TRƯỜNG ĐẠI HỌC KINH TẾ

**KHOA THỐNG KÊ – TIN HỌC**

–––––––––––––––––––––––––––––––

**BÁO CÁO THỰC TẬP NHẬN THỨC**

**TÊN ĐỀ TÀI**

**NHẬN DẠNG VÀ PHÂN LOẠI CÁC THỰC THỂ TRONG VĂN BẢN TIẾNG VIỆT**

**Nhóm: 3**

**Sinh viên thực hiện:**

1.  Thành viên: Thái Đức Tân          Lớp: 45K21.1

2.  Thành viên: Lưu Thị Bảo Châu  Lớp: 45K21.2

**Giảng viên hướng dẫn:**  Th.s Nguyễn Thành Thủy

**Đà Nẵng, 6/2021**

**LỜI CẢM ƠN**

Lời cảm ơn đầu tiên chúng em xin gửi đến Quý thầy cô Khoa Thống kê – Tin học trường Đại học Kinh tế - Đại học Đà Nẵng đã truyền dạy cho chúng em những kiến thức trong thời gian qua để chúng em có thể hoàn thành khóa học thực tập nhận thức để có thể tiếp thu được nhiều kiến thức mới. Và hơn hết chúng em xin chân thành cảm ơn thầy Nguyễn Thành Thủy, người đã tận tình hướng dẫn cho em trong suốt quá trình làm đề tài nghiên cứu.

Trong quá trình thực tập, cũng như là trong quá trình làm bài báo cáo thực tập, khả năng nghiên cứu và kinh nghiệm thực tế còn hạn chế nên bài báo cáo sẽ có thiếu sót. Nhóm em rất mong nhận được sự đóng cũng như sự cảm thông, chia sẻ của Quý thầy, cô để chúng em có điều kiện bổ sung, nâng cao kiến thức tốt hơn cho việc học tập, nghiên cứu và công việc sau này.

Cuối cùng chúng em kính chúc Quý thầy cô dồi dào sức khỏe, niềm tin để tiếp tục sự nghiệp cao quý của mình là truyền đạt kiến thức cho thể hệ mai sau.

Chúng em xin chân thành cảm ơn!

**LỜI CAM ĐOAN**

Nhóm chúng em xin cam đoan những kết quả đạt được trong báo cáo này là do nhóm nghiên cứu, tổng hợp và thưc hiện dựa trên sự hướng dẫn của Ths. Nguyễn Thành Thủy, không sao chép lại bất kỳ điều gì của người khác. Những nội dung được trình bày trong báo cáo hoặc là của nhóm, hoặc được tham khảo và tổng hợp từ các nguồn tài liệu khác nhau. Tất cả tài liệu tham khảo, tổng hợp đều được trích dẫn rõ ràng nguồn thông tin ở mục tài liệu tham khảo.

Nhóm xin hoàn toàn chịu trách nhiệm về lời cam đoan của mình. Nếu có điều gì sai trái, nhóm chúng em xin chịu hoàn toàn trách nhiệm.

**MỤC LỤC**

[**MỤC LỤC** iv](#_Toc76719198)

[**DANH MỤC HÌNH ẢNH** vi](#_Toc76719199)

[**DANH MỤC BẢNG BIỂU** vii](#_Toc76719200)

[**DANH MỤC CÁC TỪ VIẾT TẮT** viii](#_Toc76719201)

[**LỜI MỞ ĐẦU** 1](#_Toc76719202)

[**CHƯƠNG 1: TỔNG QUAN VỀ PYTHON VÀ NAMED ENTITY RECOGNITION** 3](#_Toc76719203)

[**1.1** **Ngôn ngữ lập trình Python** 3](#_Toc76719204)

[**1.2** **Named Entity Recognition** 5](#_Toc76719205)

[**CHƯƠNG 2: LÝ THUYẾT** 6](#_Toc76719206)

[**2.1** **Ngôn ngữ lập trình Python** 6](#_Toc76719207)

[2.1.1 Ứng dụng 6](#_Toc76719208)

[2.1.2 Ưu điểm 7](#_Toc76719209)

[2.1.3 Hạn chế 8](#_Toc76719210)

[2.1.4 Ứng dụng của Python vào Machine Learning 8](#_Toc76719211)

[**2.2** **Named Entity Recognition** 9](#_Toc76719212)

[2.2.1 Vấn đề cơ bản trong NER 9](#_Toc76719213)

[2.2.2 Định dạng dữ liệu 11](#_Toc76719214)

[2.2.3 Nhận dạng thực thể 12](#_Toc76719215)

[**2.3** **Nhận dạng đối tượng dựa trên BERT** 18](#_Toc76719216)

[2.3.1 Bidirectional Encoder Representations from Transformers (BERT) 18](#_Toc76719217)

[2.3.2 Nguyên lý hoạt động của BERT 18](#_Toc76719218)

[2.3.3 Ứng dụng của BERT vào NER 18](#_Toc76719219)

[2.3.4 Bộ dữ liệu cho NER 19](#_Toc76719220)

[**2.4** **Giải pháp xây dựng** 20](#_Toc76719221)

[2.4.1 Phân đoạn câu văn - Sentence Segmentation 20](#_Toc76719222)

[2.4.2 Mã hoá các từ - Word Tokenization 20](#_Toc76719223)

[2.4.3 Xác định các từ dừng - Identifying Stop Words 20](#_Toc76719224)

[2.4.4 Phân tích sự phụ thuộc về cú pháp - Dependency Parsing 21](#_Toc76719225)

[2.4.5 Tìm các cụm danh từ - Finding Noun Phrases 21](#_Toc76719226)

[2.4.6 Nhận dạng thực thể được đặt tên - Named Entity Recognition (NER) 21](#_Toc76719227)

[**CHƯƠNG 3: TRIỂN KHAI THỰC HIỆN** 22](#_Toc76719228)

[**3.1** **Nghiên cứu về code có sẵn** 22](#_Toc76719229)

[3.1.1 Github 22](#_Toc76719230)

[3.1.2 Mì AI 22](#_Toc76719231)

[3.1.3 Youtube 22](#_Toc76719232)

[**3.2** **Các cài đặt cần thiết** 23](#_Toc76719233)

[3.2.1 Cài đặt Python và Pytorch tối thiểu 23](#_Toc76719234)

[3.2.2 Cài các thư viện liên quan 23](#_Toc76719235)

[**3.3** **Thực hiện train dữ liệu** 26](#_Toc76719236)

[**3.4** **Thực hiện xử lý tách từ ngôn ngữ Tiếng Việt** 27](#_Toc76719237)

[**3.5** **Gán nhãn các thực thể đã tách (Part-of-speech tagging)** 28](#_Toc76719238)

[**CHƯƠNG 4: KẾT QUẢ** 32](#_Toc76719239)

[**4.1** **Sử dụng mô hình BERT để pre-trained model** 32](#_Toc76719240)

[**4.2** **Kết quả train model** 35](#_Toc76719241)

[**4.3** **Thực hiện nhận diện thực thể bằng mô hình BERT** 38](#_Toc76719242)

[**KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN** 40](#_Toc76719243)

[**TÀI LIỆU THAM KHẢO** 43](#_Toc76719244)

# **DANH MỤC HÌNH ẢNH**

[Hình 1.1 Top các ngôn ngữ lập trình phổ biến 4](#_Toc76719077)

[Hình 1.2 Xếp hạng các ngôn ngữ lập trình của PYPL Index 4](#_Toc76719078)

[Hình 2.1 Nhãn thực thể được gán theo cấu trúc BIO 12](#_Toc76719079)

[Hình 2.2 Nhận dạng tên riêng chỉ từng cá nhân 13](#_Toc76719080)

[Hình 2.3 Nhận dạng tên rút gọn còn 2 thành phần 14](#_Toc76719081)

[Hình 2.4 Nhận dạng tên rút gọn còn 1 thành phần 15](#_Toc76719082)

[Hình 2.5 Nhận dạng tên địa lí 17](#_Toc76719083)

[Hình 2.6 Bộ dữ liệu cho NER 20](#_Toc76719084)

[Hình 3.1 Tạo mô hình ngôn ngữ có mặt nạ 25](#_Toc76718472)

[Hình 3.2 Dự đoán câu tiếp theo 26](#_Toc76718473)

[Hình 3.3 Ví dụ về thẻ POS NLTK 30](#_Toc76718474)

[Hình 4.1 Tiền xử lí bằng word segment 32](#_Toc76718475)

[Hình 4.2 List label và mask gồm các giá trị 0 và 1 33](#_Toc76718476)

[Hình 4.3 Code load model BERT 33](#_Toc76718477)

[Hình 4.4 Kết quả load model BERT 34](#_Toc76718478)

[Hình 4.5 Code train model BERT 35](#_Toc76718479)

[Hình 4.6 Kết quả sau khi train model BERT (Epoch 0 - 1) 36](#_Toc76718480)

[Hình 4.7 Kết quả sau khi train model BERT (Epoch 2 - 3) 37](#_Toc76718481)

[Hình 4.8 Kết quả sau khi train model BERT (Epoch 4 - 5) 37](#_Toc76718482)

[Hình 4.9 Kết quả thực hiện tách và gán nhãn thực thể 39](#_Toc76718483)

**DANH MỤC BẢNG BIỂU**

[Bảng 4.1 Điểm đánh giá sau khi train 6 epochs 38](#_Toc76718695)

# **DANH MỤC CÁC TỪ VIẾT TẮT**

**AI** : Artificial Intelligence

**API** : Application Programming Interface

**BERT** : Bidirectional Encoder Representations from Transformers

**GPU** : Graphics Processing Unit

**ML** : Machine Learning

**NER** : Named Entity Recognition

**NLP** : Natural Language Processing

**NLTK** : Natural Language Toolkit

**NSP** : Next Sentence Prediction

**POS** : Part-of-speech

# **LỜI MỞ ĐẦU**

Ngày nay, cùng với sự phát triển mạnh mẽ của công nghệ phần cứng và truyền thông các hệ thống dữ liệu phục vụ cho các lĩnh vực kinh tế - xã hội cũng không ngừng tăng lên, lượng dữ liệu được tạo ra ngày càng lớn. Sự phong phú về dữ liệu, thông tin cùng với khả năng kịp thời khai thác chúng đã mang đến những năng suất và chất lượng mới cho công tác quản lý, hoạt động kinh doanh,… Nhưng rồi các yêu cầu về thông tin trong các lĩnh vực hoạt động đó, đặc biệt trong lĩnh vực ra làm quyết định, ngày càng đòi hỏi cao hơn, người quyết định không những cần dữ liệu mà còn cần có thêm nhiều hiểu biết, nhiều tri thức để hỗ trợ cho việc ra quyết định của mình. Cho đến những năm 90 của thế kỉ trước, theo cùng nhu cầu của thế giới để xử lí các lớp dữ liệu, hàng loạt các lĩnh vực nghiên cứu về tổ chức các kho dữ liệu và kho thông tin, hỗ trợ giúp ra quyết định, lọc các thông tin chính xác,… và đặc biệt là Xử lý ngôn ngữ tự nhiên (Natural Language Processing).

Kể từ khi ra đời, xử lý ngôn ngữ tự nhiên đã trở thành một trong những hướng nghiên cứu phổ biến trong lĩnh vực trí tuệ nhân tạo và khoa học máy tính. Nhiều kết quả nghiên cứu, ứng dụng xử lý ngôn ngữ tự nhiên trong khoa học, kinh tế, xã hội, luật và các lĩnh vực khác. Xử lý ngôn ngữ tự nhiên bao gồm nhiều hướng nghiên cứu quan trọng, nhận dạng thực thể được đặt tên là một trong số đó. Nhận dạng thực thể trong văn bản tự nhiên là một quá trình phân tách và gắn nhãn các thực thể trong các mẫu dữ liệu tự nhiên trong cơ sở dữ liệu lớn. Công nghệ được áp dụng để phân cụm dữ liệu phần lớn được kế thừa từ lĩnh vực máy học và được sử dụng để giải quyết các vấn đề trong kinh tế và luật. Trong những năm gần đây, trong lĩnh vực nhận dạng thực thể trong văn bản tự nhiên, người ta chủ yếu tập trung vào phân tích và phân tích các dữ liệu phức tạp như dữ liệu văn bản lớn.

1. **Tính cấp thiết của vấn đề**.

Hiện nay với sự phát triển của công nghệ, nguồn thông tin trên internet ngày càng trở nên đa dạng và cần thiết đối với đời sống con người. Tuy nhiên, thông tin trên các trang web thường thiếu khả năng diễn đạt ngữ nghĩa về các đối tượng xuất hiện, gây khó khăn trong việc tìm kiếm tài liệu. Do đó việc nghiên cứu xây dựng các ứng dụng xử lý văn bản tự động là việc cần thiết cho quá trình giao tiếp thông tin của người và máy tính. Một hệ thống tự động trích rút các thực thể, các khái niệm xuất hiện dựa vào một nguồn tri thức về các thực thể và các khái niệm phổ biến trong thế giới thực là tiền đề phát triển của các hệ thống xử lý ngôn ngữ máy tính.

1. **Mục tiêu nghiên cứu**.

Mục tiêu của đề tài là hoàn thiện hệ thống nhận diên thực thể trong câu khi xử lý ngôn ngữ tự nhiên giúp con người dễ dàng tiếp cận thông tin cần tìm dựa trên các từ khóa, tăng độ chính xác cho kết tìm kiếm. Ngoài ra, đây là bài toán căn bản để phát triển các ứng dụng xử lý khác như: trích chọn thông tin, xử lý nội dung, phân tích cảm xúc văn bản,…

1. **Phạm vi nghiên cứu**.

Hệ thống nhận diện thực thể được áp dụng trong phạm vi văn bản Tiếng Việt và được ứng dụng rộng rãi đối với tất cả các văn bản được đưa vào, đặc biệt là thông tin của các trang web trên internet.

1. **Kết cấu đề tài**

Đề tài được tổ chức gồm phần mở đầu, 4 chương nội dung và phần kết luận.

* Mở đầu
* Chương 1: Tổng quan về Python và Named Entity Recognition
* Chương 2: Lý thuyết
* Chương 3: Triển khai thực hiện
* Chương 4: Kết quả
* Kết luận và hướng phát triển

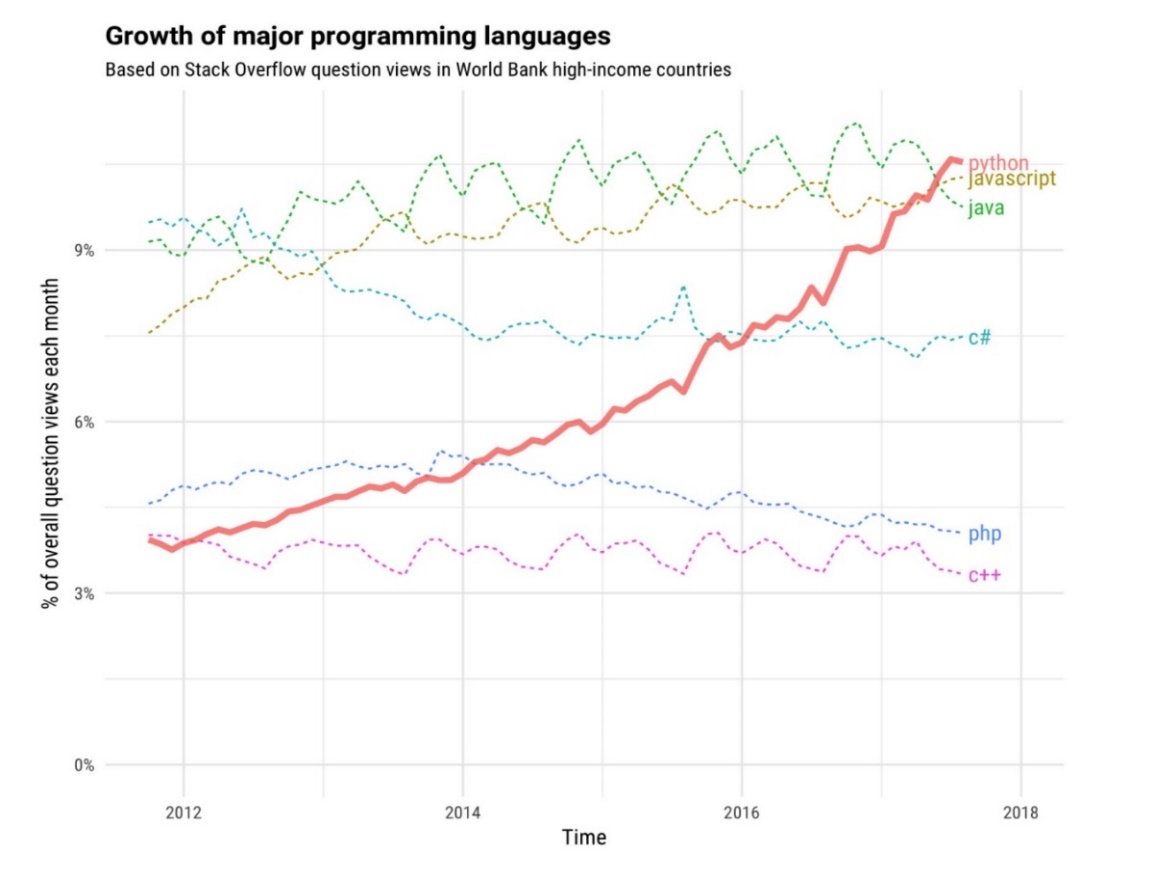
# **CHƯƠNG 1: TỔNG QUAN VỀ PYTHON VÀ NAMED ENTITY RECOGNITION**

* 1. **Ngôn ngữ lập trình Python**

Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do *Guido van Rossum* tạo ra và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình và là ngôn ngữ lập trình dễ học. được dùng rộng rãi trong phát triển trí tuệ nhân tạo hoặc học máy. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu. Vào tháng 7 năm 2018, *Van Rossum* đã từ chức lãnh đạo trong cộng đồng ngôn ngữ Python sau 30 năm làm việc.

Trong một phân tích của 39 phòng khoa học máy tính hàng đầu (được xếp hạng bởi U.S. News in 2014), các ngôn ngữ lập trình có mục đích thay thế Java để trở thành ngôn ngữ được tiếp xúc đầu tiên khi bắt đầu nghiên cứu về khoa học máy tính. 8 trong số 10 phòng ban nghiên cứu về Khoa học máy tính và 27 trong top 39 phòng nghiên cứu đều dạy Python cho khóa học mở đầu của họ.

Được tạo ra cách đây 23 năm, Python giống như một công cụ tuyệt vời trong những năm gần đây cho những người mới sử dụng lần đầu tiên. *Raspberry Pi* vận dụng Python như một công cụ giảng dạy, và Python còn càng ngày càng có ảnh hưởng tới các tổ chức như Google, Yahoo và NASA, khẳng định giá trị của nó ngay cả sau khi một lập trình viên không còn là một người mới bắt đầu. Hiện nay, nó đã thường xuyên được xếp hạng là một trong tám ngôn ngữ lập trình phổ biến nhất kể từ năm *2008 -  eight most popular programming languages.*



Hình 1.1 Top các ngôn ngữ lập trình phổ biến [1]



Hình 1.2 Xếp hạng các ngôn ngữ lập trình của PYPL Index [2]

* 1. **Named Entity Recognition**

Nhận dạng thực thể định danh (Named Entity Recognition - NER), còn gọi là nhận dạng thực thể có tên, là nhiệm vụ nhận biết các từ xuất hiện trong văn bản là tên gọi của một đối tượng nào đó, như tên người (nhân danh), tên đất (địa danh, địa điểm), tên tổ chức, tên tác phẩm, tên sự kiện, thời gian, tiền tệ, v.v. NER đóng vai trò quan trọng trong các ứng dụng tự động trích xuất thông tin, khai phá dữ liệu, dịch máy, v.v.

NER đã được quan tâm nghiên cứu trên thế giới từ đầu những năm 1990. Hiện nay các hệ thống NER cho tiếng Anh, tiếng Đức, tiếng Hà Lan, v.v. đã được xây dựng và được đánh giá cao. Điểm chung của các hệ thống NER này là tập trung đi vào nhận dạng 3 loại thực thể định danh: tên người, tên đất và tên tổ chức.

Đối với tiếng Việt, có một số đơn vị thuộc lĩnh vực công nghệ thông tin đã xây dựng hệ thống NER, nhưng với quy mô công ti, có tính chất nội bộ và thường không phổ biến. Cho đến nay vẫn chưa có thông tin cụ thể về hệ thống NER trong văn bản tiếng Việt đã được xây dựng như thế nào. Vì lẽ đó, bài báo này đưa ra hướng nhận dạng 3 loại thực thể định danh điển hình là tên người, tên đất và tên tổ chức trong văn bản tiếng Việt, nhằm mục đích xây dựng nguồn tài nguyên huấn luyện cho máy học (Machine Learning). Theo đó, các ứng dụng xử lí ngôn ngữ tự nhiên (Natural Language Processing) biết cách tự động trích xuất thông tin thực thể định danh trong các văn bản tiếng Việt, phục vụ cho một số yêu cầu thiết yếu của đời sống.

# **CHƯƠNG 2: LÝ THUYẾT**

## **Ngôn ngữ lập trình Python**

* + 1. Ứng dụng
       1. *Lập trình ứng dụng:*

Python có thể giúp bạn lập trình ứng dụng web, tạo ra các web app có khả năng mở rộng thông qua việc sử dụng Framework và CMS được tích hợp cho python. Framework phổ biến được dùng để tạo web app: djanho, flask, pyramid, plone, Django CMS,…

* + - 1. *Tạo nguyên mẫu phần mềm*

Python được đánh giá chậm hơn so với các ngôn ngữ như C++ hay Java. Nếu nguồn lực bị giới hạn và yêu cầu về hiệu quả là bắt buộc, thì việc sử dụng python thực sự không phải là lựa chọn tốt nhất. Tuy nhiên nếu dùng nó để tạo ra những nguyên bản chạy thử, thực sự là lựa chọn tuyệt vời.

* + - 1. *Sử dụng vào Khoa học dữ liệu*

Python đã trở nên phổ biến rộng rãi như một ngôn ngữ lập trình back-end có mục đích chung, cấp cao để tạo nguyên mẫu và phát triển các ứng dụng. Khả năng đọc, tính linh hoạt và tính phù hợp của Python với các hoạt động của Khoa học dữ liệu đã khiến nó trở thành một trong những ngôn ngữ được các nhà phát triển ưa thích nhất.

Có thông tin cho rằng Python đang được các nhà phát triển sử dụng rộng rãi trong việc tạo ra các trò chơi, PC độc lập, ứng dụng di động và các ứng dụng doanh nghiệp khác. Thư viện Python đơn giản hóa các công việc phức tạp và làm cho việc tích hợp dữ liệu dễ dàng hơn nhiều với ít mã hơn trong thời gian ngắn hơn. Nó bao gồm hơn 137.000 thư viện rất mạnh mẽ và được sử dụng rộng rãi để đáp ứng các yêu cầu của khách hàng và doanh nghiệp. Các thư viện này đã giúp các nhà khoa học và nhà phát triển của chúng tôi phân tích lượng dữ liệu khổng lồ, tạo ra thông tin chi tiết, ra quyết định quan trọng và hơn thế nữa.

Dưới đây là một số thư viện Python được sử dụng rộng rãi trong các lĩnh vực liên quan đến Khoa học dữ liệu.

* *Numpy:* là một thư viện Python cơ bản dành cho khoa học điện toán. Nó đi kèm với sự hỗ trợ cho một đối tượng mảng N-Chiều mạnh mẽ và các chức năng broadcasting.
* *Pandas:* cung cấp các cấu trúc dữ liệu có ý nghĩa, nhanh chóng và linh hoạt. Pandas nhằm mục đích thực hiện phân tích dữ liệu trong thế giới thực bằng Python.
* *Matplotlib:* là một thư viện mô phỏng hai chiều cho ngôn ngữ lập trình Python. Nó có khả năng tạo ra các số liệu có giá trị cao và môi trường đa nền tảng tương tác.
* *SciPy:* là thư viện Python đi kèm với một lượng module cho đại số tuyến tính, và số liệu thống kê.
* *Scikit-Learn:* có những tính năng đơn giản đơn giản nhưng hiệu quả để hoàn thành các nhiệm vụ phân tích và khai phá dữ liệu.
* *TensorFlow:* là một thư viện toán học mã nguồn mở sử dụng để tính toán số bằng các biểu đồ luồng dữ liệu.
  + 1. Ưu điểm
* *Đơn giản:* Là một ngôn ngữ có hình thức sáng sủa, cấu trúc rõ ràng, cú pháp ngắn gọn giúp người lập trình dễ dàng đọc và tìm hiểu.
* *Tốc độ xử lý:* Khá nhanh, và được đánh giá nhanh hơn so với ngôn ngữ PHP. Với tốc độ xử lý cực nhanh, Python có thể tạo ra những chương trình từ những script siêu nhỏ tới những phần mềm cực lớn như Biender 3D.
* *Chất lượng:* Thư viện có tiêu chuẩn cao, Python có khối cơ sở dữ liệu khá lớn nhằm cung cấp giao diện cho tất cả các CSDL thương mại.
* *Thuận tiện:* Python được biên dịch và chạy trên tất cả các nền tảng lớn. Nó có trên tất cả các nền tảng hệ điều hành từ UNIX, MS – DOS, Mac OS, Windows và Linix và các OS khác thuộc họ Unix.
* *Độ tương thích:* Tương thích mạnh mẽ với Unix, hardware, thirt-party software với số lượng thư viện khổng lồ (400 triệu người sử dụng).
* *Mở rộng:* Với tính năng này, Python cho phép người lập trình có thể thêm hoặc tùy chỉnh các công cụ nhằm tối đa hiệu quả có thể đạt được trong công việc.
* *GUI Programming:* Giúp cho việc thực hiện ảnh minh họa di động một cách tự nhiên và sống động hơn.
  + 1. Hạn chế

Python không có các thuộc tính như: protected, private hay public, không có vòng lặp do…while và switch….case.

Mặc dù tốc độ xử lý của Python nhanh hơn PHP nhưng không bằng JAVA và C++.

* + 1. Ứng dụng của Python vào Machine Learning
       1. *Python rất dễ học và dễ sử dụng*

Không ai thích những thứ quá phức tạp và vì vậy việc dễ dàng sử dụng Python là một trong những lý do chính khiến Python trở nên phổ biến đối với Machine Learning. Python rất đơn giản, nó có cú pháp dễ đọc và điều đó làm cho nó được yêu thích bởi cả các lập trình dày dạn và sinh viên công nghệ thông tin.

Nhiều lập trình viên nói rằng Python trực quan hơn các ngôn ngữ lập trình khác. Những người khác nói là Python có nhiều Framework, Thư viện và phần mở rộng giúp đơn giản hóa việc thực hiện các chức năng khác nhau.

Và vì Python là ngôn ngữ có mục đích chung, nó có thể thực hiện một tập hợp các tác vụ Học máy phức tạp và cho phép bạn xây dựng các nguyên mẫu nhanh chóng, kiểm thử sản phẩm của mình cho mục đích học máy tốt hơn.

* + - 1. *Python có nhiều lựa chọn về Thư viện và Framework*

Python đã khá phổ biến và do đó, nó có hàng trăm thư viện và Framework khác nhau có có sẵn, miễn phí để các Lập trình viên có thể sử dụng ngay. Các thư viện và Framework này thực sự hữu ích trong việc tiết kiệm thời gian làm việc, từ đó gián tiếp làm cho Python càng trở nên phổ biến hơn

Có nhiều thư viện Python đặc biệt hữu ích cho Trí tuệ nhân tạo và Học máy. Một số trong số này được đưa ra dưới đây:

* *Keras:* Là một thư viện mã nguồn mở đặc biệt tập trung vào thử nghiệm với các mạng lưới thần kinh sâu (Deep Neural Networks).
* *TensorFlow:* Là một thư viện phần mềm miễn phí được sử dụng cho nhiều ứng dụng học máy như neural network.
* *Scikit-learn:* Là một thư viện phần mềm miễn phí cho Machine Learning bao gồm các thuật toán phân loại, hồi quy và phân cụm khác nhau. Ngoài ra, Scikit-learn có thể được sử dụng kết hợp với NumPy và SciPy.
  + - 1. *Độc lập với nền tảng*

**Độc lập với nền tảng** tức là một ngôn ngữ lập trình hoặc framework cho phép các lập trình viên triển khai mọi thứ trên một máy và sử dụng chúng trên một máy khác mà không có bất kỳ thay đổi nào (hoặc chỉ thay đổi tối thiểu).

Python được hỗ trợ bởi nhiều nền tảng bao gồm Linux, Windows và macOS. Code Python có thể được sử dụng để tạo các chương trình thực thi độc lập cho hầu hết các hệ điều hành phổ biến, điều này có nghĩa là phần mềm Python có thể dễ dàng phân phối và sử dụng trên các hệ điều hành đó mà không cần trình thông dịch Python.

Tuy nhiên,chúng ta vẫn thường có thể tìm thấy các công ty và nhà khoa học dữ liệu sử dụng máy của riêng họ với Bộ xử lý đồ họa (GPU) mạnh mẽ để đào tạo các mô hình ML của họ. Và thực tế là Python là nền tảng độc lập làm cho việc đào tạo này rẻ hơn và dễ dàng hơn rất nhiều.

* 1. **Named Entity Recognition**
     1. Vấn đề cơ bản trong NER

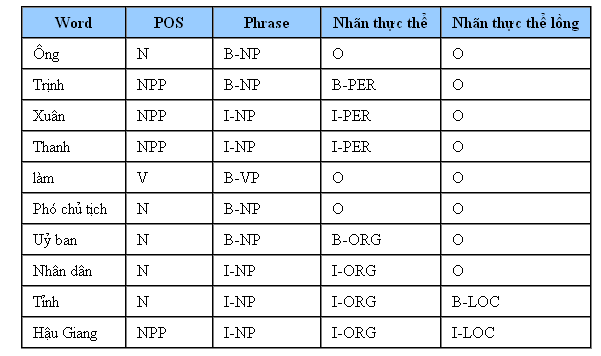
Nhận dạng ba loại thực thể định danh trong tiếng Việt được xác định tương thích với các loại thực thể được mô tả trong hệ thống NER của tiếng Anh và tiếng Đức.

* Tên người (Person) gồm các loại tên sau
* Tên, tên đệm và họ của một người: Nguyễn Văn A.
* Tên hiệu (biệt hiệu), bí danh (mật danh), biệt danh,…
* Tên các nhân vật hư cấu: Ngọc Hoàng, Thiên Lôi, Sa Tăng,…
* Tên địa lí (Địa danh - Location) bao gồm các thực thể có toạ độ địa lí nhất định, ghi lại được trên bản đồ (trừ các địa danh tưởng tượng):
* Tên gọi các hành tinh: Mặt Trăng, Mặt Trời, Trái Đất,…
* Tên gọi các thực thể mang yếu tố địa lí tự nhiên và địa lí lịch sử (quốc gia, vùng lãnh thổ, châu lục), các vùng quần cư (làng, thị trấn, thành phố, tỉnh, giáo khu, giáo xứ),…
* Tên gọi các thực thể tự nhiên: đèo, núi, dãy núi, rừng, sông, suối, hồ, biển, vịnh, vũng, eo biển, đại dương, thung lũng, cao nguyên, đồng bằng, bãi biển, khu bảo tồn thiên nhiên, khu sinh thái,…
* Tên gọi các thực thể là công trình xây dựng, công trình kiến trúc công cộng: cầu, đường, lâu đài, quảng trường, bảo tàng, trường học, nhà trẻ, thư viện, bệnh viện, nhà hát, nhà máy,…
* Tên gọi địa điểm, địa chỉ thương mại: nhà hàng, khách sạn, hiệu thuốc, quán bar,…
* Một số địa danh tưởng tượng khác: Vườn Địa Đàng, Sông Ngân, Cầu Ô Thước,…
* Tên tổ chức (Organization) bao gồm các loại tên sau:
* Các cơ quan chính phủ: các bộ ngành, uỷ ban nhân dân, hội đồng nhân dân, toà án, phòng ban,…
* Các tổ chức chính trị: cơ quan báo chí, đảng phái chính trị, đoàn thể chính trị, hội nghề nghiệp,…
* Kinh doanh, sản xuất: ngân hàng, thị trường chứng khoán, công ti, hãng phim, nhà sản xuất, hợp tác xã,…
* Các thương hiệu: Honda, Sony, Samsung,…
  + 1. Định dạng dữ liệu

Ngữ liệu trước khi được gán nhãn NER sẽ được tách từ và/hoặc gán nhãn từ loại một cách tự động. Mỗi từ được đặt trên một dòng riêng biệt. Mỗi dòng bao gồm năm cột:

1. Đơn vị từ
2. Nhãn từ loại của từ
3. Nhãn cụm từ
4. Nhãn thực thể (mức 1)
5. Nhãn thực thể lồng (mức 2)

Nhãn thực thể được gán theo cấu trúc BIO. Có 7 nhãn: B-PER và I-PER cho tên người, B-LOC và I-LOC cho địa danh, B-ORG và I-ORG cho tên tổ chức, và O cho các phần tử khác. Kí hiệu B: Begin, dùng cho từ đầu tiên của thực thể. I: Inside, dùng cho các từ tiếp theo trong cụm thực thể. O: Other, dùng cho từ không thuộc bất cứ thực thể nào. Các nhãn từ loại: N: noun. NPP: proper noun. V: verb. A: adjective, v.v. Các nhãn cụm từ: NP, VP, AP, v.v. Ví dụ:



Hình 2.1 Nhãn thực thể được gán theo cấu trúc BIO [3]

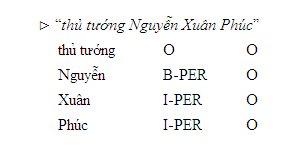
* Ở ví dụ trên, nhãn B-PER (Trịnh) chỉ ra đơn vị đầu tiên của tên người, nhãn I-PER là đơn vị tiếp theo và (cho đến khi) kết thúc của tên người (Xuân, Thanh). Nhãn B-ORG (Uỷ ban) chỉ ra đơn vị đầu tiên của tên tổ chức, nhãn I-ORG là đơn vị tiếp theo và kết thúc của tên tổ chức (Nhân dân, Tỉnh, Hậu Giang). Các đơn vị “Tỉnh, Hậu Giang”, về bản chất lại mang nhãn địa lí (location), chúng sẽ được miêu tả ở mức “Nhãn thực thể lồng” là: B-LOC (Tỉnh) - chỉ ra đơn vị đầu tiên của tên địa lí, và I-LOC (Hậu Giang) - chỉ ra đơn vị tiếp theo và kết thúc của tên địa lí.
* Trong bài báo này, để dễ hình dung, chúng tôi chỉ trình bày định dạng ngữ liệu ở dạng có ba cột: 1. Đơn vị từ; 2. Nhãn thực thể; 3. Nhãn thực thể lồng.
  + 1. Nhận dạng thực thể
       1. Tên người (Person)

Tên người (nhân danh) được xem là tên riêng. Viết hoa tên riêng là để chỉ ra rằng người đó chỉ có một mà thôi, không giống với người khác.

Thường đi trước tên người có các danh từ chung như ông, bà, anh, chị, chú, bác, thằng, chủ tịch, giám đốc, trưởng phòng, v.v. Các danh từ chung này được dùng để chỉ hoặc gọi người nào đó tuỳ theo mối quan hệ. Chẳng hạn, cùng một người tên Thanh, nhưng có người gọi anh Thanh, có người gọi ông Thanh, giám đốc Thanh, v.v. Chúng tôi cho rằng, các danh từ loại này không nằm trong cấu tạo tên người, vì chúng không có tính cố định.

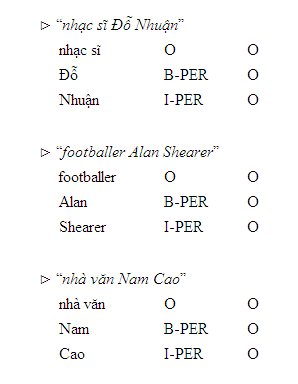
* Tên người là tên riêng chỉ từng cá nhân.

Dạng đầy đủ, tên người gồm 3 thành phần: họ + chữ đệm + tên. Không phân biệt họ, chữ đệm và tên vì coi chúng đều được riêng hoá, vì thế viết hoa chữ cái đầu của các âm tiết.



Hình 2.2 Nhận dạng tên riêng chỉ từng cá nhân [3]

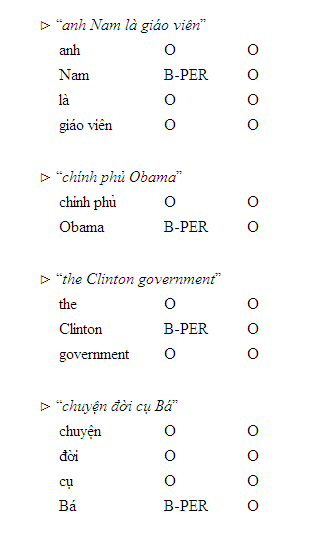
* Dạng rút gọn còn 2 thành phần: họ + tên, hoặc tên + họ (các ngôn ngữ Ấn-Âu), hoặc chữ đệm + tên



Hình 2.3 Nhận dạng tên rút gọn còn 2 thành phần [3]

* Dạng rút còn 1 thành phần

Trường hợp có các danh từ chung là từ xưng hô đứng trước bộ phận tên (hoặc họ với các ngôn ngữ Ấn-Âu) thì các danh từ này không được coi là thuộc tên người. Trường hợp danh từ chung chỉ chức vụ, công việc, v.v. được dùng để gọi thay cho tên người đảm nhiệm chức vụ, công việc đó trong một không gian cụ thể (bối cảnh của câu chuyện) thì cũng được coi là tên người (có thể viết hoa theo phong cách: ông Hàn, cụ Bá, cụ Đề, anh Lí, v.v.).



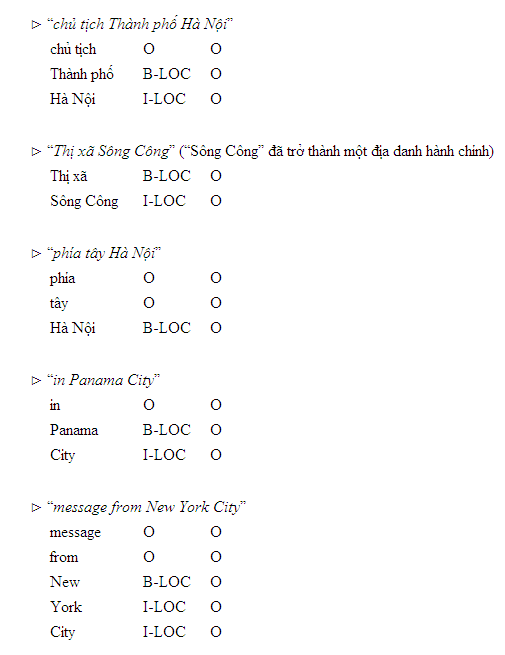
Hình 2.4 Nhận dạng tên rút gọn còn 1 thành phần [3]

* + - 1. *Tên địa lí (Location)*

Tên địa phương được phân chia theo khu vực địa lí của một nước (như làng, xã, phường, huyện, quận, tỉnh, thành phố…), cũng là các đơn vị địa danh hành chính của một nước, thuộc tri thức nền mang tính cộng đồng cao. Chẳng hạn: New York, Paris, Canbera, Jakacta, Bangkok, Hà Nội, Hải Phòng, Hà Nam, Hoà Bình, Thanh Hoá, Quảng Nam, Kon Tum, Đắc Lắc, Lâm Đồng, Đồng Nai, Tiền Giang, Cà Mau, Cầu Dền, Cầu Giấy, Ba Vì, Gia Lâm, Củ Chi, Nhà Bè, Kiến An, Hải Châu, An Khê, Bát Xát, Mèo Vạc, Tân Trào, v.v.

Với các kiểu cấu tạo Tỉnh Nam Định, Thành phố Nam Định, Thành phố Hà Nội, Thủ đô Hà Nội, Thành phố Hồ Chí Minh, Thành phố New York (New York City), v.v. là bao hàm ý phân biệt vị thế, cấp độ của một địa danh. So sánh:

* Tỉnh Nam Định: tỉnh gồm Thành phố Nam Định và 9 huyện.
* Thành phố Nam Định: trung tâm của Tỉnh Nam Định, không bao gồm 9 huyện.
* Với các cấu tạo Thành phố Hà Nội, Thủ đô Hà Nội, Hà Nội thì cả 3 kiểu đều chỉ chung một thực thể, do vậy chúng đồng nhất với nhau về tri nhận (Hà Nội là cách gọi rút gọn của Thành phố Hà Nội hoặc Thủ đô Hà Nội). Trong khi, Thành phố Hồ Chí Minh và Hồ Chí Minh lại chỉ hai thực thể khác nhau: một chỉ địa danh, một chỉ nhân danh. Từ những lí do đó, chúng tôi cho rằng, yếu tố “thành phố, thủ đô, tỉnh, thị xã, thị trấn” là thành phần tham gia vào cấu tạo nên địa danh, và vì vậy coi cả khối Thành phố Nam Định, Thành phố Hồ Chí Minh là một đơn vị để phân biệt với Tỉnh Nam Định (bao hàm Thành phố Nam Định) và Hồ Chí Minh (chỉ người).
* Xử lí tương tự với các trường hợp: Tỉnh Đồng Nai, Quận Cầu Giấy, Quận 3, Quận Hai Bà Trưng (phân biệt với Phố Hai Bà Trưng), Phường Minh Khai (phân biệt với Phố Minh Khai), Huyện Cầu Kè, Huyện Sông Cầu (phân biệt với Sông Cầu), Thị trấn Chợ Đồn, Thị xã Sông Công, v.v.



Hình 2.5 Nhận dạng tên địa lí [3]

Với những thành tựu trong việc gán nhãn ngữ nghĩa tự động cho ngữ liệu đang diễn ra hiện nay, vấn đề chưa thống nhất trong cách viết tên riêng tiếng Việt tuy là trở ngại, nhưng hoàn toàn có thể vượt qua. Ý nghĩa của đơn vị ngôn ngữ ngày càng được khai phá một cách toàn diện và có chiều sâu, báo hiệu một xu thế phát triển mới về khoa học dữ liệu trong tương lai gần.

* 1. **Nhận dạng đối tượng dựa trên BERT**
     1. Bidirectional Encoder Representations from Transformers (BERT)

BERT được dịch theo tiếng Việt là “Đại diện bộ mã hóa hai chiều từ Transformers”. Được đào tạo trước về lượng lớn văn bản, BERT đã trình bày một loại mô hình ngôn ngữ tự nhiên mới. Tận dụng sự chú ý và kiến ​​trúc máy biến áp, BERT đã đạt được kết quả hiện đại tại thời điểm xuất bản, do đó đã cách mạng hóa lĩnh vực này. Đây là một thuật toán học liên quan đến xử lý ngôn ngữ tự nhiên, trong đó mạng nơ-ron sử dụng mô hình hai chiều với biểu diễn ngôn ngữ không có giám sát.. Nó giúp tối ưu hóa hệ thống tìm kiếm, hiểu được gần chính xác sắc thái, ý nghĩa ngữ cảnh trong câu.

BERT được thiết kế để đào tạo trước các biểu diễn hai chiều sâu từ văn bản không được gắn nhãn bằng cách điều chỉnh chung trên cả ngữ cảnh bên trái và bên phải trong tất cả các lớp. Do đó, mô hình BERT được đào tạo trước có thể được tinh chỉnh chỉ với một lớp đầu ra bổ sung để tạo ra các mô hình hiện đại cho một loạt các nhiệm vụ, chẳng hạn như trả lời câu hỏi và suy luận ngôn ngữ mà không cần thực hiện nhiệm vụ quan trọng - sửa đổi kiến trúc cụ thể.

* + 1. Nguyên lý hoạt động của BERT

Đầu vào là một chuỗi các mã thông báo, đầu tiên được nhúng vào các vectơ và sau đó được xử lý trong mạng neural. Đầu ra là một chuỗi các vectơ có kích thước H, trong đó mỗi vectơ tương ứng với một mã thông báo đầu vào có cùng chỉ mục.

Khi đào tạo các mô hình ngôn ngữ, có một thách thức trong việc xác định mục tiêu dự đoán. Nhiều mô hình dự đoán từ tiếp theo theo trình tự (ví dụ: Đứa trẻ về nhà từ “<mask>” ), một cách tiếp cận định hướng vốn hạn chế việc học theo ngữ cảnh. Để vượt qua thử thách này, BERT sử dụng hai chiến lược đào tạo: Mask LM (MLM - dự đoán các từ được che dấu) và Dự đoán câu tiếp theo (Next Sentence Prediction - NSP)

* + 1. Ứng dụng của BERT vào NER

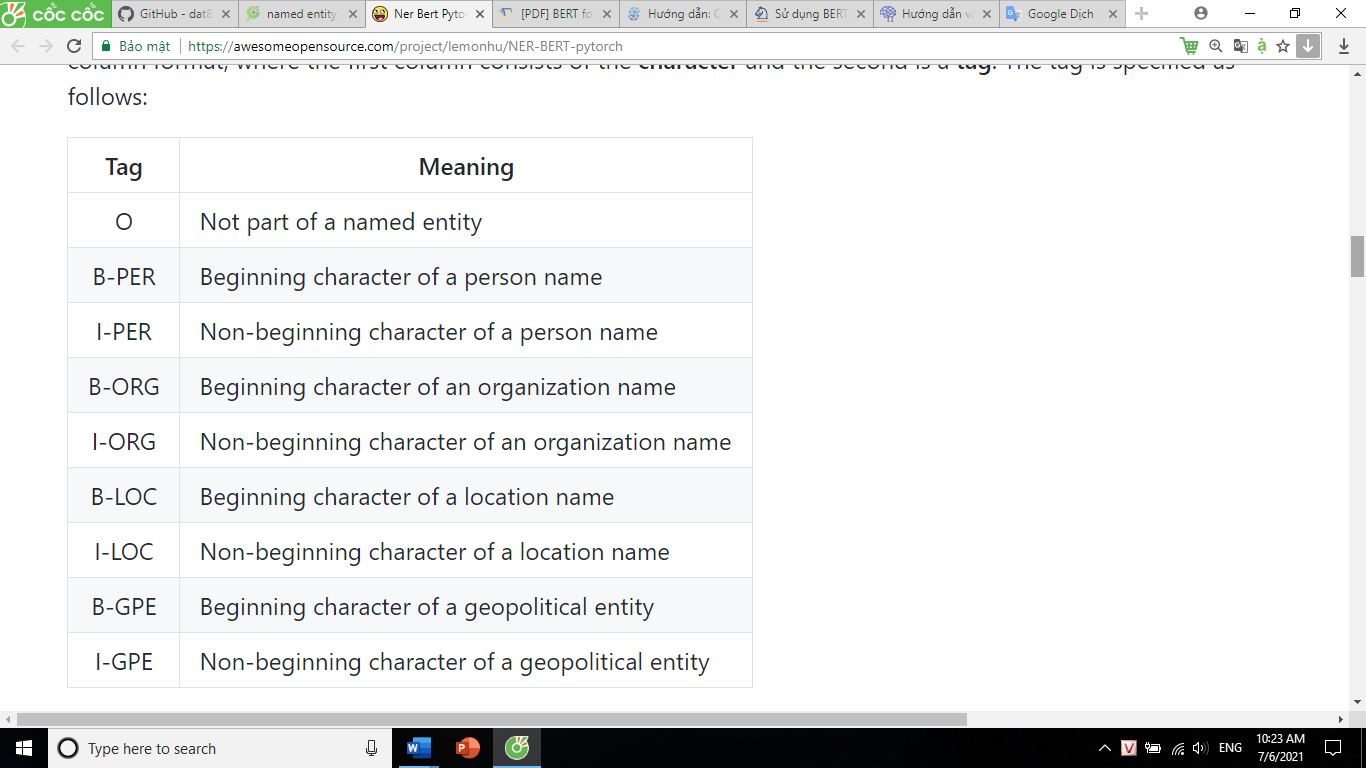
Đánh giá hiệu năng của một nhiệm vụ gán nhãn cho từng tokens, BERT đã tinh chỉnh trên tập dữ liệu CoNLL 2003 Named Entity Recognition (NER). Đây là tập dữ liệu gồm 200 ngàn từ được gán nhãn bằng tay với các nhãn là Person (tên người), Organization (tên tổ chức), Location (tên địa danh), Miscellaneous (tên sản phẩm,...) và Other (không phải thực thể có tên).

Để tinh chỉnh, chúng ta cần sử dụng trọng số ở lớp mặt nạ cuối cùng cho mỗi token i là T\_iTi (T\_i \in R^HTi​∈RH) cho vào một lớp phân loại với tập nhãn là số nhãn NER.

* + 1. Bộ dữ liệu cho NER

Có nhiều bộ dữ liệu để tinh chỉnh mô hình BERT được giám sát. Các tập dữ liệu cơ bản thường tập trung vào bốn loại thực thể được đặt tên liên quan đến người, địa điểm, tổ chức và tên của các thực thể khác.

Lược đồ BIO (Bắt đầu, Bên trong, Bên ngoài) là định dạng gắn thẻ phổ biến để gắn thẻ các mã câu cho NER, chứa bốn cột được phân tách bằng một khoảng trắng. Ở đây tiền tố B chỉ ra rằng thẻ ở đầu mỗi đoạn. Tiền tố I giống nhau cho Inside of Chunk và tiền tố O cho không có thực thể nào bên trong chunk.



Hình 2.6 Bộ dữ liệu cho NER

* 1. **Giải pháp xây dựng**
     1. Phân đoạn câu văn - Sentence Segmentation

Bước đầu tiên trong đường ống là chia văn bản thành cách câu riêng biệt. Và sau đó sẽ có được các câu lẻ riêng biệt. Chúng ta có thể cho răng, mỗi câu trong Tiếng Việt mang một ý nghĩa hoặc một ý tưởng đặc biệt. Và sẽ dễ dàng hơn để viết một chương trình có thể hiểu một câu đơn hơn là hiểu liền một lúc cả một đoạn văn.

* + 1. Mã hoá các từ - Word Tokenization

Sau đó, tiến hành chia các câu nhỏ của đã tách thành các câu văn riêng lẻ, và do đó, chúng ta cần phải xử lý từng câu văn một. Tiếp theo chia các câu văn theo các words và tokens. những việc này gọi là tokenization.

* + 1. Xác định các từ dừng - Identifying Stop Words

Tiếp theo sẽ muốn xem xét tầm quan trọng của từng từ trong câu. Tiếng Việt có rất nhiều từ nối và được sử dụng rất thường xuyên như "à", "á", "ớ",.... Khi thực hiện việc thống kê trên văn bản, những từ này sẽ mang lại rất nhiều nhiễu vì chúng xuất hiện thường xuyên hơn các từ khác

Các từ dừng thường được xác định chỉ bằng cách kiểm tra danh sách hardcoded của các từ dừng đã biết. Nhưng không có danh sách từ vựng tiêu chuẩn phù hợp cho tất cả các ứng dụng. Danh sách các từ có thể bỏ qua còn phụ thuộc vào từng ứng dụng cụ thể.

* + 1. Phân tích sự phụ thuộc về cú pháp - Dependency Parsing

Mục tiêu là xây dựng một cây có thể gán một từ đơn duy nhất làm nguồn cho mỗi từ trong câu. Từ gốc của cây này sẽ là động từ chính trong câu.

Nó cũng quan trọng để nhắc lại rằng, nhiều câu trong Tiếng Việt là mơ hồ và thực sự khó phân tích. Trong những trường hợp đó, mô hình sẽ đưa ra dự đoán dựa trên phiên bản phân tích cú pháp của câu đó, và có vẻ như nó không hoàn hảo và đôi khi mô hình sẽ dự đoán sai. Nhưng theo thời gian, mô hình sẽ tiếp tục trở nên tốt hơn trong việc phân tích văn bản một cách hợp lý.

* + 1. Tìm các cụm danh từ - Finding Noun Phrases

Đến đây máy đã coi mỗi từ trong câu của chúng tôi là một thực thể riêng biệt. Nhưng đôi khi chúng sẽ có ý nghĩa hơn để nhóm các từ lại với nhau thành một cụm đại diện cho một ý tưởng hoặc một điều duy nhất nào đó. Chúng ta có thể sử dụng thông tin từ cây phân tích sự phụ thuộc để tự động nhóm các từ mà tất cả đang nói về cùng một điều.

* + 1. Nhận dạng thực thể được đặt tên - Named Entity Recognition (NER)

Mục tiêu của nhận dạng tên của thực thể, là việc phát hiện và gán nhãn cho các danh từ này với các khái niệm trong thế giới thực mà chúng đại diện. Ở đây, câu của chúng tôi trông như thế nào sau khi chạy mỗi từ vựng qua mô hình NER đã được train. NER có hàng triệu cách để sử dụng vì nó dễ dàng lấy dữ liệu có cấu trúc ra khỏi văn bản.

# **CHƯƠNG 3: TRIỂN KHAI THỰC HIỆN**

* 1. **Nghiên cứu về code có sẵn**

Với sự phát triển của internet, việc tìm kiếm thông tin trở nên dễ dàng hơn. Việc tìm hiểu về các code đã có sẵn sẽ tận dụng được những nền móng đã có của đề tài, từ đó thực hành phát triển nên hệ thống hoàn thiện hơn, giúp tiết kiệm thời gian nghiên cứu. Nhóm đã thực hiện nghiên cứu các code sẵn có qua 3 trang web chính:

* + 1. Github

GitHub là một hệ thống quản lý dự án và phiên bản code, được sử dụng chủ yếu cho dự án có nhiều người cùng hợp tác và cần giám sát toàn bộ thay đổi của dự án. Đây là nền tảng lưu trữ online lớn nhất thế giới nên có thể dễ dàng tìm được những code về đề tài nghiên cứu tại đây.

Github đã cung cấp các code về xử lý tách câu, từ Tiếng Việt bằng thư viện Underthesea và BERT làm tiền đề cho quá trình xử lý tách văn bản. Đây là nguồn cung cấp chính các source code giúp nhóm tham khảo và hoàn thiện đề tài.

* + 1. Mì AI

Mì AI cung cấp các thông tin cần thiết của việc phát triển đề tài “Nhận dạng và phân loại các thực thể trong văn bản Tiếng Việt”, bên cạnh đó còn xác định hướng thực hiện đề tài và các code về vấn đề loại bỏ từ thừa, dấu câu không cần thiết trong văn bản.

Từ nguồn thông tin code có sẵn trên Mì AI giúp nhóm thực hiện crawl dữ liệu từ các trang web, thực hiện train\_model cũng như loại bỏ từ thừa khỏi đoạn văn bản.

* + 1. Youtube

Youtube là trang mạng video lớn nhất thế giới nên việc tìm kiếm các video hướng dẫn thực hiện viết code trở nên dễ dàng với nguồn thông tin rộng lớn. Việc thực hiện chạy code theo youtube sẽ chi tiết quá trình hơn so với những trang web lý thuyết. Bên cạnh đó youtube cũng cung cấp các thông tin hữu ích về cách sử dụng Google Colaboratory cũng như hướng dẫn các bước train model.

* 1. **Các cài đặt cần thiết**
     1. Cài đặt Python và Pytorch tối thiểu

Đối với đề tài “Nhận dạng và phân loại các thực thể trong văn bản tiếng Việt”, yêu cầu về phiên bản của ngôn ngữ lập trình là Python 3.6+ và Pytorch 1.4+.

PyTorch là một framework được xây dựng dựa trên python cung cấp nền tảng tính toán khoa học phục vụ lĩnh vực Deep learning. Pytorch tập trung vào 2 khả năng chính: sự thay thế cho bộ thư viện numpy để tận dụng sức mạnh tính toán của GPU và platform Deep learning phục vụ trong nghiên cứu, mang lại sự linh hoạt và tốc độ. Việc sử dụng Pytorch trong đề tài mang lại khả năng debug dễ dàng hơn và có thể kết hợp cả các API cấp cao và cấp thấp.

* + 1. Cài các thư viện liên quan

Thế mạnh của ngôn ngữ lập trình Python chính là sự đa dạng của Libraries và Packages hơn các ngôn ngữ C hay C++, chúng hỗ trợ trong việc quản lý và dễ dàng chỉnh sửa các dòng lệnh hơn. Sau đây là các thư viện chính được sử dụng trong đề tài:

* + - 1. *Underthesea*

Underthesea là một bộ dữ liệu và hướng dẫn module Python mã nguồn mở hỗ trợ nghiên cứu và phát triển trong xử lý ngôn ngữ tự nhiên tiếng Việt. Đây là thư viện được áp dụng vào các mô hình NLP đã được xử lý cho văn bản tiếng Việt như: phân đoạn từ, gắn thẻ một phần giọng nói (PoS), nhận dạng thực thể có tên (NER), phân loại văn bản và phân tích cú pháp phụ thuộc. Trong đề tài này, thư viện Underthesea được sử dụng ở phiên bản 1.1.17.

* + - 1. *Natural Language Toolkit (NLTK)*

NLTK là một nền tảng hàng đầu để xây dựng các chương trình Python để làm việc với dữ liệu ngôn ngữ của con người. Nó cung cấp các giao diện dễ sử dụng cho hơn 50 tài nguyên ngữ liệu và từ vựng cùng với một bộ thư viện xử lý văn bản để phân loại, mã hóa, tạo gốc, gắn thẻ, phân tích cú pháp và lập luận ngữ nghĩa, … Bên cạch đó, NLPK được sử dụng trong Machine Learning với tác dụng làm sạch dữ liệu, xử lý dữ liệu đầu vào cho các thuật toán Machine Learning. Một chức năng khác của NLTK là mô phỏng đồ hoạ và dữ liệu mẫu hữu ích. NLTK cung cấp các xử lý như classification, tokenization, stemming, tagging, parsing, và semantic reasoning...

Thư viện NLTK được sử dụng trong đề tài nhằm hỗ trợ cho vấn đề xử lý văn bản được đưa vào, giúp mã hóa và gắn thẻ và nhận diện các thực thể trong văn bản. Đề tài đã sử dụng phiên bản NLTK 3.4.5.

* + - 1. *BertTokenizer từ thư viện Transformers*

PyTorch-Transformers là một thư viện gồm các mô hình hiện đại được đào tạo trước để Xử lý Ngôn ngữ Tự nhiên (NLP) như: BERT, GPT, GPT-2, Transformer-XL, XLNet,XLM.

Trong đề tài “Nhận dạng và phân loại các thực thể trong văn bản tiếng Việt”, chúng ta sẽ sử dụng BertTokenizer từ thư viện Transformers.

Khung BERT, một mô hình biểu diễn ngôn ngữ mới từ Google AI, sử dụng đào tạo trước và tinh chỉnh để tạo ra các mô hình NLP hiện đại cho một loạt các nhiệm vụ. Những nhiệm vụ này bao gồm hệ thống trả lời câu hỏi, phân tích tình cảm và suy luận ngôn ngữ.

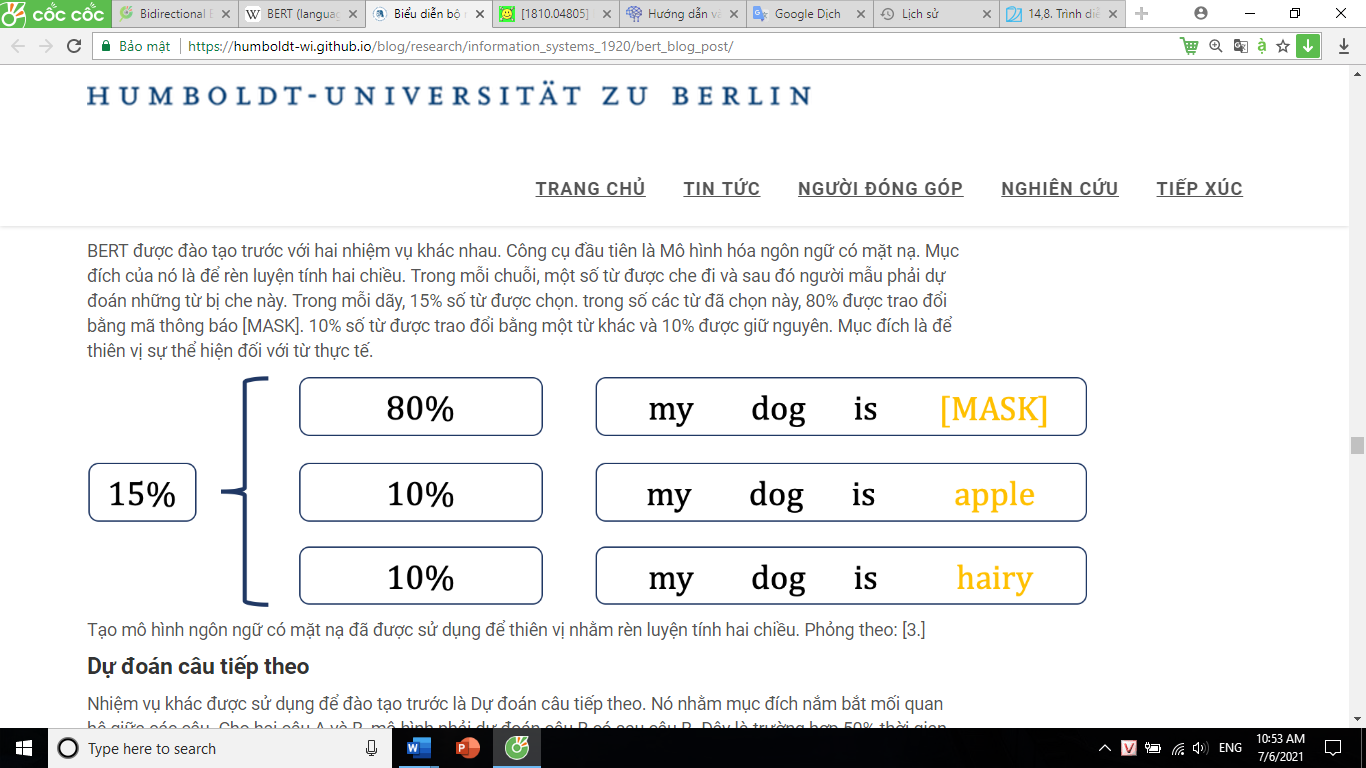
BERT được đào tạo trước bằng cách sử dụng hai nhiệm vụ dự đoán không giám sát sau:

* Tạo mô hình ngôn ngữ có mặt nạ

Để mã hóa ngữ cảnh theo hai chiều để đại diện cho từng mã thông báo, BERT che giấu mã thông báo một cách ngẫu nhiên và sử dụng mã thông báo từ ngữ cảnh hai chiều để dự đoán các mã được che giấu. Tác vụ này được gọi là một mô hình ngôn ngữ được che giấu. BertTokenizer được sử dụng trong đề tài với mục đích dự đoán các từ ngữ thiếu dựa trên ngữ nghĩa đoạn văn bản được đưa vào. Nói rõ hơn, khi đưa và một văn bản bất kì có những từ ngữ thiếu, mô hình sẽ thực hiện dự đoán các từ ngữ đó dựa trên các từ rõ ràng khác và ngữ cảnh trong chuỗi văn bản.

Trong mỗi dãy, 15% số từ được chọn ngẫu nhiên làm mã thông báo được che giấu để dự đoán. Để dự đoán từ bị che mà không gian lận bằng cách sử dụng nhãn, một cách tiếp cận đơn giản là luôn thay thế nó bằng mã thông báo “<mask>” đặc biệt trong chuỗi đầu vào BERT. Tuy nhiên, từ đặc biệt nhân tạo “<mask>” sẽ không bao giờ xuất hiện trong quá trình tinh chỉnh. Để tránh sự không phù hợp như vậy giữa đào tạo trước và tinh chỉnh, nếu từ được che đi để dự đoán (ví dụ: “tuyệt vời” được chọn để che và dự đoán trong “bộ phim này thật tuyệt”), trong đầu vào, nó sẽ được thay thế bằng:

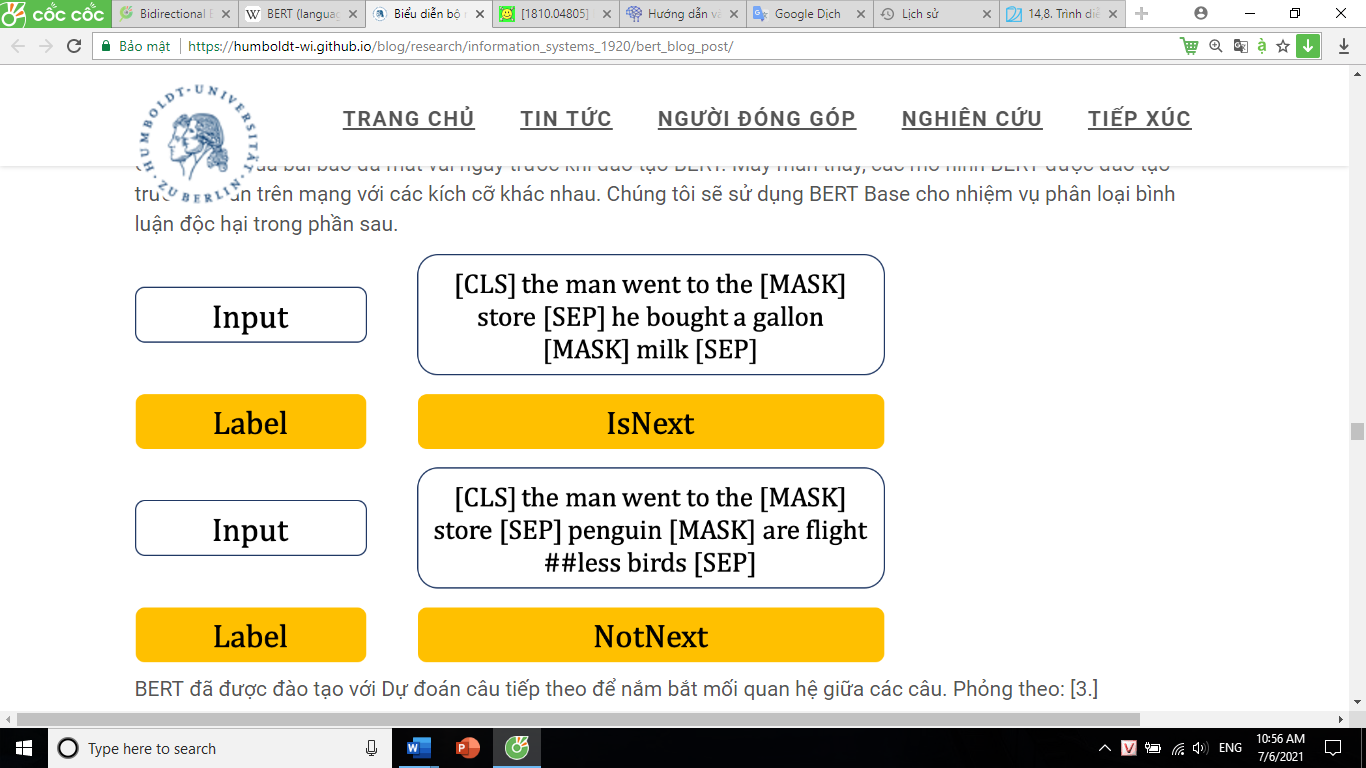
* 80% được trao đổi bằng mã thông báo (ví dụ: “bộ phim này rất hay” trở thành “bộ phim này là <mặt nạ>”).
* 10% số từ được trao đổi bằng một từ khác (ví dụ: “bộ phim này rất hay” trở thành “bộ phim này là đồ uống”).
* 10% được giữ nguyên (ví dụ: “bộ phim này rất hay” trở thành “bộ phim này thật tuyệt”).



Hình 3.1 Tạo mô hình ngôn ngữ có mặt nạ [4]

* Dự đoán câu tiếp theo

Mặc dù mô hình ngôn ngữ được che giấu có thể mã hóa ngữ cảnh hai chiều để biểu diễn các từ, nhưng nó không mô hình hóa rõ ràng mối quan hệ logic giữa các cặp văn bản. Để giúp hiểu mối quan hệ giữa hai chuỗi văn bản, BERT xem xét nhiệm vụ phân loại nhị phân, dự đoán câu tiếp theo, trong quá trình đào tạo trước của nó. Khi tạo các cặp câu để luyện trước, trong một nửa thời gian, chúng thực sự là các câu liên tiếp với nhãn “Đúng”; trong khi nửa thời gian còn lại, câu thứ hai được lấy mẫu ngẫu nhiên từ kho ngữ liệu có nhãn “Sai”.



Hình 3.2 Dự đoán câu tiếp theo [4]

* 1. **Thực hiện train dữ liệu**

Sau các bước xác định xây dựng mạng neural và đưa dữ liệu vào, quá trình train dữ liệu giúp đánh giá độ chính xác của mô hình. Quá trình này dùng để tìm kiếm mối quan hệ giữa x và y, đưa ra những dự đoán (optimizer) và đánh giá độ chính xác (loss), lặp đi lặp lại với số lần xác định (epochs).

Dữ liệu huấn luyện phải có các câu trả lời đúng, được biết đến như một mục tiêu hay mục tiêu thuộc tính. Thuật toán học tìm các mẫu trong dữ liệu đào tạo ánh xạ các thuộc tính dữ liệu đầu vào cho mục tiêu và nó xuất ra một mô hình Machine Learning nắm bắt các mẫu này.

Ở những lần lặp đầu tiên thì giá trị loss còn khá cao, tức là phương trình dự đoán mà function optimizer chưa gần với phương trình chính xác. Sau nhiều lần lặp thì giá trị loss giảm dần, càng lúc càng về 0. Tức là model của chúng ta đã dự đoán đúng mối quan hệ giữa x và y.

* 1. **Thực hiện xử lý tách từ ngôn ngữ Tiếng Việt**

Xử lý tách từ là bước quan trọng trong hệ thống xử lý ngôn ngữ tự nhiên (NLP). Đây là một quá trình xử lý nhằm mục đích xác định ranh giới của các từ trong câu văn, cũng có thể hiểu đơn giản rằng tách từ là quá trình xác định các từ đơn, từ ghép… có trong câu. Đối với xử lý ngôn ngữ, để có thể xác định cấu trúc ngữ pháp của câu, xác định từ loại của một từ trong câu, yêu cầu nhất thiết đặt ra là phải xác định được đâu là từ trong câu.

Đối với ngôn ngữ Tiếng Việt, ranh giới từ không chỉ đơn giản là những khoảng trắng như trong các ngôn ngữ thuộc loại hình hòa kết như tiếng Anh…, mà có sự liên hệ chặt chẽ giữa các tiếng với nhau, một từ có thể cấu tạo bởi một hoặc nhiều tiếng. Vì vậy vấn đề của bài toán tách từ là khử được sự nhập nhằng trong ranh giới từ. Tách từ chính xác hay không là công việc rất quan trọng, nếu không chính xác rất có thể dẫn đến việc ý nghĩa của câu sai, ảnh hưởng đến tính chính xác của chương trình.

Ngày nay, cùng với sự phát triển mạnh mẽ của công nghệ phần cứng và truyền thông các hệ thống dữ liệu phục vụ cho các lĩnh vực kinh tế - xã hội cũng không ngừng tăng lên, lượng dữ liệu được tạo ra ngày càng lớn. Sự phong phú về dữ liệu, thông tin cùng với khả năng kịp thời khai thác chúng đã mang đến những năng suất và chất lượng mới cho công tác quản lý, hoạt động kinh doanh,… Nhưng rồi các yêu cầu về thông tin trong các lĩnh vực hoạt động đó, đặc biệt trong lĩnh vực ra làm quyết định, ngày càng đòi hỏi cao hơn, người quyết định không những cần dữ liệu mà còn cần có thêm nhiều hiểu biết, nhiều tri thức để hỗ trợ cho việc ra quyết định của mình. Cho đến những năm 90 của thế kỉ trước, theo cùng nhu cầu của thế giới để xử lí các lớp dữ liệu, hàng loạt các lĩnh vực nghiên cứu về tổ chức các kho dữ liệu và kho thông tin, hỗ trợ giúp ra quyết định, lọc các thông tin chính xác,… và đặc biệt là xử lý ngôn ngữ tự nhiên (Natural Language Processing).

Từ khi ra đời, xử lý ngôn ngữ tự nhiên trở thành một trong những hướng nghiên cứu phổ biến trong lĩnh vực trí tuệ nhân tạo và khoa học máy tính. Nhiều kết quả nghiên cứu, ứng dụng của xử lý ngôn ngữ tự nhiên trong các lĩnh vực khoa học, kinh tế, xã hội, luật pháp,… Xử lý ngôn ngữ tự nhiên bao gồm nhiều hướng nghiên cứu quan trọng, một trong số đó là nhận dạng thực thể trong văn bản tự nhiên (Named Entity Recognition). Nhận dạng thực thể trong văn bản tự nhiên là quá trình phân tách và gán nhãn các thực thể trong các mẫu dữ liệu tự nhiên trong cơ sở dữ liệu lớn. Các kĩ thuật được áp dụng trong phân cụm dữ liệu phần lớn được kế thừa từ lĩnh vực học máy cho việc giải quyết các vấn đề như kinh tế, luật pháp, … Trong thời gian gần đây, trong lĩnh vực nhận dạng thực thể trong văn bản tự nhiên, người ta tập trung chủ yếu và nghiên cứu phân tích các dữ liệu phức tạp như dữ liệu văn bản lớn.

NER thực hiện những gì được gọi là phân tích cú pháp bề mặt, phân định các chuỗi mã thông báo trả lời những câu hỏi quan trọng này. NER cũng có thể được sử dụng như bước đầu tiên trong một chuỗi các bộ xử lý: cấp độ xử lý tiếp theo có thể liên quan đến hai hoặc nhiều NER, hoặc thậm chí có thể cung cấp ngữ nghĩa cho mối quan hệ đó bằng cách sử dụng một động từ. Bằng cách này, quá trình xử lý sâu hơn có thể khám phá ra "cái gì" và "như thế nào" của một câu hoặc nội dung văn bản. Thử nghiệm hoàn chỉnh đã được tiến hành trên dữ liệu thực cho thấy kết quả cạnh tranh cho các hệ thống NER thu được.

Việc xử lý tách từ ngôn ngữ Tiếng Việt sẽ được thực hiện dưới sự trợ giúp của thư viện Underthesea. Đây là một bộ dữ liệu và hướng dẫn mô-đun Python mã nguồn mở hỗ trợ nghiên cứu và phát triển trong xử lý ngôn ngữ Tiếng Việt, giúp xác định nghĩa của các từ trong câu để thực hiện phân tách.

* 1. **Gán nhãn các thực thể đã tách (Part-of-speech tagging)**

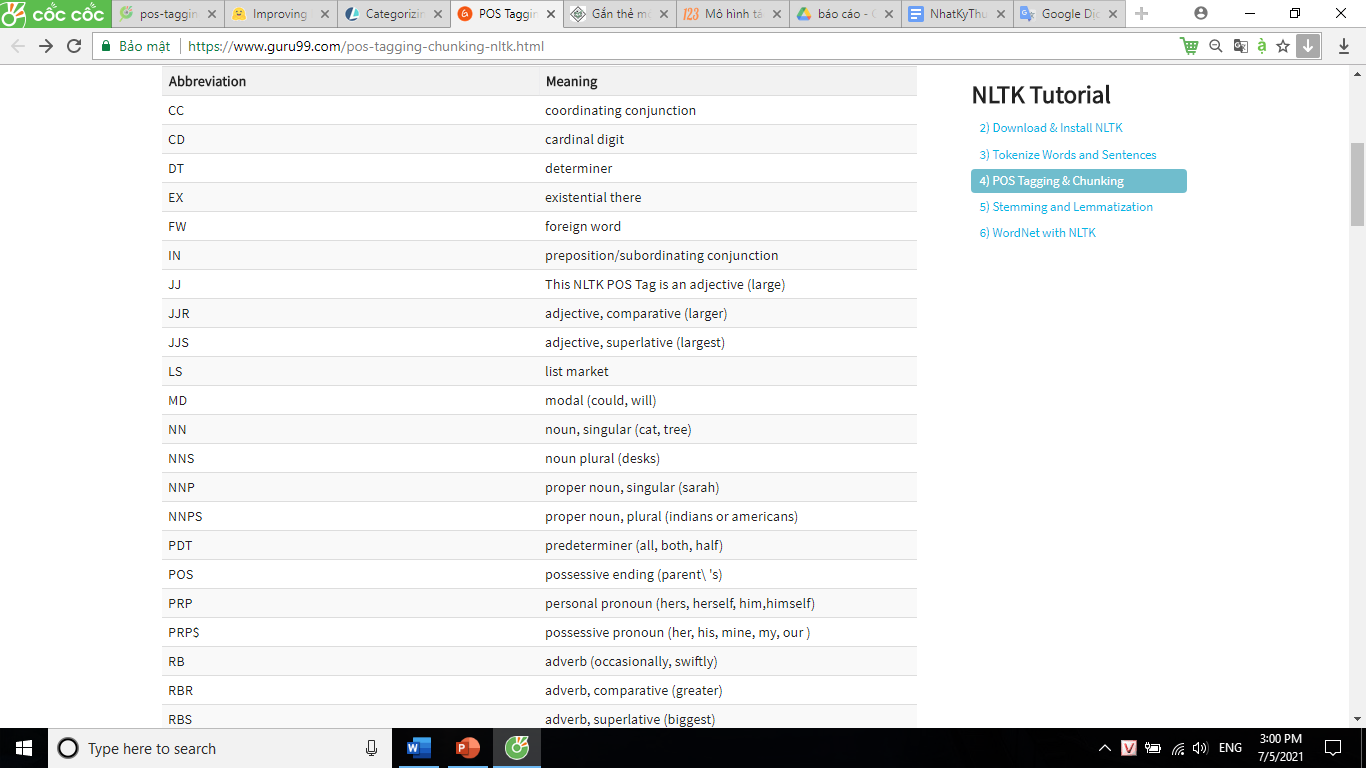
Trong ngôn ngữ học, gắn thẻ phần lời nói hay văn bản, còn được gọi là gắn thẻ ngữ pháp là quá trình đánh dấu một từ trong văn bản (ngữ liệu) tương ứng với một phần cụ thể của lời nói hoặc văn bản được đưa vào dựa trên cả định nghĩa và ngữ cảnh của nó. Hiện nay, việc gắn thể POS đã được thực hiện trong ngữ cảnh ngôn ngữ học tính toán , sử dụng các thuật toán liên kết các thuật ngữ rời rạc, cũng như các phần ẩn của văn bản, bằng một tập hợp các thẻ mô tả. Các thuật toán gắn thẻ POS được chia thành hai nhóm đặc biệt: dựa trên quy tắc và ngẫu nhiên.

Gãn nhãn từ loại là một công việc quan trọng và bắt buộc phải có đối với hầu hết các ứng dụng xử lý ngôn ngữ tự nhiên. Công việc gán nhãn từ loại cho một văn bản là xác định từ loại của mỗi từ trong phạm vi văn bản đó, tức là phân loại các từ vào các lớp từ loại của ngôn ngữ đó:

* Input: Một chuỗi các từ và tập nhãn từ loại.
* Output: Một nhãn phù hợp cho từng từ trong câu.

Quá trình gán nhãn từ loại có thể chia làm 3 bước như sau:

* Giai đoạn tiền xử lý: Phân tách xâu ký tự thành chuỗi các từ. Giai đoạn này có thể đơn giản hay phức tạp tuỳ theo ngôn ngữ và quan niệm về đơn vị từ vựng. Đối với đề tài “Nhận dạng và phân loại các thực thể trong văn bản tiếng Việt”, vấn đề tiền xử lý văn bản gặp khó khăn hơn khi khoảng trắng không phải là dấu hiệu để xác định ranh giới các đơn vị từ vựng do tần số xuất hiện từ ghép rất cao.
* Khởi tạo gán nhãn: Tức là tìm cho mỗi từ tập tất cả các nhãn từ loại mà nó có thể có. Tập nhãn này được hình thành dựa trên các cơ sở dữ liệu có sẵn, bên cạnh đó có thể tạo lập một bộ nhãn gán mới theo phương pháp thủ công dùng suy nghĩ con người.



Hình 3.3 Ví dụ về thẻ POS NLTK [5]

* Quyết định kết quả gán nhãn: Đây là giai đoạn lựa chọn nhãn gán phù hợp cho mỗi từ trong chuỗi được đưa vào dựa trên các từ ngữ khác và ngữ nghĩa của đoạn văn.

Một trong những kỹ thuật gắn thẻ lâu đời nhất là gắn thẻ POS dựa trên quy tắc. Trình gắn thẻ dựa trên quy tắc sử dụng từ điển hoặc từ vựng để nhận các thẻ có thể có để gắn thẻ từng từ. Nếu từ có thể có nhiều hơn một thẻ, thì các trình gắn thẻ dựa trên quy tắc sẽ sử dụng các quy tắc viết tay để xác định đúng thẻ. Việc phân định cũng có thể được thực hiện trong việc gắn thẻ dựa trên quy tắc bằng cách phân tích các đặc điểm ngôn ngữ của một từ cùng với các từ đứng trước cũng như theo sau của nó.

Như tên cho thấy, tất cả các loại thông tin như vậy trong gắn thẻ POS dựa trên quy tắc được mã hóa dưới dạng quy tắc. Các quy tắc này có thể là:

* Quy tắc mẫu ngữ cảnh
* Biểu thức chính quy được biên dịch thành tự động ở trạng thái hữu hạn, xen kẽ với biểu diễn câu không rõ ràng về mặt từ vựng.

Có thể hiểu việc gắn thẻ POS dựa trên Quy tắc bằng kiến ​​trúc hai giai đoạn của nó:

* **Giai đoạn đầu tiên** - Trong giai đoạn đầu tiên, nó sử dụng từ điển để gán cho mỗi từ một danh sách các phần tiềm năng của giọng nói.
* **Giai đoạn thứ hai** - Trong giai đoạn thứ hai, nó sử dụng danh sách lớn các quy tắc phân định được viết tay để sắp xếp danh sách thành một phần của giọng nói duy nhất cho mỗi từ.

# **CHƯƠNG 4: KẾT QUẢ**

* 1. **Sử dụng mô hình BERT để pre-trained model**

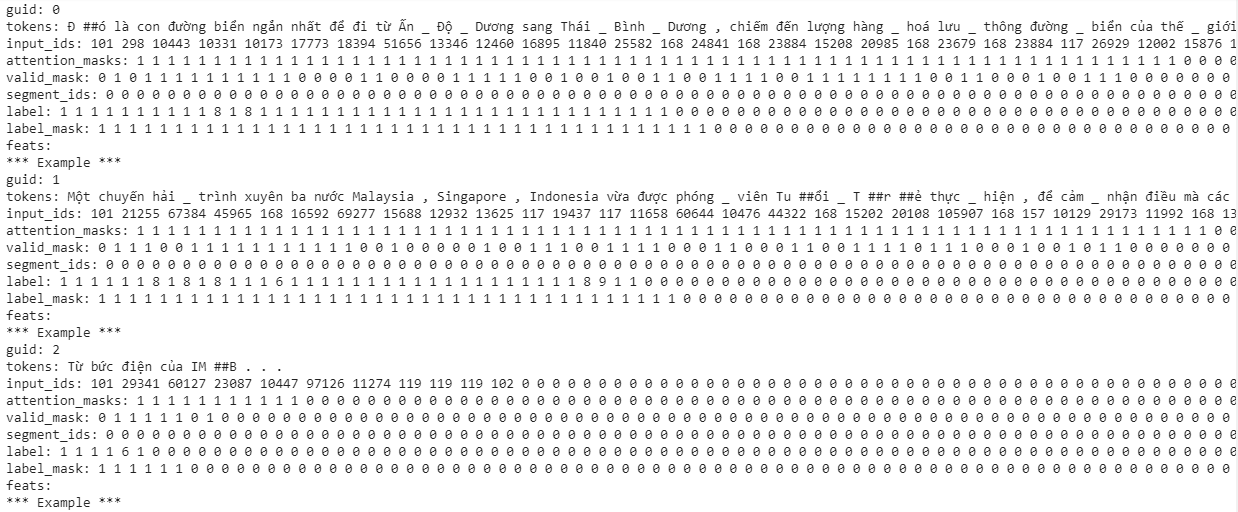
Để có thể thực hiện xử lí pre-trained model của BERT, đầu tiên cần đọc dữ liệu huấn luyện và dữ liệu test, tiền xử lý chúng bằng cách sử dụng wordsegment.

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 4.1 Tiền xử lí bằng word segment

Sau khi xong công đoạn xử lí sẽ có một list chứa các dữ liệu đã qua tách từ, ứng với list label là các nhãn dán 0 và 1. Đến đây, các bước tiền xử lí được xem đã xong. Tiếp theo, từ các dữ liệu mẫu đã có, các câu ngắn hơn 128 từ được padding 0 ở cuối, những câu dài hơn sẽ được cắt đi cho đủ 128. Tạo một mask gồm các giá trị 0, 1 để làm đầu vào cho thư viện transformers, mask này cho biết các giấ trị nào của chuỗi đã được padding.



Hình 4.2 List label và mask gồm các giá trị 0 và 1

Cuối cùng, việc chuẩn bị dữ liệu đầu vào đã xong. Tiếp tục đến với việc load model BERT với thư viện Transformers.

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 4.3 Code load model BERT

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 4.4 Kết quả load model BERT

Vậy là đã đi đến phần trainning model để máy có thể học được cách tự phân tách và gán nhãn từ loại trong văn bản. Dưới đây là phần code cho phần train model:

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 4.5 Code train model BERT

* 1. **Kết quả train model**

Với bộ dữ liệu khá lớn nên để train xong 1 epoch thì sẽ mất khoảng 35-40 phút để train xong 6 epoch và dưới đây là kết quả sau 3 tiếng 50 phút.

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 4.6 Kết quả sau khi train model BERT (Epoch 0 - 1)

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 4.7 Kết quả sau khi train model BERT (Epoch 2 - 3)

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 4.8 Kết quả sau khi train model BERT (Epoch 4 - 5)

|  |  |
| --- | --- |
| Epoch 0 | Train loss: 195,45  F1-Score: 0,49 |
| Epoch 1 | Train loss: 28,5  F1-Score: 1.0 |
| Epoch 2 | Train loss: 13,9  F1-Score: 1.0 |
| Epoch 3 | Train loss: 7,7  F1-Score: 1.0 |
| Epoch 4 | Train loss: 4,3  F1-Score: 1.0 |
| Epoch 5 | Train loss: 28,5  F1-Score: 1.0 |

Bảng 4.1 Điểm đánh giá sau khi train 6 epochs

Qua bảng trên độ chính xác của mô hình rất tốt thể hiện qua 6 epochs và nhận được F1-Score=1.0 ngay từ epoch 1 đến epoch 5. Kết quả có vẻ khá đáng khích lệ vì dường như đã tạo ra một mô hình gần như hoàn hảo để nhận diện thực thể trong văn bản. Với mô hình đã được train bằng BERT này với độ chính xác cao, thì chúng ta có thể áp dụng vào việc nhận diện thực thể trong ngôn ngữ tự nhiên là rất tốt.

* 1. **Thực hiện nhận diện thực thể bằng mô hình BERT**
* Dữ liệu đầu vào:

*“Biến căng: Ngô Diệc Phàm bị hotgirl tố lập group chat chuyên "săn" gái vị thành niên, lựa chọn nữ sinh như món hàng”*

*“Tang lễ cố diễn viên Đức Long: Diễn ra trong vắng lặng, bố mẹ ngồi thất thần, Cao Thái Hà xúc động chia sẻ về người bạn quá cố”*

*“9h30: Lễ nhập quan của cố diễn viên đã được hoàn tất. Cao Thái Hà bức xúc vì tin đồn thất thiệt liên quan tới người bạn quá cố ”*

Ảnh có chứa văn bản, biên lai

Mô tả được tạo tự động

Hình 4.9 Kết quả thực hiện tách và gán nhãn thực thể

Từ mô hình BERT đã train, sau khi predict file để thực hiện tách và gán nhãn thực thể, chúng ta có thể thấy độ chính xác của mô hình khá tốt. Như vậy với mô hình BERT chúng ta có thể rút ngắn nội dung và đưa ra các keyword của các dữ liệu lớn tốn nhiều thời gian để đọc.

# **KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

* **Kết luận**
* *Những vấn đề đã được giải quyết*

Báo cáo đã hệ thống hóa vấn đề lý thuyết về ngôn ngữ lập trình Python và nhận dạng phân loại các thực thể trong văn bản, đồng thời trình bày, phân tích, đánh giá từng bước trong quá trình tiếp cận bài toán NER. Các thư viện Python phục vụ cho bài toán xử lý ngôn ngữ tự nhiên đã được hệ thống hóa, phân tích các mặt lý thuyết về ưu nhược điểm cũng như các cách ứng dụng thực tiễn thư viện hỗ trợ xây dựng mô hình Machine Learning. Kết quả của đề tài này chính là hoàn thiện được mô hình tách từ trong văn bản tiếng Việt và thực hiện gán nhãn dựa theo ngữ nghĩa của từ trong hoàn cảnh câu văn. Sau đây là một số vấn đề chính báo cáo đã tập trung giải quyết:

Chương 1 đưa ra cái nhìn khái quát về ngôn ngữ lập trình Python và bài toán nhận diện thực thể trong văn bản tiếng Việt, là bước đầu tiếp cận về tổng quan về một ngôn ngữ lập trình mới cũng như những hiểu biết cơ bản về Named Entity Recognition.

Chương 2 xem xét và đi sâu về lý thuyết Python cũng như các mô hình, thư viện cần thiết trong việc xử lý văn bản ngôn ngữ tiếng Việt. Lý thuyết về Python giúp nhận biết được các ứng dụng, tiện ích của ngôn ngữ lập trình trong mô hình Machine Learning, song song đó là ưu nhược điểm của ngôn ngữ này. Lý thuyết về NER cũng đã được làm rõ hơn về nhiều khía cạnh, đặc biệt là hai phần định dạng dữ liệu và nhận dạng thực thể. Bên cạnh đó, trong nội dung chương 2, nhóm đã trọng tâm làm rõ lý thuyết về việc ứng dụng BERT vào bài toán nhận dạng và xử lý thực thể văn bản tiếng Việt. Từ đó làm rõ cấu trúc mô hình Machine Learning được xây dựng cũng như xác định được hướng thức hiện của đề tài.

Chương 3 đã định hướng những bước thực hiện của đề tài, xác định được các nguồn thông tin để tìm kiếm code phù hợp thực hiện chạy thử. Ở chương này, nhóm đã xác định rõ hơn về việc cài đặt phiên bản Python và Pytorch phù hợp cho việc thực hiện đề tài cũng như tìm hiểu về chức năng các thư viện cần thiết giúp việc thực hành trở nên dễ dàng hơn. Việc xác định rõ từng bước triển khai thực hiện giúp nhóm định hướng được đề tài và dễ dàng hơn trong việc chỉnh sửa lỗi. Ngoài ra, các quá trình thực hiện đào tạo dữ liệu, xử lý tách từ cũng như gán nhãn đã được nêu rõ trong chương kể cả phần lý thuyết và hướng thực hiện cụ thể. Chương triển khai thực hiện chính là bước quan trong trong việc bắt đầu thực hành xây dựng mô hình nhận dạng và phân loại thực thể trong văn bản tiếng Việt bằng Google Colaboratory.

Tại chương 4, phần kết quả được thể hiện qua 3 mục chính: sử dụng mô hình BERT để train dữ liệu, kết quả pre-trained model và thực hiện nhận diện thực thể bằng mô hình BERT. Chương này đã nêu rõ các bước tiền xử lý dữ liệu cũng như phân tách từ ứng với list label là các nhãn dán 0 và 1, đây là những bước chính trong quá trình chuẩn bị dữ liệu đầu vào. Phần train model đã cho thấy độ chính xác của mô hình rất tốt thể hiện qua 6 epochs và nhận được F1-Score=1.0. Đối với phần gán nhãn thực thể, mô hình đã hoàn thiện và đưa ra được kết quả với độ chính xác khá tốt, đây cũng chính là kết quả cuối cùng đề tài hướng đến.

* *Những vấn đề trong tương lai*

Đề tài đã nêu rõ cũng như hoàn thành mô hình nhận dạng và phân loại thực thể trong văn bản tiếng Việt. Trong tương lai, nhóm chúng em mong muốn sẽ có thể phát triển mô hình giúp tăng cường độ chính xác trong công đoạn gắn nhãn từ được tách, áp dụng được bộ dữ liệu mang tính hoàn chỉnh và chi tiết hơn để có thể áp dụng thực tiễn tại nhiều lĩnh vực. Song song đó là cải thiện khả năng dự đoán từ không rõ ràng (viết tắt, sai chính tả, …) và dự đoán câu văn tiếp theo trong đoạn văn. Với những lý thuyết và mô hình đã được tìm hiểu và hoàn thành trong đề tài, đây sẽ là tiền đề để áp dụng vào các bài toán phức tạp khác trong lĩnh vực Machine Learning.

* *Thuận lợi và khó khăn*

Với sự hướng dẫn tận tình về kì thực tập của các thầy cô trong khoa Thống kê – Tin học nói chung cũng như sự chỉ dẫn đề tài của thầy Nguyễn Thành Thủy nói riêng, nhóm chúng em đã có thể giải đáp được nhiều thắc mắc trong quá trình thực hiện đề tài và hoàn thành báo cáo. Bên cạnh đó, sự phát triển của internet đã cung cấp một nguồn thông tin rộng lớn và hữu ích trong vấn đề tìm hiểu về lý thuyết và tìm kiếm thực hành các source code, giúp vấn đề xây dựng mô hình trở nên thuận lợi và mất ít thời gian hơn.

Song song với những thuận lợi, vấn đề về dịch bệnh chính là trở ngại lớn nhất trong kì thực tập nhận thức của chúng em khi việc giải đáp thắc mắc qua internet sẽ đem lại nhiều khó khăn hơn so với trao đổi trực tiếp cùng thầy cô. Thành viên trong nhóm cũng họp và góp ý qua internet, làm hạn chế khả năng trao đổi và sửa lỗi chương trình cùng nhau.

* **Ý nghĩa và hướng phát triển**

Bài toán nhận dạng thực thể đã được quan tâm nghiên cứu và đạt được một số kết quả nhất định, tuy nhiên các kết quả này hầu hết chỉ xử lý cho các thực thể thông thường trong văn bản tiếng Anh. Trong khi đấy, nhận dạng thực thể với miền dữ liệu văn bản thông thường tiếng Việt vẫn còn nhiều vấn đề lớn đặt ra như không có một tập dữ liệu chuẩn cùng những mô tả rõ ràng về khái niệm liên quan đến các thực thể hay các tài nguyên công cụ phục vụ cho việc nhận dạng. Việc phát triển hệ thống nhận diện thực thể ngôn ngữ tiếng Việt sẽ giúp ích cho máy tính trong việc đọc hiểu văn bản, nâng cao khả năng phân loại từ và tăng độ chính xác cho hệ thống.

Đề tài hướng đến mục tiêu cải thiện độ chính xác khả năng nhận diện thực thể, có thể áp dụng rộng rãi ở nhiều lĩnh vực cũng như dự án khác. Đây là tiền đề cho hướng phát triển tương lai của các hệ thống nâng cao khác như: nhận diện cảm xúc văn bản, chatbot, …

# **TÀI LIỆU THAM KHẢO**

1. <https://www.docstring.fr/blog/5-raisons-apprendre-python/>
2. <http://www.e-ptit.edu.vn/top-10-ngon-ngu-lap-trinh-pho-bien-nhat-2020/>
3. <http://www.vietlex.com/xu-li-ngon-ngu/152-NHAN_DANG_THUC_THE_DINH_DANH_TRONG_VAN_BAN_TIENG_VIET>
4. <https://humboldt-wi.github.io/blog/research/information_systems_1920/bert_blog_post/>
5. <https://www.guru99.com/pos-tagging-chunking-nltk.html>
6. Hoàng Thị Vân Anh. (2012). *Trích rút thực thể tên từ các văn bản tiếng Việt trên internet.* Địa chỉ:

<https://123docz.net//document/1040887-trich-rut-thuc-the-tu-cac-van-ban-tieng-viet-tren-internet.htm>

1. Học viện đào tạo CNTT NIIT - ICT Hà Nội. (2020). TOP 10 Thư viện Khoa học Dữ liệu của Python. Địa chỉ:

<https://niithanoi.edu.vn/10-thu-vien-khoa-hoc-du-lieu-tot-nhat-cua-python.html>

1. Lê Minh Dũng. (2017). *Hướng dẫn NLP bằng Python NLTK (Ví dụ đơn giản).* Địa chỉ:

<https://helpex.vn/article/huong-dan-nlp-bang-python-nltk-vi-du-don-gian-5c663a0eae03f60128764f35>

1. Mohd Sanad Zaki Rizvi. (2019). *Introduction to PyTorch-Transformers: An Incredible Library for State-of-the-Art NLP (with Python code).* Địa chỉ:

<https://www.analyticsvidhya.com/blog/2019/07/pytorch-transformers-nlp-python/>

1. Nguyễn Bá Đạt. (2013). *Nhận dạng thực thể trong văn bản tiếng Việt.* Địa chỉ: <http://www.luanvan.co/luan-van/luan-van-nhan-dang-thuc-the-trong-van-ban-tieng-viet-39286/>
2. Phạm Hữu Quang. (2018). *BERT- bước đột phá mới trong công nghệ xử lý ngôn ngữ tự nhiên của Google*. Địa chỉ:

<https://viblo.asia/p/bert-buoc-dot-pha-moi-trong-cong-nghe-xu-ly-ngon-ngu-tu-nhien-cua-google-RnB5pGV7lPG>

1. QuangPH. (2018). *Hiểu hơn về BERT: Bước nhảy lớn của Google*. Địa chỉ:

<https://laptrinhx.com/hieu-hon-ve-bert-buoc-nhay-lon-cua-google-2060621192/>

1. Tram Ho. (2020). *Nhận dạng thực thể - NER.* Địa chỉ:

<https://itzone.com.vn/vi/article/seri-nlp-nhan-dang-thuc-the-ner-phan-1/>

1. Trần Thị Oanh. (2015). *Mô hình tách từ gán nhãn từ loại và hướng tiếp cận tích hợp cho tiếng Việt.* Địa chỉ:

<https://123docz.net/document/2596019-mo-hinh-tach-tu-gan-nhan-tu-loai-va-huong-tiep-can-tich-hop-cho-tieng-viet.htm>

1. Vũ Anh. (2021). *Underthesea*. Địa chỉ:

<https://pypi.org/project/underthesea/>