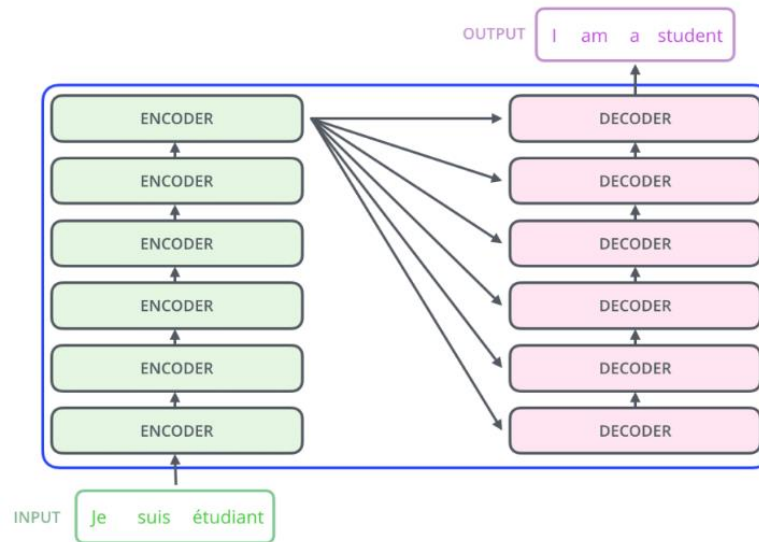
Hình 1.4 Kiến trúc đơn giản mô hình transformer cho bài toán dịch máy

Nếu mở hộp đen ra, ta sẽ thấy một thành phần mã hóa, và một thành phần giải mã, và một kết nối giữa chúng như trong Hình 1.5.



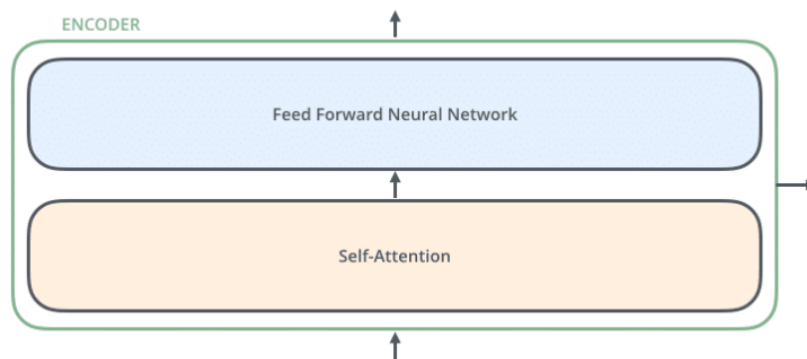
Hình 1.5 Kiến trúc encoder - decoder bên trong mô hình Transformer cho bài toán dịch máy

Thành phần mã hóa là một ngăn xếp encoder (các encoder xếp chồng lên nhau) (bài báo gốc sử dụng 6 encoder - đây không phải là một con số đặc biệt, ta hoàn toàn có thể thử nghiệm với các cấu hình khác). Thành phần giải mã là một ngăn xếp decoder với cùng số lượng. Hai ngăn xếp encoder và decoder bên trong mô hình transformer được biểu diễn trong Hình 1.6.



Hình 1.6 Các ngăn xếp encoder - decoder bên trong mô hình Transformer

Các encoder có kiến trúc giống nhau (nhưng không có cùng trọng số). Mỗi encoder lại được tạo nên bởi hai lớp con như mô tả ở Hình 1.7.

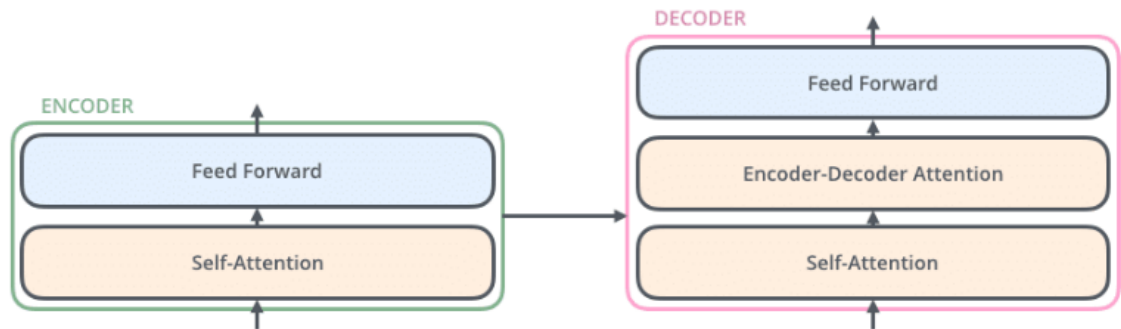


Hình 1.7 Hai lớp con bên trong một encoder của mô hình transformer

Đầu vào của encoder đầu tiên sẽ đi qua một lớp self-attention - một lớp giúp cho encoder nhìn vào các từ khác khi đang mã hóa một từ cụ thể. Chúng ta sẽ phân tích kỹ self-attention ở đoạn sau của luận văn này. Đầu ra của self-attention được truyền vào một mạng nơ ron truyền thẳng (feed - forward). Tất cả các vị trí khác nhau đều sử dụng chung một mạng truyền thẳng.

Decoder cũng có hai thành phần đó (self-attention và feed - forward) nhưng nằm giữa chúng là một lớp attention giúp decoder tập trung vào phần quan trọng

của câu đầu vào.



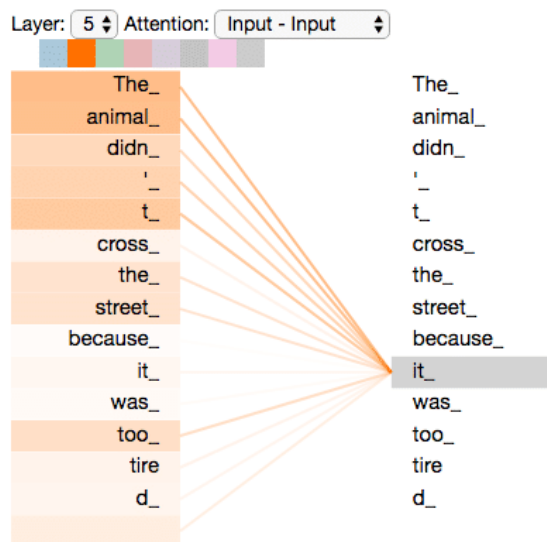
Hình 1.8 Ba lớp con bên trong một decoder của mô hình Transformer

1.4.2. Cơ chế self-attention

Giả sử câu sau là câu đầu vào mà chúng ta cần dịch từ tiếng Anh sang tiếng Việt: “*The animal didn't cross the street because it was too tired*”. Từ “*it*” trong câu trên đại diện cho cái gì? “Con vật” (*animal*) hay “đường phố” (*street*)? Câu hỏi này đơn giản với con người nhưng không đơn giản với các thuật toán.

Khi mô hình xử lý từ “*it*”, self-attention cho phép nó liên kết “*it*” với “*animal*”. Khi mô hình xử lý từng từ (từng vị trí trong câu đầu vào), self-attention cho phép nó quan sát các vị trí khác trong câu để tìm ra ý tưởng cho việc mã hóa từ hiện tại tốt hơn. Self-attention là cách mà Transformer sử dụng để duy trì hiểu biết về các từ khác có liên quan đến từ hiện tại.

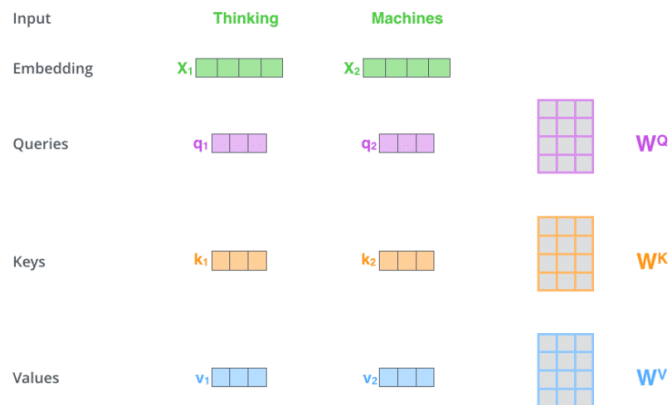
Trong Hình 1.9 dưới đây, tại layer thứ 5 (layer trên cùng của ngăn xếp encoder trong mô hình transformer), chúng ta có thể nhận thấy mô hình đã học và liên kết từ “*it*” với “*The animal*”, phần lớn tập trung vào “*The animal*” (màu đậm) và ít tập trung hơn vào các từ khác có trong câu.



Hình 1.9 Mô tả cơ chế self-attention giữa từ “it” và các từ khác trong câu tại layer thứ 5 của ngăn xếp encoder

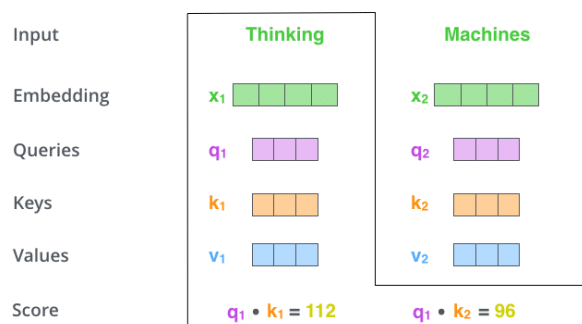
Các bước để tính toán self-attention được diễn tả cụ thể như sau:

- **Bước đầu tiên** để tính self-attention là tạo ra ba vector từ mỗi vector đầu vào của encoder (trong trường hợp này là embedding của mỗi từ). Với mỗi từ, ta sẽ tạo một vector truy vấn (Query), một vector khóa (Key), và một vector giá trị (Value). Các vector này được tạo ra bằng cách nhân embedding với ba ma trận được cập nhật trong quá trình huấn luyện. Chú ý rằng, các vector mới này có chiều nhỏ hơn vector embedding. Chiều của chúng là 64, trong khi vector embedding cũng như đầu vào và đầu ra của encoder có chiều 512. Mặc dù chiều của chúng không nhất thiết phải nhỏ hơn, đây là một lựa chọn trong kiến trúc để việc tính toán multihead attention (gần như) cố định.



Hình 1.10 Mô tả trừu tượng ba vector truy vấn (query), khóa (key) và giá trị (value) tạo ra từ các vector đầu vào

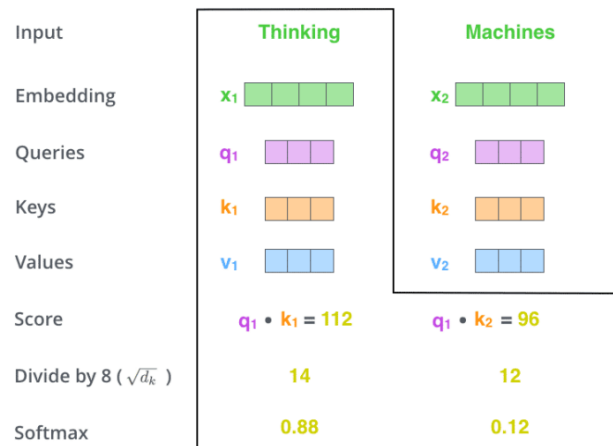
- **Bước thứ hai** để tính self-attention là tính điểm. Giả sử chúng ta tính self-attention cho từ đầu tiên trong ví dụ, “Thinking”. Ta cần tính điểm cho mỗi từ trong câu đầu vào so với từ này. Điểm sẽ quyết định cần chú ý bao nhiêu vào các phần khác của câu đầu vào khi ta đang mã hóa một từ cụ thể. Điểm được tính bằng phép nhân vô hướng giữa véc tơ truy vấn với véc tơ khóa của từ mà ta đang tính điểm. Nếu ta tiến hành self-attention cho từ ở vị trí thứ nhất, điểm đầu tiên sẽ là tích vô hướng của q1 và k1. Điểm thứ hai là tích vô hướng của q1 và k2.



Hình 1.11 Tính điểm mối liên hệ giữa từ hiện tại và các từ khác trong câu

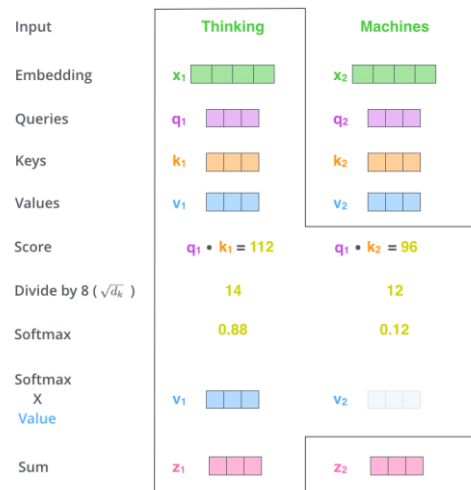
- **Bước thứ ba và bước thứ tư** là chia điểm cho 8 (căn bậc hai của số chiều của véc tơ khóa, trong bài báo gốc là 64). Điều này giúp cho độ dốc ổn định hơn. Có thể có các giá trị khả dĩ khác, nhưng đây là giá trị mặc định. Sau đó truyền kết quả qua một phép softmax. Softmax chuẩn hóa các điểm để chúng là các số dương có tổng bằng 1. Điểm softmax sẽ quyết định mỗi từ sẽ được thể hiện nhiều

hay ít tại vị trí hiện tại. Rõ ràng là từ tại vị trí này sẽ có điểm softmax cao nhất, nhưng đôi khi, chú ý đến các từ khác là cần thiết để hiểu từ hiện tại.



Hình 1.12 Cập nhật điểm bằng cách chia cho căn bậc 2 của số chiều của vector khóa và qua hàm softmax để chuẩn hóa

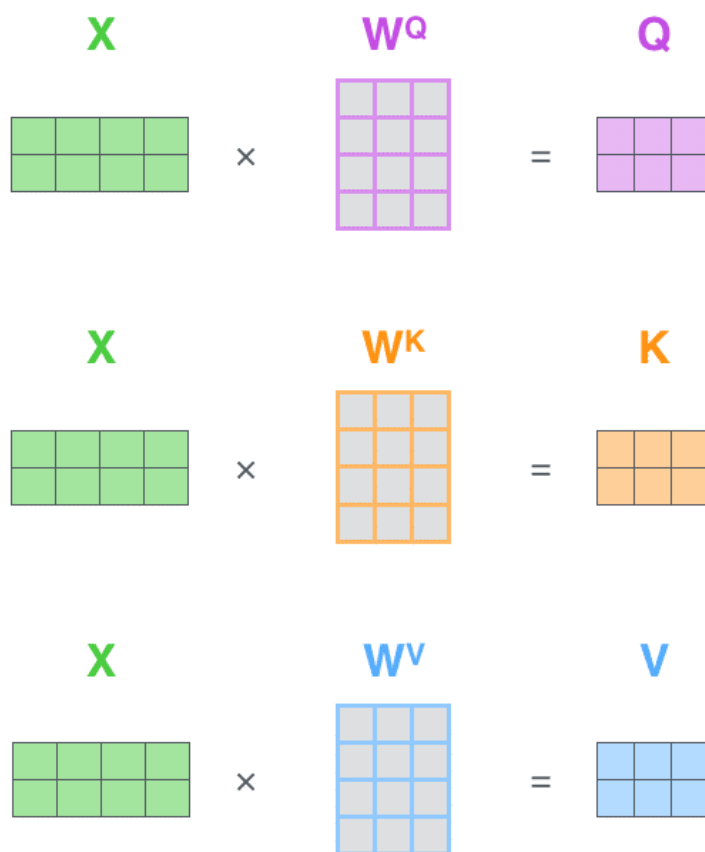
- **Bước thứ năm** là nhân mỗi véc tơ giá trị với điểm softmax (trước khi cộng chúng lại). Một cách trực giác, việc này bảo toàn giá trị của các từ mà ta muốn chú ý và bỏ qua các từ không liên quan (nhân chúng với một số rất nhỏ, ví dụ 0.001).
- **Bước thứ sáu** là cộng các véc tơ giá trị đã được nhân trọng số. Kết quả chính là đầu ra của lớp self-attention tại vị trí hiện tại (từ đầu tiên trong ví dụ của ta).



Hình 1.13 Các bước tính toán self-attention hoàn chỉnh

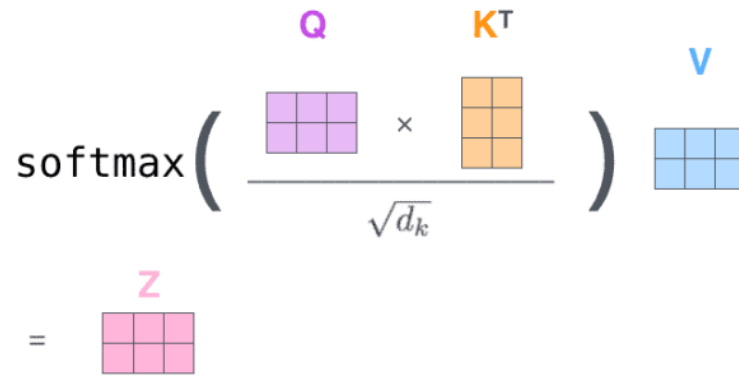
Đến đây là kết thúc việc tính toán self-attention. Vector kết quả có thể được gửi tới mạng truyền thẳng (feed - forward). Trong cài đặt thực tế, việc tính toán này được thực hiện với ma trận để đảm bảo có thể thực hiện tính toán song song trên GPU (Central Processing Units) hay TPU (Tensor Processing Unit). Việc tính toán self-attention bằng nhân ma trận được mô tả khái quát như sau:

- **Bước đầu tiên** là tính các ma trận truy vấn (Query), khóa (Key), và giá trị (Value). Ta thực hiện điều này bằng cách gộp các embedding vào ma trận X , và nhân chúng với ma trận trọng số sẽ được huấn luyện (W_Q , W_K , W_V).



Hình 1.14 Tính toán ma trận truy vấn Q , khóa K và giá trị V từ ma trận đầu vào X

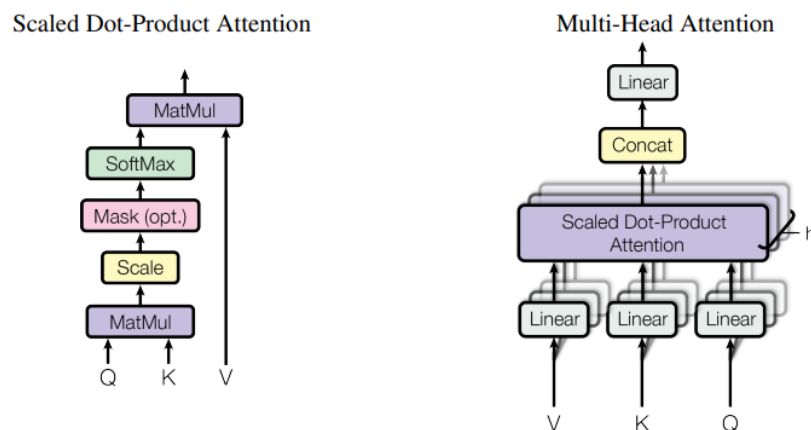
- Cuối cùng, do ta đang thực hiện trên ma trận, ta có thể gộp bước từ 2 đến 6 trong một công thức duy nhất để tính đầu ra của lớp self-attention.



Hình 1.15 Tính toán ma trận attention từ các ma trận truy vấn Q , khóa K và giá trị V

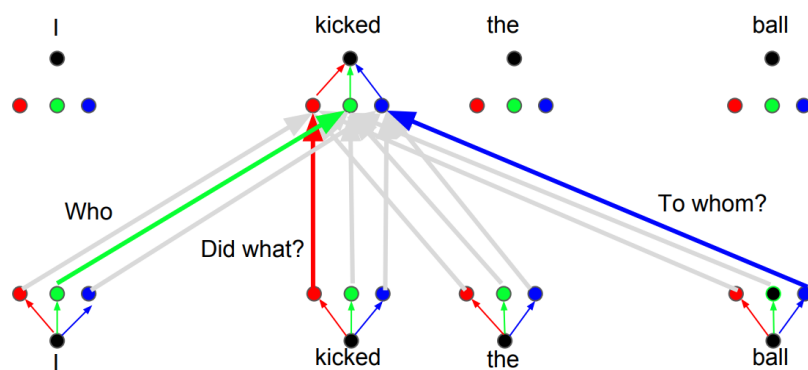
1.4.3. Multi-head attention

Trong phần trước, chúng ta đã tìm hiểu về cơ chế self-attention của transformer để có thể tạo mối liên hệ giữa từ hiện tại và các từ khác trong câu. Có một vài chi tiết khác giúp mô hình transformer có thể hoạt động tốt hơn nữa. Đó chính là Multi-head attention.



Hình 1.16 Scaled Dot - Product Attention và Multi-head attention

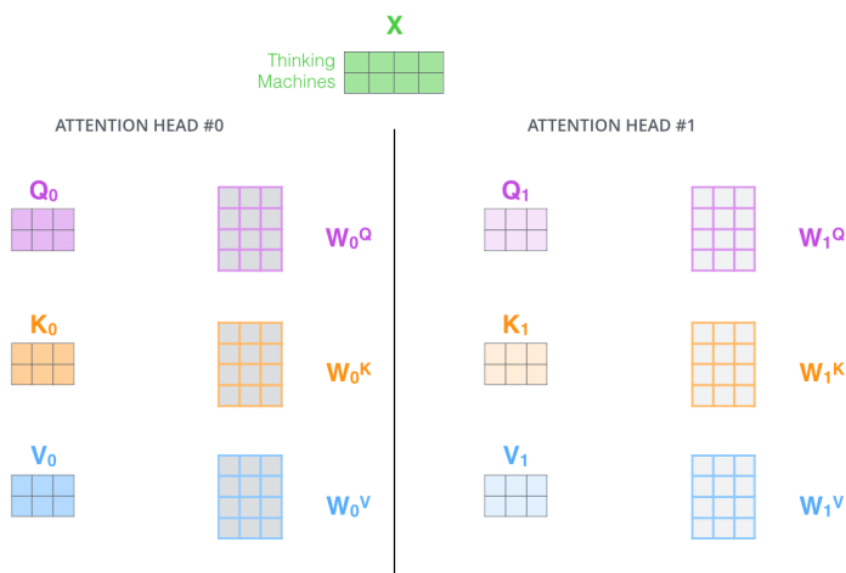
Ý tưởng đằng sau Multi-head attention là bất cứ khi nào chúng ta dịch một từ, chúng ta có thể chú ý đến từng từ khác nhau dựa trên loại câu hỏi mà mô hình đang được hỏi. Như trong Hình 1.17, tùy thuộc vào câu hỏi mà sự chú ý sẽ tập trung vào từng phần khác nhau trong câu.



Hình 1.17 Minh họa Multi-head attention dựa trên các câu hỏi khác nhau thì chú ý vào các từ khác nhau trong một câu

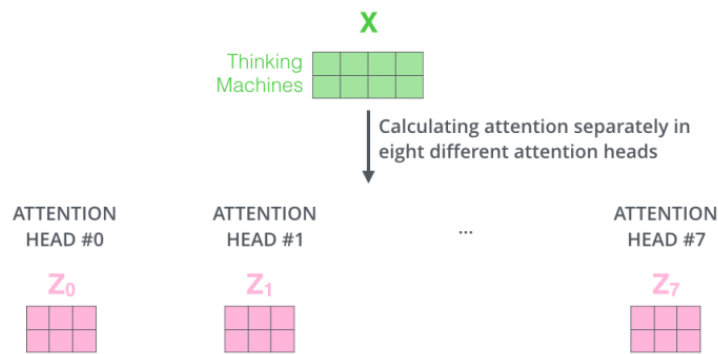
Các bước để thực hiện việc tính toán Multi-head attention trong mô hình transformer được mô tả cụ thể như sau:

- **Bước 1:** Với multi-headed attention, ta duy trì các ma trận trọng số $Q / K / V$ riêng biệt cho mỗi đầu. Như đã giới thiệu, ta nhân X với ma trận $W_Q / W_K / W_V$ để tạo ra các ma trận $Q / K / V$.



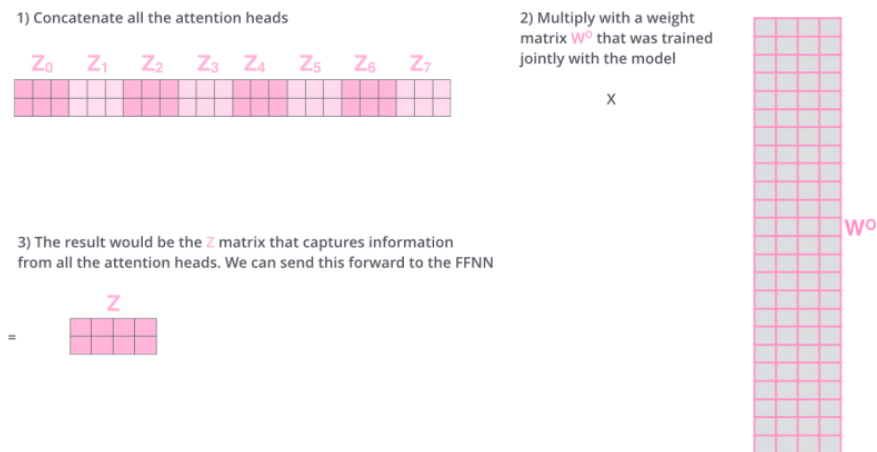
Hình 1.18 Minh họa hai đầu head #0 và #1 khi thực hiện Multi-head attention

- **Bước 2:** Nếu ta thực hiện self-attention như đã vạch ra bên trên, với 8 lần (như trong bài báo gốc) tính toán với các ma trận khác nhau, ta có 8 ma trận Z khác nhau.



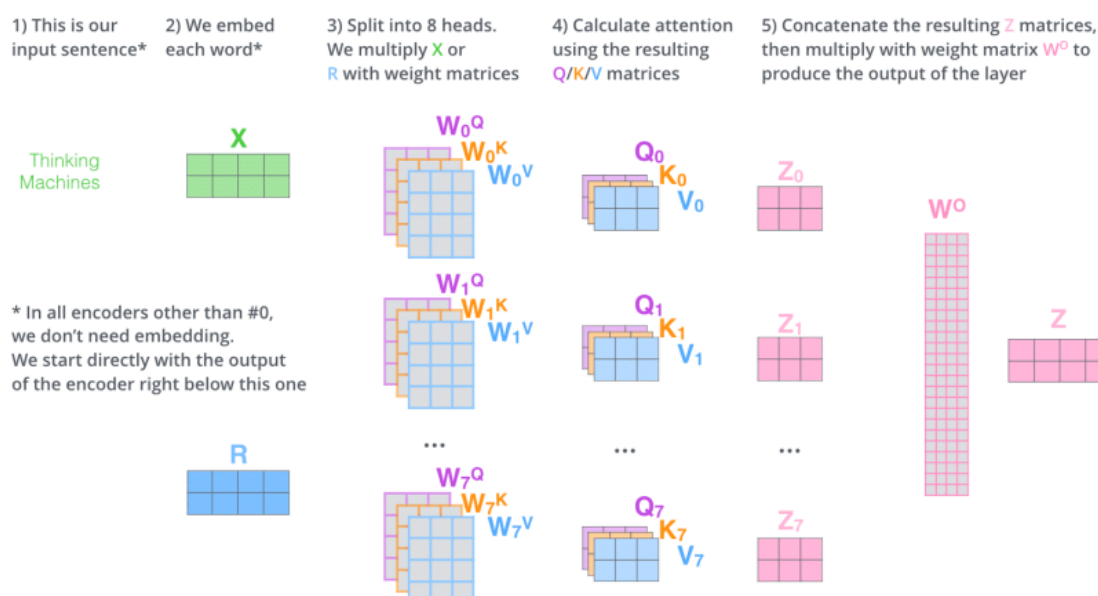
Hình 1.19 Tám đầu ra sau khi tính toán Multi-head attention

- Bước 3: biến đổi 8 ma trận về 1 ma trận duy nhất bằng cách nối các ma trận lại và nhân chúng với một ma trận trọng số được bổ sung W^O .



Hình 1.20 Nối tám ma trận Z_i và nhân với ma trận trọng số W^O để tạo 1 ma trận đầu ra duy nhất cho bước tính toán Multi-head attention

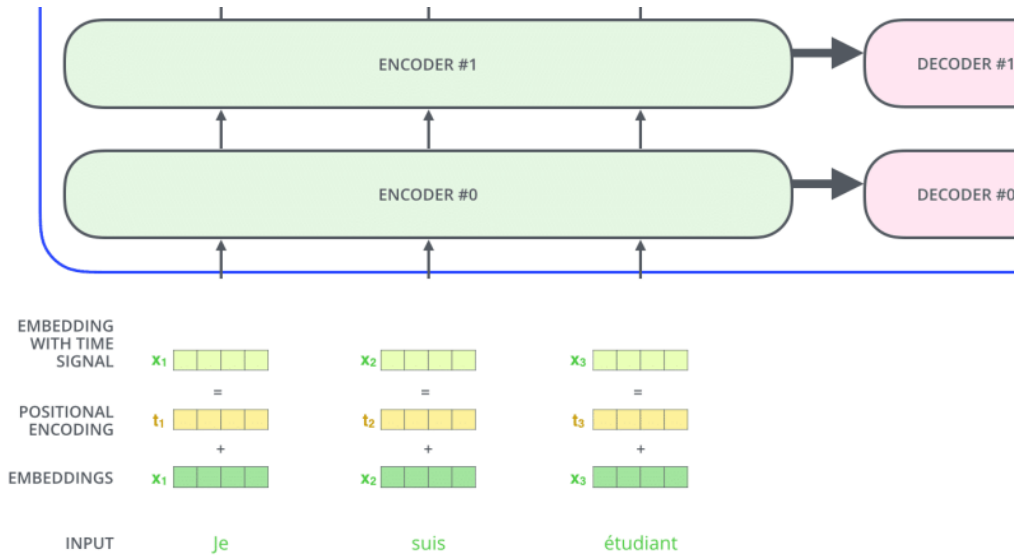
Toàn bộ 3 bước kể trên được mô tả đầy đủ như trong Hình 1.21.



Hình 1.21 Toàn bộ quá trình tính toán Multi-head attention

1.4.4. Biểu diễn thứ tự trong chuỗi với Positional Encoding

Một điều chưa được đề cập đến trong mô hình của chúng ta là cách xử lý thứ tự của các từ trong chuỗi đầu vào. Để giải quyết vấn đề này, transformer thêm một vector vào mỗi embedding đầu vào. Các véc tơ này tuân theo một mẫu cố định mà mô hình học được, giúp nó xác định vị trí của từng từ hoặc khoảng cách của các từ khác nhau trong chuỗi. Ý tưởng ở đây là việc thêm các giá trị đó sẽ cung cấp thông tin về khoảng cách giữa các vector embedding khi chúng được phản ánh thông qua các vector $Q / K / V$ và thông qua phép lấy tích vô hướng.



Hình 1.22 Vector mã hóa vị trí (positional encoding) trong mô hình transformer

Trong bài báo gốc được giới thiệu bởi (Vaswani, et al., 2017), công thức tính toán vector mã hóa vị trí của mô hình transformer được mô tả bởi hai phương trình sau:

$$P(k, 2i) = \sin\left(\frac{k}{n^{\frac{2i}{d}}}\right)$$

$$P(k, 2i + 1) = \cos\left(\frac{k}{n^{\frac{2i}{d}}}\right)$$

Trong đó:

- k: vị trí của từ trong chuỗi đầu vào
- d: số chiều của vector embedding
- n: một số thực được định nghĩa trước. Trong bài báo gốc, $n = 10000$.
- i: được sử dụng để liên kết tới index của cột. Giá trị của i nằm trong đoạn $[0, d/2)$.
- $P(k, j)$: hàm vị trí kết nối vị trí của từ thứ k trong câu với ma trận vị trí

Sequence	Index of token, k	Positional Encoding Matrix with $d=4, n=100$			
		$i=0$	$i=0$	$i=1$	$i=1$
I	0	$P_{00}=\sin(0)$ = 0	$P_{01}=\cos(0)$ = 1	$P_{02}=\sin(0)$ = 0	$P_{03}=\cos(0)$ = 1
am	1	$P_{10}=\sin(1/1)$ = 0.84	$P_{11}=\cos(1/1)$ = 0.54	$P_{12}=\sin(1/10)$ = 0.10	$P_{13}=\cos(1/10)$ = 1.0
a	2	$P_{20}=\sin(2/1)$ = 0.91	$P_{21}=\cos(2/1)$ = -0.42	$P_{22}=\sin(2/10)$ = 0.20	$P_{23}=\cos(2/10)$ = 0.98
Robot	3	$P_{30}=\sin(3/1)$ = 0.14	$P_{31}=\cos(3/1)$ = -0.99	$P_{32}=\sin(3/10)$ = 0.30	$P_{33}=\cos(3/10)$ = 0.96

Positional Encoding Matrix for the sequence 'I am a robot'

Hình 1.23 Ví dụ tính toán ma trận vị trí với $d = 4$ và $n = 100$

1.5. BERT

1.5.1. BERT là gì

BERT là viết tắt của Bidirectional Encoder Representations from Transformers là một mô hình học sâu, học ra các véc tơ đại diện theo ngữ cảnh 2 chiều của từ, được sử dụng để điều chỉnh sang các bài toán khác trong lĩnh vực xử lý ngôn ngữ tự nhiên. BERT đã thành công trong việc cải thiện những công việc gần đây trong việc tìm ra đại diện của từ trong không gian số (không gian mà máy tính có thể hiểu được) thông qua ngữ cảnh của nó.

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 pre-trained BERT đã kéo theo sự cải tiến đáng kể cho rất nhiều bài toán như Question Answering, Sentiment Analysis, ...

Cải tiến quan trọng của BERT chính là việc áp dụng huấn luyện hai chiều của Transformer vào mô hình ngôn ngữ. Khác với các mô hình directional (các mô hình chỉ đọc dữ liệu theo một chiều duy nhất trái sang phải hoặc phải sang trái)

đọc dữ liệu theo dạng tuần tự, Encoder đọc toàn bộ dữ liệu trong một lần, việc này làm cho BERT có khả năng huấn luyện dữ liệu theo cả hai chiều, qua đó mô hình có thể học được ngữ cảnh (context) của từ tốt hơn bằng cách sử dụng những từ xung quanh nó (phải - trái).

1.5.2. Sự ra đời của BERT

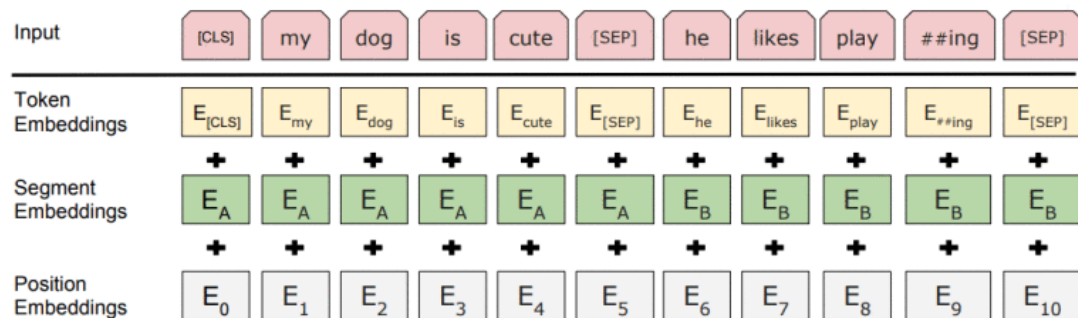
Một trong những thách thức lớn nhất của NLP là vấn đề dữ liệu. Trên internet có hàng tá dữ liệu, nhưng những dữ liệu đó không đồng nhất; mỗi phần của nó chỉ được dùng cho một mục đích riêng biệt, do đó khi giải quyết một bài toán cụ thể, ta cần trích ra một bộ dữ liệu thích hợp cho bài toán của mình,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

1.5.3. Nền tảng của BERT

BERT dựa trên transformer (cơ chế attention học các mối quan hệ theo ngữ cảnh giữa các từ trong văn bản). Một transformer cơ bản bao gồm một bộ mã hóa (encoder) để đọc đầu vào văn bản và một bộ giải mã (decoder) để đưa ra dự đoán cho nhiệm vụ. Vì mục tiêu của BERT là tạo mô hình biểu diễn ngôn ngữ, nên nó chỉ cần phần bộ mã hóa. Đầu vào cho bộ mã hóa cho BERT là một chuỗi token, trước tiên được chuyển đổi thành vector và sau đó được xử lý trong mạng neural.

Tuy nhiên, trước khi quá trình xử lý có thể bắt đầu, BERT cần đầu vào được tiền xử lý bằng một số siêu dữ liệu bổ sung:

- **Token embeddings:** Một token [CLS] được thêm vào chuỗi token đầu vào tại vị trí đầu tiên của chuỗi và một token [SEP] được thêm vào tại vị trí cuối cùng của chuỗi.
- **Segment embeddings:** Một điểm đánh dấu cho biết Câu A hoặc Câu B được thêm vào mỗi token. Điều này cho phép bộ mã hóa phân biệt giữa các câu.
- **Positional embeddings:** được thêm vào mỗi token để chỉ ra vị trí của nó trong câu.



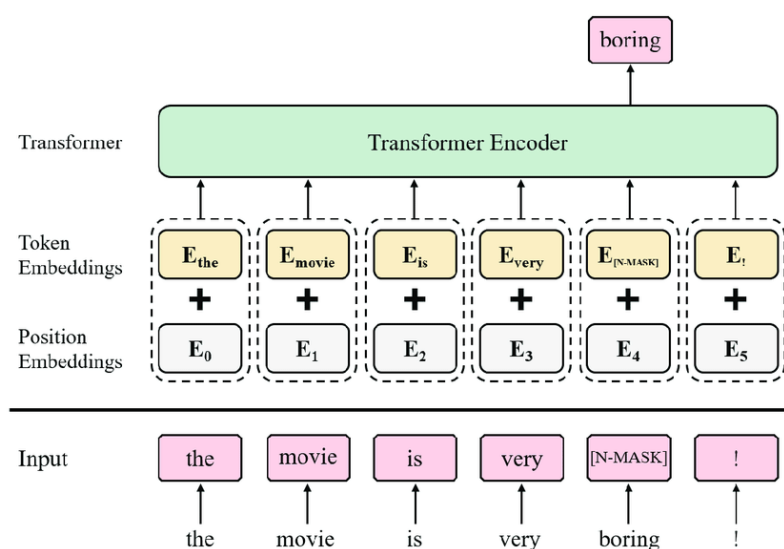
Hình 1.24 Minh họa token embeddings, segment embeddings và position embeddings trong mô hình BERT

BERT được huấn luyện đồng thời hai 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 mất mát (loss) tổng sẽ là kết hợp mất mát của cả hai nhiệm vụ và mô hình sẽ cố gắng tối giản tổng mất mát này. Chi tiết hai nhiệm vụ này như sau:

- **Masked Language Model (MLM):** Với nhiệm vụ này, việc huấn luyện mô hình BERT sẽ thực hiện che đi tầm 15% số từ trong câu và đưa vào mô hình. Và ta sẽ huấn luyện để mô hình dự đoán ra các từ bị che đó dựa vào các từ còn lại. Cụ thể là:

- Thêm một lớp classification lên trên đầu ra của encoder

- Đưa các vector trong đầu ra của encoder về vector bằng với tổng số từ có trong từ điển (vocab size), sau đó qua hàm softmax để chọn ra từ tương ứng tại mỗi vị trí trong câu.
- Mất mát (loss) sẽ được tính tại vị trí che và bỏ qua các vị trí khác (để đánh giá xem model dự đoán từ bị che đúng sai như thế nào).



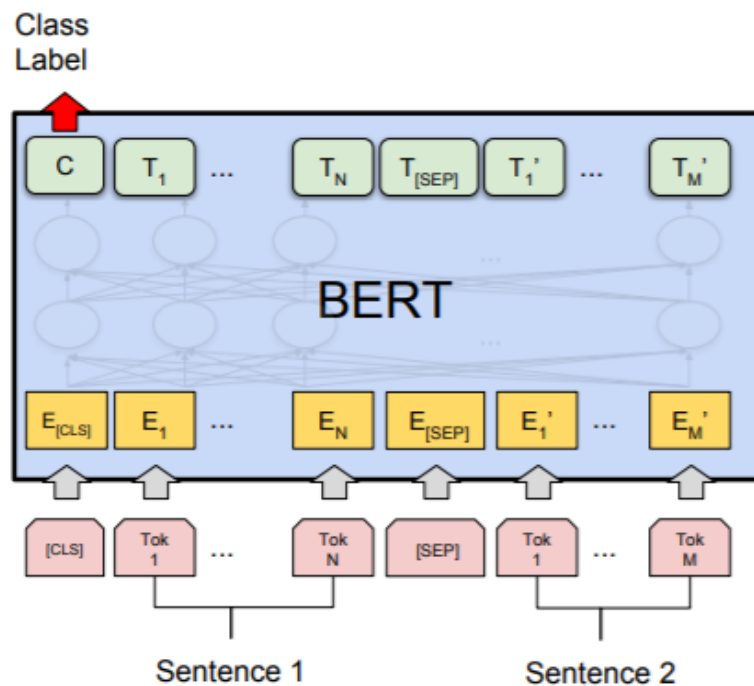
Hình 1.25 Minh họa quá trình huấn luyện BERT cho bài toán MLM

- **Next Sentence Prediction (NSP):** Với nhiệm vụ này thì mô hình sẽ được cung cấp cho một cặp câu và nhiệm vụ của nó là dự đoán ra giá trị 1 nếu câu thứ hai đúng là câu đi sau câu thứ nhất và 0 nếu không phải. Trong quá trình huấn luyện, ta chọn 50% mẫu là Positive (output là 1) và 50% còn lại là Negative được ghép linh tinh (output là 0). Cụ thể cách huấn luyện như sau:

- Bước 1: Ghép 2 câu vào nhau và thêm 1 số token đặc biệt để phân tách các câu. Token [CLS] thêm vào đầu câu thứ nhất, token [SEP] thêm vào cuối mỗi câu. Ví dụ ghép 2 câu “Hôm nay em đi học” và “Học ở trường rất hay” thì sẽ thành [CLS] Hôm nay em đi học [SEP] Học ở trường rất vui [SEP]
- Bước 2. Mỗi token trong câu sẽ được cộng thêm một vector gọi là Segment Embedding, thực ra là đánh dấu xem từ đó thuộc câu Thứ nhất hay câu thứ 2 thôi. Ví dụ nếu thuộc câu Thứ nhất thì cộng thêm

1 vector toàn số “0” có kích thước bằng chiều của vector embedding, 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 Positional Embedding vào để đánh dấu vị trí từng từ trong câu đã ghép.
- Bước 4. Đưa chuỗi sau bước 3 vào mạng.
- Bước 5. Lấy đầu ra của encoder tại vị trí token [CLS] được chuyển đổi sang một vector có 2 phần tử [c1 c2].
- Bước 6. Tính softmax trên vector đó và đầu ra là xác suất của hai class: 1 - “Đi sau” và 0 - “Không đi sau” (nghĩa là câu thứ hai có đi sau câu thứ nhất hay không)



Hình 1.26 Minh họa quá trình huấn luyện BERT cho bài toán Next Sentence Prediction

1.6. PhoBERT

Đây là một pre-trained model được huấn luyện cho đơn ngôn ngữ

(monolingual language), tức là chỉ huấn luyện dành riêng cho tiếng Việt. Việc huấn luyện dựa trên kiến trúc và cách tiếp cận giống RoBERTa của Facebook được Facebook giới thiệu giữa năm 2019, nghĩa là PhoBERT chỉ sử dụng nhiệm vụ Masked Language Model để huấn luyện, bỏ đi nhiệm vụ Next Sentence Prediction.

PhoBERT được huấn luyện trên khoảng 20 GB dữ liệu bao gồm khoảng 1GB Vietnamese Wikipedia corpus và 19 GB còn lại lấy từ Vietnamese news corpus. Đây là một lượng dữ liệu khá ổn để train một mô hình như BERT.

PhoBERT sử dụng RDRSegmenter của VnCoreNLP để tách từ cho dữ liệu đầu vào trước khi qua BPE encoder.

Về kiến trúc thì PhoBERT có hai mô hình là PhoBERTbase và PhoBERTlarge có kiến trúc tương tự như hai mô hình BERTbase và BERTlarge. Tổng số tham số cho mô hình PhoBERTbase là 135M và PhoBERTlarge là 370M.

1.7. Cách gán nhãn thực thể có tên

Nhãn thực thể được gán theo cấu trúc BIO (beginning - inside - outside). Trong lĩnh vực bất động sản chúng tôi định nghĩa có tất cả 51 nhãn tất cả, bao gồm B-transaction và I-transaction cho loại giao dịch, B-real_estate_type và I-real_estate_type cho mã loại bất động sản, B-price và I-price cho giá, v.v... Bảng liệt kê chi tiết các nhãn được mô tả như bên dưới.

#	Tên chung	Nhãn	Diễn giải	Ví dụ
1		O		
2	transaction	B-transaction	Loại giao dịch	bán, cho thuê, thuê, mua, nhượng
		I-transaction		
3	real_estate_type	B-real_estate_type	Mã loại bất động sản	căn hộ, nhà, nhà phố, đất, đất thổ cư, biệt thự
		I-real_estate_type		

4	real_estate_sub_type	B-real_estate_sub_type	Mã loại bất động sản phụ	chung cư, ShopHouse, PenHouse, Officetel
		I-real_estate_sub_type		
5	price	B-price	Giá	3,5 tỷ, 2.5 tỷ, 200 triệu, 500tr, 30 tr / tháng
		I-price		
6	area	B-area	Diện tích	100m2, 200 m2, 100 ha
		I-area		
7	direction	B-direction	Hướng	Bắc, Tây Bắc, ...
		I-direction		
8	street	B-street	Đường	Nguyễn Thị Minh Khai, CMT8, ...
		I-street		
9	ward	B-ward	Phường / xã	Bình Hòa, Tân Đông Hiệp,...
		I-ward		
10	district	B-district	Quận / huyện	Nhà Bè, Bình Chánh, Q1, ...
		I-district		
11	city	B-city	Thành phố / tỉnh	Bình Dương, ...
		I-city		
12	email	B-email	email	abc@gmail.com
		I-email		

13	phone	B-phone	Điện thoại	0938038621, 0938 038 621, 0938-038-621, 0938.038.621
		I-phone		
14	usage	B-usage	Mục đích sử dụng	để ở, kinh doanh, mở phòng mạch, ...
		I-usage		
15	floor	B-floor	kết cấu tầng / lầu, số tầng / lầu	1 trệt, 2 lầu, 1 sân thượng
		I-floor		
16	bath_room	B-bath_room	phòng tắm / toilet	3 toilet, 2 phòng tắm, ...
		I-bath_room		
17	living_room	B-living_room	phòng khách	2 PK, 1 phòng khách
		I-living_room		
18	bed_room	B-bed_room	phòng ngủ	1 PN, 2 phòng ngủ, ...
		I-bed_room		
19	position	B-position	Vị trí	Mặt tiền, hẻm, ngõ
		I-position		
20	author	B-author	Người đăng tin	Mr Minh. LH: A. Nam: 0987 396 990.
		I-author		
21	house_number	B- house_number	Số địa chỉ	243 Bình Hòa 08 quốc lộ 13, Bình Dương
		I- house_number		
22	project_name	B-project_name	Tên dự án, tên chung cư	Park Hill Times City, chung cư Becamex.
		I-project_name		

23	front_length	B-front_length	Bề rộng mặt tiền	nhà phố mặt tiền 5m
		I-front_length		
24	road_width	B-road_width	Bề rộng đường	đường rộng 5m
		I-road_width		
25	surrounding	B-surrounding	Tiện ích xung quanh	Trường học, chợ, bệnh viện, công viên
		I-surrounding		
26	legal	B-legal	Thông tin pháp lý	Sổ đỏ chính chủ, sổ hồng
		I-legal		

Bảng 1.1 Nhãn các thực thể theo cấu trúc BIO

1.8. Chỉ số đánh giá hệ thống

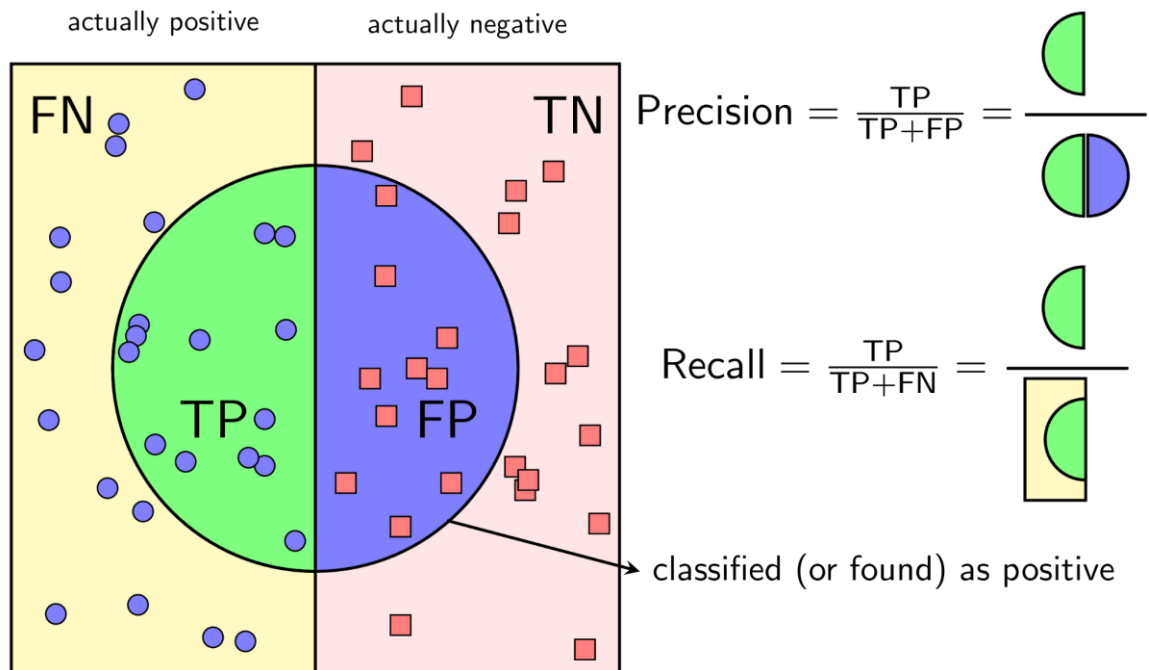
Để đo lường hiệu quả của hệ thống hoặc so sánh hệ thống này với hệ thống kia thì ta cần sử dụng các chỉ số đánh giá hệ thống. Hệ thống NER thường được đánh giá bằng cách so sánh kết quả đầu ra của chúng với chú thích hay nhãn được gán bởi con người. Các chỉ số thường được sử dụng để đo lường hiệu quả của một hệ thống NER là độ chính xác (accuracy), độ chuẩn xác (precision), độ phủ (recall) và điểm f1 (f1-score).

Để hiểu rõ các chỉ số trên, trước hết chúng ta cần nắm rõ các khái niệm cơ bản sau đây:

- True Positive (TP): số lượng điểm của lớp *positive* được phân loại đúng là *positive*.
- True Negative (TN): số lượng điểm của lớp *negative* được phân loại đúng là *negative*.
- False Positive (FP): số lượng điểm của lớp *negative* bị phân loại

nhầm thành *positive*.

- False Negative (FN): số lượng điểm của lớp *positive* bị phân loại nhầm thành *negative*



Hình 1.27 Cách tính Precision và Recall

Độ chính xác (accuracy): Khi xây dựng mô hình phân loại chúng ta sẽ muốn biết một cách khái quát tỷ lệ các trường hợp được dự báo đúng trên tổng số các trường hợp là bao nhiêu. Tỷ lệ đó được gọi là độ chính xác và sẽ được tính như sau:

$$Accuracy = \frac{TP + TN}{\text{tổng số mẫu}}$$

Độ chuẩn xác (precision) trả lời cho câu hỏi trong các trường hợp được dự báo là positive thì có bao nhiêu trường hợp là đúng? Và tất nhiên độ chuẩn xác càng cao thì mô hình của chúng ta càng tốt. Độ chuẩn xác sẽ được tính như sau:

$$Precision = \frac{TP}{\text{tổng số mẫu dự đoán là positive}} = \frac{TP}{TP + FP}$$

Độ phủ (recall) đo lường tỷ lệ dự báo chính xác các trường hợp positive trên

toàn bộ các mẫu thuộc nhóm positive. Công thức tính độ phủ như sau:

$$Recall = \frac{TP}{\text{tổng số mẫu thực sự là positive}} = \frac{TP}{TP + FN}$$

Điểm f1 (f1-score) là trung bình điều hòa giữa *độ chuẩn xác* và *độ phủ*, sẽ được tính như sau:

$$f1 - score = \frac{2}{\frac{1}{precision} + \frac{1}{recall}}$$

Chương 2. CÁC CÔNG TRÌNH LIÊN QUAN

Đối với các kỹ thuật được áp dụng trong NER, có bốn phương pháp chính:

- Phương pháp tiếp cận dựa trên quy tắc (rule-based approach) không cần dữ liệu đã được chú thích hay gán nhãn vì chúng dựa trên các quy tắc được tạo thủ công, thường là xài regex (regular expression) để bắt các đặc trưng dựa trên các quy luật đã được định nghĩa trước.
- Phương pháp tiếp cận học tập không giám sát (unsupervised learning approach) dựa trên các thuật toán không giám sát mà không có các dữ liệu huấn luyện được gán nhãn thủ công.
- Phương pháp tiếp cận học tập có giám sát dựa trên đặc trưng (feature-based supervised learning approach).
- Phương pháp tiếp cận dựa trên mạng nơ-ron học sâu tự động khám phá các biểu diễn cần thiết cho việc phân loại và phát hiện thực thể từ các văn bản hay câu văn thô.

Hệ thống NER đã được nghiên cứu và phát triển rộng rãi trong nhiều thập kỷ, nhưng các hệ thống chính xác sử dụng mạng nơ-ron học sâu (DNN - deep neural network) như trong nghiên cứu (Lample, Ballesteros, Subramanian, Kawakami, & Dyer, 2016) mới chỉ được giới thiệu trong khoảng mười lăm năm gần đây. Trong những năm gần đây, các mô hình NER dựa trên mạng nơ-ron học sâu trở nên chiếm ưu thế và đạt được những kết quả tiên tiến nhất. So với các phương pháp tiếp cận dựa trên đặc trưng (feature-based), mô hình học sâu có lợi trong việc tự động khám phá các đặc trưng ẩn.

Trong nghiên cứu của mình, từ những phương pháp đã nêu ở trên tôi sử dụng các phương pháp sau cho bài toán nhận dạng thực thể có tên là:

- Phương pháp tiếp cận dựa trên quy tắc (rule-based approach): Cụ thể là xài regex để nhận dạng các thực thể dựa trên các bộ quy tắc đã được định nghĩa trước, gán phần lớn nhãn như email, số điện thoại, diện tích, v.v.. cho tập dữ liệu

huấn luyện và kiểm thử. Sau đó, các nhãn đã được gán sẽ được kiểm tra lại thủ công và sửa lại cho đúng. Nhờ ứng dụng phương pháp này mà thời gian gán nhãn cho tập dữ liệu được rút ngắn đáng kể và công đoạn gán nhãn trở nên đơn giản hơn.

- Phương pháp tiếp cận dựa trên mạng nơ-ron học sâu: sau khi đã có bộ dữ liệu huấn luyện và kiểm thử hoàn hảo, mạng nơ-ron học sâu được áp dụng để huấn luyện và đánh giá hệ thống nhận dạng thực thể có tên. Đó chính là BERT.

2.1. Phương pháp tiếp cận dựa trên quy tắc (rule-based approach)

Phương pháp tiếp cận dựa trên quy tắc không cần dữ liệu đã được chú thích hay gán nhãn vì chúng dựa trên các quy tắc được tạo thủ công để bắt các đặc trưng dựa trên các quy luật đã được định nghĩa trước. Phương pháp tiếp cận dựa trên quy tắc trong nhận dạng thực thể có tên hoạt động như sau: một tập các quy tắc / quy luật được định nghĩa sẵn hay tự động phát sinh. Mỗi token trong văn bản sẽ được biểu diễn dưới dạng tập các đặc trưng. Văn bản đầu vào sẽ đem so sánh với tập quy tắc này, nếu quy tắc khớp thì sẽ thực hiện rút trích. Một quy tắc như vậy gồm khuôn mẫu (pattern) cộng hành động (action). Khuôn mẫu thường là regular expression định nghĩa trên tập đặc trưng của token. Khi Khuôn mẫu này khớp thì hành động sẽ được kích hoạt. Chúng ta có thể tự lập trình các quy tắc của mình hoặc sử dụng một số thư viện hỗ trợ sẵn. Một trong những framework/thư viện khá nổi tiếng là Duckling của Facebook.

2.2. Phương pháp mạng neural học sâu

(Collobert & Weston, 2008) đã đề xuất một trong những kiến trúc mạng nơ-ron đầu tiên cho NER, với các vector đặc trưng được xây dựng từ các đặc trưng trực quan (ví dụ: viết hoa của ký tự đầu tiên), từ điển và từ vựng. Không lâu sau đó, (Collobert, et al., 2011) đã thay thế các vector đặc trưng được xây dựng theo cách thủ công này bằng các phép nhúng từ, là các biểu diễn của các từ trong không

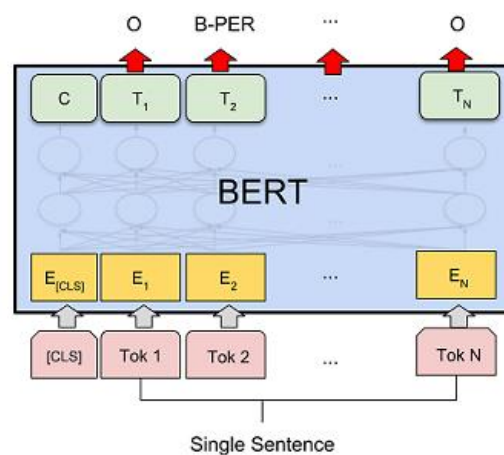
gian n chiều, thường được học qua các bộ sưu tập dữ liệu lớn không được gán nhãn thông qua một quy trình không giám sát như mô hình skip-gram. Các nghiên cứu đã chỉ ra tầm quan trọng to lớn của việc nhúng từ được huấn luyện trước đối với các hệ thống NER dựa trên mạng nơ-ron (Habibi, Weber, Neves, Wiegandt, & Leser, 2017), và tương tự đối với nhúng ký tự được huấn luyện trước trong các ngôn ngữ dựa trên ký tự như tiếng Trung Quốc (Li, Li, Sun, & Li, 2015).

Kiến trúc mạng nơ-ron học sâu hiện đại cho NER có thể được phân loại tùy thuộc vào biểu diễn của các từ trong một câu. Ví dụ: các biểu diễn có thể dựa trên các từ, ký tự, các đơn vị từ phụ khác hoặc bất kỳ sự kết hợp nào của những từ này.

2.3. Phương pháp BERT fine-tune

Đối với các nhiệm vụ phân loại câu, BERT được fine-tuning rất đơn giản. Để có được biểu diễn của một chuỗi đầu vào với số chiều cố định, chúng ta chỉ cần lấy hidden state ở lớp cuối cùng, tức là đầu ra của lớp Transformer cho token đầu tiên (token đặc biệt [CLS] được xây dựng cho đầu chuỗi).

Trong mô hình này, các chuỗi câu đầu vào sẽ được chuyển thành chuỗi các tokens mà mô hình BERT có thể hiểu được, rồi qua mạng.



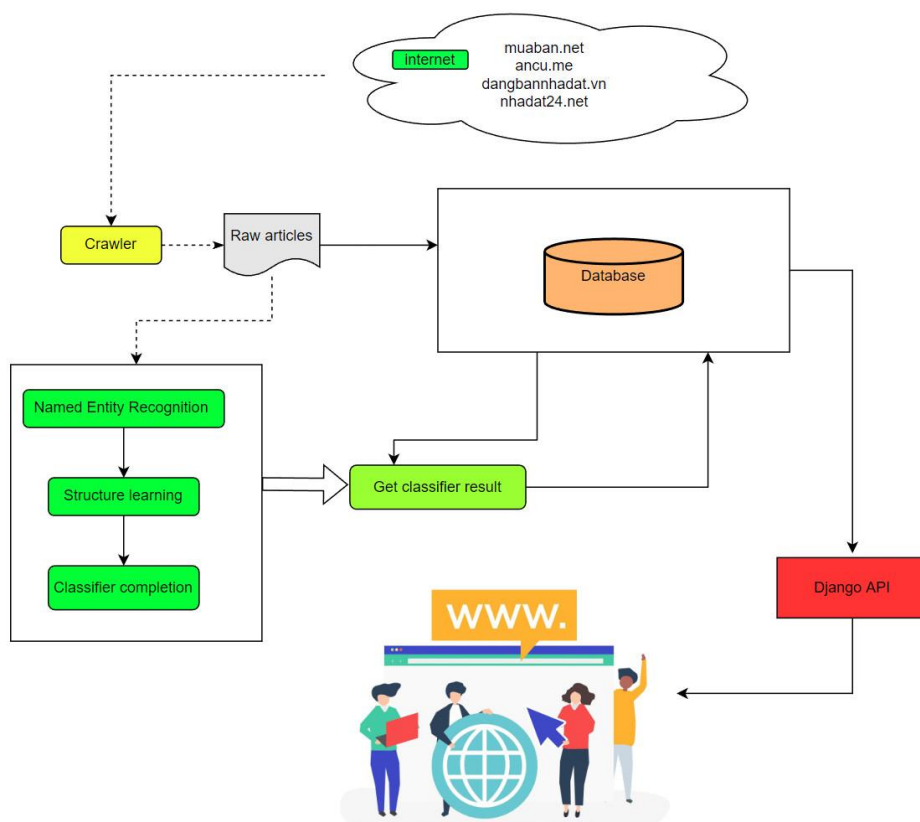
Hình 2.1 Phương pháp BERT fine-tune

Chương 3. MÔ TẢ HỆ THỐNG

Mô tả và phân tích kỹ kiến trúc cũng như chức năng các thành phần của một hệ thống sẽ giúp chúng ta tiết kiệm thời gian và công sức để hiện thực hệ thống. Trong chương này, tôi sẽ đề cập đến việc đặc tả các thành phần trong bài toán nhận dạng thực thể có tên với tập dữ liệu về bất động sản cũng như mối liên hệ giữa chúng.

Tổng quan về kiến trúc và chức năng

Một hệ thống truy xuất thông tin dữ liệu dùng cho nhu cầu tìm kiếm thông tin về bất động sản cần có nhiều thành phần khác nhau, mỗi thành phần đảm nhiệm một vai trò nhất định. Các thành phần và mối liên hệ giữa chúng được mô tả ở Hình 3.1.



Hình 3.1 Kiến trúc tổng quát của hệ thống bài toán

Các chức năng của mỗi thành phần:

- **Crawler:** Thu thập các bài đăng rao bán bất động sản từ internet để làm nguồn dữ liệu tư vấn cho hệ thống. Tính tới thời điểm viết luận văn, hiện tại hệ thống đã và đang cào dữ liệu từ bốn website sau:

- muaban.net
- ancu.me
- dangbannhadat.vn
- nhadat24h.net

- **Entity Extractor:** rút trích các thông tin cần thiết từ một bài đăng rao bán bất động sản hoặc từ một yêu cầu mà người dùng nhập vào trong quá trình tương tác với hệ thống (Ví dụ: giá 10 tỷ, diện tích 20m²...).

- **Named Entity Recognition:** Tiến hành gán nhãn cho dữ liệu đã được crawler về từ các trang web bất động sản theo những nhãn đã được khai báo ở mục 1.7 chương I, việc gán nhãn cho từng dòng dữ liệu được tiến hành thủ công.

- **Structure learning, classifier:** Xác định mô hình và tiền xử lý theo mô hình BERT với tập dữ liệu đầu vào là các mẫu dữ liệu đã được gán nhãn ở bước trên.

- **Classifier Result:** kết quả sau khi thực hiện training theo mô hình BERT. Với kết quả có được này chúng ta sẽ thực hiện gán nhãn cho những dữ liệu thô đã cào về được và lưu lại vào database.

- **Database:** Lưu trữ các thông tin đã chuẩn hóa mà Crawler thu thập được, để dễ dàng truy xuất thông tin khi thực hiện truy vấn cho người dùng.

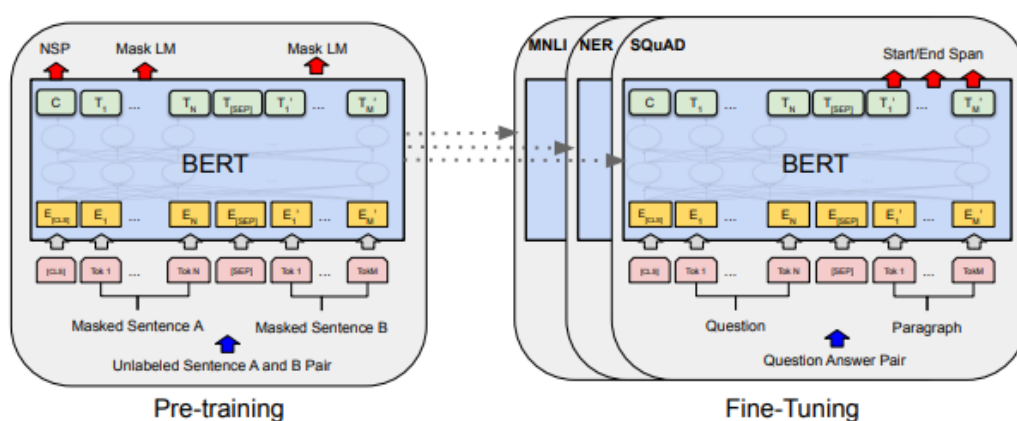
- **Django Api:** Là hệ thống Backend kết dùng để truy xuất thông tin từ hệ thống database đã được crawler từ database.

- **Front-end:** Giao diện hiển thị thông tin dữ liệu truy xuất từ api cho phép người dùng có thể thao tác.

Chương 4. PHƯƠNG PHÁP NGHIÊN CỨU

4.1. Sử dụng PhoBERT để huấn luyện

Một điểm đặc biệt ở BERT mà các mô hình nhúng từ trước đây chưa từng có đó là kết quả huấn luyện có thể fine-tuning được. Nghĩa là chúng ta có thể thêm vào kiến trúc mô hình một lớp đầu ra để tùy biến theo tác vụ huấn luyện.



Hình 4.1 Toàn bộ tiến trình pre-training và fine-tuning của BERT

Toàn bộ tiến trình pre-training và fine-tuning của BERT:

- **Bước 1:** Embedding toàn bộ các token của cặp câu bằng các véc-tơ embedding tiền huấn luyện mô hình. Các token embedding bao gồm cả 2 token là [CLS] và [SEP] để đánh dấu vị trí bắt đầu của câu hỏi và vị trí ngăn cách giữa 2 câu. 2 token này sẽ được dự báo ở output để xác định các phần **Start/End Span** của câu output.
- **Bước 2:** Các embedding vector sau đó sẽ được truyền vào kiến trúc fine-tune với nhiều block code (thường là 6, 12 hoặc 24 blocks tùy theo kiến trúc BERT). Ta thu được một vector output ở encoder.
- **Bước 3:** Để dự báo phân phối xác suất cho từng vị trí từ ở decoder, ở mỗi time step chúng ta sẽ truyền vào decoder vector output của encoder và vector embedding input của decoder để tính encoder-decoder attention. Sau đó projection

qua liner layer và softmax để thu được phân phối xác suất cho output tương ứng ở time step t.

- **Bước 4:** Trong kết quả trả ra ở output của Transformer ta sẽ hiển thị thông tin lên cho người dùng, tương ứng với dữ liệu từ câu input.

4.2. Minh họa sử dụng thực tế

Dưới đây là ví dụ về việc sử dụng PhoBERT để bóc tách và gán nhãn một mẫu dữ liệu với mẫu dữ liệu đầu vào như sau:

Raw data	ĐIJA THẾ THỊNH VƯƠNG - ĐIỂM VÀNG KẾT NỐI Richland Residence tọa lạc tại vị trí đắc địa, gần nút giao Vành đai 4 (quy mô 6-8 làn xe, rộng 74,5m) và DT 741, liền kề Thành phố mới Bình Dương. Kết nối trực tiếp với những trục đường huyết mạch: Quốc lộ 13, Quốc lộ 14, đại lộ Mỹ Phước - Tân Vạn, DT 742, DT 744 dễ dàng di chuyển đến các khu vực trọng điểm tại TP. HCM, Bình Dương, Đồng Nai và các đô thị lân cận. Sắp tới tuyến Metro Bến Thành - Suối Tiên kéo dài đến thành phố mới Bình Dương và Bến Cát, giúp lưu thông càng thuận tiện Với vị trí vàng, cư dân Richland Residence còn hưởng trọn loạt tiện ích quanh khu vực chỉ trong vài phút kết nối: UBND phường Hòa Lợi, trường THCS Hòa Lợi, Đại học Quốc tế Việt Đức, Trung tâm Hành chính thành phố mới, chợ Chánh Lưu, chợ Hòa Lợi, bệnh viện Đa khoa Mỹ Phước, Trung tâm Thương mại thể thao Bình Dương (lớn nhất Việt Nam)... Đồng thời giá trị sản phẩm Richland Residence còn hứa hẹn tăng mạnh trong thời gian tới.Hotline : 0908.766.167 em Lộc-----RICHLAND RESIDENCEVun đắp giá trị - Kiến tạo phồn hoa#Richland #RichlandResidence #KimOanhGroup
----------	---

Hình 4.2 Mẫu dữ liệu đầu vào

Ta được kết quả như sau:

Label	Content
O	ĐỊA THỂ THỊNH VƯỢNG - ĐIỂM VÀNG KẾT NỐI Richland Residence tọa lạc tại vị trí đắc địa , gần nút giao
street	Vành đai 4
O	(quy mô 6-8 làn xe , rộng
area	74,5 m
O) và
street	ĐT 741
O	, liền kề Thành phố mới Bình Dương . Kết nối trực tiếp với những trục đường huyết mạch :
street	Quốc lộ 13
O	,
street	Quốc lộ 14
O	,
street	đại lộ Mỹ Phước - Tân Vạn
O	, ĐT
street	742
O	, ĐT
street	744
O	dễ dàng di chuyển đến các khu vực trọng điểm tại TP. HCM , Bình Dương , Đồng Nai và các đô thị lớn lân cận . Sắp tới tuyến
surrounding	Metro Bến Thành - Suối Tiên
O	kéo dài đến thành phố mới Bình Dương và Bến Cát , giúp lưu thông càng thuận tiện Với vị trí vàng , cư dân Richland Residence còn hưởng trọn loạt tiện ích quanh khu vực chỉ trong vài phút kết nối :
surrounding	UBND phường Hoà Lợi
O	,
surrounding	trường THCS Hoà Lợi
O	,
surrounding	Đại học Quốc tế Việt Đức
O	,
surrounding	Trung tâm Hành chính thành phố mới
O	,
surrounding	chợ Chánh Lưu
O	,
surrounding	chợ Hoà Lợi
O	,
surrounding	bệnh viện Đa khoa Mỹ Phước
O	, Trung tâm
surrounding	Thương mại thế giới Bình Dương
O	(lớn nhất Việt Nam) ... Đồng thời giá trị sản phẩm Richland Residence còn hứa hẹn tăng mạnh trong thời gian tới.Hotline :
phone	0908.766.167
O	em
author	Lộc
O	----- - RICHLAND RESIDENCEVun đắp giá trị - Kiến tạo phồn hoa # Richland # RichlandResidence # KimOanhGroup

Hình 4.3 Sử dụng mô hình PhoBERT để tiên đoán

Như kết quả ở trên, ta được một số nhãn như label “street” có nội dung là “Vành đai 4” và một số kết quả tương ứng với mỗi label.

Chương 5. CÁC CÔNG NGHỆ SỬ DỤNG

5.1. Ngôn ngữ lập trình

5.1.1. Python

Python là một ngôn ngữ lập trình bậc cao được Guido van Rossum tạo ra và lần đầu ra

mất vào năm 1991, được sử dụng rất thành công trên hàng ngàn ứng dụng thương mại trên thế giới, bao gồm cả những hệ thống rất lớn và quan trọng.

Python thuộc dạng ngôn ngữ thông dịch mang tính hướng đối tượng, với nhiều đặc điểm:

- Dễ đọc, dễ học và dễ nhớ với thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình.
- Quy trình phát triển phần mềm nhanh vì dòng lệnh được thông dịch thành mã máy và thực thi ngay lập tức.
- Rất phù hợp cho xử lý các bài toán Machine Learning vì việc hiện thực giải thuật được
- Thực hiện dễ dàng với số dòng lệnh rất ít, giúp cho người lập trình có thể tập trung vào việc thiết kế giải thuật thay vì những vấn đề thấp hơn như quản lý và cấp phát bộ nhớ.
- Có hệ thống thư viện hỗ trợ phong phú, hỗ trợ mạnh về tính toán trong lĩnh vực học máy như Numpy, Pandas, Sklearn, TensorFlow, Keras...



Hình 5.1 Các thư viện Machine Learning nổi tiếng có hỗ trợ Python

Trong quá trình thực hiện đề tài này, nhóm tôi sử dụng phiên bản Python 3.8 là phiên bản được đánh giá hoạt động ổn định ở thời điểm hiện tại để hiện thực các API.Server phục vụ cho việc quản lí tương tác hội thoại.

5.1.2. Javascript, HTML & CSS

Javascript là ngôn ngữ lập trình được phát triển bởi Brendan Eich tại Hãng truyền thông

Netscape từ năm 1995 với cái tên đầu tiên Mocha, rồi sau đó đổi tên thành LiveScript, và cuối cùng thành JavaScript. Giống Java, JavaScript có cú pháp tương tự C, nhưng nó gần với Self hơn Java. ".js" là phần mở rộng thường được dùng cho tập tin mã nguồn JavaScript.



Hình 5.2 Xây dựng giao diện chatbot bằng ngôn ngữ HTML / CSS / JS

5.2. Thư viện - Framework

Trong quá trình giải quyết các vấn đề trong lĩnh vực học máy thì ta không thể không nhắc đến sự hỗ trợ của các thư viện lập trình. Chúng giúp chúng ta tập trung xử lý vấn đề ở mức trừu tượng cao mà không phải quan tâm đến phần hiển thị thực cấp thấp. Trong mục này, tôi sẽ giới thiệu một số thư viện hỗ trợ đã được sử dụng trong quá trình làm luận văn.

5.2.1. Scrapy

Scrapy là một thư viện Python được tạo ra để quét và xây dựng các trình thu thập dữ liệu web. Nó nhanh chóng, đơn giản và có thể điều hướng qua nhiều trang web mà không mất nhiều công sức. Scrapy có sẵn thông qua thư viện Pip Installs Python (PIP). Lợi thế của scrapy không chỉ ở việc hỗ trợ sẵn các hàm thư viện, mà nó còn đặc biệt ở chỗ nó định nghĩa luôn cả quy trình cũng như kiến trúc để lấy dữ liệu về.

5.2.2. Django

Django là 1 web framework khá nổi tiếng được viết hoàn toàn bằng ngôn

ngữ Python. Nó là 1 framework với đầy đủ các thư viện, module hỗ trợ các web-developer. Django sử dụng mô hình MVC và được phát triển bởi Django Software Foundation (DSF một tổ chức phi lợi nhuận độc lập) Mục tiêu chính của Django là đơn giản hóa việc tạo các website phức tạp có sử dụng cơ sở dữ liệu. Django tập trung vào tính năng “có thể tái sử dụng” và “có thể tự chạy” của các component, tính năng phát triển nhanh, không làm lại những gì đã làm. Một số website phổ biến được xây dựng từ Django là Pinterest, Instagram, Mozilla, và Bitbucket.

5.2.3. Nodejs

NodeJS là một môi trường runtime chạy JavaScript đa nền tảng và có mã nguồn mở, được sử dụng để chạy các ứng dụng web bên ngoài trình duyệt của client. Nền tảng này được phát triển bởi Ryan Dahl vào năm 2009, được xem là một giải pháp hoàn hảo cho các ứng dụng sử dụng nhiều dữ liệu nhờ vào mô hình hướng sự kiện (event-driven) không đồng bộ.

5.2.4. Reactjs

React.JS là một thư viện Javascript dùng để xây dựng giao diện người dùng, nó không phải là một framework js nào hết. React hỗ trợ việc xây dựng những thành phần (components) UI có tính tương tác cao, có trạng thái và có thể sử dụng lại được. React được xây dựng xung quanh các component. React không chỉ hoạt động trên phía client, mà còn được render trên server và có thể kết nối với nhau...

5.2.5. Tensorflow

TensorFlow là một thư viện phần mềm mã nguồn mở dành cho máy học trong nhiều loại hình tác vụ nhận thức và hiểu ngôn ngữ. Nó hiện đang được sử dụng cho cả nghiên cứu lẫn sản xuất bởi 50 đội khác nhau trong hàng tá sản phẩm thương mại của Google, như nhận dạng giọng nói, Gmail, Google Photos, và tìm kiếm, nhiều trong số đó đã từng sử dụng chương trình tiên nhiệm DistBelief của nó.

5.2.6. Pytorch

PyTorch là một khung công tác học máy mã nguồn mở dựa trên thư viện

Torch, được sử dụng cho các ứng dụng như thị giác máy tính và xử lý ngôn ngữ tự nhiên, chủ yếu được phát triển bởi Meta AI.

5.2.7. VnCoreNLP

VnCoreNLP là hệ thống cung cấp chú thích NLP nhanh và chính xác trong Tiếng Việt, cung cấp các chú thích ngôn ngữ phong phú thông qua các thành phần NLP chính của phân đoạn từ, POS tag, entity recognition (NER) và phân tích cú pháp phụ thuộc.

5.3. Database

PostgreSQL là hệ thống quản trị cơ sở dữ liệu quan hệ và đối tượng (Object - Relational Database Management System) có mục đích chung, là hệ thống cơ sở dữ liệu mã nguồn mở miễn phí tiên tiến nhất hiện nay. PostgreSQL sở hữu hệ thống tính năng đa dạng giúp hỗ trợ các nhà quản trị bảo vệ toàn vẹn dữ liệu, tạo ra một môi trường chịu lỗi giúp bạn quản lý cả tập dữ liệu lớn lẫn tập dữ liệu nhỏ.

Trong đề tài này, PostgreSQL database được sử dụng để lưu trữ dữ liệu cào được trên bốn trang web bất động sản.

5.4. Công cụ

5.4.1. Docker

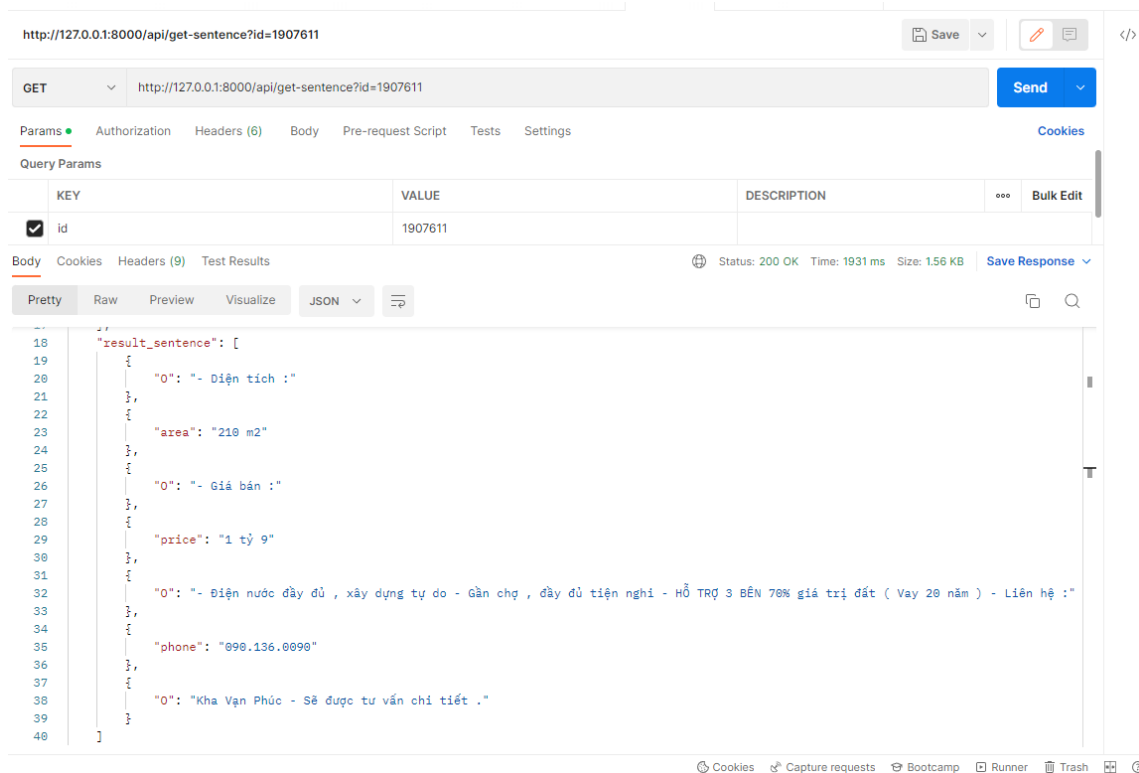
Docker là một nền tảng để cung cấp cách để building, deploying và running ứng dụng dễ dàng hơn bằng cách sử dụng các containers (trên nền tảng ảo hóa).

Trong hệ thống sử dụng docker để triển khai hệ thống crawler, django và web API để triển khai một cách dễ dàng.

5.4.2. Postman

Postman là một công cụ cho phép chúng ta thao tác với API, hỗ trợ nhiều phương thức khác nhau như GET, POST, PUT, ...

Trong hệ thống sử dụng Postman để hỗ trợ kiểm thử kết quả từ Django API.



Hình 5.3 Sử dụng Postman để hỗ trợ kiểm thử kết quả từ Django API

Chương 6. HIỆN THỰC HỆ THỐNG

Từ những đặc tả ở **Error! Reference source not found.** và những công cụ đã liệt kê tại **Error! Reference source not found.**, tôi đã tiến hành hiện thực thuật toán tra cứu thông tin bất động sản. Dưới đây là mô tả chi tiết cách hiện thực của từng thành phần chức năng trong hệ thống:

6.1. Hệ thống cào dữ liệu (Data Crawler)

Như đã đề cập trong **Error! Reference source not found.**, hệ thống cào dữ liệu (data crawler) sẽ bao gồm bốn thành phần chính:

- PostgreSQL Database: lưu trữ dữ liệu sau khi cào được từ năm website bất động sản.
- Crawler: Cào dữ liệu và đổ dữ liệu thô vào PostgreSQL Database
- Backend: Truy vấn PostgreSQL Database và cung cấp các Restful API endpoint để frontend có giao tiếp nhằm thu thập các dữ liệu cần thiết xuất ra màn hình
- Frontend: giao tiếp với backend qua Restful APIs để thu thập dữ liệu cần thiết và hiển thị lên giao diện người dùng.

Bình Dương Real Estate Scraper Tài Liệu Về Bình Dương Real Estate Scraper

Nhập số trang để quét dữ liệu của website tương ứng

ANCU.me Số Trang Ví dụ: 5 Quét Dữ Liệu

Nhập số trang cần quét

Hiện Thị Mô Tả: Tất Hiện Thị Địa Chỉ: Tất

Cập Nhật Copy Excel CSV PDF Xóa Dữ Liệu Đã Chọn Hiện 10 hàng mỗi trang (cũng là số dòng khi tải file về) Search:

<input type="checkbox"/>	Số Thứ Tự	Thời Gian Đăng Bài	Tiêu Đề	Loại BDS	Mô Tả	Diện Tích	Giá	Địa Chỉ Cụ Thể	Bản Đồ	Đường Liên Kết	Người Đăng Bài	Số Điện Thoại
<input type="checkbox"/>	2011349	2022-08-20 07:00:00	Cần bán đất minh hòa giá rẻ đường bê tông có nhà sản thổ cư	Mua bán Đất	minh hòa 8.5x25 thổ cư 100 giá 960tr chỉ cách nhà 100m dân cư hui, cách trung tâm xã 2km. Đây đủ tiện ích xung quanh bán kính 2km có căn nhà cấp 4 sẵn , giá rẻ không nơi nào bằng , đầu tư an cư đều được th 0386554243	257	960 Triệu	đường Tỉnh lộ 749, xã Minh Hòa, huyện Dầu Tiếng, Bình Dương	Ấn Để Xem Vĩ Độ	https://ancu.me/ancu-ban-dat-minh-hoa-01a-nc-duong-be-tong-co-nha-san-tho-cu-ad2011349.html	Dương Ngọc Hiếu	386554243
<input type="checkbox"/>	2038730	2022-08-19 07:00:00	Chỉ 750tr sở hữu căn hộ mặt tiền q11k, 60m2 2pn	Mua bán chung cư	Bạn đang Ồ THUẾ , 100% khi đọc qua dự án này bạn sẽ chính thức sở hữu ngôi nhà của riêng mình tại Thành Phố Dĩ An : 1.Nhân nhà sau 12 tháng vẫn chưa cần đóng tiền. 2. Tầng nổi thật cao cấp - Video call, we toto, gô ăn cường... 3. Giá bán sốc nhất thị trường Tp Dĩ An chỉ từ 39tr/m2 có vat 4. Pháp lý 100% đầy đủ : Giấy Phép xây dựng , 1/500 , Chấp thuận chủ trương số công thương. 2 NGÂN HÀNG ĐỒNG HÀNH hỗ trợ CHO VAY : Vietcombank, ACB ----- - Sở hữu vị trí dự án mặt tiền đường : Quốc lộ 1k- đối diện chợ Đồng Hoà - Qui mô dự án : Hơn 6000m2 - Đa dạng mẫu căn hộ cho KH lựa chọn : 1PN , 1WC - Diện tích 53m2 - 2PN , 2WC - Diện tích 58m2 - 72m2 - 3PN , 2WC - Diện tích 78m2 - 87m2 - Penthouse : Diện tích 131m2 - 141m2 . Sở hữu chuỗi tiện ích đẳng	60	2.4 Tỷ	Quốc lộ 1K Đồng Hòa	Ấn Để Xem Vĩ Độ	https://ancu.me/chi-750tr-so-huu-can-ho-mat-tien-q11k-60m2-2pn-ad2038730.html	Anh Phát Kiều	901410280

Hình 6.1 Giao diện web browser của hệ thống cào dữ liệu từ trang web An Cư

ĐĂNG BÁN NHÀ ĐẤT WWW.DANGBANNHADAT.VN Số Trang Ví dụ: 5 Quét Dữ Liệu

Nhập số trang cần quét

Cập Nhật Copy Excel CSV PDF Xóa Dữ Liệu Đã Chọn Hiện 10 hàng mỗi trang (cũng là số dòng khi tải file về) Search:

<input type="checkbox"/>	Số Thứ Tự	Thời Gian Đăng Bài	Tiêu Đề	Loại BDS	Mô Tả	Diện Tích	Giá	Địa Chỉ Cụ Thể	Đường Liên Kết	Người Đăng Bài	Số Điện Thoại
<input type="checkbox"/>	319850	2022-10-08 07:00:00	Chính chủ bán gấp căn hộ Eco Xuân tại QL 13, Phường Lái Thiêu, Thành phố Thuận An, Bình Dương- Diện tích : 66.9m2 , 2PN,2WC, có lò gas, thông thoáng- View quốc lộ 13, hướng đông mát mẻ.- Nội thất hoàn thành, chỉ cần xách vali vào ở. * Tiện ích+ Hồ bơi người lớn trẻ em, khu vui chơi trẻ em, gym, yoga, siêu thị bán lẻ Vinmart, cửa hàng tiện ích rau củ quả.+ Các tiện ích đầy đủ khác lân cận bán kính 1km trở lại.+ Cách QL13 100m, siêu thị mua sắm Lotte Mart và BV quốc tế Becamex kề bên, TTHC Lái Thiêu 1km. + Cách TTMT Aeon Mall, KCN Vasp1, KCN Đồng An và sân Golf Sóng Bé 5phút, chợ Lái Thiêu 700m, Bách hóa xanh, điện máy xanh 03 phút... + Không gian yên tĩnh, mát mẻ, KDC nằm ngay TT Lái Thiêu, Tp Thuận An. Thuận tiện đi lại về TpHCM 20 phútTDM 15 phút. Giá bán: 2.36 tỷ (chủ đầu tư đang gấp rút ra số)- Hỗ trợ vay NH tối đa, thủ tục nhanh chóng 24h.Liên hệ chính chủ : 0989162971 Ms Thuận	Bán căn hộ chung cư tại Quốc Lộ 13Phường Lái Thiêu Thuận An Bình Dương, Giá cực tốt	67	2.36 Tỷ	Quốc Lộ 13 - Phường Lái Thiêu - Thuận An - Bình Dương	https://dangbannhadat.vn/ban-can-ho-chung-cu-hinh-chu-ban-anc-can-ho-eco-xuan-trung-tam-tp-thuan-an-hinh-duong-gia-cuc-tot-qr319850.html	Ms Thuận	989162971	

Hình 6.2 Giao diện web browser của hệ thống cào dữ liệu từ trang web đăng bán nhà đất

6.2. Gán nhãn và training model

6.2.1. Gán nhãn dữ liệu

Với tập dữ liệu đã được cào về từ hệ thống cào dữ liệu ở mục 6.1. Chúng ta tiến hành gán nhãn cho dữ liệu theo các nhãn được khai báo trước đó cho toàn bộ mẫu tin đã được cào về.

Sau đây là ví dụ về việc gán nhãn cho một mẫu tin:

```
{
  "id": 1,
  "data": "Cần bán nhà Tân Uyên tại Trung tâm Hội Nghĩa 3 phòng ngủ Nhà phố Tân Uyên Sổ sẵn Công chứng ngay Nhà mặt tiền kinh doanh đối diện chợ Hội Nghĩa Ngang diện tích sử dụng 160m2 Có 3PN 3WC phòng thờ sân vườn Sổ hoàn công sang tên ngayNgân hàng hỗ trợ 70 Thanh toán 50 được nhận nhà sử dụng ở hoặc cho thuê gt Liên hệ 0364159638 Công để xem chi tiết",
  "label": [
    [25, 44, "surrounding", "Trung tâm Hội Nghĩa"],
    [53, 59, "real_estate_type", "ngủ Nhà"],
    [100, 108, "position", "mặt tiền"],
    [129, 148, "surrounding", "chợ Hội Nghĩa Ngang"],
    [167, 172, "area", "160m2"],
    [176, 179, "bed_room", "3PN"],
    [180, 183, "bath_room", "3WC"],
    [285, 286, "usage", "ở"],
    [292, 300, "usage", "cho thuê"],
    [312, 322, "phone", "0364159638"]
  ]
}
```

Hình 6.3 Gán nhãn cho dữ liệu đã được cào về

Phương thức gán nhãn được mô tả như sau:

- Id: được đánh số để xác định theo từng mẫu tin nhằm xác định nếu mẫu dữ liệu bị lỗi trong quá trình training.
- Data: dữ liệu thô đã được cào về
- Label: danh sách các nhãn được định danh theo data trong đó mỗi label gồm 4 trường chính, trong đó [(1), (2), (3), (4)] được mô tả như sau: (1) vị trí bắt đầu của nội dung label được gán, (2) vị trí kết thúc của nội dung label được gán, (3) label được gán cho nội dung được chọn, (4) giá trị của label được xác định trong dữ liệu thô.

6.2.2. Training model

Sau khi gán nhãn cho toàn bộ dữ liệu đã được cào về, chúng ta tiến hành training model theo mô hình PhoBERT.

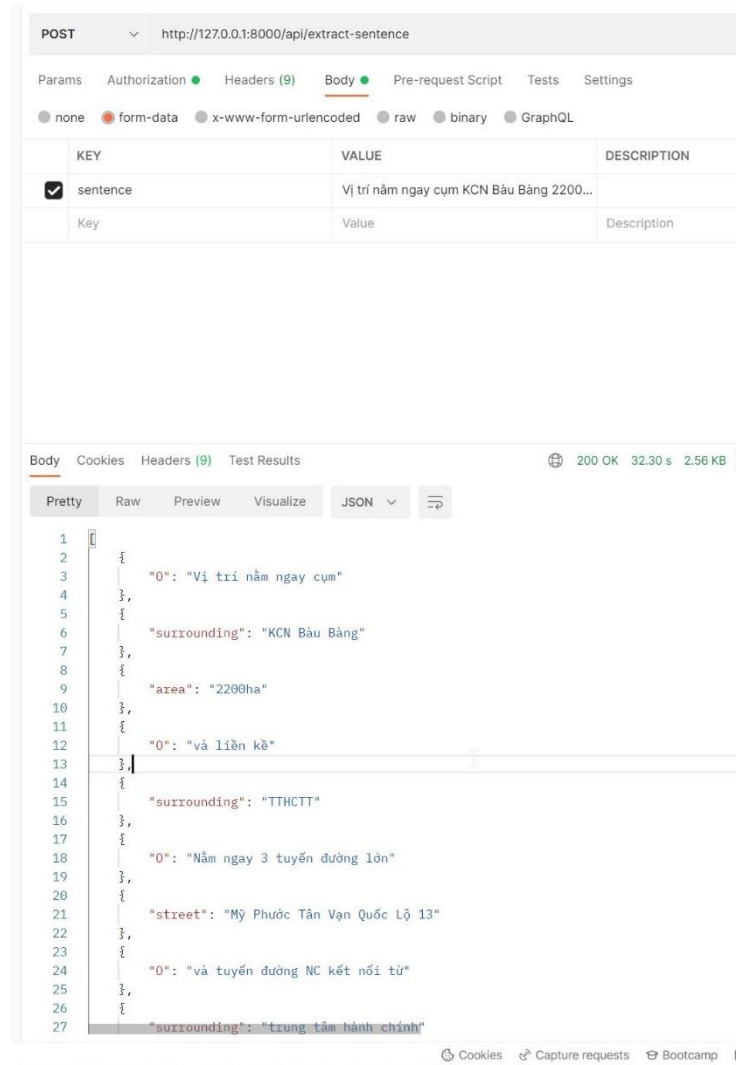
```
Validation F1-Score: 0.9350302918926319
Epoch: 64% | 32/50 [1:42:37<57:25, 191.41s/it]A
Average Train Loss: 0.81817671468199383
Average Train Accuracy: 0.9909620126397666
Average Train F1-Score: 0.99108014409949703
saving the best performance model ...
saving the best performance model ...
Validation loss: 0.24626137880303568
Validation Accuracy: 0.9355177851760095
Validation F1-Score: 0.9348337064597698
Epoch: 66% | 34/50 [1:49:08<51:02, 191.42s/it]A
Average Train Loss: 0.81668581508161117
Average Train Accuracy: 0.99181376376330286
Average Train F1-Score: 0.9918527240773883
saving the best performance model ...
Validation loss: 0.2487702127546072
Validation Accuracy: 0.937576660418083
Validation F1-Score: 0.9378186278573074
Epoch: 70% | 35/50 [1:52:12<47:50, 191.39s/it]A
Average Train Loss: 0.816139126574021896
Average Train Accuracy: 0.9918408685788365
Average Train F1-Score: 0.9918559755570734
saving the best performance model ...
Validation loss: 0.2595148785807025
Validation Accuracy: 0.9358867168073775
Validation F1-Score: 0.9343714765124195
Epoch: 72% | 36/50 [1:55:23<44:40, 191.43s/it]A
Average Train Loss: 0.815576233291560186
Average Train Accuracy: 0.9922881218603143
Average Train F1-Score: 0.9923084453012475
saving the best performance model ...
Validation loss: 0.25571783840970444
Validation Accuracy: 0.935619988318439
Validation F1-Score: 0.93493322994889
Epoch: 74% | 37/50 [1:58:34<41:27, 191.35s/it]A
```

Hình 6.4 Quá trình training model

Sau quá trình training ta được model với khoảng 500MB dung lượng.

6.3. Named Entity Recognition Service

Đây là một backend, sử dụng pretrained model đạt được sau khi huấn luyện mô hình AI là BERT để nhận dạng thực thể có tên từ đầu vào là một đoạn văn bản được gửi tới hệ thống thông qua Restful API.



Hình 6.5 Sử dụng postman để giả lập gửi API request tới NER service

6.4. Hệ thống Django backend

Đây là hệ thống kết nối trực tiếp với database để truy xuất dữ liệu. Hiện tại dữ liệu được lấy từ bốn trang web được lưu vào bốn bảng khác nhau. Mỗi bảng đều có các trường khác nhau để lưu trữ dữ liệu phù hợp đối với trang web.

Hệ thống backend được thiết kế với một số API chính như sau:

- /house: dùng để get toàn bộ thông tin dữ liệu đã crawl được từ database
- delete/house: dùng để xóa một dòng dữ liệu crawl từ database

- `api/get-sentence`: dùng để lấy thông tin dữ liệu sau khi sử dụng phobert để tiên đoán.

6.4.1. Hệ thống tự động cào dữ liệu tự động

Hệ thống được xây dựng cơ chế tự động cào dữ liệu từ các trang web bất động sản có đăng các mẫu tin về bất động sản Bình Dương là muaban.net, ancu.me, dangbannhadat.vn và nhadat24h.net. Hệ thống được cài đặt, khi triển khai sẽ tự động cào những mẫu tin mới nhất chưa nằm trong cơ sở dữ liệu. Được cài đặt mỗi 5 phút sẽ thực thi một lần.

6.4.2. Hệ thống tự động gán nhãn cho mẫu tin đã cào về

Hệ thống được xây dựng để xử lý những mẫu tin đã được hệ thống tự động cào về ở mục 6.4.1. Để tiến hành gán nhãn, ta sử dụng model đã được training trước đó (mô tả ở mục 6.2). Sau quá trình gán nhãn tự động, kết quả sẽ được lưu trữ vào cơ sở dữ liệu và hiển thị lên hệ thống webapp frontend sẽ được mô tả ở mục tiếp theo.

6.5. Hệ thống webapp frontend

Hệ thống webapp frontend được xây dựng trên ngôn ngữ lập trình ReactJs. Có giao diện trực quan, người dùng dễ dàng thao tác, cho phép người dùng có thể xóa từng mẫu tin hoặc xóa nhiều mẫu tin. Hệ thống webapp frontend còn cho phép người dùng có thể tìm kiếm thông tin theo “Chủ sở hữu”, “Loại hình”, “Địa chỉ” và “Chứng nhận sở hữu”.

House price table

Chủ sở hữu	Loại hình	Search	Reset Search
Địa chỉ	Chứng nhận sở hữu		

ID	Loại hình ^	Chủ sở hữu	Số điện thoại	Diện tích	Giá tiền	Địa chỉ	Chứng nhận sở hữu	Xóa
<input type="checkbox"/> 1986944	Mua bán Đất nền dự án	Trần Tấn Lộc Nguyễn	908766167	70	1.3 Tỷ	DT741 Hòa Lợi		
<input type="checkbox"/> 1987014	Mua bán nhà riêng	Phan Công	364159638	65	2.898 Tỷ	đường 747, xã Hội Nghĩa, huyện Tân Uyên, Bình Dương		
<input type="checkbox"/> 1909635	Mua bán Đất	Trần Hương	335423802	300	1.3 Tỷ	xã Thới Hòa, thị xã Bến Cát, Bình Dương		
<input type="checkbox"/> 1933982	Mua bán Đất	Trần Tuyền	974618124	70	650 Triệu	đường ĐT 741, phường Phước Vĩnh, huyện Phú Giáo, Bình Dương		
<input type="checkbox"/> 1987332	Mua bán Đất	Trần Trọng	768384384	100	830 Triệu	đường Nguyễn Chí Thanh, xã Thới Hòa, thị xã Bến Cát, Bình Dương		
<input type="checkbox"/> 1987358	Mua bán Đất	Văn Hải Vũ	339191006	100	680 Triệu	ĐT 750 Lai Uyên		
<input type="checkbox"/> 1982133	Mua bán Đất nền dự án	Nguyễn Hiếu	969574107	150	500 Triệu	đường ĐT 741, xã Chánh Phú Hòa, thị xã Bến Cát, Bình Dương		
<input type="checkbox"/> 1982140	Mua bán Đất	Thị Minh Châu Trần	345531472	94	800 Triệu	DA1-1 Mỹ Phước		

0 items selected delete selected items

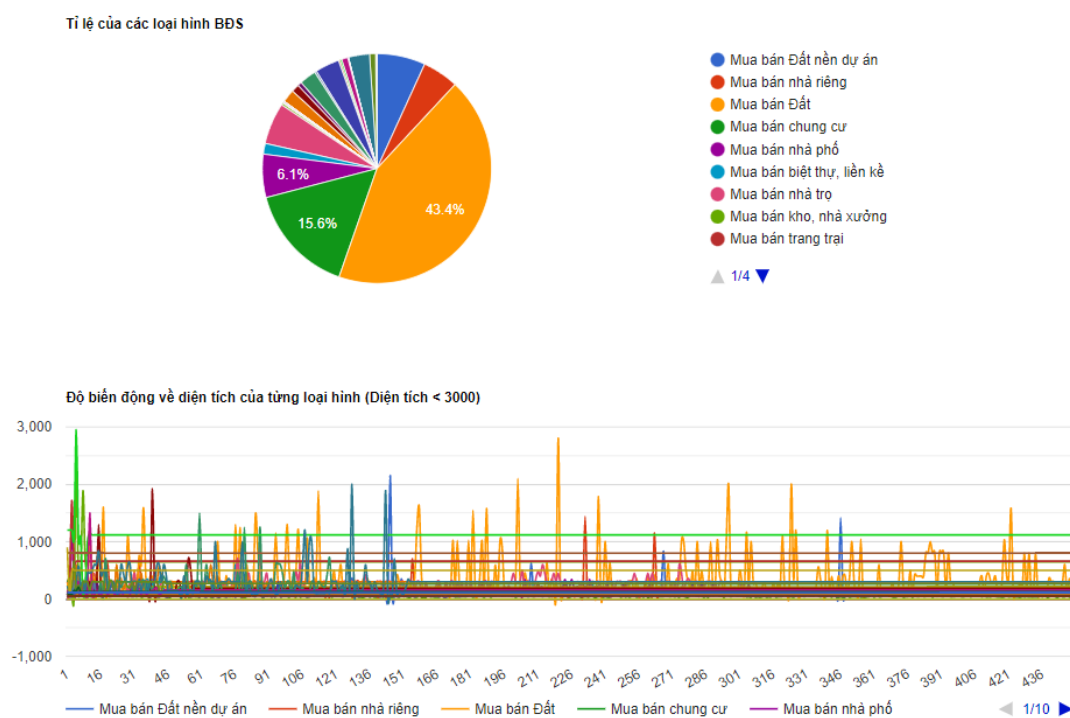
< previous **1** 2 3 ... 646 647 next >

Hình 6.6 Giao diện web hiển thị thông tin các mẫu tin được cào về

Detail information of house with ID: 1987332

Raw data	- Gia đình cần bán lại lô đất gần ở KCN Mỹ Phước 3.Thuộc Thới Hoà- Bến Cát- Chính chủ, sổ hồng riêng,thổ cư 100%.- Diện tích: 100m2(5x20m)- Giá bán: 830 triệu/nền. (Bao sổ sách giấy tờ).- Ngay chợ, tiện kinh doanh, đối diện ngay KCN đang hoạt động.- Cơ sở hạ tầng hoàn thiện 100%. Điện nước đã có, hệ thống đèn đường chiếu sáng, vỉa hè lát gạch.- Tiện ích xung quanh đầy đủ, dân cư đông đúc.- Mặt tiền đường 7m, thông dài ra KCN, gần QL13 giao thông đi lại thuận tiện.- Đất ở vị trí đẹp tiện để ở, đầu tư, xây nhà cho thuê ngay.- Hỗ trợ ngân hàng 70%.- Sang tên chuyển nhượng tại phòng công chứng Tỉnh Bình Dương. Nhận sổ đỏ trong vòng 7 ngày.- Liên hệ 0768384384 gặp Trọng để xem đất và các giấy tờ liên quan.
Label	Content
O	- Gia đình cần bán lại lô đất gần ở
surrounding	KCN Mỹ Phước 3
O	. Thuộc Thới Hoà - Bến Cát - Chính chủ ,
legal	sổ hồng riêng
O	, thổ cư 100% . - Diện tích :
area	100 m2 (5 x 20 m
O) - Giá bán :
price	830 triệu
O	/ nền . (Bao sổ sách giấy tờ) . - Ngay
surrounding	chợ
O	, tiện kinh doanh , đối diện ngay KCN đang hoạt động . - Cơ sở hạ tầng hoàn thiện 100% . Điện nước đã có , hệ thống đèn đường chiếu sáng , vỉa hè lát gạch . - Tiện ích xung quanh đầy đủ , dân cư đông đúc . - Mặt tiền đường 7m , thông dài ra KCN , gần
street	QL13
O	giao thông đi lại thuận tiện . - Đất ở vị trí đẹp tiện để ở ,
usage	đầu tư
O	,
usage	xây nhà cho thuê
O	ngay . - Hỗ trợ ngân hàng 70% . - Sang tên chuyển nhượng tại phòng công chứng Tỉnh Bình Dương . Nhận sổ đỏ trong vòng 7 ngày . - Liên hệ
phone	0768384384
O	gặp Trọng để xem đất và các giấy tờ liên quan .

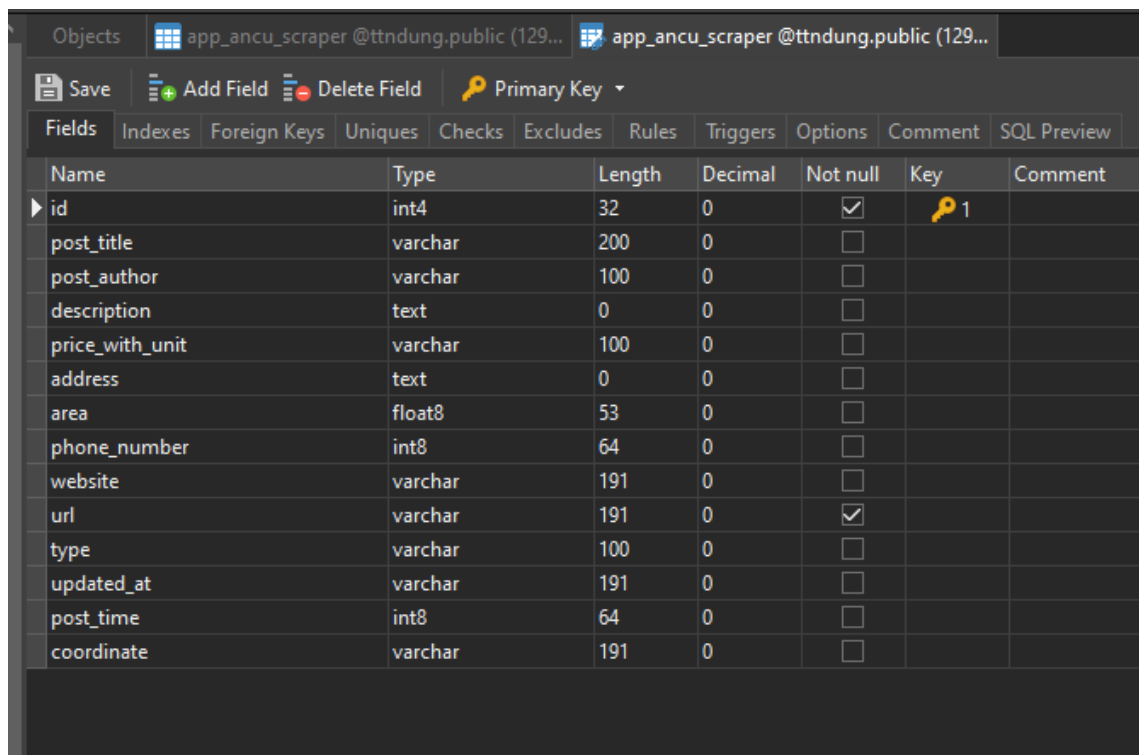
Hình 6.7 Thông tin chi tiết một mẫu tin cào về được gán nhãn



Hình 6.8 Thông tin được thống kê qua biểu đồ

6.6. Kết quả trả về

- Kết quả trả về từ hệ thống cào dữ liệu được lưu vào database như sau:



Name	Type	Length	Decimal	Not null	Key	Comment
id	int4	32	0	<input checked="" type="checkbox"/>	1	
post_title	varchar	200	0	<input type="checkbox"/>		
post_author	varchar	100	0	<input type="checkbox"/>		
description	text	0	0	<input type="checkbox"/>		
price_with_unit	varchar	100	0	<input type="checkbox"/>		
address	text	0	0	<input type="checkbox"/>		
area	float8	53	0	<input type="checkbox"/>		
phone_number	int8	64	0	<input type="checkbox"/>		
website	varchar	191	0	<input type="checkbox"/>		
url	varchar	191	0	<input checked="" type="checkbox"/>		
type	varchar	100	0	<input type="checkbox"/>		
updated_at	varchar	191	0	<input type="checkbox"/>		
post_time	int8	64	0	<input type="checkbox"/>		
coordinate	varchar	191	0	<input type="checkbox"/>		

Hình 6.9 Database lưu trữ dữ liệu sau khi crawler về

Dữ liệu được lưu với các fields như hình trên. Dữ liệu thô được lưu trong trường description, dữ liệu ở trường này sẽ được gán nhãn thông qua mô hình PhoBERT.

Objects app_ancu_scraper @ttndung.public (129... app_ancu_scraper @ttndung.public (129...				
Begin Transaction Text Filter Sort Import Export				
id	post_title	post_author	description	pr
1986944	Chỉ 520 triệu sở hữu Trần Tấn Lộc Nguyễn		đắp giá trị - Kiến tạo phần hoa#Richland #RichlandResidence #KimOanhGroup	1.3
1987014	Cần bán nhà Tân Uyên Phan Công		Cần bán nhà Tân Uyên, tại Trung tâm Hội Nghĩa, 3 phòng ngủNhà phố Tân Uyên. Số sã 2.8	
1909635	Qua năm gia đình Trần Hương		Qua năm gia đình tôi về Đà Nẵng định cư nên cần bán miếng đất chỗ KCN Bình Dương 1.3	
1933982	Bán lô đất cách chợ Trần Tuyền		- Mặt tiền đường DT741, tt. Phước Vĩnh, Phú Giáo- Diện tích 5x12 thổ cư 100%- Gần TTH 650	
1987332	Cần bán gấp đất sã Trần Trọng		- Gia đình cần bán lại lô đất gần ở KCN Mỹ Phước 3.Thuộc Thời Hòa- Bến Cát- Chính ch 830	
1987358	Bán đất ngay chợ Văn Hải Vũ		Cần bán đất ở Bình Dương.Ngay chợ, cấp sát quốc lộ 13.Dân cư đông, Tiện ích đầy đủ.1.680	
1982133	Đất nền khu công nghiệp Nguyễn Hiếu		👉👉 TIN ĐƯỢC KHÔNG /-li /-li /-li 📞CHỈ 500 TRIỆU 📞SUẤT ĐẦU TƯ ĐẤT VSIP 2C/ 500	
1982140	Bán nhanh 94m2 đ Thị Minh Châu Trần		Lô đất có diện tích 94m2 (5m x19m), nằm ngay trên trục đường chính HL604 và DA1-1 r 800	
1982154	Cần hộ chất phát n Trúc Đào Nguyễn		HT Pearl chuẩn an cư chuẩn đầu tư tại Đông Sài Gòn.HT Pearl căn hộ sở hữu vị trí tâm đ 2.5	
1977207	Mặt tiền DT741 An Phan Tùng		Cần bán 2 lô mặt tiền DT741 An Bình, Phú Giáo khu dân cư đông đúc, kinh doanh buôn 120	
1982249	Kim oanh group m Thắng Phát Đỗ		Vừa ra mắt thị trường, Richland Residence đã nhanh chóng thu hút sự quan tâm của đ 1.5	
1853677	Chỉ 8tr/m2, sở hữu hunng nguyên		Chỉ 8TR/M2, Sở Hữu Căn Nhà Phố Thương Mại Sát Chợ, Công Viên, TT.Thương Mại - 09 1.2	
1982243	Cần hộ astral city c Hoàng Thiện Đỗ		Đa dạng các sản phẩm: shophouse, officetel, TTTM, Hồ bơi chàn mây tầng 20, rạp chiếu 2.3	
1977338	Đầu tư giá hời ngay lâm Đăng		Đầu tư giá hời ngay trung tâm hành chính Bàu Bàng, Bình Dương. Em còn lô vị trí 2 mặt 1.1	
1973843	Bcons Miền Đông t Sơn		Chung cư Bcons ngay gần đại học An Ninh, cách khu công nghệ cao 7p đi xeBcons đã 1.7	
1977285	Đất thổ cư ngay m: hoang dam		Tôi cần bán vài lô đất thổ cư mặt tiền đường DX132,Khu đất đã có sổ sã,xung quanh đi 1.9	
1907611	CHỦ cần bán GẤP Đình Xuân Kha		- Diện tích: 210 m2- Giá bán: 1 tỷ 9- Diện nước đầy đủ, xây dựng tự do- Gần chợ, đầy đủ 1.9	

ĐỊA THỂ THỊNH VƯỢNG - ĐIỂM VÀNG KẾT NỐI Richland Residence tọa lạc tại vị trí đắc địa, gần nút giao Vành đai 4 (quy mô 6-8 làn xe, rộng 74,5m) và DT 741, liền kề Thành phố mới Bình Dương. Kết nối trực tiếp với những trục đường huyết mạch: Quốc lộ 13, Quốc lộ 14, đại lộ Mỹ Phước - Tân Vạn, DT 742, DT 744 dễ dàng di chuyển đến các khu vực trọng điểm tại TP. HCM, Bình Dương, Đồng Nai và các đô thị lớn lân cận. Sắp tới tuyến Metro Bến Thành - Suối Tiên kéo dài đến thành phố mới Bình Dương và Bến Cát, giúp lưu thông càng thuận tiện Với vị trí vàng, cư dân Richland Residence còn hưởng trọn loạt tiện ích quanh khu vực chỉ trong vài phút kết nối: UBND phường Hòa Lợi, trường THCS Hòa Lợi, Đại học Quốc tế Việt Đức, Trung tâm Hành chính thành phố mới, chợ Chánh Lưu, chợ Hòa Lợi, bệnh viện Đa khoa Mỹ Phước, Trung tâm Thương mại thế giới Bình Dương (lớn nhất Việt Nam)... Đồng thời giá trị sản phẩm Richland Residence còn hứa hẹn tăng mạnh trong thời gian tới.Hotline : 0908.766.167 em Lộc-----RICHLAND RESIDENCEvun đắp giá trị - Kiến tạo phần hoa#Richland #RichlandResidence #KimOanhGroup

Hình 6.10 Dữ liệu sau khi được crawler về

Chương 7. KIỂM THỬ VÀ ĐÁNH GIÁ

7.1. Mô tả tập dữ liệu

Tập dữ liệu của tôi sử dụng trong đề tài được thu thập từ bốn trang chuyên đăng tin mua bán bất động sản Bình Dương là muaban.net, ancu.me, dangbannhadat.vn, nhadat24h.net. Và số dữ liệu thu thập được từ các trang web trên có tổng cộng 3458 mẫu dữ liệu. Trong đó, các mẫu dữ liệu ở các trang theo từng trang web được mô tả như sau:

Tập dữ liệu gồm 10% (345 mẫu) dùng để test và 90% (3118 mẫu) dùng để huấn luyện. Các nhãn gán trong tập dữ liệu được thống kê như bảng dưới đây:

Bảng các thực thể có tên cần xác định

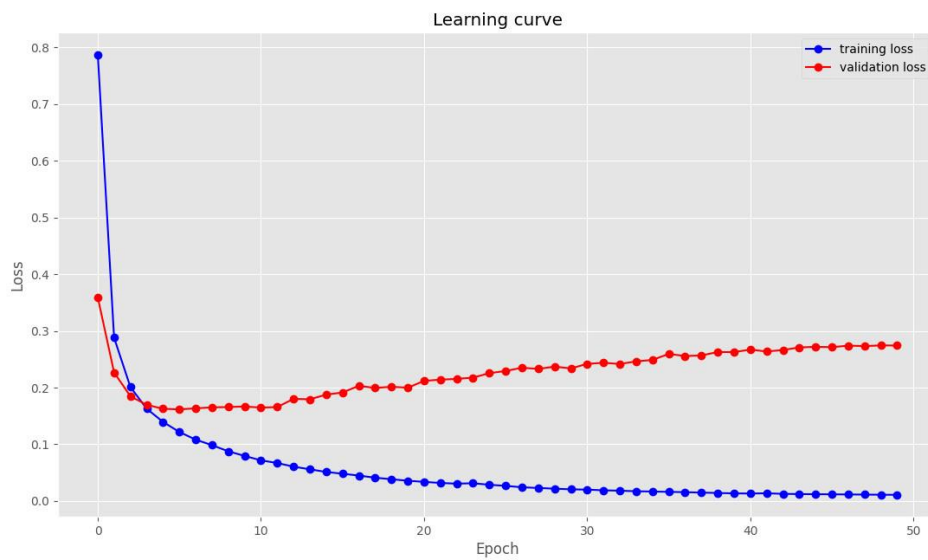
Tên thực thể	Diễn giải	Số lượng thực thể được gán
transaction	Loại giao dịch	1747
real_estate_type	Mã loại bất động sản	2017
real_estate_sub_type	Mã loại bất động sản phụ	358
price	Giá	6651
area	Diện tích	5770
direction	Hướng	147
street	Đường	2557
ward	Phường / xã	937

district	Quận / huyện	417
city	Thành phố/ tỉnh	1866
email	email	6
phone	Điện thoại	2832
usage	Mục đích sử dụng	2509
floor	kết cấu tầng / lầu, số tầng / lầu	1056
bath_room	phòng tắm / toilet	634
living_room	phòng khách	143
bed_room	phòng ngủ	1043
position	Vị trí	1317
author	Người đăng tin	1389
house_number	Số địa chỉ	66
project_name	Tên dự án, tên chung cư	1200
front_length	Bề rộng mặt tiền	119
road_width	Bề rộng đường	1213
surrounding	Tiện ích xung quanh	9699
legal	Thông tin pháp lý	2570

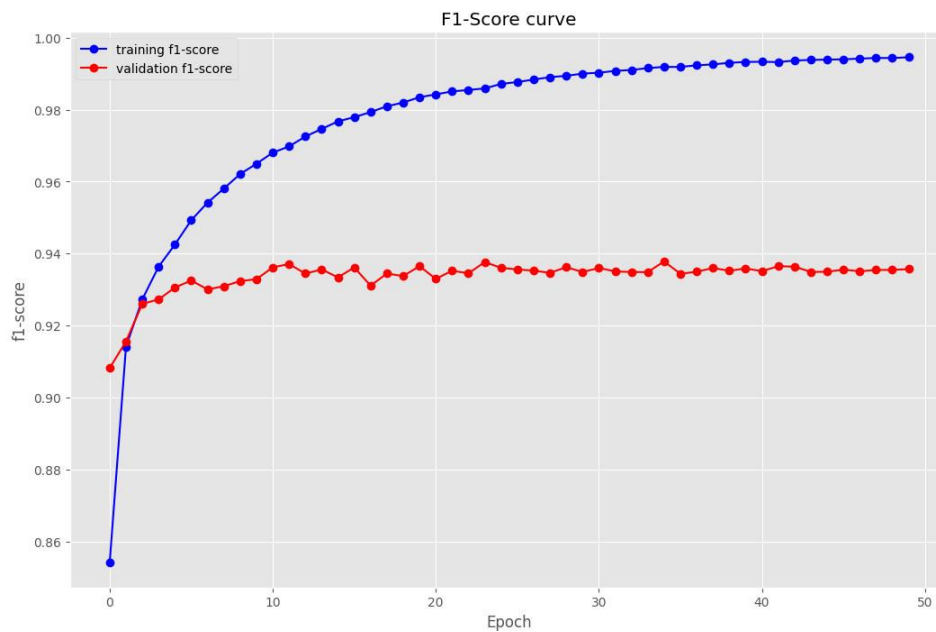
Bảng 7.1 Bảng các thực thể có tên cần xác định

7.2. Kết quả thí nghiệm Mô hình PhoBERT

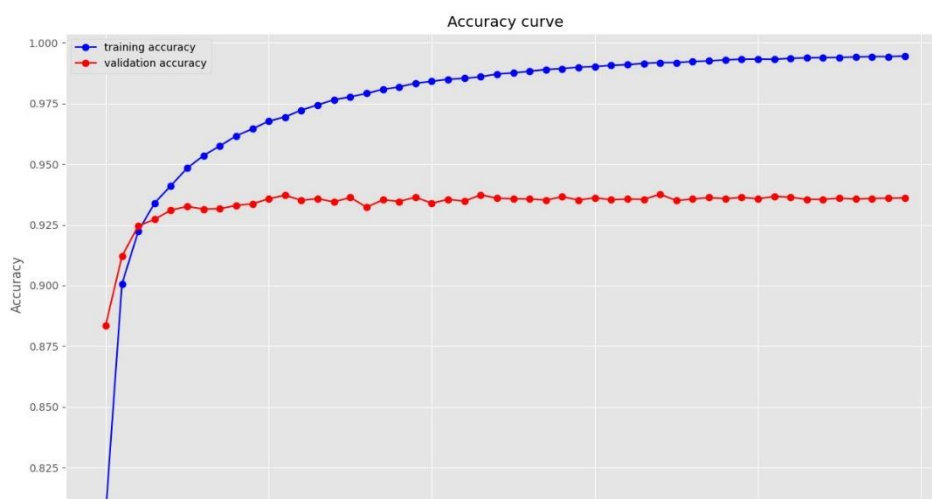
Các chỉ số độ chuẩn xác, độ phủ và điểm F1 của mô hình PhoBERT được thể hiện dưới đây.



Hình 7.1 Đồ thị training và validation loss theo epoch cho mô hình PhoBERT



Hình 7.3 Đồ thị training và validation f1-score theo epoch cho mô hình PhoBERT



Hình 7.2 Đồ thị training và validation accuracy theo epoch cho mô hình PhoBERT

Bảng thể hiện các chỉ số và kết quả trong đó, các kết quả có số liệu support có giá trị thấp (có giá trị là 0 và 1) được loại bỏ khỏi danh sách.

Label	precision	recall	f1-score	support
B-area	0.88	0.88	0.88	591
B-author	0.89	0.78	0.80	150
B-bath_room	0.90	0.75	0.77	64
B-bed_room	0.91	0.93	0.91	88
B-city	0.92	0.67	0.68	189
B-direction	0.93	0.62	0.70	13
B-district	0.94	0.53	0.58	40
B-floor	0.96	0.87	0.88	105
B-front_length	0.97	0.36	0.43	14
B-house_number	0.98	0.33	0.40	3
B-legal	0.99	0.79	0.82	296
B-living_room	0.100	1.00	1.00	18
B-phone	0.101	0.87	0.90	289
B-position	0.102	0.71	0.74	123
B-price	0.103	0.86	0.89	666
B-project_name	0.104	0.64	0.61	116
B-real_estate_sub_type	0.105	0.81	0.76	32

B-real_estate_type	0.106	0.66	0.70	214
B-road_width	0.107	0.74	0.81	137
B-street	0.108	0.72	0.72	248
B-surrounding	0.109	0.82	0.83	902
B-transaction	0.110	0.79	0.82	212
B-usage	0.111	0.79	0.83	320
B-ward	0.112	0.64	0.66	100
I-area	0.113	0.93	0.93	1263
I-author	0.114	0.75	0.77	148
I-bath_room	0.115	0.81	0.83	108
I-bed_room	0.116	0.94	0.91	158
I-city	0.117	0.68	0.68	309
I-direction	0.118	0.71	0.80	14
I-district	0.119	0.54	0.64	112
I-floor	0.89	0.92	0.93	198
I-front_length	0.50	0.33	0.90	6
I-legal	0.90	0.79	0.84	662
I-living_room	1.00	1.00	1.00	36
I-phone	0.95	0.90	0.92	1113

I-position	0.67	0.72	0.69	137
I-price	0.93	0.86	0.89	1414
I-project_name	0.57	0.66	0.61	320
I-real_estate_sub_type	0.61	0.77	0.68	30
I-real_estate_type	0.66	0.61	0.63	140
I-road_width	0.92	0.72	0.81	80
I-street	0.76	0.79	0.78	595
I-surrounding	0.89	0.89	0.89	3778
I-transaction	0.63	0.81	0.71	54
I-usage	0.93	0.85	0.89	819
I-ward	0.71	0.62	0.66	313
O	0.96	0.97	0.96	51746

Bảng 7.2 Bảng chỉ số và kết quả của các thực thể có tên

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Các công việc đạt được

Mô hình BERT dự đoán khá chính xác trên tập dữ liệu kiểm thử đúng như kỳ vọng. Mô hình BERT đạt được kết quả tốt.

2. Các hạn chế

Vì mô hình sử dụng thư viện `vncorenlp` để thực hiện thao tác word segmentation, do đó kết quả đầu ra của mô hình phụ thuộc lớn vào độ chính xác của thao tác word segmentation của thư viện `vncorenlp`.

3. Bước phát triển

Tuy hệ thống đã có thể hoạt động khá ổn định nhưng để đưa vào phục vụ tư vấn thực tế thì cần có thêm một số cải tiến:

- Tích hợp vào các nền tảng bất động sản lớn (`batdongsan`, `muaban`, `muban24h...`), mở rộng được phạm vi phục vụ.
- Mở rộng thêm nguồn dữ liệu dùng cho truy vấn.
- Tăng cường hỗ trợ thêm nhiều dạng ý định khác của người dùng.

TÀI LIỆU THAM KHẢO

- Aone, C. (1999). A trainable summarizer with knowledge acquired from robust NLP techniques. *The MIT Press*, pp. 71-80.
- Babych, B., & Hartley, A. (2003). Improving machine translation quality with automatic named entity recognition. *Proceedings of the 7th International EAMT workshop on MT and other language technology tools, Improving MT through other language technology tools, Resource and tools for building MT at EACL 2003*.
- Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. *Proceedings of the 25th international conference on Machine learning*, 160-167.
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011, August 12). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 2493-2537.
- Cheng, P., & Erk, K. (2020). Attending to entities for better text understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05), 7554-7561.
- Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., . . . Yates, A. (2005). Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*, 165(1), 91-134.
- Guo, J., Xu, G., Cheng, X., & Li, H. (2009). Named entity recognition in query. *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 267-274.
- Habibi, M., Weber, L., Neves, M., Wiegandt, D. L., & Leser, U. (2017). Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, 33(14), i37-i48.
- Lafferty, J., McCallum, A., & Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proc. 18th International Conf. on Machine Learning* (pp. 282-289). Morgan Kaufmann.
- Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016, January). Neural architectures for named entity recognition. *HLT-NAACL*.
- LeCun, Y., Yoshua, B., & Geoffrey, H. (2015). Deep learning. *nature*, 436-444.
- Li, J., Sun, A., Han, J., & Li, C. (2020). A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*.
- Li, Y., Li, W., Sun, F., & Li, S. (2015). Component-enhanced Chinese character embeddings. *arXiv preprint arXiv:1508.06669*.

- Moll, D., Zaanen, M. V., & Smith, D. (2006). Named entity recognition for question answering. *Proceedings of the Australasian Language Technology Workshop 2006*, pp. 51-58.
- Nguyen, T., Nguyen, L., & Tran, X. (2016). Vietnamese named entity recognition at vlsp 2016 evaluation campaign. *Proceedings of The Fourth International Workshop on Vietnamese Language and Speech Processing*.
- Pham, H. T., & Le, P. H. (2017). End-to-end recurrent neural network models for Vietnamese named entity recognition: Word-level vs. character-level. *arXiv preprint arXiv:1705.04044*.
- Vaswani, A., Noam, S., Niki, P., Jakob, U., Llion, J., Aidan, G. N., . . . Illia, P. (2017). Attention is all you need. *Advances in neural information processing systems*.
- Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., & Liu, Q. (2019). ERNIE: Enhanced language representation with informative entities. *ACL*, 1441–1451.