



SẢN PHẨM DỰ THI
SÁNG KIẾN AN TOÀN GIAO THÔNG 2024



Vita - TRỢ LÝ ẢO GIAO THÔNG THÔNG MINH

Cung cấp thông tin về tình trạng giao thông tại một địa điểm
theo thời gian thực bằng cách sử dụng trợ lý ảo thông minh
và đưa ra phương án di chuyển tối ưu

Dơn vị: Trường đại học Bách Khoa - Đại học Quốc gia TP. HCM

Nhóm trưởng: Lữ Xuân Minh

Thành viên

Phan Thanh Huy

Lê Nguyễn Gia Hưng

Nguyễn Tuấn Kiệt

Hồ Thành Nhân

Mục lục

1 LÝ DO CHỌN ĐỀ TÀI	3
2 MỤC TIÊU CỦA ĐỀ TÀI	3
3 HIỆN THỰC	4
3.1 Intent detection	4
3.1.1 Tóm tắt	4
3.1.2 Giới thiệu	4
3.1.3 Phương pháp	5
3.1.4 Cách thực hiện	6
3.1.5 Thí nghiệm	8
3.1.6 Kết luận	10
3.2 Trích xuất thực thể có tên (NER)	10
3.2.1 Nguyên tắc cơ bản về Trích xuất Thực thể (NER - Named Entity Recognition) . .	10
3.2.2 Lựa chọn mô hình và lý do	11
3.2.3 Tiền xử lý dữ liệu	11
3.2.4 Triển khai NER với loại LOC	12
3.2.5 Hậu xử lý kết quả	14
3.2.6 Các siêu tham số huấn luyện mô hình	14
3.2.7 Dánh giá và tối ưu hóa	15
3.3 Google Map Auto Complete	16
3.3.1 Giới thiệu	16
3.3.2 Chức năng của Google Maps Autocomplete	17
3.3.3 Lợi ích của việc tích hợp Google Maps Autocomplete trong dự án	18
3.3.4 Tích hợp Google Maps Autocomplete vào dự án	18
3.3.5 Kết luận	19
3.4 Lấy tọa độ của địa điểm	19
3.4.1 Công nghệ sử dụng	19
3.4.2 Thông số API	20
3.4.3 Ví dụ	20
3.5 Lấy toạ độ của các camera	26
3.5.1 Giới thiệu	26
3.5.2 Ví dụ	26
3.5.3 Quy Trình gọi API	30
3.6 Google Map Distance Matrix	30
3.6.1 API Distance Matrix là gì?	30
3.6.2 Hoạt động của API và ứng dụng vào mô hình	31
3.6.3 Xác định ngưỡng threshold và tính toán các giá trị đặt trưng của từng con đường truy xuất được từ điểm ban đầu	33
4 TÍNH MỚI, TÍNH SÁNG TẠO	36
5 MÔ HÌNH HÓA HỆ THỐNG VITA	37

6 PHƯƠNG HƯỚNG PHÁT TRIỂN	38
7 THAM KHẢO	39

1 LÝ DO CHỌN ĐỀ TÀI

Giao thông là điều kiện cần thiết cho sự phát triển kinh tế - xã hội. Trong bối cảnh hiện nay, giao thông, đặc biệt là giao thông đường bộ, đóng vai trò chiến lược và là nền tảng quan trọng cho quá trình phát triển. Không chỉ hỗ trợ việc di chuyển của con người và hàng hóa, giao thông đường bộ còn kết nối các vùng miền, tạo điều kiện cho giao lưu văn hóa, thúc đẩy hợp tác kinh tế và hội nhập quốc tế.

Tuy nhiên dân số tăng, cùng với điều kiện kinh tế xã hội phát triển nhưng sự chưa đáp ứng kịp của hạ tầng giao thông đã gây ra nhiều trở ngại. Diễn hình là tình trạng ùn tắc giao thông ở các thành phố lớn. Những tài xế, người tham gia giao thông thường sử dụng dài VOH để nghe thông tin về tình hình giao thông. Thông tin này phần nào cho người nghe dài biết về tình hình giao thông ở các đoạn đường phổ biến tuy nhiên còn nhiều hạn chế vì dài VOH không phải lúc nào cũng trực tuyến. Mặt khác, tài xế hoàn toàn không có thông tin về những đoạn đường mà họ quan tâm nhưng không được trình bày trên VOH.

Tài xế có thể lựa chọn sử dụng công nghệ như Google Map. Tuy nhiên, trong điều kiện đang phải lái xe trên đường, sử dụng thiết bị di động không được khuyến khích vì có thể gây mất tập trung. Trường hợp xấu nhất là có thể gây tai nạn không mong muốn.

Liệu rằng có giải pháp nào có thể cung cấp các thông tin về tình hình giao thông thời gian thực tại một địa điểm cụ thể và có thể tương tác “không chạm” với tài xế để họ có thể tập trung vào việc lái xe hay không?

Dó là câu hỏi thôi thúc nhóm VIS chúng tôi phát triển dự án “Vita - Trợ lý ảo giao thông thông minh”. Trợ lý ảo sẽ cung cấp thông tin về tình trạng giao thông tại một địa điểm thời gian thực bằng cách tương tác qua giọng nói với tài xế và giúp họ đưa ra phương án di chuyển tối ưu hơn.

2 MỤC TIÊU CỦA ĐỀ TÀI

Đề tài “Vita - Trợ lý ảo giao thông thông minh” nhằm mục tiêu xây dựng và phát triển một hệ thống trợ lý ảo cung cấp thông tin giao thông thời gian thực cho tài xế thông qua tương tác bằng giọng nói. Các mục tiêu cụ thể bao gồm:

- **Cung cấp thông tin giao thông tức thời:** Giúp tài xế nắm bắt kịp thời tình trạng giao thông tại các địa điểm cụ thể, bao gồm thông tin về ùn tắc, sự cố, và tình trạng đường xá, giúp tối ưu hóa lộ trình di chuyển.
- **Tương tác “không chạm” an toàn:** Phát triển khả năng tương tác giọng nói để tài xế có thể nhận thông tin giao thông mà không cần sử dụng tay hoặc nhìn vào thiết bị, giúp giảm thiểu nguy cơ mất tập trung và nâng cao an toàn khi lái xe.
- **Thúc đẩy ứng dụng công nghệ trong giao thông:** Góp phần vào việc ứng dụng các công nghệ hiện đại như trí tuệ nhân tạo và xử lý ngôn ngữ tự nhiên vào lĩnh vực giao thông, nhằm hỗ trợ hiệu quả cho các tài xế và giảm thiểu ùn tắc giao thông tại các thành phố lớn.

3 HIỆN THỰC

3.1 Intent detection

3.1.1 Tóm tắt

Hiểu được ý định truy vấn giúp các công cụ tìm kiếm hiện đại cải thiện kết quả tìm kiếm cũng như hiển thị câu trả lời tức thời cho người dùng. Trong công trình này, chúng tôi giới thiệu một phương pháp phân loại truy vấn chính xác để phát hiện ý định của truy vấn tìm kiếm của người dùng. Chúng tôi đề xuất sử dụng mạng nơ-ron tích chập (CNN) để trích xuất các biểu diễn vectơ truy vấn làm các đặc điểm cho phân loại truy vấn. Trong mô hình này, các truy vấn được biểu diễn dưới dạng vectơ để có thể nắm bắt các truy vấn tương tự về mặt ngữ nghĩa bằng cách nhúng chúng vào không gian vectơ.

3.1.2 Giới thiệu

Các công cụ tìm kiếm hiện đại không chỉ dừng lại ở việc tìm kiếm các tài liệu có liên quan. Để đáp ứng nhu cầu thông tin của người dùng, họ cũng hướng đến mục tiêu hiển thị câu trả lời ngắn gọn cho truy vấn của người dùng. Do đó, họ yêu cầu hiểu sâu sắc về truy vấn của người dùng. Xác định mục đích đằng sau truy vấn là một bước quan trọng để đạt được mục tiêu này. Điều này không chỉ giúp hiển thị kết quả tìm kiếm được làm giàu về mặt ngữ nghĩa mà còn giúp cải thiện kết quả xếp hạng bằng cách kích hoạt công cụ tìm kiếm theo chiều dọc trong một miền nhất định (ví dụ: tìm kiếm hình ảnh, tìm kiếm tin tức và công cụ tìm kiếm việc làm)

Phát hiện ý định truy vấn là một nhiệm vụ đầy thách thức vì các truy vấn thường ngắn và việc xác định chính xác ý định của người dùng đòi hỏi nhiều ngữ cảnh hơn ngoài các từ khóa. Hơn nữa, số lượng danh mục trong ý định có thể rất cao. Hầu hết các phương pháp đều đòi hỏi rất nhiều nỗ lực của con người để đáp ứng những thách thức này, bằng cách xác định các mẫu cho từng lớp ý định hoặc bằng cách xác định các tính năng phân biệt đối xử cho các truy vấn để chạy các mô hình thống kê.

Cách tiếp cận của chúng tôi là sử dụng học sâu để tìm các biểu diễn vectơ truy vấn; sau đó sử dụng chúng như các tính năng để phân loại các truy vấn theo ý định. Một trong những lợi thế của việc sử dụng biểu diễn vectơ truy vấn là nhúng các truy vấn vào một không gian sao cho các truy vấn tương tự về mặt ngữ nghĩa gần nhau. Ví dụ, các truy vấn “tình hình giao thông ở đường Lý Thường Kiệt” và “đường Lý Thường Kiệt đang như thế nào?” sẽ gần nhau; do đó, chúng có thể được gán cùng một ý định. Học sâu chủ yếu được sử dụng trong các tác vụ xử lý văn bản thông qua việc tận dụng các biểu diễn vectơ từ về cơ bản là các vectơ có các đặc điểm ngữ nghĩa được mã hóa của các từ trong các chiều của chúng. Trong bài báo này, chúng tôi tìm hiểu các biểu diễn vectơ truy vấn sử dụng mạng nơ-ron tích chập (CNN) được đào tạo trên các biểu diễn vectơ từ. CNN ban đầu được phát minh cho mục đích thị giác máy tính và gần đây đã được chứng minh là hữu ích trong nhiều tác vụ xử lý ngôn ngữ tự nhiên và truy xuất thông tin, chẳng hạn như phân tích cú pháp ngữ nghĩa, mô hình hóa câu, xếp hạng tài liệu, độ tương tự tài liệu và định dạng lại truy vấn. Nhiệm vụ của chúng tôi khác với các câu nhúng theo nghĩa là các truy vấn thường ngắn và không có cấu trúc. Nó cũng khác với các truy vấn nhúng ở chỗ nó không sử dụng

nhiều loại tính năng khác nhau, chẳng hạn như dữ liệu nhập qua hoặc phiền tìm kiếm.

Trong dự án này, chúng tôi trình bày cách tiếp cận của mình để xác định ý định truy vấn như một nhiệm vụ phân loại đa lớp với các biểu diễn vectơ truy vấn được trích xuất làm các đặc điểm. Chúng tôi trình bày hiệu quả của cách tiếp cận này trong các thiết lập thử nghiệm khác nhau. Kết quả cho thấy các vectơ truy vấn thu được từ mô hình CNN được đào tạo trước hoạt động tốt trong việc phát hiện ý định truy vấn và tương đương với các đặc điểm bag-of-word. Hơn nữa, mặc dù chúng được học tự động từ dữ liệu đào tạo, chúng vượt trội hơn đáng kể so với đường cơ sở dựa trên quy tắc được thiết kế cẩn thận.

3.1.3 Phương pháp

Trong phần này, chúng tôi mô tả mô hình đề xuất của chúng tôi và một tập hợp hệ thống con-trasive mà chúng tôi sử dụng làm cơ sở để phát hiện ý định truy vấn.

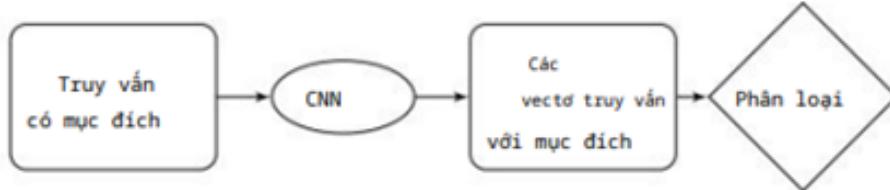
3.1.3.1. Giới thiệu phương pháp của chúng tôi

1. Kiến trúc mô hình

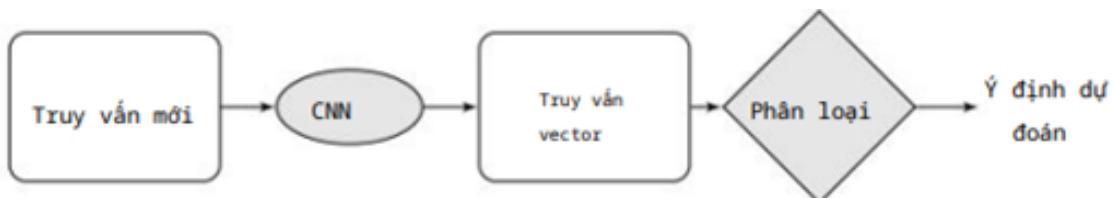
Mô hình đề xuất của chúng tôi có hai bước chính:

B1: Đào tạo các tham số mô hình trong thời gian ngoại tuyến

B2: Chạy mô hình trên các truy vấn mới trong thời gian trực tuyến



Hình 1: Thời gian đào tạo/ ngoại tuyến



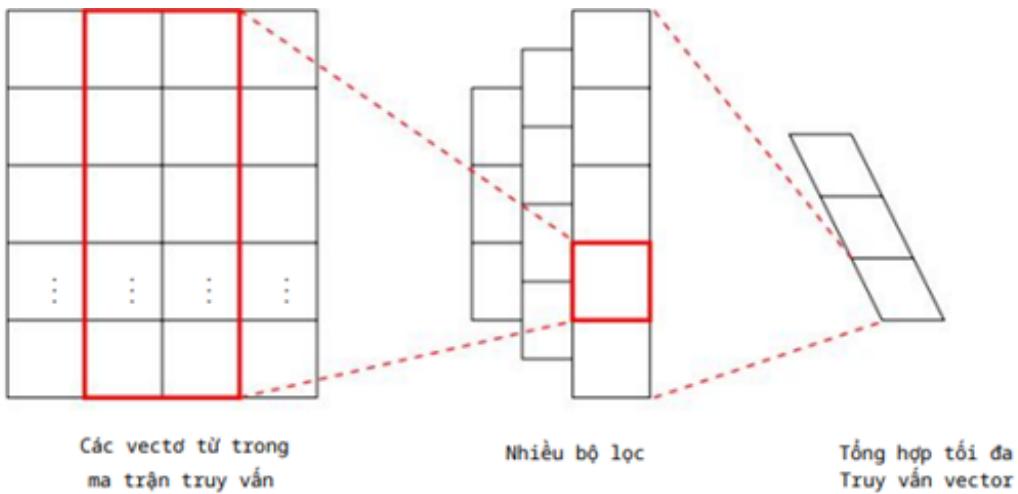
Hình 2: Thời gian kiểm tra/ trực tuyến

Trong quá trình đào tạo, chúng tôi sử dụng các truy vấn được gắn nhãn để tìm hiểu các tham số của mạng nơ-ron tích chập và bộ phân loại ý định. Trong thời gian chạy, chúng tôi chuyển truy vấn qua hai thành phần này để phát hiện ý định của nó

2. Mạng nơ-ron tích chập (CNN)

Với một truy vấn, mục tiêu của thành phần mạng nơ-ron tích chập là tìm biểu diễn vectơ của nó.

Chúng tôi sử dụng kiến trúc CNN của Collobert et al và triển khai biến thể của nó. Chúng tôi đã sửa đổi một chút mô hình trong giai đoạn thử nghiệm để có được các biểu diễn truy vấn sau lớp gộp tối đa bằng cách di chuyển lại lớp softmax. Việc có các biểu diễn vectơ truy vấn sau đó sẽ giúp chúng tôi kết hợp chúng với các tính năng truy vấn khác như mô hình ngôn ngữ.



Hình 3: Model CNN

3.1.4 Cách thực hiện

Các biểu diễn truy vấn được đào tạo trên các vectơ từ được đào tạo trước được cập nhật trong quá trình đào tạo CNN. Chúng tôi sử dụng các vectơ word2vec có sẵn công khai được đào tạo trên Google News. Trong giai đoạn đào tạo, các tham số của mô hình CNN được học bằng cách truyền nhiều bộ lọc qua các vectơ từ và sau đó áp dụng hoạt động gộp tối đa theo thời gian để tạo các tính năng được sử dụng trong một lớp softmax được kết nối đầy đủ. Cuối cùng, chúng tôi sử dụng hàm mất mát entropy chéo để học các tham số của mô hình. Chúng tôi sử dụng dropout để điều chỉnh sự thay đổi của các tham số bằng cách đặt ngẫu nhiên một số trọng số bằng không để ngăn chặn quá trình khớp quá mức.

3.1.4.1. Phương pháp đối chiếu

Chúng tôi đã so sánh phương pháp của mình với cả mô hình dựa trên quy tắc và mô hình phân loại sử dụng đặc trưng từ các vector từ hoặc mô hình ngôn ngữ của câu.

- **Phương pháp dựa trên quy tắc (Rule-based method)**

Phương pháp dựa trên quy tắc được thiết kế trong một công cụ tìm kiếm thương mại để gắn nhãn ý định của truy vấn. Phương pháp này sử dụng một tập hợp các biểu thức chính quy được tạo thủ công và tự động. Ngoài ra, nó còn sử dụng một công cụ gắn thẻ thực thể để liên kết các từ trong truy vấn với các thực thể tên tương ứng. Khi các truy vấn có mẫu câu tương tự nhau, phương pháp này hoạt động khá tốt. Ví dụ, nếu truy vấn là “tình hình đường Điện Biên Phủ?” hoặc “Điện Biên Phủ như thế nào?”, ý định sẽ được nhận diện là “tình trạng giao thông” vì các truy vấn này chứa thực thể là vị trí và từ khóa đặc biệt. Các từ khóa này chủ yếu được các chuyên gia thu thập thủ công. Tuy nhiên, hệ thống dựa trên quy tắc thiếu khả năng mở rộng. Khi xuất hiện một truy vấn

mỗi không khớp với bất kỳ mẫu nào trước đó, hệ thống dựa trên quy tắc không thể gán ý định cho nó. Do đó, nó sẽ bỏ sót truy vấn ngay cả khi truy vấn chỉ là một biến thể nhỏ của các mẫu trước.

- **Đặc trưng Bag-of-Words**

Là một phương pháp cơ bản khác để phát hiện ý định truy vấn, chúng tôi sử dụng các đặc trưng bag-of-words trong mô hình phân loại. Mô hình ngôn ngữ được xem là một phương pháp cơ bản mạnh mẽ cho hầu hết các ứng dụng xử lý ngôn ngữ tự nhiên (NLP). Trong phương pháp này, chúng tôi xem xét các N-grams (n-gram) trong truy vấn làm đặc trưng và huấn luyện một bộ phân loại để học các tham số. Hơn nữa, các truy vấn có cùng từ sẽ có xác suất cao có cùng ý định. Tuy nhiên, mô hình này sẽ bỏ qua các N-grams tương tự nếu chúng không xuất hiện trong dữ liệu huấn luyện.

- **Đặc trưng vecto tích lũy**

Tích lũy các embedding từ của truy vấn là một tập hợp đặc trưng đơn giản khác. Mục tiêu là tìm một embedding đại diện cho truy vấn và sử dụng nó làm đặc trưng để huấn luyện bộ phân loại ý định. Thay vì đưa các vector từ của truy vấn vào một mạng nơ-ron tích chập, chúng ta có thể đơn giản lấy các vector từ của mỗi từ trong truy vấn và cộng chúng lại với nhau (Sum w2v) hoặc lấy trung bình của chúng (Average w2v). Embedding truy vấn kết quả sẽ có cùng kích thước với các vector từ. Trong các thí nghiệm của chúng tôi, chúng tôi sử dụng các embedding word2vec.

3.1.4.2. So sánh độ hiệu quả của 2 phương pháp (CNN và Rule-Based)

Dưới đây chúng tôi đã lập bảng so sánh giữa 2 phương pháp như sau:

Method	Accuracy(%)	Avg. Precision	Avg. Recall	Avg. F-score
Rule-based	75	0.46	0.39	0.40
Sum w2v	74.5	0.48	0.34	0.36
Average w2v	75.2	0.49	0.34	0.36
Unigram	81.3	0.56	0.45	0.47
Unigram+bigram	81.2	0.56	0.44	0.46
CNN	81.6	0.62	0.47	0.50

Hình 4: Kết quả Intent Detection với 125 lớp ý định cấp thấp

Nhận xét bảng 1:

- + **Phương pháp Rule-based** đạt độ chính xác 75% với F-score trung bình là 0.40, cho thấy khả năng nhận diện ý định ở mức khá.
- + **Phương pháp CNN** vượt trội với độ chính xác 81.6%, cùng F-score trung bình đạt 0.50. Đây là kết quả tốt nhất trong các phương pháp so sánh, cho thấy CNN có khả năng phân loại tốt hơn khi xử lý nhiều lớp ý định.
- + **Các phương pháp khác** (Sum w2v, Average w2v, Unigram, Unigram+bigram) có độ chính xác và F-score trung bình thấp hơn so với CNN.

Kết luận: CNN là phương pháp hiệu quả nhất cho bài toán phát hiện ý định với số lượng lớn các lớp ý định, vượt trội hơn so với phương pháp Rule-based và các phương pháp dựa trên vector từ.

Method	Accuracy(%)	Avg. Precision	Avg. Recall	Avg. F-score
Rule-based	77	0.45	0.34	0.36
Sum w2v	86	0.49	0.24	0.28
Average w2v	85.9	0.49	0.25	0.29
Unigram	89.9	0.71	0.37	0.43
Unigram+bigram	89.9	0.68	0.34	0.40
CNN	90.3	0.56	0.43	0.47

Hình 5: Kết quả Intent Detection với 14 lớp ý định cấp cao

Nhận xét bảng 2:

- + **Phương pháp Rule-based** đạt độ chính xác 77% với F-score trung bình 0.36, cho thấy hiệu quả khá trong việc nhận diện ý định cấp cao.
- + **Phương pháp CNN** đạt độ chính xác 90.3%, với F-score trung bình là 0.47, cao nhất trong các phương pháp so sánh. Điều này khẳng định ưu thế của CNN trong bài toán nhận diện ý định với ít lớp hơn.
- + **Các phương pháp Unigram và Unigram+bigram** cũng đạt kết quả khá cao, đặc biệt là Unigram với F-score trung bình 0.43.

Kết luận: CNN tiếp tục là phương pháp hiệu quả nhất khi số lượng lớp ý định ít hơn, thể hiện độ chính xác và F-score vượt trội so với phương pháp Rule-based và các phương pháp khác.

***Kết luận:** Sau khi so sánh CNN và các phương pháp khác, tôi đã chọn CNN vì nó cho thấy hiệu quả vượt trội trong việc nhận diện ý định. CNN đạt độ chính xác và F-score cao nhất trong cả hai bảng, cho thấy khả năng phân loại tốt hơn so với phương pháp Rule-based và các phương pháp dựa trên vector từ. Điều này đặc biệt quan trọng khi xử lý nhiều lớp ý định phức tạp, giúp tăng cường độ chính xác và ổn định cho hệ thống.

3.1.5 Thí nghiệm

Các truy vấn được thu thập từ nhật ký của một công cụ tìm kiếm thương mại. Các ý định truy vấn được xây dựng dựa trên hệ thống phân loại được thiết kế riêng, với hơn 3800 ý định ở cấp thấp nhất. Ở cấp cao hơn của hệ thống phân loại này, có 350 danh mục ý định. Bảng 3 cho thấy một mẫu các danh mục ý định

Bảng 1: Phân loại ý định cấp cao và cấp thấp

Ý định cấp cao	Ý định cấp thấp
Bộ phim	Xếp hạng
	Dàn diễn viên
	Chiều dài của phim
	Ngày phát hành
Con người	Vợ chồng
	Ngày sinh
	Chiều cao
	Nghề nghiệp

Trong thí nghiệm của chúng tôi, chúng tôi đã chọn 10,000 truy vấn mà các nhãn vàng của chúng xuất hiện ít nhất 10 lần trong tập dữ liệu đã gán nhãn. Thông kê về dữ liệu của chúng tôi được thể hiện trong bảng 4 bên dưới.

3.1.5.1. Thiết lập thí nghiệm

Bảng 2: Thông kê dữ liệu truy vấn

Thông tin	Giá trị
Câu truy vấn	10.000
Lớp ý định cấp thấp	125
Lớp ý định cấp cao	14

Trong các thí nghiệm của chúng tôi, để thực hiện phân loại, chúng tôi sử dụng phương pháp phân chia 10 lần (10-fold cross-validation) với mô hình standard Random Forest (số lượng cây $n_{estimators} = 100$) trong bộ công cụ scikit-learn.

Trong mô hình CNN, chúng tôi sử dụng 3 bộ lọc với kích thước cửa sổ lần lượt là 2, 3 và 4, mỗi kích thước cửa sổ có 100 bản đồ đặc trưng. Những kích thước cửa sổ này sẽ giúp nắm bắt thông tin của bigram, trigram và 4-gram từ truy vấn. Chúng tôi áp dụng kỹ thuật dropout để điều chỉnh với tỷ lệ dropout là 0.5, đồng thời sử dụng hàm kích hoạt ReLU và các mini-batches có kích thước 50.

Chúng tôi báo cáo các số đo hiệu suất bao gồm độ chính xác (accuracy), độ chính xác trung bình (precision), độ nhạy (recall), và điểm F (F-score). Độ chính xác của phương pháp được tính như sau:

Accuracy

$$\text{Độ chính xác (accuracy)} = \frac{\text{Số ý định được dự đoán chính xác}}{\text{Tổng số ý định}}$$

Độ chính xác (precision), độ nhạy (recall) và điểm F (F-score) được tính cho từng lớp ý định (cả cấp độ thấp và cấp độ cao) và sau đó lấy trung bình để có được các chỉ số tổng thể. Độ chính xác và độ nhạy của một lớp ý định i (ví dụ: xếp hạng phim) được tính như sau:

Precision

$$\text{Precision}_i = \frac{\text{Số lượng ý định được dự đoán đúng với ý định } i}{\text{Tổng số ý định dự đoán với ý định } i}$$

Recall

$$\text{Recall}_i = \frac{\text{Số lượng ý định được dự đoán đúng với ý định } i}{\text{Tổng số truy vấn được chú thích bởi con người với ý định } i}$$

Điểm F (F-Score) cho lớp ý định i có thể được tính là trung bình điều hòa của độ chính xác (precision) và độ nhạy (recall):

F1_Score

$$\text{F-score}_i = 2 \times \frac{\text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}$$

3.1.5.2. Tóm tắt

Để đánh giá tính hữu ích của phương pháp phát hiện truy vấn được đề xuất của chúng tôi, chúng tôi đã tiến hành 2 thí nghiệm phân loại và phân nhóm. Đối với các thí nghiệm phân loại, chúng tôi xác thực mô hình CNN bằng cách sử dụng các biện pháp phân loại tiêu chuẩn và so sánh nó với các đường cơ sở tương phản. Đối với nhiệm vụ phân cụm, chúng tôi sử dụng biểu diễn truy vấn dưới dạng các tính năng để nhóm các truy vấn.

3.1.6 Kết luận

Để hiểu ý định truy vấn tìm kiếm của người dùng, chúng tôi đã sử dụng biểu diễn vectơ truy vấn được học bởi mô hình mạng nơ-ron tích chập trong tác vụ phân loại. Đóng góp chính của công trình của chúng tôi là chứng minh rằng chỉ sử dụng vectơ truy vấn làm đặc điểm là đủ để phân loại truy vấn và đơn giản hóa nỗ lực thiết kế đặc điểm. Các vectơ truy vấn cung cấp cho chúng tôi một biểu diễn ngữ nghĩa của các truy vấn, giúp phát hiện ý định truy vấn. Chúng cũng hữu ích để nhóm các truy vấn.

3.2 Trích xuất thực thể có tên (NER)

3.2.1 Nguyên tắc cơ bản về Trích xuất Thực thể (NER - Named Entity Recognition)

Named Entity Recognition (NER), hay còn gọi là nhận diện thực thể có tên, là một trong những nhiệm vụ quan trọng nhất trong xử lý ngôn ngữ tự nhiên (NLP). Mục tiêu của NER là phát hiện và phân loại các thực thể được nhắc đến trong văn bản vào các loại cụ thể, chẳng hạn như tên người (Person), tổ chức (Organization), địa điểm (Location), thời gian (Time), và nhiều loại thực thể khác. Trong bối cảnh tiếng Việt, loại thực thể "Địa điểm" (LOC) thường bao gồm các địa danh, thành phố, con đường và tên quốc gia, phù hợp với việc nhận diện thông tin vị trí trong truy vấn của khách hàng.

Cách hoạt động của NER:

- Phân đoạn và nhận diện thực thể: Hệ thống chia văn bản thành các token (từ hoặc cụm từ nhỏ hơn) và phân tích từng token để xác định xem nó có thuộc vào loại thực thể nào không.
- Phân loại thực thể: Nếu token là một thực thể, mô hình NER sẽ gán cho nó một nhãn tương ứng, chẳng hạn như LOC cho địa điểm, PER cho tên người, hay ORG cho tổ chức. Các mô hình

NER hiện đại thường sử dụng mạng nơ-ron sâu (deep learning), chẳng hạn như mô hình BERT, ELECTRA, và các mô hình Transformer khác, được huấn luyện trên dữ liệu ngôn ngữ lớn để tự động học ngữ cảnh và mối quan hệ giữa các từ.

Thách thức trong NER đối với tiếng Việt

- Đặc trưng về ngữ pháp và cấu trúc:** Tiếng Việt là ngôn ngữ đơn âm, không phân cách giữa các từ và cụm từ bằng khoảng trắng như tiếng Anh, khiến việc tách câu và nhận diện thực thể trở nên phức tạp hơn.
- Thiếu dấu hiệu phân cách thực thể rõ ràng:** Các thực thể trong tiếng Việt thường không có dấu hiệu phân cách rõ ràng như chữ in hoa ở đầu các danh từ riêng (như trong tiếng Anh), dẫn đến thách thức trong nhận diện các thực thể.

3.2.2 Lựa chọn mô hình và lý do

Mô hình NlpHUST/ner-vietnamese-electra-base là một mô hình dựa trên kiến trúc ELECTRA, được thiết kế đặc biệt cho tác vụ nhận diện thực thể (NER) trong tiếng Việt. Mô hình này được huấn luyện trên một tập dữ liệu lớn bằng tiếng Việt, cho phép nó nắm bắt tốt các đặc điểm ngữ nghĩa và ngữ pháp của ngôn ngữ, từ đó cải thiện khả năng nhận diện và phân loại các thực thể như tên địa điểm, tên người, và tổ chức.

Lý do chọn mô hình

- Khả năng hiểu ngữ cảnh tiếng Việt:** Mô hình đã được tối ưu hóa cho ngữ liệu tiếng Việt, giúp nhận diện chính xác các thực thể địa điểm trong các câu phức tạp và đa nghĩa. Điều này rất quan trọng trong việc trích xuất thông tin từ các câu truy vấn khách hàng, nơi mà ngữ cảnh có thể ảnh hưởng đến ý nghĩa.
- Độ chính xác cao:** Các nghiên cứu chỉ ra rằng mô hình ELECTRA thường đạt độ chính xác cao hơn so với các mô hình truyền thống khác trong tác vụ NER. Điều này giúp cải thiện chất lượng của các dự đoán về thực thể địa điểm.
- Khả năng xử lý các tên riêng:** Mô hình này có khả năng phát hiện tốt các danh từ riêng, như tên địa danh, giúp việc trích xuất thông tin chính xác hơn. Sự kết hợp giữa cấu trúc Transformer và khả năng phân tích ngữ nghĩa giúp mô hình hiểu rõ hơn về các mối quan hệ giữa các từ trong câu.
- Tính linh hoạt:** Mô hình có thể được điều chỉnh và huấn luyện lại trên dữ liệu cụ thể để nâng cao hiệu suất cho các ngữ cảnh đặc thù, điều này rất hữu ích trong việc tùy biến cho các ứng dụng cụ thể như dịch vụ khách hàng.

3.2.3 Tiền xử lý dữ liệu

- Chuẩn bị dữ liệu đầu vào: Đọc và chuẩn bị văn bản tiếng Việt từ file hoặc database, đảm bảo văn bản được chuẩn hóa (loại bỏ các ký tự đặc biệt, kiểm tra mã hóa UTF-8).

- Chia câu thành token: Sử dụng tokenizer của ELECTRA để chia câu thành các token phù hợp. Tokenizer giúp biến đổi văn bản thành các mã số đầu vào mà mô hình có thể xử lý.

Ví dụ:

Tiền xử lý dữ liệu

```
1 from transformers import AutoModelForTokenClassification # type: ignore
2 model =
3     AutoModelForTokenClassification.from_pretrained("NlpHUST/ner-vietnamese-electra-base")
```

3.2.4 Triển khai NER với loại LOC

- Dự đoán thực thể:

Dầu tiên, sử dụng mô hình "NlpHUST/ner-vietnamese-electra-base" để phân tích và dự đoán thực thể trong câu. Mô hình sẽ trả về danh sách các từ, mỗi từ có nhãn thực thể và độ tin cậy của nhãn.

Triển khai NER

```
1 from transformers import AutoTokenizer # type: ignore
2 tokenizer = AutoTokenizer.from_pretrained("NlpHUST/ner-vietnamese-electra-base")
3
4 from transformers import AutoModelForTokenClassification # type: ignore
5 model = AutoModelForTokenClassification.from_pretrained("NlpHUST/ner-vietnamese-electra-base")
6
7 nlp = pipeline("ner", model=model, tokenizer=tokenizer)
```

Kết quả từ bước này sẽ là danh sách các từ hoặc cụm từ trong câu thuộc tất cả các lớp (PERSON, LOCATION, ORGANIZATION,...), mỗi từ hoặc cụm từ kèm theo nhãn và độ tin cậy dự đoán.

- Khởi tạo biến lưu kết quả thực thể vị trí (LOCATION)

Chúng ta sẽ khởi tạo các danh sách locations, current_location và biến current_score để lưu các thông tin cần thiết:

- + locations: Danh sách chứa các cụm từ là địa điểm và độ tin cậy
- + current_location: Danh sách tạm thời để lưu các từ thuộc cùng một cụm địa điểm
- + current_score: Biến lưu độ tin cậy cao nhất cho cụm địa điểm hiện tại

Khởi tạo biến

```

1 locations = [] # Danh sách lưu các cụm địa điểm cuối cùng
2 current_location = [] # Danh sách tạm để lưu từ thuộc một cụm địa điểm
3 current_score = 0 # Biến lưu độ tin cậy cao nhất của cụm hiện tại

```

- Lọc các thực thể loại LOCATION

Duyệt qua từng thực thể mà mô hình dự đoán, kiểm tra nhãn entity của nó:

- + Nếu nhãn là I - LOCATION, thêm từ vào current_location và cập nhật current_score bằng độ tin cậy cao nhất
- + Nếu gặp từ không phải I-LOCATION, ghép current_location thành cụm, lưu vào locations với current_score, và reset current_location và current_score.

Lọc các location

```

1 for result in ner_results:
2     if result['entity'] == 'I-LOCATION':
3         # Thêm từ vào danh sách hiện tại
4         current_location.append(result['word'])
5         current_score = max(current_score, result['score']) # Lưu độ tin cậy cao nhất
6     else:
7         # Nếu không còn thực thể địa điểm, lưu kết quả
8         if current_location:
9             locations.append({
10                 'name': ' '.join(current_location), # Ghép các từ lại thành một cụm
11                 'score': current_score
12             })
13         # Reset danh sách
14         current_location = []
15         current_score = 0

```

- Xử lý ngoại lệ:

Nếu câu kết thúc mà vẫn còn từ trong current_location, ta ghép lại thành cụm cuối cùng và lưu vào locations để đảm bảo không bị thiếu các cụm từ ở cuối câu.

Xử lý ngoại lệ

```

1 # Kiểm tra nếu còn địa điểm chưa lưu
2 if current_location:
3     locations.append({
4         'name': ' '.join(current_location),
5         'score': current_score
6     })

```

3.2.5 Hậu xử lý kết quả

Để đảm bảo kết quả chính xác và trực quan, các bước hậu xử lý bao gồm:

- Ghép các token lại thành thực thể hoàn chỉnh: Do mỗi từ được xử lý như một token riêng biệt, cần ghép lại các từ liên tiếp có cùng nhãn để tạo thành cụm từ hoàn chỉnh, như “Phan Văn Tri” cho tên đường.
- Tính toán độ tin cậy: Để đảm bảo tính chính xác, lấy giá trị xác suất của các token và tính độ tin cậy trung bình của mỗi thực thể.

Hậu xử lý

```

1 # Hiển thị kết quả
2 for loc in locations:
3     print(f"\nTên địa điểm: {loc['name']}, Độ tin cậy: {loc['score']:.4f}")

```

3.2.6 Các siêu tham số huấn luyện mô hình

Trong quá trình huấn luyện mô hình nhận diện thực thể có tên (NER) cho loại thực thể vị trí (LOCATION), các siêu tham số dưới đây đóng vai trò quyết định đến hiệu suất và độ chính xác của mô hình. Mỗi siêu tham số được thiết lập nhằm đạt sự cân bằng giữa độ chính xác và hiệu quả tính toán.

3.2.6.1 Learning Rate (Tốc độ học)

learning_rate = 5e-05: Đây là một trong những tham số quan trọng nhất trong việc huấn luyện mô hình. Learning rate nhỏ giúp giảm rủi ro nhảy qua cực tiểu cục bộ nhưng có thể làm chậm quá trình hội tụ. Với giá trị 5e-05, mô hình tiến hành cập nhật trọng số từng bước nhỏ, giúp đạt hiệu quả cao mà không rơi vào nguy cơ quá mức (overfitting).

3.2.6.2 Train Batch Size (Kích thước batch huấn luyện)

train_batch_size = 16: Batch size trong huấn luyện ảnh hưởng đến tốc độ và tính ổn định của quá trình cập nhật. Với batch size là 16, mô hình có khả năng tối ưu hóa nhanh hơn trong mỗi lần cập nhật trọng số so với việc sử dụng batch nhỏ hơn, giúp tiết kiệm bộ nhớ GPU mà vẫn đạt được độ chính xác tốt.

3.2.6.3. Eval Batch Size (Kích thước batch đánh giá)

eval_batch_size = 4: Batch size trong đánh giá (evaluation) được giữ nhỏ hơn để giảm tải cho bộ nhớ khi kiểm tra mô hình trên tập dữ liệu đánh giá. Điều này giúp cải thiện tốc độ đánh giá mà vẫn đảm bảo tính chính xác.

3.2.6.4 Seed (Hạt khởi tạo ngẫu nhiên)

seed = 42: Seed giúp duy trì tính nhất quán trong việc tạo ra các giá trị ngẫu nhiên trong quá trình huấn luyện, từ đó tạo ra một quá trình huấn luyện có thể tái lập. Seed này đảm bảo rằng khi huấn luyện lại mô hình với cùng dữ liệu và cấu hình, kết quả đạt được sẽ gần như giống nhau.

3.2.6.5 Optimizer (Bộ tối ưu hóa)

optimizer = Adam với betas=(0.9, 0.999) và epsilon=1e-08: Adam là bộ tối ưu hóa phổ biến cho mô hình học sâu, giúp tối ưu hóa tốc độ hội tụ bằng cách kết hợp giữa Moment và RMSprop. Các giá trị beta điều khiển trọng số các yếu tố động lượng của tốc độ học, và epsilon giúp tránh chia cho số 0 khi cập nhật trọng số. Bộ tối ưu hóa này mang lại tốc độ hội tụ nhanh và độ chính xác cao cho bài toán NER.

3.2.6.6 Learning Rate Scheduler (Điều chỉnh tốc độ học)

lr_scheduler_type = linear: Bộ điều chỉnh learning rate theo dạng tuyến tính giảm dần giúp mô hình hội tụ ổn định, đảm bảo tốc độ học sẽ giảm từ từ trong suốt quá trình huấn luyện để tránh bị dao động quá mức trong các epoch cuối.

3.2.6.7 Number of Epochs (Số lượng epoch)

num_epochs = 10.0: Epoch là số lần toàn bộ tập dữ liệu huấn luyện được chạy qua mô hình. Với 10 epoch, mô hình có đủ số lần lặp để học tốt các mẫu trong dữ liệu mà không dễ dàng bị quá khớp (overfitting).

3.2.7 Đánh giá và tối ưu hóa

3.2.7.1 Đánh giá mô hình

Trong quá trình đánh giá mô hình nhận diện thực thể có tên (NER) với mục tiêu nhận diện vị trí (LOCATION), các chỉ số đánh giá như Loss, Precision, Recall, và F1 Score đóng vai trò rất quan trọng trong việc xác định khả năng và độ chính xác của mô hình. Dưới đây là phân tích chi tiết dựa trên các kết quả nhận được:

- **LOSS:** $Loss = 0.0580$: Loss (hàm mất mát) cho thấy mức độ sai lệch của dự đoán so với nhãn thực tế trong quá trình huấn luyện. Giá trị loss thấp như 0.0580 cho thấy mô hình đang học khá tốt từ dữ liệu, ít xảy ra lỗi và có độ chính xác cao trong việc phân loại các thực thể loại vị trí. Việc giữ loss thấp là điều kiện cần thiết để các chỉ số khác (Precision, Recall, F1 Score) có thể đạt hiệu quả cao.
- **Precision (Độ chính xác)**
- + Location Precision = 0.9353: Precision là tỷ lệ giữa số thực thể loại vị trí (LOCATION) được mô hình nhận diện đúng và tổng số thực thể mô hình dự đoán là vị trí.

+ Với độ chính xác 93.53%, mô hình có khả năng dự đoán đúng khá cao khi nhận diện các thực thể địa điểm. Điều này nghĩa là trong các trường hợp mô hình xác định một thực thể là vị trí, hơn 93% trong số đó thực sự là địa điểm theo nhãn thực tế.

+ Precision cao cho thấy mô hình ít xảy ra dự đoán sai đối với các thực thể loại vị trí, giúp giảm tỷ lệ sai sót khi gán nhãn cho các từ/cụm từ không phải là địa điểm thành địa điểm.

- Recall (Khả năng tìm kiếm)

+ Location Recall = 0.9377: Recall là tỷ lệ giữa số thực thể vị trí mà mô hình nhận diện đúng và tổng số thực thể vị trí trong tập dữ liệu thực tế.

+ Với giá trị Recall đạt 93.77%, mô hình có khả năng bao phủ tốt các thực thể vị trí trong dữ liệu. Điều này cho thấy mô hình đã nhận diện đúng và không bỏ sót quá nhiều thực thể vị trí, giúp đảm bảo rằng phần lớn các địa danh được mô hình phát hiện và gán nhãn đúng.

+ Recall cao như thế này đặc biệt quan trọng trong các ứng dụng thực tế, giúp đảm bảo mô hình có khả năng nhận diện hầu hết các thông tin quan trọng về vị trí.

- F1 Score

+ Location F1 Score = 0.9365: F1 Score là trung bình điều hòa giữa Precision và Recall, giúp đưa ra một thước đo cân bằng khi mô hình có thể có độ chính xác hoặc khả năng tìm kiếm tốt nhưng không đồng đều.

+ F1 Score đạt 93.65% cho thấy mô hình không chỉ chính xác mà còn có khả năng bao quát tốt, đồng thời cân bằng giữa việc giảm sai sót và không bỏ sót các thực thể quan trọng. Đây là chỉ số quan trọng vì nó kết hợp cả Precision và Recall để đánh giá hiệu quả tổng thể của mô hình trong nhận diện thực thể vị trí.

3.2.7.2 Đánh giá tổng thể Các chỉ số cao này cho thấy mô hình có hiệu suất rất tốt trong việc nhận diện các thực thể loại vị trí với:

+ Độ chính xác cao (Precision) giúp giảm thiểu dự đoán sai cho các thực thể khác thành vị trí.

+ Khả năng tìm kiếm tốt (Recall) giúp nhận diện phần lớn các địa danh cần thiết trong dữ liệu đầu vào.

3.3 Google Map Auto Complete

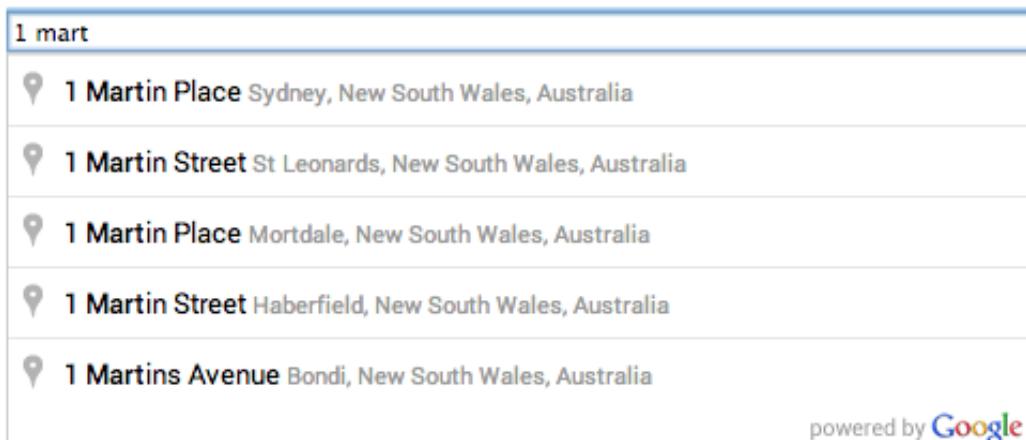
3.3.1 Giới thiệu

Google Maps Autocomplete là một tính năng mạnh mẽ trong Google Maps API, cho phép tự động hoàn thành địa chỉ khi người dùng nhập vào ô tìm kiếm. Tính năng này giúp cải thiện trải nghiệm người dùng, giảm thiểu lỗi nhập liệu và tiết kiệm thời gian tìm kiếm địa điểm. Trong bối cảnh dự án dự đoán tình trạng tắc nghẽn giao thông, việc tích hợp Google Maps Autocomplete sẽ được hiện thực trên đầu ra của mô hình NER sau khi nhận dạng địa chỉ trong nội dung của khách hàng

3.3.2 Chức năng của Google Maps Autocomplete

Google Maps Autocomplete cung cấp các gợi ý địa điểm dựa trên ký tự người dùng đã nhập. Các chức năng chính của nó bao gồm:

- + Autocomplete thêm trường nhập văn bản vào ứng dụng, và theo dõi trường đó để xem các mục nhập ký tự. Khi người dùng nhập văn bản, tính năng tự động hoàn thành sẽ trả về các cụm từ gợi ý địa điểm ở dạng danh sách lựa chọn thả xuống. Khi người dùng chọn một địa điểm trong danh sách, thông tin về địa điểm sẽ được trả về đối tượng tự động hoàn thành và có thể được truy xuất theo ứng dụng của bạn.



Hình 6: Trường văn bản tự động hoàn thành và danh sách chọn

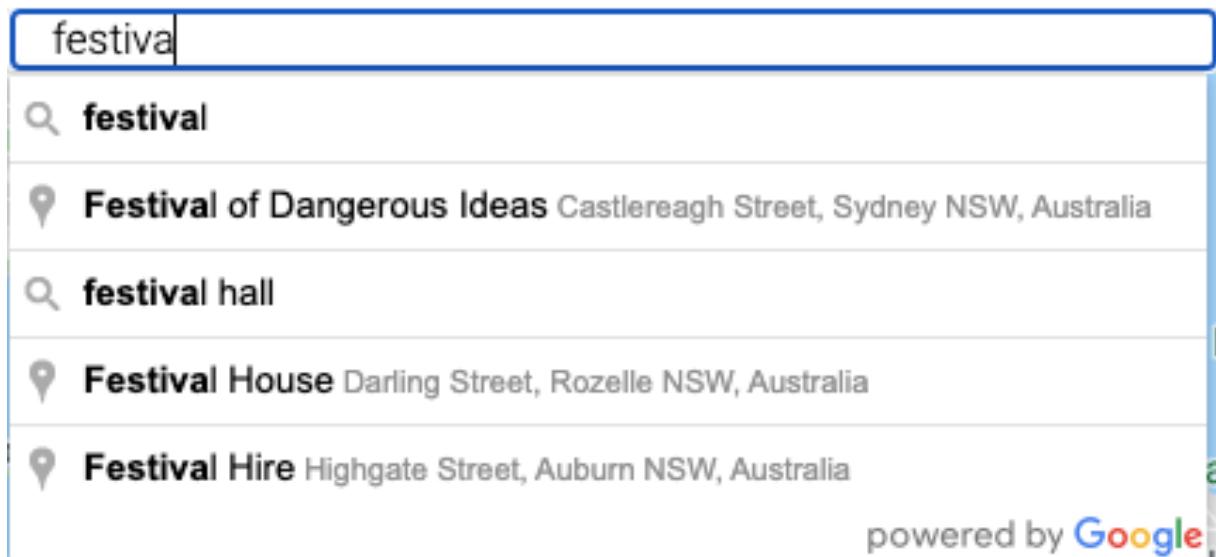
1 Martin Place, Sydney, New South Wales, Australia			
Street address	1 Martin Pl		
City	Sydney		
State	NSW	Zip code	2000
Country	Australia		

Hình 7: Biểu mẫu địa chỉ đã hoàn thành

- + SearchBox thêm trường nhập văn bản vào trang web của bạn, tương tự như Autocomplete. Có những điểm khác biệt như sau:

- Điểm khác biệt chính nằm ở kết quả xuất hiện trong danh sách chọn. SearchBox nguồn cung cấp một danh sách mở rộng các dự đoán, có thể bao gồm các địa điểm (như được xác định bởi API Địa điểm) cùng với các cụm từ tìm kiếm được đề xuất. Ví dụ: nếu người dùng nhập "pizza ở new", danh sách chọn có thể bao gồm cụm từ "pizza ở New York, NY" cũng như tên của nhiều cửa hàng pizza.
- SearchBox cung cấp ít tuỳ chọn hơn Autocomplete để hạn chế nội dung tìm kiếm. Trong trường hợp trước, bạn có thể nghiêng về một LatLangBounds nhất định. Trong trường hợp thứ hai,

bạn có thể hạn chế tìm kiếm ở một quốc gia cụ thể và một số loại địa điểm cụ thể, cũng như đặt giới hạn.



Hình 8: SearchBox hiển thị cụm từ tìm kiếm và thông tin dự đoán về địa điểm

3.3.3 Lợi ích của việc tích hợp Google Maps Autocomplete trong dự án

Việc tích hợp Google Maps Autocomplete vào mô hình nhận dạng thực thể có tên (NER) không chỉ hỗ trợ người dùng hoàn thành địa chỉ nhanh chóng, mà còn mang lại nhiều lợi ích thiết thực cho quá trình dự đoán và giảm thiểu lỗi. Những ưu điểm chính bao gồm:

- + **Nâng cao độ chính xác của dữ liệu:** Tự động hoàn thành địa chỉ giúp giảm thiểu lỗi nhập liệu do người dùng gõ sai hoặc thiếu thông tin địa điểm, qua đó nâng cao tính chính xác của dữ liệu đầu vào cho mô hình dự đoán tắc nghẽn giao thông.
- + **Tối ưu trải nghiệm người dùng:** Với Autocomplete, người dùng chỉ cần nhập vài ký tự đầu tiên và có thể dễ dàng chọn đúng địa chỉ từ danh sách gợi ý. Điều này giúp tiết kiệm thời gian và mang lại trải nghiệm thuận tiện hơn, đặc biệt trong các tình huống cần nhập địa chỉ nhiều lần hoặc trên thiết bị di động. Hơn nữa, Autocomplete cũng sẽ tự động hoàn thành nội dung của người dùng sau khi dùng voice chat. Đây là 1 lợi ích cực kì tốt vì nó khắc phục trực tiếp vào điểm yếu của ứng dụng này, đó là xử lý âm thanh đầu vào.
- + **Tăng hiệu quả cho các mô hình dự đoán:** Thông qua các gợi ý địa điểm được tối ưu, dữ liệu đầu vào được chuẩn hóa và dễ dàng xử lý hơn trong các mô hình NER, cải thiện hiệu quả trong quá trình dự đoán tình trạng giao thông.

3.3.4 Tích hợp Google Maps Autocomplete vào dự án

Quá trình tích hợp Google Maps Autocomplete vào dự án dự đoán tắc nghẽn giao thông bao gồm các bước cơ bản sau:

- + **Cấu hình API:** Để sử dụng Google Maps Autocomplete, cần tạo một dự án trên Google Cloud Platform, sau đó kích hoạt Google Maps Places API và tạo khóa API. API này cho phép dự án của chúng tôi truy xuất các gợi ý địa điểm từ cơ sở dữ liệu của Google Maps.
- + **Tạo trường Autocomplete:** Trường Autocomplete được thêm vào giao diện nhập liệu của người dùng, cho phép theo dõi các ký tự đầu vào. Khi người dùng nhập, API sẽ tự động gợi ý các địa điểm khả thi trong danh sách thả xuống. Bên cạnh đó, Autocomplete cũng sẽ đảm nhận trách nhiệm xử lý dữ liệu sau khi thực hiện NER bằng mô hình nếu người dùng thực hiện voice chat.
- + **Lọc và xử lý đầu ra từ API:** Khi người dùng chọn một địa điểm, dữ liệu được lưu lại và có thể chuyển trực tiếp sang mô hình NER để tiếp tục xử lý. Điều này giúp hệ thống dễ dàng nắm bắt địa chỉ chính xác và cung cấp thông tin dự đoán về tình trạng giao thông tại địa điểm đó.
- + **Tùy chỉnh các tham số tìm kiếm:** Dựa trên yêu cầu cụ thể của dự án, có thể tùy chỉnh các tham số như giới hạn quốc gia, loại địa điểm, hoặc phạm vi tìm kiếm nhằm tối ưu hóa kết quả trả về. Điều này cho phép hệ thống xử lý chính xác các địa điểm và con đường quan trọng, hỗ trợ người dùng dễ dàng xác định vị trí và nhận diện những khu vực giao thông phức tạp hoặc dễ ùn tắc.

3.3.5 Kết luận

Google Maps Autocomplete là một công cụ hữu ích và có tính ứng dụng cao trong các dự án yêu cầu độ chính xác và hiệu quả về địa chỉ nhập liệu, đặc biệt trong dự án dự đoán tắc nghẽn giao thông. Việc sử dụng Autocomplete kết hợp với mô hình NER giúp tiết kiệm thời gian cho người dùng và đảm bảo chất lượng đầu vào, từ đó cải thiện độ chính xác trong dự đoán và nâng cao hiệu quả của dự án.

3.4 Lấy tọa độ của địa điểm

3.4.1 Công nghệ sử dụng

Google Maps Geocoding API là một dịch vụ của Google cung cấp khả năng chuyển đổi giữa địa chỉ (như tên đường, thành phố, quốc gia) và tọa độ địa lý (latitude và longitude). API này là một công cụ mạnh mẽ giúp xác định vị trí chính xác của một địa chỉ hoặc lấy tên địa điểm từ tọa độ, tạo nền tảng cho các ứng dụng cần định vị địa lý, bản đồ, hoặc chỉ đường.

API có hai chức năng chính:

- **Geocoding:** Chuyển đổi từ địa chỉ văn bản (ví dụ: số nhà, tên đường, tên thành phố) thành tọa độ (latitude và longitude).

Geocoding: Địa chỉ → Latitude, Longitude

- **Reverse Geocoding:** Chuyển đổi từ tọa độ sang địa chỉ dễ đọc, giúp hiển thị địa chỉ chi tiết dựa trên vị trí hiện tại hoặc vị trí đã xác định.

Reverse Geocoding: Latitude, Longitude → Địa chỉ

3.4.2 Thông số API

API URL

<https://maps.googleapis.com/maps/api/geocode/>

Bảng 3: Tham số Geocoding API

Tham số	Ý nghĩa
outputFormat	Dịnh dạng xuất response, có thể là json hoặc xml
address	Địa chỉ chính xác được Google quy định, địa chỉ này lấy từ API Google Places AutoComplete.
extra_computations	Tham số yêu cầu Google tính toán thêm các vị trí landmark lân cận phù hợp với loại đường hoặc địa điểm người dùng muốn đến. Bao gồm 2 thông số, ADDRESS_DESCRIPTORS và BUILDING_AND_ENTRANCES.
key	API key

3.4.3 Ví dụ

REQUEST

<https://maps.googleapis.com/maps/api/geocode/json?address=Nhà thờ Đức Bà, Công xã Paris, Bến Nghé, District 1, Ho Chi Minh City, Vietnam&key=abcs1234567>

RESPONSE

```
{
  "results": [
    {
      "address_components": [
        {
          "long_name": "01",
          "short_name": "01",
          "types": [
            "street_number"
          ]
        },
        ...
      ],
      ...
    }
  ]
}
```

RESPONSE(CONT)

```
{  
    "long_name": "Công xã Paris",  
    "short_name": "Công xã Paris",  
    "types": [  
        "route"  
    ],  
},  
{  
    "long_name": "Quận 1",  
    "short_name": "Quận 1",  
    "types": [  
        "administrative_area_level_2",  
        "political"  
    ],  
},  
{  
    "long_name": "Hồ Chí Minh",  
    "short_name": "Hồ Chí Minh",  
    "types": [  
        "administrative_area_level_1",  
        "political"  
    ],  
},  
{  
    "long_name": "Vietnam",  
    "short_name": "VN",  
    "types": [  
        "country",  
        "political"  
    ],  
},  
{  
    "long_name": "70000",  
    "short_name": "70000",  
    "types": [  
        "postal_code"  
    ]  
},  
}
```

RESPONSE(CONT)

```
        ]
    }
],


"address_descriptor": {
    "areas": [
        {
            "containment": "WITHIN",
            "display_name": {
                "language_code": "en",
                "text": "Bến Nghé"
            },
            "place_id": "ChIJ483BI0kvdTERGk6ufQ_-3Nc"
        },
        {
            "containment": "NEAR",
            "display_name": {
                "language_code": "en",
                "text": "People's Committee of District 1"
            },
            "place_id": "ChIJGXTMOTcvdTERupmRw5uvQbk"
        }
    ],
    "landmarks": [
        {
            "display_name": {
                "language_code": "en",
                "text": "Saigon Central Post Office"
            },
            "place_id": "ChIJRxocgI4udTERspse3PV-LA",
            "spatial_relationship": "NEAR",
            "straight_line_distance_meters": 97.49818420410156,
            "types": [
                "establishment",
```

RESPONSE(CONT)

```
        "finance",
        "point_of_interest",
        "post_office"
    ],
},
{
    "display_name": {
        "language_code": "en",
        "text": "Diamond Plaza"
    },
    "place_id": "ChIJ75zeyjcvdTERzMaTAtkcoYk",
    "spatial_relationship": "NEAR",
    "straight_line_distance_meters": 145.3543395996094,
    "types": [
        "establishment",
        "point_of_interest",
        "shopping_mall"
    ]
},
{
    "display_name": {
        "language_code": "en",
        "text": "People's Committee of District 1"
    },
    "place_id": "ChIJGXTMOTcvdTERyEB4_0y58uA",
    "spatial_relationship": "NEAR",
    "straight_line_distance_meters": 98.79651641845703,
    "types": [
        "establishment",
        "point_of_interest"
    ]
},
{
    "display_name": {
```

RESPONSE(CONT)

```
    "language_code": "en",
    "text": "Ho Chi Minh City Post Office"
},
"place_id": "ChIJG0t3qXcmdTERJuyQfLLtQD0",
"spatial_relationship": "NEAR",
"straight_line_distance_meters": 103.8990859985352,
"types": [
    "establishment",
    "finance",
    "point_of_interest",
    "post_office"
]
},
{
    "display_name": {
        "language_code": "en",
        "text": "CGV Vincom Đồng Khởi"
    },
    "place_id": "ChIJpQeTd0gvdTERe-_j6WTtops",
    "spatial_relationship": "NEAR",
    "straight_line_distance_meters": 381.5486145019531,
    "types": [
        "establishment",
        "movie_theater",
        "point_of_interest"
    ]
}
],
},
"formatted_address": "01 Công xã Paris, Bến Nghé, Quận 1, Hồ Chí Minh 70000, Vietnam",
"geometry": {
    "location": {
        "lat": 10.7797855,
        "lng": 106.6990189
    }
},
```

RESPONSE(CONT)

```

    "location_type": "ROOFTOP",
    "viewport": {
        "northeast": {
            "lat": 10.7808395302915,
            "lng": 106.7006990302915
        },
        "southwest": {
            "lat": 10.7781415697085,
            "lng": 106.6980010697085
        }
    },
    "partial_match": true,
    "place_id": "ChIJJUSTY5jcvdTERRVvtbJNZT-g",
    "plus_code": {
        "compound_code": "QMHX+WJ District 1, Ho Chi Minh City, Vietnam",
        "global_code": "7P28QMHX+WJ"
    },
    "types": [
        "church",
        "establishment",
        "place_of_worship",
        "point_of_interest",
        "tourist_attraction"
    ]
},
],
"status": "OK"
}

```

Sau khi đã có respond, có lấy các giá trị *landmarks.straight_lines_distant_meters*. Các giá trị trong respond gồm: 97.49818420410156, 145.3543395996094, 98.79651641845703, 103.8990859985352, 381.5486145019531

Tính trung bình của các giá trị này, ta được: 164.8.., đây là giá trị radius để truyền vào Utraffic.

3.5 Lấy toạ độ của các camera.

3.5.1 Giới thiệu

UTraffic (UTraffic: Urban Traffic Estimation System): Hệ thống dự báo tình trạng giao thông đô thị dựa vào dữ liệu từ cộng đồng đã nghiên cứu và đề xuất các giải pháp hợp lý, vận dụng sức mạnh của công nghệ 4.0 và sức mạnh của cộng đồng nhằm giải quyết các vấn đề cốt lõi, mới và chưa được giải quyết tốt bởi các giải pháp hiện có, góp phần giảm thiểu ùn tắc giao thông (UTGT), nâng cao sự an toàn và thuận lợi cho người tham gia giao thông. Utraffic bao gồm các giải pháp nổi bật sau:

- Dữ liệu về giao thông được thu thập từ nhiều nguồn, trong đó có nguồn quan trọng là từ cộng đồng thông qua ứng dụng trên điện thoại di động và ứng dụng web.
- Ước lượng và dự báo tình trạng giao thông (TTGT) chính xác và kịp thời bằng các kỹ thuật phân tích dữ liệu lớn và học máy.
- Cảnh báo ùn tắc giao thông (UTGT) cho người dân một cách kịp thời.
- Hỗ trợ chức năng tìm đường (routing) hiệu quả có xét đến tình trạng giao thông.
- Cung cấp thông tin thống kê, dự báo, hỗ trợ công tác ra quyết định của nhà quản lý.

Tham số:

Bảng 4: Tham số và ý nghĩa

Tham số	Ý nghĩa
lat	Kinh độ của địa điểm được chọn
lng	Vĩ độ của địa điểm được chọn
radius	Bán kính dùng để xác định các đoạn đường gần nhất đối với địa điểm được chọn

Dữ liệu trả về: là json chứa các tham số đáng lưu tâm

Bảng 5: Tham số và ý nghĩa

Tham số	Ý nghĩa
data.cordinations	Toạ độ của điểm bắt đầu và kết thúc của đoạn đường được chọn

3.5.2 Ví dụ

REQUEST

```
api.bktraffic.com/api/cameras/get-nearby-cameras?lat=10.792894534527097&lng=106.65333299109086&radius=100
```

RESPONSE

```
"code": 200,  
"message": "success",  
"data": [  
    {  
        "_id": "587478d8b807da0011e33cf3",  
        "Title": "TTH 28.1",  
        "Code": "TTH 28.1",  
        "Location": {  
            "type": "Point",  
            "coordinates": [  
                106.653594374657,  
                10.7928387592472  
            ]  
        },  
        "SnapshotUrl": "http://camera.thongtingiaothong.vn/api/snapshot/587478d8b807da0011e33cf3",  
        "CamType": "tth_axis",  
        "Publish": "True",  
        "ManagementUnit": null,  
        "CamStatus": "NOT_IMAGE",  
        "PTZ": "False",  
        "Angle": 300,  
        "DisplayName": "Nút giao Bảy Hiền 1 (Cách Mạng Tháng Tám)",  
        "VideoUrl": null,  
        "VideoStreaming": 0,  
        "DataId": null,  
        "NodeId": "ecf3a4ae-3187-42f5-9305-8e22130b2456",  
        "Path": "/root/vdms/tangthu/data/layerdata/camera/7b32f46f-8ede-49a4-9ec7-298da1404ae9",  
        "CreatedDate": "2017-05-20T04:14:33.754Z",  
        "ModifiedDate": "2022-12-28T04:02:42.545Z",  
    }]
```

RESPONSE(CONT)

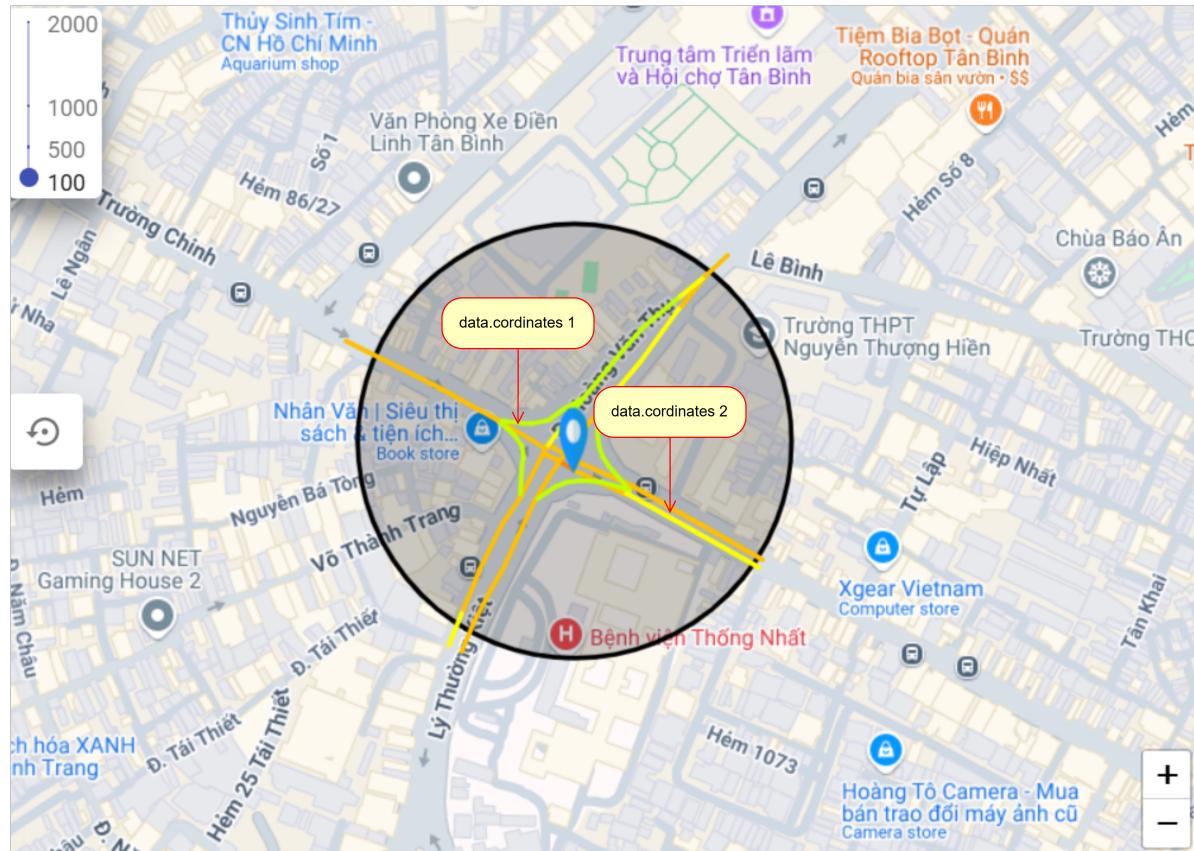
```
"DynamicProperties": [],
    "District": "Quận Tân Bình"
},


{
    "_id": "58747923b807da0011e33cf5",
    "Title": "TTH 28.2",
    "Code": "TTH 28.2",
    "Location": {
        "type": "Point",
        "coordinates": [
            106.653615832329,
            10.7931970867358
        ]
    },
    "SnapshotUrl": "http://camera.thongtingiaothong.vn/api/snapshot/58747923b807da0011e33cf5",
    "CamType": "tth",
    "Publish": "True",
    "ManagementUnit": null,
    "CamStatus": "UP",
    "PTZ": "False",
    "Angle": 210,
    "DisplayName": "Nút giao Bảy Hiền 2 (Hoàng Văn Thụ)",
    "VideoUrl": null,
    "VideoStreaming": 0,
    "DataId": null,
    "NodeId": "a0ee6be4-4d59-41f2-a606-6e5ce69be700",
    "Path": "/root/vdms/tangthu/data/layerdata/camera/36425b93-52bf-4bff-9c0f-d745efb52a9e",
    "CreatedDate": "2017-05-20T04:14:35.105Z",
    "ModifiedDate": "2024-02-11T12:11:03.207Z",
    "DynamicProperties": [],
    "District": "Quận Tân Bình"
}
]
```

Các coordinates của các cameras là:

```
{"coordinates": [106.653594374657, 10.7928387592472]}
```

```
{"coordinates": [106.653615832329, 10.7931970867358]}
```



Hình 9: Bảng đồ mô phỏng điểm XYZ và tìm 4 đường gần nhất

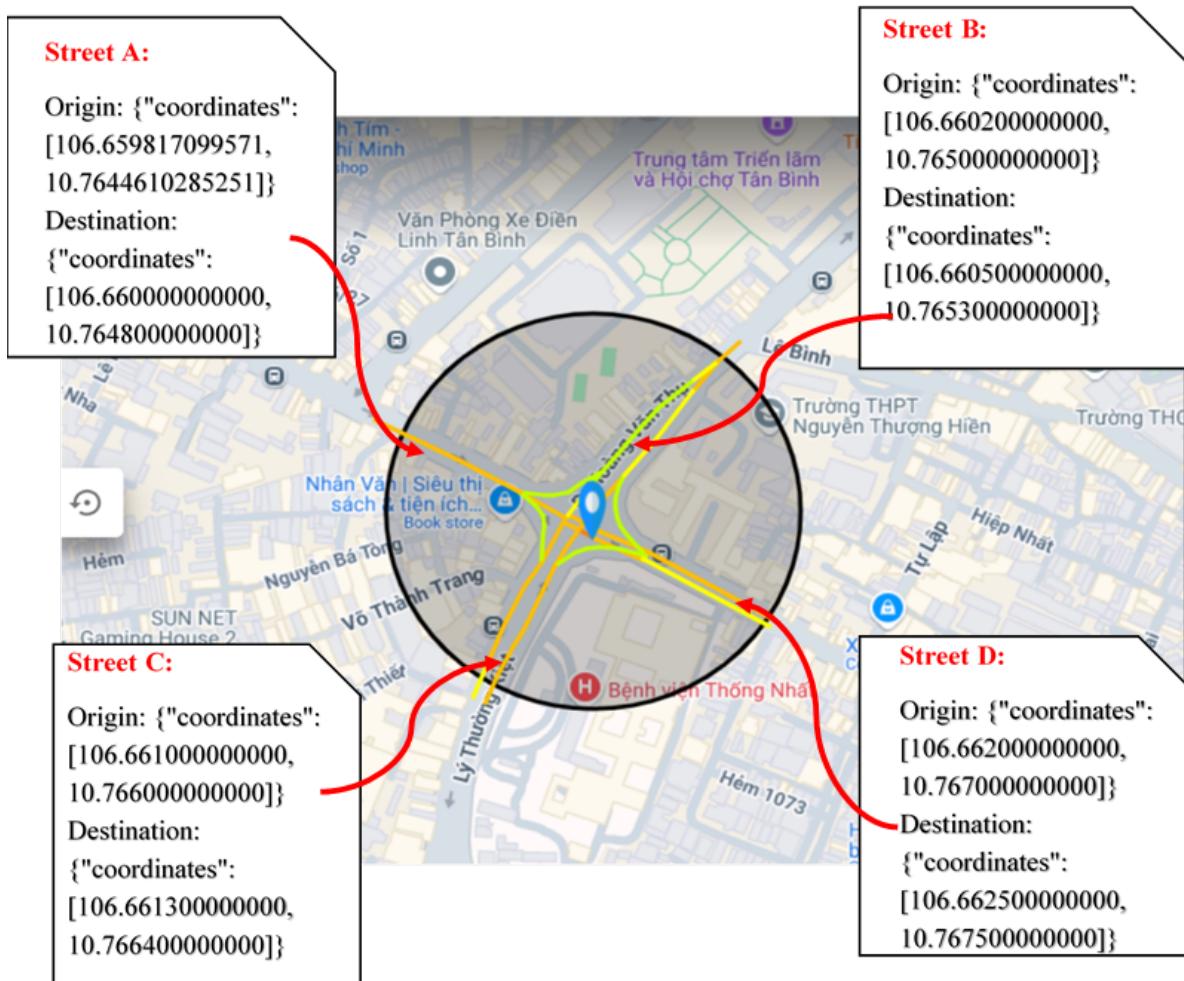
3.5.3 Quy Trình gọi API

API URL

<https://bktraffic.com/home/apis/cameras/get-nearby-cameras>

3.6 Google Map Distance Matrix

Sau khi đã có dữ liệu về những con đường gần mục tiêu cần truy vấn, tiến hành thay đổi định dạng Json, ta có dạng dữ liệu được mô phỏng như Hình 10 dưới đây:



Hình 10: Bảng đồ mô phỏng điểm XYZ với 4 đường gần nhất
(dữ liệu chỉ để mô phỏng không đúng với thực tế)

3.6.1 API Distance Matrix là gì?

Nó là một API do Google Map cung cấp. API Distance Matrix sử dụng số lượng điểm xuất phát (origin) và điểm đến (destination) tùy ý, đồng thời trả về khoảng cách distance và thời gian (duration) di chuyển giữa mỗi điểm xuất phát và tất cả điểm đến được yêu cầu, bắt đầu từ điểm xuất phát đầu tiên trong yêu cầu và tiếp tục từng bước đến điểm xuất phát tiếp theo.

Ví dụ: nếu yêu cầu của bạn chỉ định OriginA là điểm xuất phát, còn DestinationA là điểm đến, thì yêu cầu đó sẽ trả về khoảng cách và thời gian di chuyển từ OriginA -> DestinationA.

3.6.2 Hoạt động của API và ứng dụng vào mô hình

Dựa trên các điểm dữ liệu đã thu thập, chúng ta sẽ sử dụng ***API Google Maps Distance Matrix*** để truy xuất tình hình giao thông bằng cách thực hiện các bước sau:

Dầu tiên, chúng ta sẽ lấy tọa độ của điểm xuất phát (origin) và điểm đích (destination). Sau đó, những tọa độ này sẽ được đưa vào ***API Google Maps Distance Matrix*** để tìm kiếm các thông tin quan trọng như khoảng cách (distance), thời gian di chuyển (duration), và thời gian di chuyển trong điều kiện giao thông hiện tại (duration_in_traffic) cho các đoạn đường gần điểm cần khảo sát.

Với dữ liệu này, chúng ta có thể đánh giá chính xác hơn về tình hình giao thông và lên kế hoạch di chuyển một cách hiệu quả. ***Fetch Distance Matrix API***

API URL

```
https://maps.googleapis.com/maps/api/distancematrix/json?origins={origin}&
destinations={destination}&traffic_model={traffic_model}&departure_time=
{departure_time}&key={api_key}
```

Bảng 6: Tham số của Distance Matrix API

Tham số	Ý nghĩa
origin	Tọa độ vị trí bắt đầu trên con đường đó (cooperation = [lat, lon])
destination	Tọa độ của vị trí kết thúc trên con đường đó (cooperation = [lat, lon])
traffic_model	Mô hình giao thông: <ul style="list-style-type: none"> • best_guess: Ước lượng tốt nhất về thời gian di chuyển. • pessimistic: Ước lượng thời gian di chuyển dài nhất. • optimistic: Ước lượng thời gian di chuyển ngắn nhất.
depature_time	Thời gian khởi hành

Cụ thể, với tọa độ xuất phát là Oringin = (10.764461, 106.659817) và tọa độ đích là Destination = (10.779843, 106.696082), chúng ta sẽ thực hiện một yêu cầu đến API này.

DEMO API URL

```
https://maps.googleapis.com/maps/api/distancematrix/json?origins=10.
7644610285251,106.659817099571&destinations=10.779843,106.696082&traffic_model=
best_guess&departure_time=now&key=YOUR_API_KEY
```

Để áp dụng cho các tuyến đường còn lại xuất phát từ một điểm cụ thể, bạn có thể tiến hành gọi API Distance Matrix để truy xuất thông tin giao thông và thời gian di chuyển tới từng điểm đích khác nhau.

Dưới đây là phần hiện thực code

Python Code

```

1 def get_traffic_situation(source, destination):
2     slat, slng = source
3     elat, elng = destination
4     traffic_model = "best_guess"
5     departure_time = "now"
6     url = (
7         f"https://maps.googleapis.com/maps/api/distancematrix/json"
8         f"?origins={slat},{slng}&destinations={elat},{elng}"
9         f"&traffic_model={traffic_model}&departure_time={departure_time}"
10        f"&key={GOOGLE_API_KEY}"
11    )
12    response = requests.get(url)
13    if response.status_code == 200 and response.json():
14        rows = response.json().get('rows')
15        if rows and len(rows) > 0:
16            elements = rows[0]['elements']
17            if elements and len(elements) > 0:
18                duration = elements[0]['duration']['value']
19                duration_in_traffic = elements[0]['duration_in_traffic']['value']
20                situation = get_route_status_base_on_duration(duration, duration_in_traffic)
21                distance = elements[0]['distance']['value'] / 1000
22                return {
23                    'situation': situation,
24                    'average_velocity': f"{distance /
25                                (elements[0]['duration_in_traffic']['value'] / 3600)} km/h"
26                }
27    return None

```

Kết quả response như sau

Kết quả mô phỏng

```
{
  "A": {
    "Name": "Street A",
    "distance": "2 km",
    "duration": "10 min",
    "duration_in_traffic": "11 min",
    "status": "OK"
  },
  "B": {
    "Name": "Street B",
    "distance": "24 km",
    "duration": "40 min",
    "duration_in_traffic": "80 min",
    "status": "OK"
  },
  "C": {
    "Name": "Street C",
    "distance": "5 km",
    "duration": "5 min",
    "duration_in_traffic": "30.5 min",
    "status": "OK"
  },
  "D": {
    "Name": "Street D",
    "distance": "13 km",
    "duration": "22 min",
    "duration_in_traffic": "30 min",
    "status": "OK"
  }
}
```

3.6.3 Xác định ngưỡng threshold và tính toán các giá trị đặc trưng của từng con đường truy xuất được từ điểm ban đầu

Sau khi trích xuất được các đặc trưng từ các tuyến đường xung quanh điểm cần truy vấn, chúng ta sẽ tiến hành tính toán các đặc trưng như tốc độ (*velocity*) và tình trạng giao thông (*situation*). Đồng

thời, chúng ta sẽ thiết lập các ngưỡng (*threshold*) để đánh giá tình trạng giao thông trên những đoạn đường đó, nhằm cung cấp những dự đoán chính xác cho người dùng.

Được tiến hành theo 3 bước

Bước 1. Các trạng thái trên đường được qui ước dựa vào threshold

Công thức ngưỡng

$$\text{Threshold} = \frac{\text{duration}}{\text{duration_in_traffic}} \quad (*)$$

Từ công thức (*), ta ước tính và quy ước mức độ giao thông dựa vào ngưỡng như sau

Bảng 7: Quy ước trạng thái giao thông

Trạng thái	Ngưỡng Threshold
Ùn tắc	> 0.9
Tương đối ùn tắc	(0.7, 0.9]
Khá thông thoáng	(0.4, 0.7]
Thông thoáng	≤ 0.4

Bước 2. Tính toán các đặc trưng từ dữ liệu vừa tìm

Bảng 8: Công thức tính toán các đặc trưng của từng con đường

Công thức tính	Ý nghĩa
$\text{Free_flow_traffic} = \frac{\text{distance}}{\text{duration}}$	Vận tốc lưu thông khi đường thông thoáng
$\text{Velocity} = \text{Current_flow_traffic} = \frac{\text{distance}}{\text{duration_in_traffic}}$	Vận tốc lưu thông hiện tại trên đường

Áp dụng công thức (*) đưa ra ration của từng tuyến đường, sau đó so sánh với threshold ở Bảng 5 để đưa ra trạng thái của từng tuyến đường đó (situation).

Kết quả được mô tả như sau:

Street A	Street B	Street C	Street D
“Velocity”: 10.9 km/h “Ratio”: 0.91 “Situation”: Ùn tắc	“Velocity”: 18 km/h “Ratio”: 0.50 “Situation”: Khá thông thoáng	“Velocity”: 9.8 km/h “Ratio”: 0.16 “Situation”: Thông thoáng	“Velocity”: 26 km/h “Ratio”: 0.73 “Situation”: Tương đối ùn tắc

Hình 11: Kết quả của từng tuyến đường

Bước 3. Đưa vào mô hình GPT-4 để đưa ra dự đoán

Mô hình GPT-4, với khả năng xử lý ngôn ngữ tự nhiên mạnh mẽ, có thể tiếp nhận dữ liệu đầu vào dưới dạng văn bản và đưa ra các nhận định, phân tích hoặc đề xuất dựa trên kiến thức đã được huấn luyện.

Cách thức hoạt động của GPT-4

- Nhập dữ liệu:** Tất cả các thông tin thu thập được từ bước trước như các đặc trưng giao thông như Velocity, status, ration của 1 đường nào đó

- **Xử lý thông tin:** GPT-4 sẽ phân tích các thông tin này để nhận diện các mô hình và xu hướng trong dữ liệu. Ví dụ, nếu một con đường có tỷ lệ lưu thông cao và nằm trong ngưỡng ùn tắc, mô hình có thể đưa ra nhận định rằng "Đường này đang bị tắc nghẽn".
- **Đưa ra dự đoán và đề xuất:** Dựa trên các phân tích, GPT-4 sẽ cung cấp các nhận định về tình trạng giao thông hiện tại và đưa ra các đề xuất phù hợp. Các đề xuất có thể bao gồm:
 - **Chỉ dẫn lộ trình:** Khuyến nghị người dùng chọn lộ trình thay thế để tránh ùn tắc.
 - **Thời gian dự kiến:** Cung cấp thời gian dự kiến cho hành trình dựa trên các yếu tố như lưu lượng giao thông hiện tại và dự đoán trong tương lai.
 - **Tình trạng giao thông:** Cập nhật người dùng về tình trạng giao thông theo thời gian thực để họ có thể điều chỉnh kế hoạch di chuyển của mình.

Dưới đây là phần hiện thực code

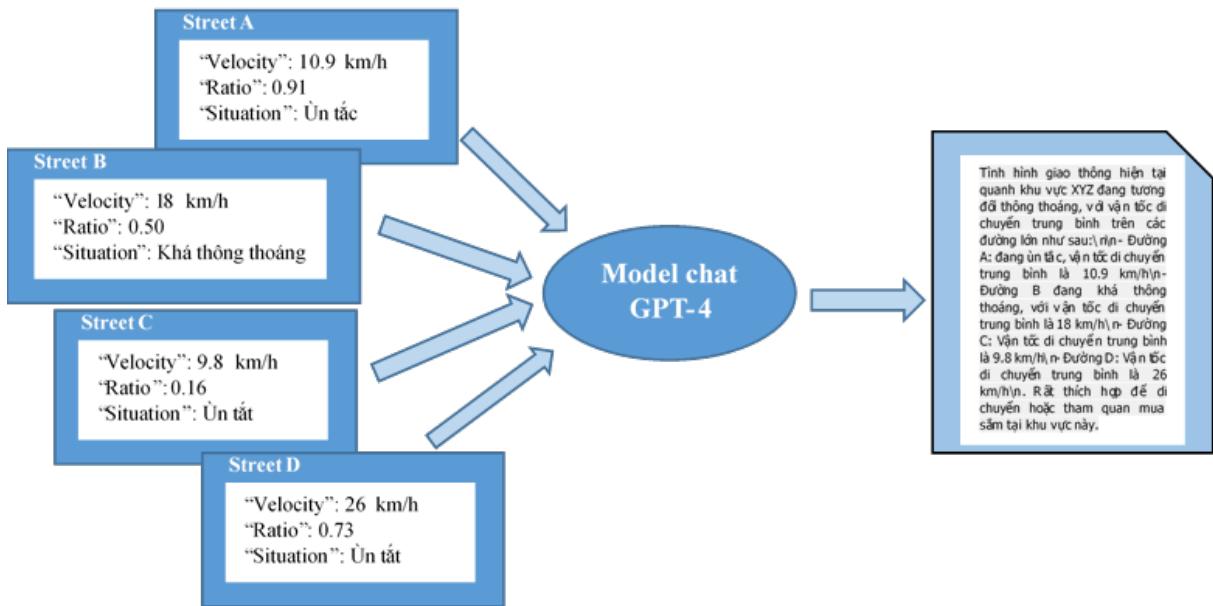
Python Code

```

1 load_dotenv()
2 OPENAI_API_KEY=os.getenv('OPENAI_API_KEY')
3 client = OpenAI(api_key=OPENAI_API_KEY)
4 def analyze_traffic_situation(traffic_situations, location, query):
5     prompt = (
6         f"Tình hình giao thông quanh {location} như sau:\n"
7     )
8     for street, details in traffic_situations.items():
9         prompt += (
10             f"- Đường {street}: Tình trạng giao thông: {details['situation']}, Vận tốc di
11             chuyển: {details['average_velocity']}%\n"
12         )
13     prompt += (
14         query
15     )
16     response = client.chat.completions.create(
17         model="gpt-4",
18         messages=[
19             {"role": "user", "content": prompt}
20         ]
21     )
22     return response.choices[0].message.content

```

Dưới đây là mô hình thực hiện và trả response



Hình 12: Phản hồi vị trí cần truy vấn

4 TÍNH MỚI, TÍNH SÁNG TẠO

Tương tác giọng nói nâng cao:

Hiểu ngữ nghĩa: Vita sẽ được trang bị công nghệ xử lý ngôn ngữ tự nhiên (NLP) tiên tiến, cho phép hiểu và phản hồi các câu hỏi phức tạp từ tài xế, không chỉ là các lệnh đơn giản.

Dự đoán tình trạng giao thông:

Sử dụng các thuật toán học máy, Vita có thể dự đoán tình trạng giao thông trong tương lai dựa trên dữ liệu lịch sử và thời gian thực, giúp tài xế chuẩn bị cho các tình huống có thể xảy ra.

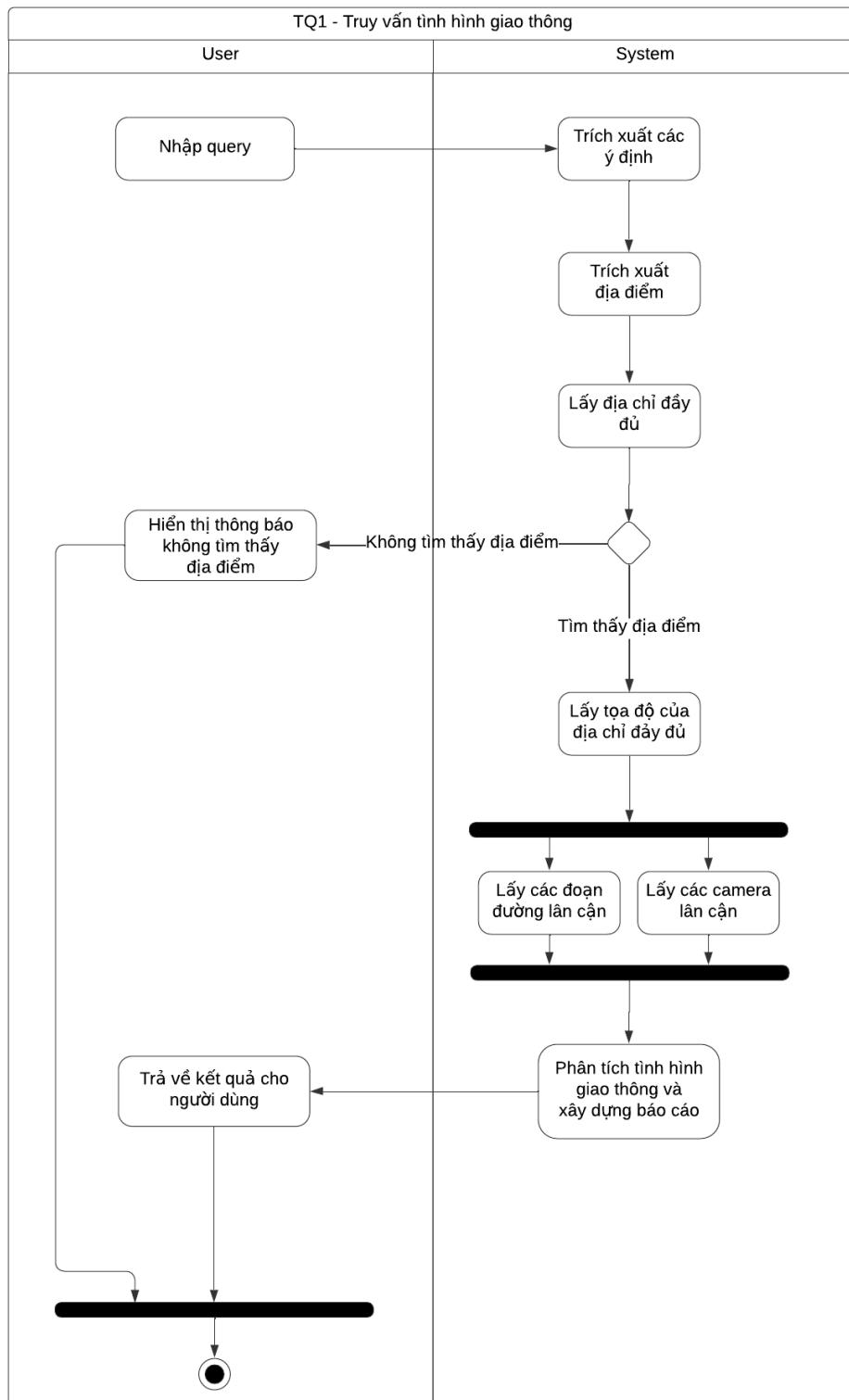
Tích hợp với hệ sinh thái giao thông thông minh:

Vita có khả năng kết nối với các hệ thống giao thông thông minh khác như đèn tín hiệu, camera giám sát giao thông, và các cảm biến trên đường để cung cấp thông tin giao thông chính xác hơn và nhanh chóng hơn.

Cập nhật thông tin theo thời gian thực:

Hệ thống sẽ cung cấp thông tin không chỉ về tình trạng giao thông mà còn về thời tiết, sự kiện đặc biệt, hoặc các yếu tố khác có thể ảnh hưởng đến hành trình, tạo ra một cái nhìn tổng quát về điều kiện di chuyển.

5 MÔ HÌNH HÓA HỆ THỐNG VITA



Hình 13: Activity diagram

6 PHƯƠNG HƯỚNG PHÁT TRIỂN

Phân tích hành vi người dùng:

Vita có thể học hỏi từ thói quen di chuyển của tài xế qua thời gian, cung cấp thông tin tùy chỉnh và các lộ trình ưa thích, giúp cá nhân hóa trải nghiệm giao thông cho từng tài xế.

Chế độ lái xe an toàn:

Vita sẽ tích hợp một chế độ lái xe an toàn, tự động ngắt các thông báo không cần thiết khi tài xế đang trong tình huống lái xe căng thẳng, chỉ tập trung vào những thông tin quan trọng.

Hỗ trợ đa ngôn ngữ và ngữ điệu:

Hỗ trợ nhiều ngôn ngữ và ngữ điệu khác nhau để phục vụ đa dạng người dùng, từ đó mở rộng đối tượng sử dụng và nâng cao trải nghiệm người dùng.

Chia sẻ thông tin giữa người dùng:

Cung cấp tính năng cho phép tài xế chia sẻ thông tin giao thông và lộ trình của họ với bạn bè hoặc gia đình, tạo thành một mạng lưới thông tin giao thông hữu ích và cộng đồng hơn.

7 THAM KHẢO

Tài liệu

- [1] Homa B. Hashemi, “Intent Detection Report,” Intelligent Systems Program, University of Pittsburgh, hashemi@cs.pitt.edu, 2021.
- [2] BKTraffic, “API từ BKTraffic,” truy cập từ: <https://bktraffic.com/home/apis/intro>.
- [3] Google, “Google Map Distance Matrix,” truy cập từ: <https://developers.google.com/maps/documentation/distance-matrix/overview>.
- [4] Google, “Google Map Auto Complete,” truy cập từ: <https://developers.google.com/maps/documentation/javascript/place-autocomplete>.
- [5] Google, “Google Geocoding,” truy cập từ: <https://developers.google.com/maps/documentation/geocoding/overview>.
- [6] OpenAI, “GPT-4 API general availability and deprecation of older models in the Completions API,” truy cập từ: <https://openai.com/api>.
- [7] H. Cao, D. H. Hu, D. Shen, D. Jiang, J.-T. Sun, E. Chen, and Q. Yang, “Context-aware query classification,” In Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 3–10, ACM, 2009.
- [8] J. C. K. Cheung and X. Li, “Sequence clustering and labeling for unsupervised query intent discovery,” In Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, pp. 383–392, ACM, 2012.
- [9] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, “Natural language processing (almost) from scratch,” The Journal of Machine Learning Research, vol. 12, pp. 2493–2537, 2011.
- [10] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, “LIBSVM: A library for large linear classification,” The Journal of Machine Learning Research, vol. 9, pp. 1871–1874, 2008.
- [11] J. Gao, P. Pantel, M. Gamon, X. He, L. Deng, and Y. Shen, “Modeling interestingness with deep neural networks,” In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2014.
- [12] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” arXiv preprint arXiv:1207.0580, 2012.