BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC THĂNG LONG**

A red and blue logo

AI-generated content may be incorrect.

**BÁO CÁO BÀI TẬP LỚN**

**PHÂN LOẠI THƯ RÁC**

**GIÁO VIÊN HƯỚNG DẪN: NGUYỄN HÙNG CƯỜNG**

**SINH VIÊN THỰC HIỆN: NGUYỄN THỊ THÙY TRANG**

**MÃ SINH VIÊN: A46483**

**BỘ MÔN: PHÁT TRIỂN ỨNG DỤNG VỚI PYTHON**

**HÀ NỘI 2025**

**MỤC LỤC**

[LỜI MỞ ĐẦU 1](#_Toc195857202)

[CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI 2](#_Toc195857203)

[1.1. Lý do chọn đề tài 2](#_Toc195857204)

[1.2. Tính cấp thiết của đề tài 2](#_Toc195857205)

[1.3. Lịch sử nghiên cứu 3](#_Toc195857206)

[1.4. Mục tiêu, đối tượng và phạm vi nghiên cứu 3](#_Toc195857207)

[1.5. Phương pháp nghiên cứu 4](#_Toc195857208)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 5](#_Toc195857209)

[2.1. Xử lý Ngôn ngữ Tự nhiên (NLP) 5](#_Toc195857210)

[2.1.1. Khái niệm NLP 5](#_Toc195857211)

[2.1.2. Ý nghĩa của NLP 5](#_Toc195857212)

[2.1.3. Quy trình của mô hình NLP 6](#_Toc195857213)

[2.1.4. Giai đoạn xử lý ngôn ngữ tự nhiên 6](#_Toc195857214)

[2.1.5. Các mô hình NLP nổi bật 18](#_Toc195857215)

[2.1.6. Các thuật toán trong NLP 19](#_Toc195857216)

[2.1.7. Ứng dụng của NLP 19](#_Toc195857217)

[2.1.8. Thách thức trong NLP 22](#_Toc195857218)

[2.2. Học máy (Machine Learning) 22](#_Toc195857219)

[2.2.1. Khái niệm về học máy 22](#_Toc195857220)

[2.2.2. Quá trình học máy 22](#_Toc195857221)

[2.2.3. Các mô hình học máy 22](#_Toc195857222)

[2.3. Học sâu (Deep Learning) 23](#_Toc195857223)

[2.3.1. Khái niệm về học sâu 23](#_Toc195857224)

[2.3.2. Các loại mô hình trong học sâu: 23](#_Toc195857225)

[2.4. Các thư viện lập trình 24](#_Toc195857226)

[2.4.1. Pandas 24](#_Toc195857227)

[2.4.2. Matplotlib 24](#_Toc195857228)

[2.4.3. seaborn 25](#_Toc195857229)

[2.4.4. Numpy 25](#_Toc195857230)

[2.4.5. nltk 26](#_Toc195857231)

[2.4.6. re 29](#_Toc195857232)

[2.4.7. Scikit-learning 30](#_Toc195857233)

[2.4.8. Tensorflow 30](#_Toc195857234)

[2.4.9. pickle 31](#_Toc195857235)

[2.5. Chỉ số đánh giá hiệu suất mô hình 31](#_Toc195857236)

[2.5.1. Confusion Matrix: 31](#_Toc195857237)

[2.5.2. Accuracy 32](#_Toc195857238)

[2.5.3. Precision: 32](#_Toc195857239)

[2.5.4. Recall 32](#_Toc195857240)

[2.5.5. F1-Score 33](#_Toc195857241)

[2.5.6. AUC-ROC 33](#_Toc195857242)

[2.6. Framework Django 33](#_Toc195857243)

[2.6.1. Khái niệm về framework Django 33](#_Toc195857244)

[2.6.2. Tính năng chính của Django 33](#_Toc195857245)

[CHƯƠNG 3. THỰC NGHIỆM 34](#_Toc195857246)

[3.1. Quy trình thực hiện 34](#_Toc195857247)

[3.2. Đọc dữ liệu 34](#_Toc195857248)

[3.3. Tiền xử lý dữ liệu 35](#_Toc195857249)

[3.4. Chia dữ liệu 37](#_Toc195857250)

[3.5. Xây dựng và huấn luyện mô hình 37](#_Toc195857251)

[3.5.1. Random Forest 38](#_Toc195857252)

[3.5.2. LSTM 38](#_Toc195857253)

[3.6. Đánh giá mô hình 40](#_Toc195857254)

[3.6.1. Độ chính xác 40](#_Toc195857255)

[3.6.2. Ma trận nhầm lẫn 41](#_Toc195857256)

[CHƯƠNG 4. MÔ PHỎNG CHƯƠNG TRÌNH 42](#_Toc195857257)

[4.1. Tổng quan về trang web mô phỏng cho bài toán phân loại thư rác 42](#_Toc195857258)

[4.1.1. Mục tiêu của trang web: 42](#_Toc195857259)

[4.1.2. Mô hình dùng cho mô phỏng: 42](#_Toc195857260)

[4.1.3. Quy trình hoạt động của trang web: 42](#_Toc195857261)

[4.1.4. Các công nghệ sử dụng 43](#_Toc195857262)

[4.2. Xây dựng trang web với Django 43](#_Toc195857263)

[4.2.1. Cài đặt Django 43](#_Toc195857264)

[4.2.2. Cài đặt thư viện liên quan 44](#_Toc195857265)

[4.2.3. Tạo giao diện cho trang web 45](#_Toc195857266)

[4.2.4. Tạo view cho trang web 45](#_Toc195857267)

[4.2.5. Demo kết quả 47](#_Toc195857268)

[4.3. Deploy với docker 48](#_Toc195857269)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 51](#_Toc195857270)

[TÀI LIỆU THAM KHẢO 53](#_Toc195857271)

**DANH MỤC HÌNH ẢNH**

[Hình 1. Minh họa cho Tokennization 7](#_Toc195849916)

[Hình 2. Minh họa rút gọn và phục hồi từ 8](#_Toc195849917)

[Hình 3. Minh họa cho stopwords removal 8](#_Toc195849918)

[Hình 4. Minh họa cho BoW 10](#_Toc195849919)

[Hình 5. Minh họa cho word2vec 11](#_Toc195849920)

[Hình 6. Minh họa cho N-grams 11](#_Toc195849921)

[Hình 7. Minh họa cho gán nhãn từ loại 12](#_Toc195849922)

[Hình 8. Minh họa cây phụ thuộc 13](#_Toc195849923)

[Hình 9. Minh họa cây thành phần 13](#_Toc195849924)

[Hình 10. Minh họa tính mạch lạc trong cấu trúc diễn ngôn phân cấp 15](#_Toc195849925)

[Hình 11. Phân tích cảm xúc có trong văn bản 21](#_Toc195849926)

[Hình 12. Số lượng email spam (1) và email hợp lệ (0) 35](#_Toc195849927)

[Hình 13. Lỗi trùng lặp trong bộ dữ liệu 35](#_Toc195849928)

[Hình 14. Đồ thị của hàm mất mát 39](#_Toc195849929)

[Hình 15. Đồ thị của hàm mất mát sau chỉnh sửa 40](#_Toc195849930)

[Hình 16. Giao diện của web 47](#_Toc195849931)

[Hình 17. Giao diện của web khi thử với nội dung spam 48](#_Toc195849932)

[Hình 18. Giao diện của web khi thử với nội dung không spam 48](#_Toc195849933)

**DANH MỤC BẢNG**

[Bảng 1. Ma trận nhầm lẫn với trường hợp phân loại nhị phân 32](#_Toc195849958)

[Bảng 2. Đánh giá mô hình 40](#_Toc195849959)

[Bảng 3. Ma trận nhầm lẫn của mô hình 41](#_Toc195849960)

# LỜI MỞ ĐẦU

Trong thời đại công nghệ thông tin phát triển mạnh mẽ như hiện nay, email đã trở thành một phương tiện liên lạc quan trọng và không thể thiếu trong hoạt động kinh doanh, học tập và đời sống hằng ngày. Tuy nhiên, cùng với sự phát triển đó, email spam (thư rác) cũng trở thành một vấn đề lớn, gây ra nhiều phiền toái cho người dùng và tiềm ẩn các nguy cơ bảo mật nghiêm trọng như lừa đảo, xâm nhập trái phép vào hệ thống thông tin.

Email spam không chỉ làm gián đoạn công việc mà còn có thể gây tổn thất tài chính và mất mát thông tin quan trọng nếu người dùng vô tình nhấp vào các liên kết độc hại. Do đó, việc phát triển một hệ thống có khả năng phân loại email spam một cách chính xác và hiệu quả là vô cùng cần thiết.

Xuất phát từ thực tiễn đó, tôi quyết định lựa chọn đề tài “Phân loại email spam” với mục tiêu xây dựng và tối ưu hóa mô hình có khả năng phân loại email spam một cách hiệu quả, từ đó giúp người dùng giảm thiểu rủi ro khi sử dụng email và nâng cao chất lượng thư điện tử. Bố cục của báo cáo gồm: Lời mở đầu, 4 chương, kết luận và tài liệu tham khảo.

**MỞ ĐẦU:** Phần này gồm có các danh mục hình ảnh, bảng, các thuật ngữ viết tắt sử dụng và giới thiệu tổng quan về bố cục nội dung trình bày trong báo cáo

**CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI:** Chương này sẽ trình bày tổng quan về đề tài: lý do lựa chọn đề tài, tính cấp thiết của đề tài, lịch sử nghiên cứu, mục đính, đối tượng, phạm vi nghiên cứu và phương pháp nghiên cứu của đề tài.

**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT:** Chương này sẽ trình bày các kiến thức về các công cụ, thuật toán, mô hình sử dụng để nghiên cứu đề tài và các chỉ số đánh giá hiệu năng mô hình.

**CHƯƠNG 3: THỰC NGHIỆM:** Chương này sẽ đưa ra các bước thực hiện phân loại email spam (đọc dữ liệu, tiền xử lý dữ liệu, xây dựng thuật toán, huấn luyện mô hình, đánh giá).

**CHƯƠNG 4: MÔ PHỎNG CHƯƠNG TRÌNH:** Chương này sẽ thực hiện xây dựng ứng dụng Python với giao diện Web cơ bản để minh hoạ kết quả nghiên cứu xử lý ngôn ngữ tự nhiên trong Python.

**PHẦN KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN:** Phần này trình bày về ưu điểm của mô hình phân loại, những tồn tại và định hướng phát triển đề tài trong tương lai.

## TỔNG QUAN VỀ ĐỀ TÀI

### Lý do chọn đề tài

Thư điện tử (email) là một hệ thống trao đổi thông tin qua mạng máy tính, cho phép gửi và nhận văn bản, hình ảnh hoặc âm thanh.

Thư rác (spam) là những email hoặc tin nhắn không mong muốn, với nội dung quảng cáo hoặc phát tán nội dung gây phiền nhiễu, tiềm ẩn nguy cơ lừa đảo, phân tán phần mềm độc hại và thường được gửi hàng loạt với số lượng lớn tới một tập hợp người nhận không phân biệt.

Trong thời đại số hiện nay, việc trao đổi thông tin bằng email trở nên cực kỳ phổ biến. Trên thực tế, thư rác (spam) đã trở thành một vấn đề đáng lo ngại. Hòm thư điện tử của chúng ta nhận được hàng trăm, thậm chí hàng nghìn thư rác mỗi ngày. Thư rác không chỉ làm mất thời gian của người dùng mà còn gây tốn kém tài nguyên, từ việc lọc thư đến bảo vệ an ninh mạng.

Vậy nên, lọc hay phân loại thư rác là rất cần thiết. Phân loại thư rác là quá trình tách biệt các email không mong muốn khỏi hộp thư hợp lệ. Việc xây dựng mô hình học máy tích hợp xử lý ngôn ngữ tự nhiên (NLP) nhằm tự động nhận diện và phân loại thư rác giúp cải thiện trải nghiệm người dùng và nâng cao hiệu quả bảo mật. Xuất phát từ thực tế đó, chúng tôi chọn đề tài “Phân loại email spam” để thực hiện nghiên cứu.

### Tính cấp thiết của đề tài

Email là một trong những phương thức giao tiếp chính trong công việc và đời sống cá nhân. Tuy nhiên, sự gia tăng nhanh chóng về số lượng email đã kéo theo vấn nạn email spam như các thư rác không mong muốn, thường chứa các nội dung quảng cáo, lừa đảo hoặc độc hại.

Tác hại của email spam:

* Giảm hiệu suất làm việc: Người dùng mất nhiều thời gian để phân loại và xử lý các email không mong muốn.
* Ảnh hưởng đến an ninh mạng: Nhiều email spam chứa mã độc, đường dẫn giả mạo, có thể dẫn đến lừa đảo trực tuyến (phishing), đánh cắp thông tin cá nhân hoặc xâm nhập hệ thống.
* Lãng phí tài nguyên: Việc xử lý và lưu trữ email spam làm tiêu tốn băng thông, dung lượng máy chủ và tài nguyên tính toán.

Các phương pháp lọc thủ công không còn hiệu quả do số lượng email quá lớn và sự tinh vi của các loại spam. Việc phát triển một hệ thống phân loại email chính xác, tự động thích nghi với các phương thức spam mới là rất cần thiết để bảo vệ người dùng và duy trì môi trường email an toàn.

### Lịch sử nghiên cứu

Những nghiên cứu về phát hiện thư rác đã bắt đầu từ những năm 1990. Ban đầu, các hệ thống lọc spam chủ yếu dựa trên các quy tắc (rule-based systems), trong đó người dùng hoặc nhà cung cấp dịch vụ email tạo ra danh sách chặn (blacklist) và danh sách trắng (whitelist) để xác định các email hợp lệ hoặc không hợp lệ. Với sự gia tăng nhanh chóng của thư rác và sự sáng tạo của những kẻ gửi thư rác (thường xuyên thay đổi địa chỉ và tên miền), phương pháp này đã không còn hiệu quả trong việc phát hiện và lọc thư rác một cách chính xác.

Vào những năm 2000, các phương pháp phân loại thư rác thông qua học máy bắt đầu được nghiên cứu và áp dụng. Các thuật toán học máy như Naive Bayes, Support Vector Machines (SVM) và K-Nearest Neighbors (KNN) bắt đầu được ứng dụng rộng rãi trong việc phân loại email. Các mô hình này có thể học từ dữ liệu, nghĩa là chúng có khả năng cải thiện hiệu suất theo thời gian khi chúng được huấn luyện với nhiều dữ liệu hơn. Điều này đã giúp giải quyết những hạn chế của phương pháp lọc dựa trên quy tắc, vốn chỉ dựa vào các danh sách đen và trắng cố định.

Sự phát triển của kỹ thuật trích xuất đặc trưng (feature extration) đóng một vai trò quan trọng trong việc cải thiện khả năng phân loại thư rác. Các phương pháp trích xuất đặc trưng như TF-IDF (Term Frequency-Inverse Document Frequency) giúp mô hình có thể phân tích tần suất từ và mức độ quan trọng của từ trong email, qua đó tăng cường khả năng phân loại chính xác hơn.

Với sự gia tăng mạnh mẽ về số lượng và tính đa dạng của thư rác, các nghiên cứu về phân loại thư rác đã chuyển sang những mô hình phức tạp hơn, sử dụng deep learning (học sâu). Những mô hình học sâu, đặc biệt là các neural networks và Recurrent Neural Networks (RNNs), đã bắt đầu được sử dụng để hiểu rõ hơn về cấu trúc ngữ nghĩa của email và phát hiện thư rác trong các tình huống phức tạp hơn.

### Mục tiêu, đối tượng và phạm vi nghiên cứu

* Mục tiêu: Nghiên cứu này nhằm xây dựng một mô hình phát hiện email spam hiệu quả, áp dụng các kỹ thuật Xử lý Ngôn ngữ Tự nhiên (NLP) và mô hình học máy, học sâu.
* Đối tượng nghiên cứu: Các email, bao gồm cả email spam và không spam, sẽ là đối tượng nghiên cứu chính. Dữ liệu email sẽ được thu thập từ nguồn công khai như các bộ dữ liệu email mở (ví dụ: từ Kaggle hoặc các dự án nghiên cứu khác) và có thể bao gồm hộp thư cá nhân để đảm bảo tính thực tiễn trong việc áp dụng mô hình.
* Phạm vi nghiên cứu: Tập trung vào các kỹ thuật xử lý dữ liệu văn bản, trích xuất đặc trưng và huấn luyện mô hình học máy để phân loại email spam. Nghiên cứu cũng bao gồm việc đánh giá hiệu suất các mô hình và triển khai mô hình trong thực tế.

### Phương pháp nghiên cứu

* Thu thập dữ liệu: Thu thập và gắn nhãn dữ liệu email từ các nguồn công khai và hộp thư cá nhân.
* Xử lý dữ liệu: Làm sạch và chuẩn hóa dữ liệu email, sử dụng các kỹ thuật NLP kết hợp với Học Máy.
* Trích xuất đặc trưng: Chuyển đổi văn bản thành vector số bằng các kỹ thuật như Bag-of-Words và TF-IDF.
* Huấn luyện mô hình: Sử dụng các mô hình học máy như Random Forest và mô hình học sâu như LSTM để huấn luyện trên tập dữ liệu.
* Đánh giá mô hình: Sử dụng các chỉ số như độ chính xác, độ nhạy, độ đặc hiệu và F1-score để đánh giá hiệu suất mô hình.
* Mô phỏng: Xây dựng ứng dụng với giao diện web cơ bản để minh họa kết quả.

## CƠ SỞ LÝ THUYẾT

### Xử lý Ngôn ngữ Tự nhiên (NLP)

#### 

#### Khái niệm NLP

Ngôn ngữ tự nhiên là ngôn ngữ mà con người sử dụng hàng ngày để giao tiếp với nhau, bao gồm các ngôn ngữ như tiếng Anh, tiếng Việt, …. Ngôn ngữ tự nhiên bao gồm cả lời nói và văn bản.

Xử lý ngôn ngữ tự nhiên (Natural Language Processing – NLP) là một công nghệ máy học, cung cấp cho máy tính khả năng diễn giải, tương tác và hiểu được ngôn ngữ của con người. NLP giúp máy tính xử lý các dạng dữ liệu ngôn ngữ tự nhiên nhằm thực hiện các tác vụ như phân loại, dịch, tìm kiếm thông tin, và nhiều ứng dụng khác.

Ngày nay, phần lớn dữ liệu đến từ các kênh liên lạc khác nhau như email, tin nhắn văn bản, các bài đăng trên mạng xã hội, tệp video, tệp âm thanh, và nhiều hơn nữa. Những dữ liệu này chủ yếu ở dạng không có cấu trúc, tạo ra một lượng thông tin khổng lồ mà các hệ thống cần phải xử lý và phân tích. Việc xử lý những dạng dữ liệu này đòi hỏi các công nghệ tiên tiến như Xử lý Ngôn ngữ Tự nhiên (NLP), Học máy (Machine Learning), và các phương pháp học sâu (Deep Learning), nhằm phân tích, hiểu và tối ưu hóa thông tin từ những dữ liệu đa dạng này.

#### Ý nghĩa của NLP

Trước đây, máy tính chỉ có thể nhận lệnh thông qua mã lập trình, nhưng nhờ NLP, con người có thể giao tiếp với máy móc bằng ngôn ngữ tự nhiên (văn bản hoặc giọng nói). Xử lý ngôn ngữ tự nhiên đóng vai trò rất quan trọng trong việc phân tích đầy đủ dữ liệu văn bản và giọng nói một cách hiệu quả. Điều này mở ra nhiều cơ hội mới trong tự động hóa, tối ưu hóa quy trình và cải thiện trải nghiệm người dùng.

NLP quan trọng bởi:

* Thu hẹp khoảng cách giữa người và máy: NLP giúp máy móc hiểu và tương tác với ngôn ngữ tự nhiên của con người.
* Tự động hóa quy trình: Phân loại tài liệu, trích xuất thông tin, phân tích cảm xúc và hỗ trợ dịch vụ khách hàng tự động
* Ứng dụng đa dạng công việc: Hỗ trợ trợ lý ảo, chatbot, dịch máy, tìm kiếm thông tin và phân tích phản hồi người dùng.
* Phân tích dữ liệu phi cấu trúc: Hệ thống trích xuất thông tin quan trọng từ văn bản, hồ sơ y tế và hợp đồng, giúp doanh nghiệp đưa ra các quyết định chính xác hơn.
* Xóa bỏ rào cản ngôn ngữ: Với khả năng dịch ngôn ngữ tự động, NLP hỗ trợ giao tiếp đa ngôn ngữ, thúc đẩy hợp tác quốc tế và kinh doanh toàn cầu.

#### Quy trình của mô hình NLP

* Thu thập dữ liệu: Dữ liệu ngôn ngữ thô được lấy từ các nguồn như văn bản, giọng nói, email, khảo sát và bản ghi âm. Nguồn dữ liệu này có thể đến từ các nền tảng trực tuyến, hệ thống CRM hoặc kho dữ liệu của doanh nghiệp.
* Xử lý dữ liệu: Dữ liệu thô được làm sạch và chuẩn bị trước khi đưa vào mô hình.
* Đào tạo mô hình: Dữ liệu đã được xử lý được đưa vào mô hình ngôn ngữ để học. Mô hình sẽ học từ dữ liệu mẫu và tối ưu hóa các tham số để đưa ra dự đoán chính xác hơn.
* Triển khai và suy luận: Mô hình được tích hợp vào hệ thống sản xuất hoặc các ứng dụng thực tế để xử lý dữ liệu trực tiếp. Mô hình ngôn ngữ này sẽ nhận dữ liệu đầu vào (văn bản, giọng nói, email,…), phân tích nó và đưa ra kết quả theo yêu cầu của ứng dụng.

#### Giai đoạn xử lý ngôn ngữ tự nhiên

##### Phân tích cấp độ từ (Word Level Analysis)

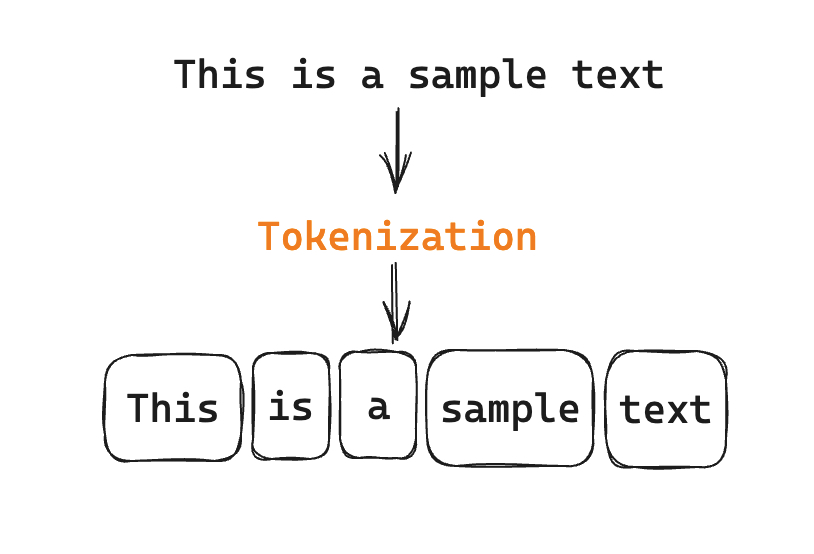
Word Level Analysis là một phần quan trọng trong NLP, đóng vai trò như bước tiền xử lý đầu tiên, giúp chuẩn bị và làm sạch văn bản để mô hình học máy có thể xử lý và hiểu được dữ liệu ngôn ngữ tự nhiên. Phân tích cấp độ từ chủ yếu tập trung vào việc tách các từ ra khỏi văn bản và xử lý các yếu tố liên quan đến chúng để làm cho dữ liệu trở nên dễ dàng sử dụng trong các bước tiếp theo của quá trình NLP. Các bước chính trong Word Level Analysis bao gồm:

* Tách câu (Sentence Segmentation): Tách văn bản ra thành các câu riêng biệt dựa theo chấm câu.

Ví dụ: “ Trước đây, máy tính chỉ có thể nhận lệnh thông qua mã lập trình, nhưng nhờ NLP, con người có thể giao tiếp với máy móc bằng ngôn ngữ tự nhiên (văn bản hoặc giọng nói). Xử lý ngôn ngữ tự nhiên đóng vai trò rất quan trọng trong việc phân tích đầy đủ dữ liệu văn bản và giọng nói một cách hiệu quả. Điều này mở ra nhiều cơ hội mới trong tự động hóa, tối ưu hóa quy trình và cải thiện trải nghiệm người dùng.”

Sau khi áp dụng tách câu, kết quả đầu ra là:

* Trước đây, máy tính chỉ có thể nhận lệnh thông qua mã lập trình, nhưng nhờ NLP, con người có thể giao tiếp với máy móc bằng ngôn ngữ tự nhiên (văn bản hoặc giọng nói)
* Xử lý ngôn ngữ tự nhiên đóng vai trò rất quan trọng trong việc phân tích đầy đủ dữ liệu văn bản và giọng nói một cách hiệu quả
* Điều này mở ra nhiều cơ hội mới trong tự động hóa, tối ưu hóa quy trình và cải thiện trải nghiệm người dùng
* Tách từ (Tokenization): tách câu thành các từ hoặc cụm từ nhỏ hơn.

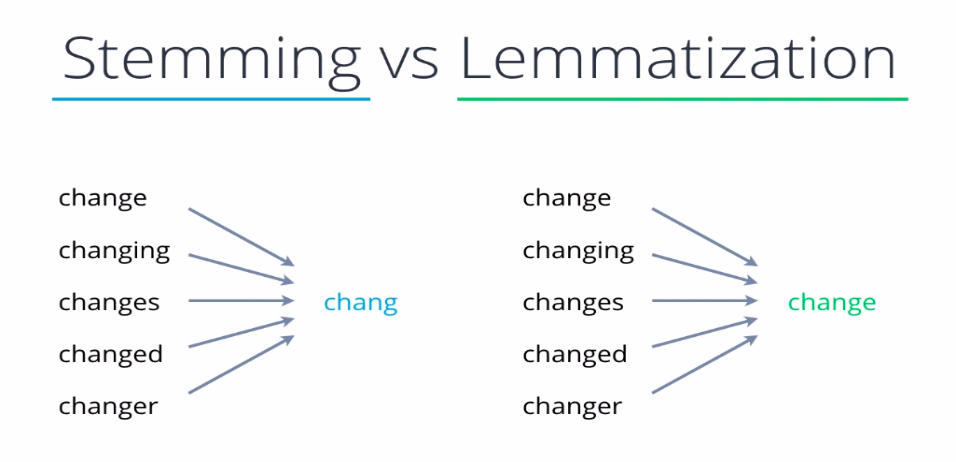


Hình . Minh họa cho Tokennization

* Việc Tokenization rất đơn giản trong tiếng Anh, chỉ cần tách từ theo khoảng trống. Khi đó, dấu câu cũng được coi là một mã thông báo riêng biệt vì các dấu này cũng có ý nghĩa.
* Tokenization dựa trên khoảng trống không hiệu quả với tiếng Việt, vì trong tiếng Việt, nhiều từ ghép có thể không có dấu cách rõ ràng.

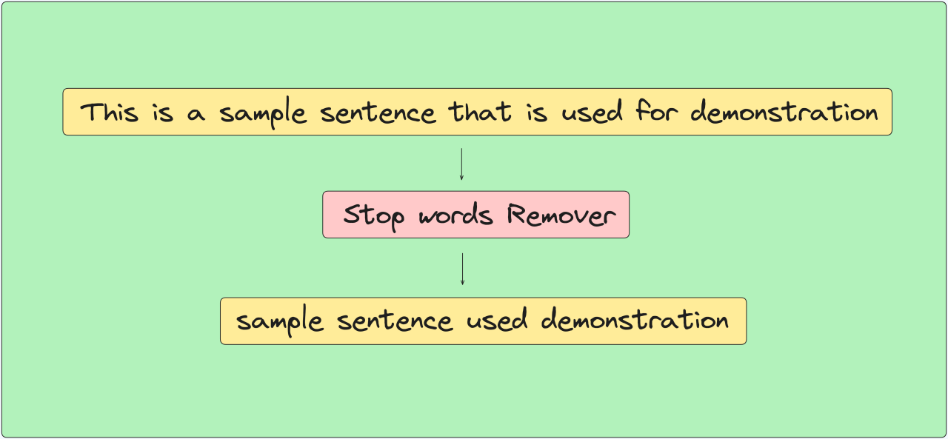
Ví dụ: “Tôi yêu lập trình” sẽ bị token hóa thành: [“Tôi”, “yêu”, “lập”, “trình”], nhưng “lập trình” là một từ ghép.

* Regular Expressions (re) là một công cụ mạnh mẽ để định nghĩa các mẫu phân tách từ.
* Rút gọn từ (stemming) và phục hồi từ gốc (lemmatization): chuyển các từ về dạng gốc



Hình . Minh họa rút gọn và phục hồi từ

* Rút gọn từ (Stemming): là quá trình rút ngắn các từ về dạng gốc của chúng bằng cách cắt bỏ hậu tố (suffix) hoặc tiền tố (prefix). Phương pháp này đơn giản và nhanh chóng nhưng có thể không luôn tạo ra từ gốc có nghĩa trong từ điển, vì vậy kết quả không phải lúc nào cũng chính xác.
* Phục hồi từ (Lemmatization): là quá trình chuyển các từ về dạng gốc chuẩn có nghĩa trong từ điển, dựa trên ngữ cảnh của từ đó trong câu. Phục hồi từ có sự hiểu biết ngữ nghĩa và ngữ pháp của từ, vì vậy kết quả của nó chính xác hơn so với rút gọn từ.
* Xử lý từ nối / từ dừng (stopwords): stopwords là những từ xuất hiện thường xuyên và không mang nhiều giá trị ngữ nghĩa. Những từ này sẽ làm nhiễu dẫn đến khó xác định được những từ có độ quan trọng.
* Thường thì sẽ sử dụng loại bỏ từ dừng (stopwords removal) để xử lý stopwords. Điều này giúp tiết kiệm tài nguyên tính toán, giảm độ phức tạp và tăng độ chính xác.



Hình . Minh họa cho stopwords removal

* Tuy nhiên, trong một số tình huống, stopwords có thể mang ý nghĩa qua trọng và không nên bị loại bỏ. Có thể dùng phương pháp đánh trọng số cho stopwords để xử lý nó.

Ví dụ: Với bài toán phân tích cảm xúc: “not good”, từ “not” rất quan trọng vì nó thay đổi cảm xúc cho câu từ tích cực thành tiêu cực.

* Chuyển thành chữ thường (lowercasing):Chuyển tất cả từ trong văn bản thành chữ thường để tránh phân biệt giữa chữ hoa với chữ thường có cùng nghĩa.

Ví dụ: “Cat” và “cat”

* Xử lý ký tự đặc biệt:Các ký tự có thể là dấu câu, ký tự đặc biệt, nháy đơn hoặc nháy kép. Các phương pháp xử lý ký tự đặc biệt:
* Có thể loại bỏ khi chúng không cung cấp giá trị thông tin quan trọng.  
  Ví dụ: “I love cat very much!!!” thành “I love cat very much”
* Có thể thay thế các ký tự đặc biệt bằng từ có nghĩa.  
  Ví dụ: “I love #cat” thành “I love hashtag cat”
* Giữ lại các dấu câu quan trọng  
  Ví dụ: “You are comming to the party”: được hiểu là câu khẳng định  
   “You are comming to the party?”: trở thành câu hỏi về việc có tham   
   gia tiệc không.
* Trích xuất đặc trưng (biểu diễn từ): là quá trình chuyển đổi dữ liệu văn bản thô thành các dạng số (vector), giúp mô hình học máy có thể hiểu và xử lý dữ liệu. Trong NLP, quá trình này giúp biến các từ và câu thành các đặc trưng có thể sử dụng trong các mô hình học máy hoặc học sâu. Quá trình này cực kỳ quan trọng trong việc biểu diễn văn bản dưới dạng số để máy tính có thể xử lý. Các phương pháp trích xuất đặc trưng phổ biến trong NLP:
* One-hot Encoding: Mỗi từ trong từ điển được biểu diễn bởi một vector nhị phân, có chiều dài bằng với số từ trong từ điển, và chỉ có một giá trị bằng 1 ở vị trí tương ứng với từ đó, trong khi tất cả các giá trị còn lại là 0.

Giả sử từ điển gồm ba từ: [‘apple’, ‘mango’, ‘kiwi’]:

* + ‘apple’ được mã hóa là: [1, 0, 0]
  + ‘mango’ được mã hóa là: [0, 1, 0]
  + ‘kiwi’ được mã hóa là: [0, 0, 1]
* Bag of Words (BoW): Phương pháp phổ biến để biểu diễn văn bản dưới dạng vector, trong đó mỗi từ trong văn bản được đại diện bằng một số tần suất xuất hiện của nó trong văn bản.

A screenshot of a document

AI-generated content may be incorrect.

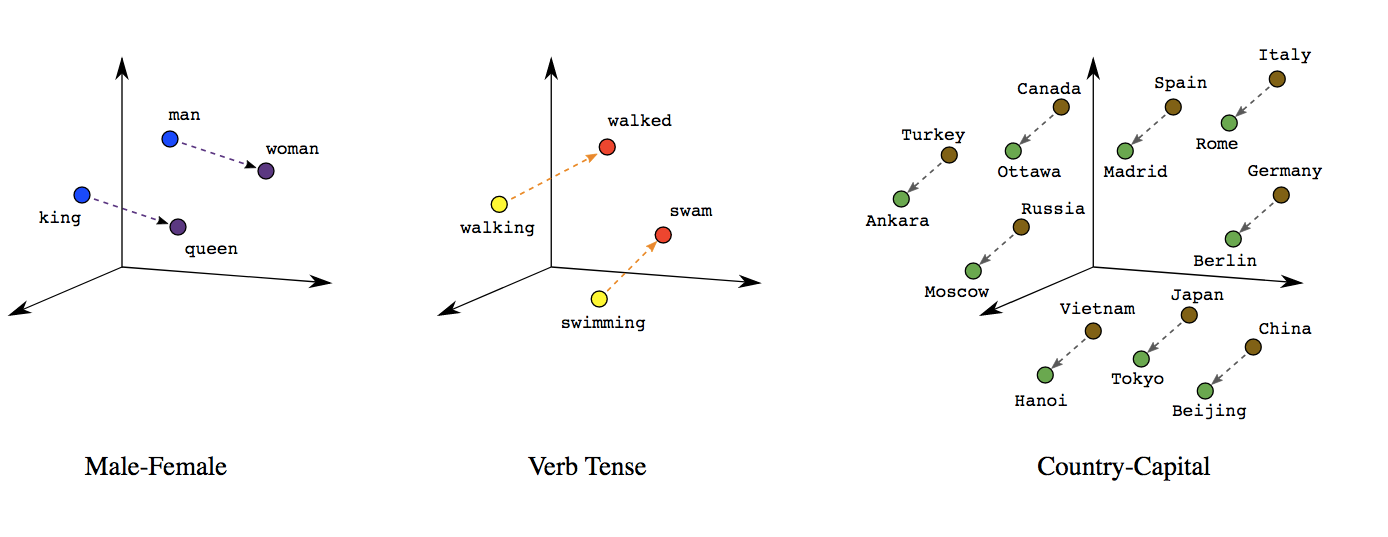
Hình . Minh họa cho BoW

* TF-IDF (Term Frequency - Inverse Document Frequency): là phương pháp tính toán mức độ quan trọng của từ trong văn bản dựa trên tần suất của từ (TF) và mức độ phổ biến của từ đó trong toàn bộ tài liệu (IDF). TF-IDF đánh giá mức độ quan trọng của một từ trong tài liệu, đồng thời cũng xem xét sự hiếm gặp của từ trong toàn bộ bộ dữ liệu. Công thức của TF-IDF:
  + Với TF(t): Tần suất của từ ‘t’ trong tài liệu
  + df(t): Số tài liệu chứa từ ‘t’
  + N: Tổng số tài liệu trong tập dữ liệu

Ví dụ: có 3 tài liệu sau: “I love cats”, “Cats are cute”, “My sister loves her cat”

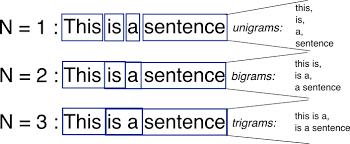
TF-IDF(cat) trong tài liệu 3 = ) = 0.0954

* Word Embeddings: là phương pháp chuyển ‘từ’ thành ‘vector số’ trong không gian đa chiều, trong đó các từ có nghĩa tương tự sẽ được đặt gần nhau trong không gian. Các phương pháp phổ biến để tạo embeddings là Word2Vec, GloVe và FastText.
  + Word2Vec: Sử dụng mạng nơ-ron để học vector từ bằng cách dự đoán từ xung quanh từ đã cho.
  + GloVe: Phương pháp matrix factorization, học các vector từ bằng cách tối ưu hóa các ma trận đồng xuất hiện của các từ trong một tập dữ liệu.
  + FastText: phát triển bởi Facebook AI Research, mở rộng Word2Vec bằng cách học vector cho cả subword (tiểu từ), như tiền tố, hậu tố, giúp xử lý tốt hơn các từ chưa gặp và từ phức tạp.



Hình . Minh họa cho word2vec

* N-grams: là phương pháp chia văn bản thành các cụm từ có độ dài cố định. Thay vì chỉ sử dụng các từ riêng lẻ (tokens), N-grams sử dụng các cụm từ để tìm hiểu mối quan hệ giữa các từ trong ngữ cảnh rộng hơn.
  + Unigrams: Mỗi từ riêng lẻ
  + Bigrams: Mỗi cặp từ liền kề
  + Trigrams: Mỗi ba từ liền kề



Hình . Minh họa cho N-grams

* Feature Hashing: phương pháp sử dụng hàm băm (hash function) để giảm số lượng đặc trưng trong dữ liệu. Thay vì lưu trữ các đặc trưng trong một không gian có kích thước lớn, ta sử dụng hàm băm để ánh xạ các từ vào không gian có kích thước cố định.

##### Phân tích cú pháp (Syntactic Analysis)

Phân tích cú pháp (Syntactic Analysis) là quá trình xác định cấu trúc ngữ pháp của một câu trong ngôn ngữ tự nhiên, giúp máy tính hiểu được mối quan hệ giữa các từ trong câu. Mục tiêu của phân tích cú pháp là xây dựng một cây cú pháp (parse tree) thể hiện cách các từ trong câu kết nối với nhau theo các quy tắc ngữ pháp của ngôn ngữ.

Phân tích cú pháp trong NLP giúp xác định cấu trúc ngữ pháp của câu và mối quan hệ giữa các thành phần trong câu. Nó bao gồm việc xác định chủ ngữ, động từ, tân ngữ, và các thành phần cú pháp khác của câu. Phân tích cú pháp có thể được thực hiện theo hai hướng chính: Dependency Parsing và Constituency Parsing.

Các kỹ thuật trong phân tích cú pháp:

* Gắn nhãn từ loại (POS Tagging): dùng để xác định từ loại của các từ trong một câu hoặc đoạn văn. Các từ loại này bao gồm danh từ, động từ, tính từ, phó từ, liên từ, v.v. Việc gán nhãn từ loại cho các từ trong văn bản giúp mô hình NLP hiểu được vai trò và chức năng ngữ pháp của chúng trong ngữ cảnh cụ thể. Các từ loại cơ bản trong POS Tagging:
* Noun (Danh từ)
* Verd (Động từ)
* Adjective (Tính từ)
* Adverb (Phó từ)
* Pronoun (Đại từ)
* Preposition (Giới từ)
* Conjunction (Liên từ)
* Interjection (Thán từ)

A blue rectangle with red and black text

AI-generated content may be incorrect.

Hình . Minh họa cho gán nhãn từ loại

* Phân tích phụ thuộc (Dependency Parsing): các từ được nối với nhau theo các mối quan hệ phụ thuộc. Cây phân tích phụ thuộc (dependency tree) sẽ thể hiện các mối quan hệ này. Mỗi từ trong câu có một cây con kết nối với từ chính (root) , và từ chính sẽ liên kết với các từ khác (đỉnh con) tùy thuộc vào vai trò của chúng trong câu.

A diagram of a diagram

AI-generated content may be incorrect.

Hình . Minh họa cây phụ thuộc

* Phân tích thành phần (Constituency Parcing): xác định các thành phần cú pháp trong câu dựa trên các quy tắc ngữ pháp (grammar rules) và xây dựng cấu trúc cây thể hiện mối quan hệ giữa các thành phần đó. Mỗi nhánh trong cây là một thành phần cú pháp, và từ gốc đến các lá của cây thể hiện mối quan hệ phụ thuộc giữa các thành phần.

A diagram of a company structure

AI-generated content may be incorrect.

Hình . Minh họa cây thành phần

##### Phân tích ngữ nghĩa (Semantic Analysis)

Phân tích ngữ nghĩa trong NLP là quá trình hiểu nghĩa của từ và câu trong văn bản. Nó không chỉ bao gồm việc hiểu các từ đơn lẻ mà còn phải hiểu ngữ cảnh mà chúng xuất hiện. Mục tiêu là giải nghĩa các từ trong văn bản sao cho mô hình có thể hiểu được ý nghĩa của văn bản và đưa ra kết quả chính xác. Các bài toán, kỹ thuật trong phân tích ngữ nghĩa:

* Phân biệt nghĩa của từ (Word Sense Disambiguation): Xác định nghĩa chính xác của từ khi từ đó có nhiều nghĩa trong ngữ cảnh. Một từ có thể có từ đồng âm, từ đồng nghĩa, trái nghĩa hoặc nhiều nghĩa.

Ví dụ: ‘bank’ có nghĩa là ngân hàng hoặc bờ sông tùy vào ngữ cảnh:  
“She go to the bank” và “The boat is floating near the bank of the river”

* Nhận diện thực thể (Named Entity Recognition – NER): Nhận diện thực thể có tên trong văn bản như người, địa điểm, tổ chức, sự kiện, ….Kỹ thuật này giúp máy tính phân loại các phần của văn bản thành các nhóm thực thể quan trọng.

Ví dụ: “VinGroup was founded in Hanoi, Vietnam by Pham Nhat Vuong.”

* VinGroup: Organization
* Hanoi, Vietnam: Location
* Pham Nhat Vuong: Person
* Gán nhãn vai trò ngữ nghĩa (Semantic Role Labeling – SRL): Xác định vai trò của ngữ nghĩa trong câu, giúp mô hình hiểu các vai trò của đối tượng.

Ví dụ: “ I usually eat Thuy Ta ice cream at Hoan Kiem lake.”

* I : Agent (người thực hiện hành động)
* Thuy Ta ice cream: Theme (vật thể bị tác động)
* Hoan Kiem lake: Location (nơi thực hiện hành động)
* Giải quyết tham chiếu (Coreference Resolution): xác định các tham chiếu trong văn bản, tức là các từ hoặc cụm từ liên kết đến cùng một đối tượng. Kỹ thuật này giúp làm rõ những tham chiếu mơ hồ trong văn bản, từ đó hiểu đúng ngữ nghĩa của câu.

Ví dụ: “This is my sister. She is 21 years old”: ‘she’ tham chiếu đến ‘my sister’

##### Xử lý diễn ngôn (Discourse Processing)

Xử lý diễn ngôn trong xử lý ngôn ngữ tự nhiên (NLP) là quá trình phân tích và xây dựng cấu trúc mạch lạc của văn bản, giúp văn bản trở nên dễ hiểu và có ý nghĩa rõ ràng. Quá trình này bao gồm một số yếu tố quan trọng:

* Tính mạch lạc (Coherence): Tính mạch lạc của văn bản được xác định bởi mối quan hệ giữa các câu và thực thể trong văn bản. Mỗi câu cần phải có sự liên kết ý nghĩa rõ ràng với các câu trước và sau để tạo thành một thể thống nhất. Mối quan hệ về tính mạch lạc trong diễn ngôn:
* Kết quả: Câu mô tả kết quả của câu trước.
* Giải thích: Câu giải thích nguyên nhân của câu trước.
* Song song: Hai câu có ý nghĩa tương tự.
* Mở rộng: Câu nói về cùng một sự kiện nhưng từ góc độ khác.
* Cơ hội: Mô tả sự thay đổi trạng thái hoặc hành động.

Ví dụ: S1: A đến ngân hàng để gửi tiền.

S2: Sau đó, anh ta đi tàu đến cửa hàng vải của B.

S3: Anh ấy muốn mua một ít quần áo.

S4: Anh ấy không có quần áo mới để dự tiệc.

S5: Anh ấy cũng muốn nói chuyện với B về sức khỏe của anh ấy.

A diagram of a diagram

AI-generated content may be incorrect.

Hình . Minh họa tính mạch lạc trong cấu trúc diễn ngôn phân cấp

* Cấu trúc diễn ngôn (Discourse Structure): Cấu trúc diễn ngôn là cách tổ chức các phần của văn bản, chia thành các đoạn văn có sự liên kết chặt chẽ về ý nghĩa. Việc phân đoạn diễn ngôn giúp cho các ứng dụng như tóm tắt văn bản hay trích xuất thông tin dễ dàng hơn.
* Phân đoạn diễn ngôn (Discourse Segmentation): Phân đoạn diễn ngôn có thể là không giám sát (dựa vào ngữ pháp và ngữ nghĩa của các câu) hoặc có giám sát (dựa vào các dữ liệu đã được gán nhãn hoặc từ ngữ chỉ thị). Mục tiêu là xác định các đơn vị có mối quan hệ ý nghĩa rõ ràng.
* Giải quyết tham chiếu (Reference Resolution): Quá trình này giúp xác định các thực thể mà các đại từ hoặc danh từ trong văn bản đề cập đến. Có hai loại tham chiếu:
* Đồng tham chiều (Coreference Resolution): Tìm các biểu thức tham chiếu liên quan đến cùng một thực thể.
* Tham chiếu anaphora (Anaphora Resolution): Xác định thực thể cho một đại từu duy nhất. Đại từ ‘he’, nhiệm vụ của tham chiếu Anaphora là tìm ra ‘A’.

##### Inception

Inception hay Natural Language Inception là quá trình bắt đầu hoặc hình thành cách thức hiểu và xử lý ngôn ngữ tự nhiên. Inception trong NLP tập trung vào việc nhận diện và phân tích các thành phần ngữ pháp của ngôn ngữ, bao gồm:

* Ngữ âm (Phonology): Nghiên cứu âm thanh của ngôn ngữ và cách phát âm các từ.
* Hình thái học (Morphology): Nghiên cứu cấu trúc từ vựng và cách hình thành các từ từ các đơn vị nhỏ hơn như gốc từ, tiền tố và hậu tố.
* Cú pháp (Syntax): Nghiên cứu cách các từ được sắp xếp trong câu và cấu trúc câu trong ngữ pháp.
* Ngữ nghĩa (Semantics): Nghiên cứu về ý nghĩa của từ và câu trong ngữ cảnh.
* Ngữ dụng học (Pragmatics): Nghiên cứu cách ngôn ngữ được sử dụng trong các tình huống giao tiếp cụ thể.

Inception cũng bao gồm các thể loại ngữ pháp như:

* Số (Number):
* Số ít (Singular): Chỉ một đối tượng.
* Số nhiều (Plural): Chỉ nhiều đối tượng. Ví dụ: dog/dogs, this/these.
* Giới tính (Gender):
* Phân biệt qua các đại từ và danh từ trong ngữ pháp. Ví dụ: he, she, it (giới tính nam, nữ, trung tính).
* Ngôi (Person):
* Ngôi thứ nhất: Người nói.
* Ngôi thứ hai: Người nghe.
* Ngôi thứ ba: Người hoặc vật được nhắc đến.
* Chức năng ngữ pháp (Case): Chức năng ngữ pháp của danh từ, đại từ hoặc cụm danh từ trong câu.
* Nominative case: Chủ ngữ (I, we, you, he, she...).
* Genitive case: Sở hữu (my, mine, his, her...).
* Objective case: Tân ngữ (me, us, him, her...).
* So sánh (Degree):
* Positive degree: So sánh đơn (big, fast).
* Comparative degree: So sánh hơn (bigger, faster).
* Superlative degree: So sánh nhất (biggest, fastest).
* Xác định và Không xác định (Definiteness and Indefiniteness):
* Xác định: Đối tượng rõ ràng (the).
* Không xác định: Đối tượng không xác định (a/an).
* Thì (Tense):
* Present tense: Thì hiện tại: Hành động xảy ra trong hiện tại (works).
* Past tense: Thì quá khứ: Hành động xảy ra trong quá khứ (rained).
* Future tense: Thì tương lai: Hành động xảy ra trong tương lai(will rain).
* Aspect: đánh giá quá trình của một hành động hoặc sự kiện trong một thời điểm nhất định.
* Perfective aspect: Hành động hoàn thành (I met my friend).
* Imperfective aspect: Hành động chưa hoàn thành (I am working).
* Thái độ (Mood): khái niệm dùng để chỉ thái độ của người nói đối với hành động hoặc sự kiện mà họ đang nói đến.
* Thể hiện thái độ của người nói, như khẳng định, nghi vấn, yêu cầu….
* Sự hòa hợp (Agreement):
* Person: Hòa hợp giữa chủ ngữ và động từ (I am, He is).
* Number: Hòa hợp giữa chủ ngữ và động từ (The boy sings, The boys sing).
* Gender: Hòa hợp giới tính (He reached his destination, The ship reached her destination).
* Case: Sự hòa hợp trong một số ngữ pháp (Who came first – he or his sister?).
* Văn nói và văn viết: Trong văn viết, các câu văn thường mạch lạc; văn nói thường gặp phải ngập ngừng, thay đổi câu hoặc sửa lại từ khi đang giao tiếp.

#### Các mô hình NLP nổi bật

* Encoder: Là mô hình tập trung vào việc mã hóa thông tin ngữ cảnh từ đầu vào và chuyển chúng thành các biểu diễn ngữ nghĩa (semantic representations). Encoder phù hợp với các tác vụ như phân loại văn bản, phân tích cảm xúc, trích xuất thông tin và nhận diện thực thể (NER). Các mô hình tiêu biểu của Encoder:
* BERT: Mô hình sử dụng cơ chế song song 2 chiều (bidirectional) để hiểu ngữ cảnh của từ dựa trên cả phía trước và phía sau.
* RoBERTa: Phiên bản cải tiến của BERT
* DistilBERT: Phiên bản nhẹ hơn và nhanh hơn của BERT
* ELECTRA: Sử dụng cách tiếp cận phát hiện lỗi trong từ thay vì phương pháp mặt nạ từ (masked token) như BERT
* Decoder: Là mô hình tập trung vào tạo và sinh ngôn ngữ từ biểu diễn ngữ nghĩa (semantic representations). Các mô hình Decoder thường được sử dụng trong các tác vụ tạo văn bản, dịch máy, tạo câu trả lời và sinh mã lập trình. Các mô hình tiêu biểu của Decoder:
* GPT (Generative Pre-trained Transformer): là mô hình tạo ngôn ngữ tự động sử dụng cơ chế hồi tiếp một chiều (autoregressive) để dự đoán từ tiếp theo dựa trên ngữ cảnh.
* LaMDA (Language Model for Dialogue Applications): là mô hình ngôn ngữ của Google, tập trung vào hội thoại thông minh và tự nhiên.
* BLOOM: là mô hình ngôn ngữ mã nguồn mở, hỗ trợ đa ngôn ngữ
* T5 (Text-to-Text Transfer Transformer): là mô hình đa năng do Google Research phát triển, có khả năng chuyển đổi bất kỳ bài toán NLP nào thành bài toán chuyển đổi văn bản. Mô hình này có khả năng xử lý đầu vào và đầu ra dưới dạng văn bản, làm cho nó linh hoạt hơn so với các mô hình chỉ xử lý riêng từng dạng đầu vào hoặc đầu ra.

#### Các thuật toán trong NLP

* Phân loại văn bản: bài toán phân loại văn bản vào một hoặc nhiều lớp dựa trên nội dung của văn bản. Ví dụ, phân loại email là spam hay không spam, phân loại cảm xúc (positive/negative), phân loại chủ đề (thể thao, chính trị, giải trí,…).
* Nhận diện thực thể có tên (Named Entity Recognition):  là bài toán xác định và phân loại các thực thể có tên trong văn bản. Những thực thể này có thể bao gồm tên người, địa điểm, tổ chức, thời gian, ngày tháng, và các thông tin đặc trưng khác.
* Dịch máy (Machine Translation): là quá trình tự động chuyển đổi một đoạn văn từ ngôn ngữ này sang ngôn ngữ khác. Ví dụ như từ tiếng Anh sang tiếng Việt hoặc ngược lại.
* Tóm tắt văn bản:  là quá trình rút gọn một đoạn văn bản dài thành một bản tóm tắt ngắn hơn, đồng thời vẫn đảm bảo giữ lại những thông tin chính yếu và bản chất của nội dung.
* Hỏi – Đáp (Question Answering – QA): là bài toán được yêu cầu trả lời các câu hỏi dựa trên một văn bản đầu vào.

#### Ứng dụng của NLP

NLP đang ngày càng phát triển và được ứng dụng rộng rãi trong nhiều lĩnh vực, giúp cải thiện các hệ thống giao tiếp giữa người và máy. Một số ứng dụng phổ biến của NLP:

* Xử lý văn bản:
* Tóm tắt tài liệu tự động: Hệ thống tự động tóm tắt các tài liệu dài, chẳng hạn như báo cáo tài chính, báo cáo nghiên cứu và hợp đồng pháp lý, giúp người dùng nhanh chóng nắm bắt các ý chính.
* Phân loại văn bản: Phân loại tài liệu theo các danh mục nhất định, chẳng hạn như phân loại tin tức thành các chủ đề: “kinh tế”, “giáo dục”,…
* Phân tích văn bản pháp lý: Trích xuất thông tin quan trọng từ các hợp đồng, hồ sơ và tài liệu pháp lý để hỗ trợ luật sư và chuyên gia pháp lý.
* Gợi ý nội dung: Thông qua việc phân tích hành vi người dùng như lịch sử tìm kiếm, nội dung đã xem và sở thích cá nhân, hệ thống có thể đưa ra các đề xuất phù hợp và chính xác hơn. Công nghệ này hiện được ứng dụng rộng rãi trên các nền tảng như YouTube, Netflix, Spotify và các trang thương mại điện tử, giúp tối ưu hóa trải nghiệm người dùng và tăng tỷ lệ tương tác.
* Phân tích ngữ nghĩa y khoa:
* Trích xuất thông tin y khoa: Tự động trích xuất thông tin từ hồ sơ bệnh án, chẳng hạn như tên thuốc, triệu chứng và chỉ định điều trị.
* Hỗ trợ chuẩn đoán: Phân tích và trích xuất dữ liệu từ báo cáo y tế và kết quả chụp X-quang, MRI để đưa ra các kết luận chẩn đoán chính xác hơn.
* Phân tích nghiên cứu y học: Nghiên cứu các bài báo y tế và tài liệu nghiên cứu để xác định các yếu tố liên quan đến sức khỏe cộng đồng và đề xuất các phương pháp điều trị.
* Chatbot và trợ lý ảo: đóng vai trò quan trọng trong việc tối ưu hóa khả năng tương tác giữa con người và máy móc, cho phép hệ thống hiểu và phản hồi các yêu cầu của người dùng một cách tự động và linh hoạt
* Trợ lý ảo (Siri, Google Assistant, Alexa): Sử dụng NLP để hiểu lệnh thoại và phản hồi theo thời gian thực.
* Chatbot dịch vụ khách hàng: Chatbot được tích hợp vào các website và ứng dụng di động, cung cấp dịch vụ hỗ trợ khách hàng 24/7. Các chatbot này có khả năng trả lời câu hỏi thường gặp (FAQs), giải quyết yêu cầu đơn giản và chuyển tiếp các trường hợp phức tạp cho nhân viên hỗ trợ.
* Dịch ngôn ngữ: giúp người dùng chuyển đổi nội dung từ ngôn ngữ này sang ngôn ngữ khác một cách nhanh chóng và tiện lợi. Việc dịch ngôn ngữ đóng vai trò quan trọng trong các lĩnh vực giao tiếp quốc tế, du lịch, giáo dục và kinh doanh toàn cầu, góp phần xóa bỏ rào cản ngôn ngữ và thúc đẩy hợp tác quốc tế. Google Translate là công cụ dịch phổ biến nhất thế giới.
* Phân tích cảm xúc: Công nghệ này có thể xác định cảm xúc của người dùng là tích cực, tiêu cực hoặc trung lập, từ đó cung cấp cho doanh nghiệp cái nhìn sâu sắc hơn về suy nghĩ và thái độ của khách hàng.
* Phân tích đánh giá sản phẩm: Nhiều nền tảng thương mại điện tử như Shopee, Lazada và Amazon sử dụng phân tích cảm xúc để đánh giá mức độ hài lòng của khách hàng đối với sản phẩm.
* Phân tích xu hướng thị trường: Phân tích các bài đăng, bình luận và tweet trên mạng xã hội giúp doanh nghiệp nắm bắt xu hướng thị trường và thái độ của công chúng đối với sản phẩm, dịch vụ hoặc sự kiện.

A screenshot of a phone

AI-generated content may be incorrect.

Hình . Phân tích cảm xúc có trong văn bản

* Tìm kiếm thông tin: là khả năng tìm và trích xuất thông tin phù hợp từ cơ sở dữ liệu hoặc trên web.
* Công cụ tìm kiếm (Google, Safari): Hệ thống xử lý ngôn ngữ cho phép các công cụ tìm kiếm hiểu được ý định tìm kiếm của người dùng, ngay cả khi người dùng nhập từ khóa sai hoặc câu hỏi mơ hồ.
* Tìm kiếm trong tài liệu nội bộ: Ứng dụng trong các công ty và tổ chức lớn, nơi cần tìm kiếm các tài liệu nội bộ và trích xuất thông tin nhanh chóng.
* Tìm kiếm dữ liệu phi cấu trúc: Tìm kiếm thông tin trong các tập dữ liệu không có cấu trúc như email, tài liệu và hồ sơ y tế.

#### Thách thức trong NLP

* Sự đa dạng và phức tạp của ngôn ngữ tự nhiên: Ngôn ngữ tự nhiên có cấu trúc phức tạp, bao gồm ngữ pháp, ngữ cảnh và ẩn ý mà con người thường sử dụng. Điều này gây khó khăn cho hệ thống trong việc phân tích các câu mơ hồ hoặc câu có ngữ cảnh không rõ ràng.
* Dữ liệu không cân bằng và thiếu dữ liệu huấn luyện
* Ngôn ngữ đa nghĩa và ngữ cảnh
* Vấn đề đạo đức và bảo mật: NLP thường được sử dụng trong trợ lý ảo và chatbot, nơi thông tin cá nhân của người dùng có thể bị thu thập. Điều này đặt ra thách thức về quyền riêng tư và bảo mật dữ liệu.

### Học máy (Machine Learning)

#### Khái niệm về học máy

Học máy (Machine Learning) là một lĩnh vực con trong trí tuệ nhân tạo (AI) nghiên cứu về các thuật toán và mô hình giúp máy tính có khả năng học hỏi từ dữ liệu và cải thiện hiệu suất của mình mà không cần phải lập trình cụ thể. Thay vì sử dụng các quy tắc được lập trình sẵn, học máy cho phép máy tính tự động học và phát triển các quy tắc từ các mẫu dữ liệu.

#### Quá trình học máy

* Thu thập dữ liệu: Các mô hình học máy cần một lượng dữ liệu lớn để học và cải thiện.
* Xử lý dữ liệu: Dữ liệu thu thập phải được xử lý và chuẩn bị để có thể sử dụng cho việc huấn luyện mô hình.
* Huấn luyện mô hình: Các thuật toán học máy sẽ tìm ra các mối quan hệ trong dữ liệu và sử dụng chúng để dự đoán hoặc phân loại thông tin mới.
* Đánh giá và thực nghiệm

#### Các mô hình học máy

* Học có giám sát: Dữ liệu huấn luyện đã được gắn nhãn. Mô hình học từ các cặp (input, output) để dự đoán giá trị đầu ra cho các dữ liệu mới. Các bài toán trong học có giám sát:
* Hồi quy (Regression): Dùng để dự đoán giá trị liên tục (ví dụ: dự đoán giá nhà, nhiệt độ, doanh thu).
* Phân loại (Classification): Dùng để phân loại các đối tượng vào các nhóm (ví dụ: phân loại email là spam hay không, phân loại bệnh nhân mắc bệnh).
* Mô hình tiêu biểu:
  + Hồi quy tuyến tính (Linear Regression)
  + Rừng quyết định (Random forest)
  + Mạng nơ-ron (Neural Networks)
* Học không giám sát:
* Clustering (Phân nhóm): Chia dữ liệu thành các nhóm sao cho các điểm dữ liệu trong mỗi nhóm có đặc điểm tương tự nhau.
* Giảm chiều (Dimensionality Reduction): Giảm số lượng đặc trưng trong dữ liệu trong khi vẫn giữ lại được thông tin quan trọng.
* Mô hình tiêu biểu: K-means, KNN
* Học tăng cường:
* Q-Learning: Một phương pháp học tăng cường không cần mô hình môi trường, học từ các hành động và phản hồi.
* Deep Q-Network (DQN): Mô hình sử dụng mạng nơ-ron sâu để học các chính sách từ dữ liệu.

### Học sâu (Deep Learning)

#### Khái niệm về học sâu

Học sâu là một phân nhánh của học máy, chủ yếu sử dụng các mạng nơ-ron nhiều lớp để học và mô phỏng các cấu trúc phức tạp từ dữ liệu lớn. Học sâu sử dụng các kiến trúc như Mạng nơ-ron tích chập (CNN), Mạng nơ-ron hồi tiếp (RNN) và Transformer để giải quyết các bài toán phức tạp mà các mô hình học máy truyền thống không thể thực hiện hiệu quả.

#### Các loại mô hình trong học sâu:

* Mạng nơ-ron tích chập (CNN): Dùng chủ yếu trong xử lý hình ảnh.
* Mạng nơ-ron hồi tiếp (RNN): Phù hợp cho các chuỗi dữ liệu (như văn bản, âm thanh).
* LSTM (Long Short-Term Memory): Biến thể của RNN, được thiết kể để giải quyết vấn đề vanishing gradient khi huấn luyện các mạng nơ-ron dài
* Transformer: Mô hình tiên tiến dùng trong NLP và các bài toán xử lý chuỗi.

### Các thư viện lập trình

#### Pandas

* Thư viện pandas trong python là một thư viện mã nguồn mở, hỗ trợ đắc lực trong thao tác dữ liệu. Đây cũng là bộ công cụ phân tích và xử lý dữ liệu mạnh mẽ của ngôn ngữ lập trình python. Thư viện này sử dụng một cấu trúc dữ liệu riêng là Dataframe. Pandas cung cấp rất nhiều chức năng xử lý và làm việc trên cấu trúc dữ liệu này.
* Cài đặt pandas:
* Sử dụng pip: pip install pandas
* Sử dụng conda: conda install pandas
* Khai báo thư viện: import pandas as pd
* Đọc file csv: dùng hàm read\_csv, dùng head(n) để in ra n bản ghi đầu tiên. Ngược lại của head là tail



* Có thể dùng hàm .info() để xem thông tin của dataframe và dùng hàm shape để xem kích thước

A close up of words

AI-generated content may be incorrect.

#### Matplotlib

Matplotlib là một thư viện vẽ đồ thị, trực quan hóa dữ liệu rất mạnh mẽ, hữu ích cho những người làm việc với Python và NumPy bởi hầu hết dữ liệu làm việc sẽ chỉ ở dạng mảng.

* Cài đặt pandas:
* Sử dụng pip: pip install matplotlib
* Sử dụng conda: conda install matplotlib
* Khai báo thư viện: import matplotlib.pyplot as plt
* Các biểu đồ, đồ thị thông dụng:
* Đồ thị đường: plot(x-axis values, y-axis values)
* Thanh: plt.bar(categorical variables, values, color)
* Tròn: plt.pie(value, categorical variables)
* Histogram: plt.hist(values, number of bins)
* Biểu đồ phân tán: plt.scatter(x-axis values, y-axis values)
* Một số lệnh cơ bản:
* plt.show() - hiển thị biểu đồ
* plt.title(“string”) - đặt tiêu đề
* plt.xlabel(“string”) - đặt nhãn cho trục x
* plt.ylabel(“string”) - đặt nhãn cho trục y

#### seaborn

Seaborn là mở rộng của matplotlib, có nghĩa là seaborn kế thừa từ matplotlib và cũng chính vì vậy, seaborn khong thể thay thế hoàn toàn được matplotlib.

* Cách cài đặt:
* Sử dụng pip: pip install seaborn
* Sử dụng conda: conda install seaborn
* Khai báo thư viện: import seaborn as sns

#### Numpy

Numpy (Numeric Python): là một thư viện toán học phổ biến và mạnh mẽ của Python. Cho phép làm việc hiệu quả với ma trận và mảng, đặc biệt là dữ liệu ma trận và mảng lớn với tốc độ xử lý nhanh.

* Cài đặt thư viện: pip install numpy
* Khai báo thư viện: import numpy as np

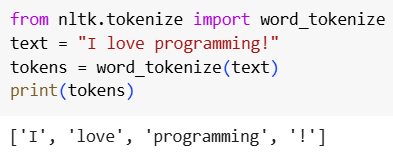
#### nltk

Thư viện NLTK - Natural Language Toolkit là một trong những thư viện open-source xử lí ngôn ngữ tự nhiên. Được viết bằng Python và với ưu điểm là dễ dàng sử dụng nên thư viện này ngày càng trở nên phổ biến và có được một cộng đồng lớn mạnh. Thư viện cung cấp hơn 50 kho dữ liệu văn bản khác nhau (corpora) và nhiều chức năng để xử lí dữ liệu văn bản để phục vụ cho nhiều mục đích khác nhau.

* Cài đặt thư viện: pip install nltk
* Khai báo thư viện: import nltk
* Trong thư viện nltk có rất nhiều tài nguyên (corpora, bộ từ điển, mô hình, v.v.), cần tải tài nguyên xuống để phục vụ cho các tác vụ xử lý ngôn ngữ tự nhiên. Các tài nguyên phổ biến và cần thiết trong nltk:
* ‘punkt’: Bộ công cụ phân tách câu và từ. Cần thiết để tách câu thành các từ (tokenization).
* ‘stopwords’: NLTK cung cấp một bộ các từ dừng (stopwords) trong nhiều ngôn ngữ. Stopwords của nltk hỗ trợ các ngôn ngữ sau:

1. English (Tiếng Anh)
2. Arabic (Tiếng Ả Rập)
3. French (Tiếng Pháp)
4. German (Tiếng Đức)
5. Spanish (Tiếng Tây Ban Nha)
6. Italian (Tiếng Ý)
7. Dutch (Tiếng Hà Lan)
8. Portuguese (Tiếng Bồ Đào Nha)
9. Russian (Tiếng Nga)
10. Chinese (Tiếng Trung)
11. Japanese (Tiếng Nhật)
12. Hindi (Tiếng Hindi)
13. Finnish (Tiếng Phần Lan)
14. Swedish (Tiếng Thụy Điển)
15. Greek (Tiếng Hy Lạp)
16. Turkish (Tiếng Thổ Nhĩ Kỳ)
17. Danish (Tiếng Đan Mạch)
18. Norwegian (Tiếng Na Uy)
19. Czech (Tiếng Séc)

* ‘wordnet’: Một từ điển từ đồng nghĩa; giúp tìm các từ đồng nghĩa, trái nghĩa và các mối quan hệ giữa các từ.
* ‘averaged\_perceptron\_tagger’: Dùng để gắn nhãn từ loại (POS tagging), giúp xác định loại từ (danh từ, động từ, tính từ, v.v.) trong câu.
* ‘words’: Một danh sách lớn các từ trong tiếng Anh. Thường được sử dụng trong các bài toán kiểm tra chính tả hoặc tìm kiếm từ trong văn bản.
* ‘maxent\_ne\_chunker’: Mô hình học máy sử dụng Maximum Entropy để nhận diện các thực thể có tên (Named Entity Recognition - NER), như tên người, tổ chức, địa điểm, …
* Để tải tài nguyên, dùng câu lệnh: nltk.download('tên\_tài\_nguyên'), hoặc dùng nltk.download() để tải tất cả tài nguyên từ nltk
* Các tính năng chính của thư viện:
* Tokenization (Tách từ): cần đến bộ tài nguyên ‘punkt’



* Stopwords (Từ dừng): cần đến bộ tài nguyên ‘stopwords’

A close-up of words

AI-generated content may be incorrect.

* Lemmatization (Rút gọn từ): cần đến bộ tài nguyên ‘wordnet’

A close-up of a computer screen

AI-generated content may be incorrect.

* Stemming (Cắt tỉa):

A close-up of a computer code

AI-generated content may be incorrect.

* POS Tagging (Gắn thẻ từ loại): cần đến bộ tài nguyên ‘averaged\_perceptron\_tagger\_eng’

A screenshot of a computer program

AI-generated content may be incorrect.

* N-grams: cần đến bộ tài nguyên ‘punkt’ để tách từ

A computer screen shot of text

AI-generated content may be incorrect.

* Named Entity Recognition (NER) - Nhận diện thực thể có tên: cần tên bộ tài nguyên ‘word’, ‘maxent\_ne\_chunker’ và ‘punkt’

A screenshot of a computer code

AI-generated content may be incorrect.

#### re

re là thư viện của Python cho phép làm việc với biểu thức chính quy (regular expressions), một công cụ mạnh mẽ dùng để tìm kiếm, phân tích, thay thế, và kiểm tra các chuỗi văn bản.

* Các hàm cơ bản trong re:
* re.search(pattern, string): Tìm kiếm mẫu trong chuỗi và trả về đối tượng khớp đầu tiên nếu có, hoặc None nếu không tìm thấy.

A close-up of a computer code

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

* re.sub(pattern, repl, string): Thay thế các chuỗi con khớp với mẫu bằng một giá trị khác trong chuỗi.

A screenshot of a computer

AI-generated content may be incorrect.

* re.findall(pattern, string): Tìm tất cả các chuỗi con trong chuỗi khớp với mẫu và trả về danh sách.

A white background with black text

AI-generated content may be incorrect.

* re.split(pattern, string): Phân tách chuỗi thành các phần nhỏ hơn dựa trên mẫu.

A white background with black text

AI-generated content may be incorrect.

* re.match(pattern, string): chỉ kiểm tra xem mẫu có khớp với phần đầu chuỗi hay không. Nếu mẫu không khớp ngay từ đầu chuỗi, nó sẽ không tìm kiếm phần còn lại của chuỗi.

#### Scikit-learning

Scikit-learning (còn được gọi là sklearn) là một thư viện phổ biến trong Python cho học máy. Nó cung cấp các công cụ đơn giản và hiệu quả để xây dựng các mô hình học máy cho các tác vụ như phân loại, hồi quy, phân cụm, và giảm chiều dữ liệu.

Một số tính năng cơ bản:

* Các thuật toán học máy:
* Phân loại (Classification): SVM, KNN, Naive Bayes.
* Hồi quy (Regression): Linear Regression, Decision Trees.
* Phân cụm (Clustering): K-means, DBSCAN.
* Tiền xử lý dữ liệu:
* StandardScaler: Chuẩn hóa dữ liệu.
* OneHotEncoder: Chuyển đổi dữ liệu phân loại thành dạng số.
* train\_test\_split: Chia dữ liệu
* Đánh giá mô hình:
* Accuracy, Precision, Recall, F1-score: Để đánh giá hiệu quả mô hình.

#### Tensorflow

TensorFlow là một thư viện mã nguồn mở mạnh mẽ cho học sâu (deep learning) và học máy (machine learning), được phát triển bởi Google. TensorFlow giúp xây dựng và triển khai các mô hình học máy, đặc biệt là các mô hình học sâu, như mạng nơ-ron nhân tạo, mạng tích chập (CNN), và mạng hồi tiếp (RNN).

Các tính năng chính của TensorFlow:

* Mạng nơ-ron nhân tạo (ANN): Tensorflow hỗ trợ xây dựng và huấn luyện các mô hình mạng nơ-ron sâu, bao gồm các lớp fully connected, convolutional layers và recurrent layers.
* Tính toán ma trận (Tensor): TensorFlow hoạt động chủ yếu với tensor, là các ma trận nhiều chiều (ndarray).
* GPU và TPU: TensorFlow hỗ trợ tính toán song song trên GPU và TPU (Tensor Processing Units), giúp tăng tốc độ huấn luyện mô hình trên các bộ xử lý mạnh mẽ.
* Keras: là một API cấp cao cho TensorFlow, giúp xây dựng và huấn luyện mô hình học sâu một cách dễ dàng và nhanh chóng.

Cách cài đặt tensorflow và khai báo thư viện:

pip install tensorflow

import tensorflow as tf

#### pickle

pickle là một thư viện có sẵn trong Python cho phép lưu trữ (serialize) và nạp (deserialize) các đối tượng Python, bao gồm các mô hình học máy. Các câu lệnh cơ bản với pickle:

* Khai báo thư viện: import pickle
* Lưu đối tượng: pickle.dump()
* Tải đối tượng: pickle.load()

### Chỉ số đánh giá hiệu suất mô hình

#### Confusion Matrix:

Là một ma trận dùng để đánh giá hiệu suất của các mô hình phân loại. Nó cho thấy số lượng dự đoán đúng và sai trong từng lớp cụ thể.

Cấu trúc của Confusion Matrix:

* True Positive (TP): Dự đoán đúng mẫu dương tính (mẫu thực sự là Positive và được dự đoán là Positive).
* True Negative (TN): Dự đoán đúng mẫu âm tính (mẫu thực sự là Negative và được dự đoán là Negative).
* False Positive (FP): Dự đoán sai mẫu dương tính (mẫu thực sự là Negative nhưng bị dự đoán là Positive).
* False Negative (FN): Dự đoán sai mẫu âm tính (mẫu thực sự là Positive nhưng bị dự đoán là Negative).

|  |  |  |
| --- | --- | --- |
| Actual\Predict | Positive | Negative |
| Positive | True Positive (TP) | False Negative (FN) |
| Negative | False Positive (FP) | True Negative (TN) |

Bảng . Ma trận nhầm lẫn với trường hợp phân loại nhị phân

#### Accuracy

* Accuracy là chỉ số đo lường tỷ lệ dự đoán chính xác của mô hình phân loại. Nó được tính bằng cách chia tổng số dự đoán chính xác cho tổng số dữ liệu.
* Công thức:

#### Precision:

* Precision là chỉ số đo lường tỷ lệ dự đoán tích cực mà mô hình đã dự đoán chính xác. Nó được tính bằng cách chia tổng số dự đoán tích cực đúng cho tổng số dự đoán tích cực.
* Công thức:

#### Recall

* Recall là chỉ số đo lường tỷ lệ số trường hợp tiêu cực mà mô hình phân loại đúng. Nó được tính bằng cách chia tổng số trường hợp tiêu cực đúng cho tổng số trường hợp tiêu cực thực tế
* Công thức:

#### F1-Score

* F1 - Score là một phép kết hợp của precision và recall, và nó đại diện cho trung bình điều hoà giữa hai số đo này. Đại diện cho sự cân bằng giữa precision và recall.
* Công thức:

#### AUC-ROC

* ROC (Receiver Operating Characteristic) → Là đồ thị biểu diễn mối quan hệ giữa TPR (True Positive Rate hay còn gọi là recall) và FPR (False Positive Rate) khi thay đổi ngưỡng (threshold) của mô hình.
* AUC (Area Under the Curve) → Là diện tích dưới đường ROC, đại diện cho khả năng phân loại của mô hình.
* Công thức:

### Framework Django

#### Khái niệm về framework Django

Django là một framework mã nguồn mở (open-source) và mạnh mẽ dành cho phát triển ứng dụng web sử dụng Python. Nó giúp các lập trình viên xây dựng các ứng dụng web một cách nhanh chóng và dễ dàng bằng cách cung cấp nhiều công cụ và thư viện tích hợp sẵn. Django tuân theo mô hình MVC (Model-View-Controller) hoặc MTV (Model-Template-View), giúp tách biệt các thành phần trong ứng dụng và dễ dàng duy trì mã nguồn.

#### Tính năng chính của Django

* Dễ dàng sử dụng và triển khai nhanh chóng
* Tuân theo mô hình MTV (Model-Template-View)
* Tích hợp ORM (Object-Relational Mapping): Django cung cấp một hệ thống ORM mạnh mẽ, cho phép lập trình viên thao tác với cơ sở dữ liệu mà không phải viết các câu lệnh SQL thủ công.
* Quản lý cơ sở dữ liệu tự động
* Bảo mật
* Phát hiện trang web động và RESTful APIs: Django hỗ trợ việc xây dựng các API RESTful một cách dễ dàng thông qua thư viện bổ trợ như Django REST framework (DRF).
* Django Admin: Django đi kèm với một hệ thống quản trị tích hợp (Django Admin), giúp các lập trình viên quản lý và thao tác với dữ liệu ứng dụng mà không cần phải xây dựng một giao diện quản trị từ đầu.
* Tính năng mở rộng: Django cho phép bạn mở rộng các tính năng của ứng dụng thông qua các app. Một ứng dụng Django có thể bao gồm một hoặc nhiều app riêng biệt, giúp việc quản lý và tái sử dụng mã trở nên dễ dàng hơn.
* Mã nguồn mở

## THỰC NGHIỆM

### Quy trình thực hiện

Đọc dữ liệu

Tiền xử dữ liệu

Chia dữ liệu

dữ liệu

Xây dựng mô hình

Đánh giá

Huấn luyện

### Đọc dữ liệu

* Bộ dữ liệu lấy từ Kaggle [1]
* Mô tả bộ dữ liệu: dữ liệu có 83448 dòng và 2 cột, trong đó text - là nội dung của email và label - nhãn của email với 1- email spam và 0 - là email hợp lệ.

A blue rectangular object with white text

AI-generated content may be incorrect.

Hình . Số lượng email spam (1) và email hợp lệ (0)

* Có thể nhận thấy rằng, qua tỷ lệ giữa email spam và email hợp lệ, email spam chiếm nhiều hơn email hợp lệ.

### Tiền xử lý dữ liệu

Trước khi tiến hành xây dựng mô hình, dữ liệu cần được làm sạch và chuyển đổi và dạng số để phù hợp với thuật toán học máy. Quá trình tiền xử lý bao gồm:

* Kiểm tra và xử lý các giá trị bị thiếu hoặc trùng lặp: sử dụng isnull() để kiểm tra trong cột text của dữ liệu, và không có giá trị nào bị lỗi missing value, sử dụng duplicate, phát hiện ra lỗi trùng lặp.

A black screen with white text

AI-generated content may be incorrect.

Hình . Lỗi trùng lặp trong bộ dữ liệu

Dữ liệu sau khi xử lý trùng lặp, dữ liệu có 83446 dòng và 2 cột.

* Tạo một hàm để xử lý dữ liệu văn bản:

A screenshot of a computer program

AI-generated content may be incorrect.

* Hàm sử dụng lower() để chuyển ký tự trong chuỗi thành chữ thường
* Hàm sử dụng re.sub() để loại bỏ ký tự đặc biệt
* Hàm dùng word\_tokenize() từ thư viện nltk để tách từ
* Loại bỏ stopwords với thư viện nltk
* Dùng WordNetLemmatizer() từ thư viện nltk để rút gọn từ
* Sử dùng hàm cho tập dữ liệu:



* Trích xuất đặc trưng:
* Bước 1: Xác định thông số cho bài:

A group of letters on a white background

AI-generated content may be incorrect.

* Bước 2: Khởi tạo một Tokenizer (bộ tách từ) để chuyển văn bản thành các chỉ số số học.



* Bước 3: Chuyển văn bản thành chuỗi số cho mô hình học sâu (LSTM)



* Bước 4: Chuyển văn bản thành vector TF-IDF cho mô hình học máy (Random forest)

A close up of words

AI-generated content may be incorrect.

* Bước 5: Lấy nhãn cho mô hình



### Chia dữ liệu

Để xây dựng và đánh giá hiệu quả các mô hình, việc chia dữ liệu thành 2 tập riêng biệt rất quan trọng. Sử dụng hàm train\_test\_split từ thư viện scikit-learn để ngẫu nhiên chia dữ liệu theo tỉ lệ:

* 80% dữ liệu cho tập huấn luyện (Training Set)
* 20% dữ liệu dành cho tập kiểm tra (Testing Test)

### Xây dựng và huấn luyện mô hình

Dùng các thuật toán học máy trên tập huấn luyện và sử dụng chúng để dự đoán trên tập kiểm tra. Hai thuật toán được sử dụng ở đây là Random Forest, LSTM.

#### Random Forest

Trong đề tài này, mô hình Random Forest được áp dụng để thực hiện bài toán phân loại, được xây dựng từ tập dữ liệu đã được xử lý làm sạch và đã được chuyển đổi thành dạng vector bằng phương pháp TF-IDF.

Ý tưởng chính:

* Mô hình sử dụng nhiều cây quyết định để tạo ra các quyết định phân loại khác nhau.
* Số lượng cây quyết định được xác định thông qua tham số

n\_estimators=100.

* Được huấn luyện bằng cách xây dựng nhiều cây quyết định độc lập.

A screen shot of a computer screen

AI-generated content may be incorrect.

#### LSTM

* Mô hình bao gồm các lớp embedding, LSTM, dropout và được huấn luyện bằng thuật toán adam và sử dụng hàm mất mát binary\_crossentropy.
* Sử dụng Sequential từ thư viện Keras để tạo mô hình kiểu xếp chồng từng lớp lên nhau.

A computer screen with text on it

AI-generated content may be incorrect.

* Với embedding\_dim = 128, tức là mỗi từ trong văn bản sẽ được chuyển đổi thành một vecto có 128 chiều. Và số lượng units (nút) trong lớp LSTM là lstm\_units = 64.
* Hàm kích hoạt sigmoid, cho ra output là một giá trị xác suất từ 0 đến 1
* Sử dụng batch\_size là 64 và huyến luyện trong 10 epoch, đồng thời đánh giá mô hình trên tập validation X\_test\_lstm, y\_test để theo dõi hiệu suất tránh overfitting. Sau mỗi epoch, sự thay đổi độ mất mát là:

A graph with a line

AI-generated content may be incorrect.

Hình . Đồ thị của hàm mất mát

* Mô hình học tốt ở giai đoạn đầu, ổn định ở giữa, nhưng bắt đầu có dấu hiệu overfitting nhẹ ở giai đoạn cuối.
* Điều này có thể do mô hình đã học quá tốt trên tập huấn luyện nhưng kém tổng quát hóa hơn trên tập kiểm tra. Để tránh overfitting, giảm epochs xuống ở mức 6-7.

A graph with a line and a red line

AI-generated content may be incorrect.

Hình . Đồ thị của hàm mất mát sau chỉnh sửa

### Đánh giá mô hình

#### Độ chính xác

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Mô hình | Accuracy | Precision | Recall | F1-score | AUC | Thời gian huấn luyện |
| Random Forest | 0.984182 | 0.984748 | 0.985197 | 0.984973 | 0.998017 | 0.8639 giây |
| LSTM | 0.984602 | 0.984761 | 0.985994 | 0.985377 | 0.997116 | 2.6525 giây |

Bảng . Đánh giá mô hình

* LSTM có hiệu suất tốt hơn Random Forest về mặt Precision, Recall, và F1-score, mặc dù sự khác biệt không lớn.
* Tuy nhiên, Random Forest lại có thời gian huấn luyện nhanh hơn rất nhiều so với LSTM, điều này có thể là một yếu tố quan trọng trong các ứng dụng yêu cầu thời gian huấn luyện ngắn.
* AUC của cả hai mô hình đều cao, với Random Forest một chút nhỉnh hơn.

#### Ma trận nhầm lẫn

|  |  |
| --- | --- |
| Mô hình | Ma trận nhầm lẫn |
| Random Forest | |  |  |  | | --- | --- | --- | |  | Dự đoán: 0 | Dự đoán: 1 | | Thực tế: 0 | 7774 | 134 | | Thực tế: 1 | 130 | 8652 | |
| LSTM | |  |  |  | | --- | --- | --- | |  | Dự đoán: 0 | Dự đoán: 1 | | Thực tế: 0 | 7725 | 183 | | Thực tế: 1 | 77 | 8705 | |

Bảng . Ma trận nhầm lẫn của mô hình

* Cả Random Forest và LSTM đều có độ chính xác cao, tuy nhiên LSTM có sai sót ít hơn về số lượng False Negative (FN) so với Random Forest. Cụ thể, LSTM có 183 lỗi dự đoán là lớp 1 nhưng thực tế là lớp 0, trong khi Random Forest có 134 lỗi nhưng lại có 130 lỗi False Positive (dự đoán lớp 0 nhưng thực tế là lớp 1).

## MÔ PHỎNG CHƯƠNG TRÌNH

### Tổng quan về trang web mô phỏng cho bài toán phân loại thư rác

Trang web mô phỏng cho bài toán phân loại thư rác nhằm giúp người dùng kiểm tra xem một email có phải là thư rác hay không bằng cách sử dụng các mô hình học máy. Đây là một ứng dụng đơn giản được xây dựng với django và sử dụng mô hình phân loại thư rác đã được huấn luyện ở phần trước.

#### Mục tiêu của trang web:

Sử dụng các mô hình học máy, trang web sẽ phân tích nội dung email và trả lại kết quả phân loại, cho biết liệu email đó có phải là spam hay không.

#### Mô hình dùng cho mô phỏng:

LSTM là sự lựa chọn hợp lý cho bài toán phân loại thư rác bởi so với Random Forest:

* LSTM hoạt động tốt hơn với văn bản dài
* LSTM có hiệu suất phân loại cao hơn với khả năng giảm thiểu các lỗi False Negative, quan trọng trong các bài toán phân loại. Nên dù có thời gian huấn luyện lâu hơn so với Random Forest, LSTM vẫn là lựa chọn hợp lý hơn.
* LSTM là mô hình học sâu với khả năng học và hiểu các mối quan hệ phức tạp giữa các từ trong văn bản.
* LSTM có thể kết hợp với các mô hình học sâu khác để nâng cao độ chính xác và giảm thời gian huấn luyện.

#### Quy trình hoạt động của trang web:

* Nhập dữ liệu: Người dùng nhập nội dung email vào ô nhập liệu và nhấn nút "Check".
* Xử lý dữ liệu: Sau khi người dùng nhấn "Check", dữ liệu văn bản sẽ được truyền vào mô hình học máy (đã huấn luyện trước) để phân loại.
* Kết quả phân loại: Mô hình sẽ phân tích văn bản và trả lại kết quả là "Spam" hoặc "Ham" (không phải spam).
* Hiển thị kết quả: Kết quả phân loại sẽ được hiển thị trực tiếp trên giao diện người dùng.

#### Các công nghệ sử dụng

* Django: Là framework web mạnh mẽ của Python, được sử dụng để xây dựng trang web, quản lý backend và xử lý các yêu cầu từ người dùng.
* HTML, CSS và Bootstrap: Sử dụng để tạo frontend cho trang web
* Các thư viện liên quan đến mô hình LSTM được huấn luyện trong chương trước.

### Xây dựng trang web với Django

#### Cài đặt Django

* Mở terminal hoặc command prompt và chạy lệnh để cài django



* Khởi tạo dự án Django

A computer screen with text

AI-generated content may be incorrect.

Khi tạo xong, tạo cửa sổ thư mục sẽ hiện như sau:

A screenshot of a computer program

AI-generated content may be incorrect.

* Tạo ứng dụng (App) trong Django:



* Đăng ký ứng dụng trong settings.py (thuộc NLP\_project): Tại file settings.py thuộc thư mục NLP\_project, thêm ứng dụng ‘home’ vào INSTALLED\_APPS.

A screen shot of a computer program

AI-generated content may be incorrect.

* Cấu hình URL: Mở tệp urls.py trong thư mục NLP\_project để thực hiện cấu hình url cho app
* Cần khai báo app trong file urls.py



* Thêm path của app



* Khởi động server phát triển của Django: Chạy lệnh này trong terminal để xem kết quả của ứng dụng.



#### Cài đặt thư viện liên quan

Cần cài đặt các thư viện sau:

* pandas 2.2.3
* numpy 2.1.3
* tensor (lưu ý, tensor hiện tại chỉ hỗ trợ đến phiên bản Python 3.11) 0.3.6
* nltk 3.9.1
* tensorflow 2.19.0

Hoặc có thể cài file requirements.txt (file chứa tất cả các gói dùng để chạy ứng dụng phân loại thư rác)

pip install -r requirements.txt

#### Tạo giao diện cho trang web

* Tạo thư mục ‘templates’ trong thư mục ‘home’. Trong thư mục này tạo tệp ‘home.html’. Tệp ‘home.html’ là giao diện của web.
* Trong ‘home.html’ cần có form để người dùng nhập văn bản và button ‘Check’ để gửi dữ liệu. Khi người dùng nhấn vào nút ‘Check’, dữ liệu sẽ được gửi tới url /submit/.

A computer screen shot of a program code

AI-generated content may be incorrect.

* Tại urls.py, cần thêm câu lệnh ánh xạ đến một hàm trong view (đây là hàm xử lý và dự đoán nội dung email trong view)



* Trong ‘home.html’ cũng cần một ô trả về kết quả xử lý.

A black background with white text

AI-generated content may be incorrect.

#### Tạo view cho trang web

view là nơi xử lý các logic của ứng dụng, nhận yêu cầu HTTP từ người dùng và trả về phản hồi (response). View có thể trả về các trang HTML, dữ liệu JSON, hoặc bất kỳ phản hồi nào khác tùy theo yêu cầu.

* Tạo view trả về trang chính (home)

A computer screen with text

AI-generated content may be incorrect.

* Tạo 1 file .py xử lý dữ liệu văn bản. Tệp này có chứa mô hình, hàm chuẩn hóa dữ liệu đầu vào rồi đưa vào mô hình.
* Khai báo các thư viện, hàm cần thiết

A screen shot of a computer program

AI-generated content may be incorrect.

* Khởi tạo mô hình

A screen shot of a computer program

AI-generated content may be incorrect.

* Tạo hàm tiền xử lý văn bản

A computer screen shot of text

AI-generated content may be incorrect.

* Tạo hàm dự đoán kết quả đầu ra

A computer screen shot of text

AI-generated content may be incorrect.

* Tạo view trả về kết quả xử lý của mô hình
* Cần khai báo file xử lý, dự đoán kết quả của văn bản



* Tạo hàm nhận vào và xử lý văn bản người dùng nhập vào (nội dung của email), dự đoán nó có phải là thư rác hay không (hàm predict\_email) và sau đó trả kết quả phân loại về trang ‘home.html’

A computer screen shot of text

AI-generated content may be incorrect.

#### Demo kết quả

* Giao diện ban đầu của web:

A screen shot of a phone

AI-generated content may be incorrect.

Hình . Giao diện của web

* Thử với nội dung của email spam

A screenshot of a computer

AI-generated content may be incorrect.

Hình . Giao diện của web khi thử với nội dung spam

* Thử với nội dung của email không phải spam

A screenshot of a computer

AI-generated content may be incorrect.

Hình . Giao diện của web khi thử với nội dung không spam

### Deploy với docker

Docker là một công cụ mạnh mẽ giúp đóng gói ứng dụng và tất cả các phụ thuộc của nó vào trong một container. Đây là một cách tuyệt vời để triển khai ứng dụng Django với môi trường độc lập.

* Bên trong project ‘NLP\_project’ tạo một file có tên là ‘Dockerfile’. Lưu ý: file này không có phần mở rộng
* Dockerfile là một file văn bản chứa các chỉ dẫn để xây dựng image.

A screenshot of a computer program

AI-generated content may be incorrect.

* Sau khi viết Dockerfile, dùng lệnh ‘docker build -t <tên> .’ để xây dựng Docker image.



* Tiếp đến là chạy container từu Docker image. Có 2 cách để chạy:
* Cách 1: Chạy với câu lệnh trong powershell hoặc terminal



* Cách 2: Vào docker image của docker desktop rồi run

A screenshot of a computer

AI-generated content may be incorrect.

Sau khi run sẽ hiện ra bảng sau:

A screenshot of a computer

AI-generated content may be incorrect.

Điền tên của container, cổng (ports) rồi nhấn run

* Sau khi chạy xong container, có thể truy cập ứng dụng qua địa chỉ: <http://localhost:8000> trên trình duyệt.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

* 1. **ƯU ĐIỂM**
* Các mô hình phân loại trên đã đạt được hiệu quả cao cho thấy khả năng phân loại tốt.
* Với độ chính xác cao
* Khả năng tổng hóa tốt
* Xử lý tốt chuỗi văn bản: các mô hình đặc biệt là LSTM, có khả năng xử lý và phân tích hiệu quả nội dung văn bản phức tạp của email.
* Tính linh hoạt và mở rộng
* Hiệu suất cao
* Mô phỏng ứng dụng dễ hiểu, dễ sử dụng
  1. **NHỮNG TỒN TẠI CỦA ĐỀ TÀI**

Mặc dù mô hình phân loại email đã đạt kết quả khả quan, nhưng vẫn còn tồn tại một số hạn chế và khắc phục sau:

* Thời gian huấn luyện dài
* Gradient dễ bị suy biến hoặc bùng nổ
* Dễ bị overfitting

Về web:

* Giao diện còn thô sơ, đơn giản.
* Khi nhấn ‘Check’ thì bị load lại trang và mất nội dung đã nhập
* Mất thời gian khá lâu với lần đầu kiểm tra khi mở web
* Hiện tại chỉ hỗ trợ với ngôn ngữ là tiếng anh.
  1. **HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI**
* Bài toán phân loại email spam sẽ được tối ưu hóa hơn để tăng cường khả năng phát hiện và lọc các email độc hại. Đồng thời, hướng phát triển trong tương lai sẽ tập trung vào việc tối ưu hóa và nâng cao khả năng mô hình phân loại với những định hướng sau:
* Kết hợp LSTM với CNN để trích xuất đặc trưng tốt hơn từ dữ liệu văn bản.
* Tối ưu hóa mô hình xử lý ngôn ngữ tự nhiên (NLP) như sử dụng các mô hình ngôn ngữ tiền huấn luyện (BERT, RoBERTa) để cải thiện khả năng hiểu ngữ cảnh và chủ ngữ của email.
* Với Random Forest, kết hợp với phương pháp Boosting (như XGBoost, LightGBM) để cải thiện khả năng phân loại. Tăng cường hiệu suất bằng ccahs sử dụng phương pháp tối ưu siêu tham số Grid Search
* Mở rộng bộ dữ liệu (thêm các ngôn ngữ khác) để có thể mở rộng nhận diện email spam.
* Với Web: tích hợp các tính năng mới, mở rộng khả năng của mô hình và cải thiện trải nghiệm người dùng.
* Sử dụng các thành phần từu Bootstrap để cải thiện giao diện như thêm cards, modals
* Sử dụng
* Tích hợp với Gmail và các dịch vụ email khác như outlook
* Phát triển thành một hệ thống quản lý thư rác (có phần sign up và log in)

# TÀI LIỆU THAM KHẢO

[1] Bộ dữ liệu Spambase trên Kaggle [Spam Email Classification Dataset](https://www.kaggle.com/datasets/purusinghvi/email-spam-classification-dataset?select=combined_data.csv)

[2] Võ Thu Trang - Xử lý ngôn ngữ tự nhiên (NLP) là gì? Tìm hiểu về lập trình ngôn ngữ tư duy - [https://vbee.vn/blog/ai/xu-ly-ngon-ngu-tu-nhien-nlp](https://vbee.vn/blog/ai/xu-ly-ngon-ngu-tu-nhien-nlp/?utm_source=BL&utm_medium=CPC&utm_term=Vietnam&utm_campaign=SM_GG_DU_NBR_AS_A_VN_B_UNK_UNK&gad_source=1&gbraid=0AAAAA-NgIlHC8rBLgoCUI8CrhMxgv44WN&gclid=CjwKCAjw5PK_BhBBEiwAL7GTPZqidF5wwkkQrdVNae-B3TGCehXBrLLwEVw3HjQrSj0J5KAZ3BAyVhoCmzMQAvD_BwE)

[3] aws - Xử lý ngôn ngữ tự nhiên (NLP) là gì? - <https://aws.amazon.com/vi/what-is/nlp/>

[4] viettelidc - [Cẩm nang AI] Xử lý ngôn ngữ tự nhiên (Natural Language Processing) trong AI là gì? - <https://viettelidc.com.vn/tin-tuc/cam-nang-ai-xu-ly-ngon-ngu-tu-nhien-natural-language-processing-trong-ai-la-gi>

[5] tutorialspoint – Natural Language Processing tutorial - <https://www.tutorialspoint.com/natural_language_processing/index.htm>

[6] GreateLearning - Introduction to NLP - <https://www.mygreatlearning.com/natural-language-processing/tutorials>

[7] trituenhantao.io – Sự thú vị của NLP - <https://trituenhantao.io/kien-thuc/su-thu-vi-cua-nlp-phan-2/>

[8] Nguyễn Văn Hiếu – Pandas Python Tutorial - <https://viblo.asia/p/pandas-python-tutorial-XL6lAxaDZek>

[9] Nguyen Van Hoang - Giới thiệu về Matplotlib (một thư viện rất hữu ích của Python dùng để vẽ đồ thị) - <https://viblo.asia/p/gioi-thieu-ve-matplotlib-mot-thu-vien-rat-huu-ich-cua-python-dung-de-ve-do-thi-yMnKMN6gZ7P>

[10] Cao Minh Hiếu - Data visualization với Seaborn - <https://viblo.asia/p/data-visualization-voi-seaborn-oOVlYP9vZ8W>

[11] CodeLearn – Tìm Hiểu Thư Viện NumPy Trong Python - <https://codelearn.io/sharing/tim-hieu-thu-vien-numpy-trong-python>

[12] Phan Văn Tấn - Tìm hiểu về Django - framework hỗ trợ Python trong lập trình web - <https://viblo.asia/p/tim-hieu-ve-django-framework-ho-tro-python-trong-lap-trinh-web-QpmlexbkZrd>

[13] Nguyễn Văn Hiến - 3 giai đoạn xử lý văn bản đầu vào trong NLP: tokenization, stop word và stemming - <https://tummosoftware.com/post/3-giai-doan-tokenization-stop-word-stemming/>