

**TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT**  
**VIỆN KỸ THUẬT CÔNG NGHỆ**



**TIỂU LUẬN MÔN HỌC**  
**KỸ THUẬT LẬP TRÌNH TRONG PHÂN TÍCH**  
**DỮ LIỆU**

**PHÂN TÍCH DỮ LIỆU CỦA TRANG**  
**VIETNAMNET**

**GV: ThS. Hồ Ngọc Trung Kiên**

**SVTH: Trần Quay Tín MSSV: 2024802010221**

**Nguyễn Hoàng Hiệp MSSV: 2024802010235**

**BÌNH DƯƠNG - 04/2023**

**TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT**  
**VIỆN KỸ THUẬT CÔNG NGHỆ**



**TIỂU LUẬN MÔN HỌC**  
**KỸ THUẬT LẬP TRÌNH TRONG PHÂN TÍCH**  
**DỮ LIỆU**

**PHÂN TÍCH DỮ LIỆU CỦA TRANG**  
**VIETNAMNET**

**GV: ThS. Hồ Ngọc Trung Kiên**

**SVTH: Trần Quay Tín MSSV: 2024802010221**

**Nguyễn Hoàng Hiệp MSSV: 2024802010235**

**BÌNH DƯƠNG - 04/2023**

# MỤC LỤC

<b>MỤC LỤC</b>	<b>ii</b>
<b>DANH MỤC HÌNH</b>	<b>iv</b>
<b>LỜI CAM ĐOAN</b>	<b>1</b>
<b>LỜI MỞ ĐẦU</b>	<b>2</b>
<b>CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI</b>	<b>3</b>
1.1. Lý do chọn đề tài	3
1.2. Mục tiêu nghiên cứu	3
1.3. Đối tượng nghiên cứu	3
1.4. Phạm vi nghiên cứu	3
<b>CHƯƠNG 2. CƠ SỞ LÝ THUYẾT</b>	<b>5</b>
2.1. Giới thiệu ngôn ngữ lập trình Python	5
2.2. Giới thiệu công cụ Google Colab	5
2.3. Giới thiệu các thư viện của python	6
2.3.1. Thư viện Numpy	6
2.3.2. Thư viện pandas	6
2.3.3. Thư viện Requests	7
2.3.4. Thư viện BeautifulSoup	7
2.3.5. Thư viện regular expression	7
2.3.6. Thư viện Regex	8
2.3.7. Thư viện Underthesea	8
2.3.8. TfidfVectorizer class	9
2.3.9. Thư viện NLTK	9
2.3.10. Thư viện Matplotlib	9
2.3.11. Giải thuật Cosine Similatory	10
<b>CHƯƠNG 3. MÔ HÌNH BÀI TOÁN</b>	<b>11</b>
3.1. Mô hình bài toán	11
3.2. Giải thích các bước	11
<b>CHƯƠNG 4. THỰC NGHIỆM</b>	<b>13</b>
4.1. Các bước trong mô hình	13
4.1.1. Code thêm thư viện	13
4.1.2. Code lấy thông tin trang web	14

4.1.3. Code load dữ liệu và xóa số.....	16
4.1.4. Code xóa tab html.....	18
4.1.5. Code chuyển văn bản thành chữ thường.....	19
4.1.6. Code chuẩn hóa bảng mã .....	20
4.1.7. Code chuẩn hóa kiểu gõ .....	22
4.1.8. Code xóa stopwords.....	26
4.1.9. Code chuyển sang dạng ngữ nghĩa .....	28
4.1.10. Code chuyển thành vector.....	29
4.1.11. Code tìm bài báo giống cụm từ.....	31
<b>KẾT LUẬN .....</b>	<b>38</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>39</b>

## DANH MỤC HÌNH

<b>Hình 2.1:</b> Công cụ Google Colab .....	6
<b>Hình 3.1:</b> Mô hình bài toán .....	11
<b>Hình 4.1:</b> Kết quả lấy thông tin từ website .....	15
<b>Hình 4.2:</b> Kết quả load dữ liệu từ excel .....	17
<b>Hình 4.3:</b> Kết quả đoạn mã xóa số.....	17
<b>Hình 4.4:</b> Kết quả đoạn mã xóa tab html .....	18
<b>Hình 4.5:</b> Kết quả đoạn mã chuyển văn bản thành chữ thường .....	19
<b>Hình 4.6:</b> Kết quả đoạn mã chuẩn hóa bảng mã.....	21
<b>Hình 4.7:</b> Kết quả đoạn mã chuẩn hóa kiểu gõ .....	26
<b>Hình 4.8:</b> Kết quả đoạn mã xóa stopword.....	27
<b>Hình 4.9:</b> Kết quả đoạn mã chuyển sang dạng ngữ nghĩa .....	29
<b>Hình 4.10:</b> Kết quả đoạn mã chuyển sang vector.....	30
<b>Hình 4.11:</b> Kết quả của đoạn mã tìm bài báo giống cụm từ .....	33
<b>Hình 4.12:</b> Kết quả của đoạn mã tìm bài báo giống cụm từ .....	36

## **LỜI CAM ĐOAN**

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi và được sự hướng dẫn khoa học của ThS. Hồ Ngọc Trung Kiên. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây.

Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong báo cáo còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung báo cáo của mình. Trường Đại học Thủ Dầu Một không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

Bình Dương, ngày tháng năm 2023

Người thực hiện

(ký tên và ghi rõ họ tên)

## **LỜI MỞ ĐẦU**

Trong thời đại của công nghệ thông tin hiện nay, các website tin tức trực tuyến trở thành một trong những nguồn thông tin được sử dụng phổ biến nhất. Tuy nhiên, với lượng thông tin khổng lồ và không ngừng tăng lên từ các trang web này, việc phân tích dữ liệu trở thành một thách thức lớn đối với các nhà nghiên cứu, chuyên gia và những người quản lý website. Phân tích dữ liệu từ các trang tin tức có thể giúp chúng ta hiểu rõ hơn về các xu hướng, sự kiện và những vấn đề nổi bật trong xã hội. Chính vì vậy, việc nghiên cứu và phân tích dữ liệu của website tin tức trở thành một lĩnh vực rất quan trọng và hữu ích. Đồng thời đây cũng là thử thách, cơ hội cho chúng em được thực hiện, traie nghiệm việc phân tích để củng cố lại các kiến thức đã học ở môn kỹ thuật lập trình trong phân tích dữ liệu. Việc thực hiện phân tích dữ liệu từ website sẽ giúp chúng em hiểu sâu và tường tận hơn về các nội dung của bài học. Chúng em lựa chọn website Vietnamnet.vn để tiến hành phân tích dữ liệu vì đây là trang báo tin tức được cập nhật liên tục, đa dạng về chủ đề, phong phú về nội dung và trang web cho phép crawl tin tức với số lượng lớn.

## CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI

### 1.1. Lý do chọn đề tài

Website vietnamnet là một trong những trang web lớn và phổ biến nhất tại Việt Nam, với nhiều thông tin đa dạng, phong phú về nhiều lĩnh vực khác nhau như: chính trị, xã hội, văn hoá, thể thao, công nghệ, giải trí,... và các thông tin được cập nhật liên tục. Chính vì thế, đây là một trang web cung cấp cho chúng em một nguồn dữ liệu vô cùng lớn và đa dạng để có thể tiến hành phân tích dữ liệu.

Phân tích dữ liệu của trang có thể giúp chúng em hiểu rõ hơn về các xu hướng, thị trường, tin tức và các vấn đề xã hội đang được quan tâm tại Việt Nam. Đây đồng thời cũng là cơ hội cho chúng em được thực hành, trải nghiệm thực tế về việc phân tích dữ liệu, giúp củng cố những kiến thức mà chúng em đã học trong thời gian qua.

### 1.2. Mục tiêu nghiên cứu

Sử dụng các công cụ và thư viện của Python để tiến hành thu thập các dữ liệu từ website như: Tiêu đề, thể loại, mô tả, nội dung.

Xử lý phân tích dữ liệu vừa thu thập được:

- Xoá tab HTML và xoá số.
- Chuyển văn bản thành chữ thường.
- Chuẩn hoá văn bản (Cách gõ dấu trong tiếng Việt).
- Xoá Stopword.
- Chuyển dữ liệu sang dạng ngữ nghĩa.
- Chuyển dữ liệu sang dạng Vector.
- Tìm kiếm và so khớp văn bản.

### 1.3. Đối tượng nghiên cứu

Nghiên cứu có thể tập trung vào các chủ đề phổ biến nhất trên website, các từ khóa được sử dụng nhiều, các bài viết, các bài đăng (ví dụ: tin mới nhất, thể thao, kinh doanh, đời sống...)

### 1.4. Phạm vi nghiên cứu

Phân tích dữ liệu được thu thập từ trang web Vietnamnet.

Xử lý dữ liệu để chuẩn hóa và làm sạch dữ liệu, loại bỏ các kí tự đặc biệt, loại bỏ các từ không có nghĩa trong tiếng việt.



Phân tích dữ liệu trên 1000 tin tức được lấy từ trang web.

Việc phân tích dữ liệu được thực thi bằng cách: thu thập dữ liệu, xử lý dữ liệu và so khớp dữ liệu.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

### 2.1. Giới thiệu ngôn ngữ lập trình Python

Python là một ngôn ngữ lập trình thông dịch, mã nguồn mở và đa nền tảng được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau. Python có cú pháp đơn giản, dễ đọc và dễ hiểu, điều này giúp người lập trình có thể tập trung vào các vấn đề cốt lõi hơn là các chi tiết kỹ thuật.

Python có thư viện phong phú và mạnh mẽ, bao gồm các thư viện đối tượng, quản lý cấu trúc dữ liệu, đồ họa, truyền thông và nhiều hơn nữa. Ngoài ra, Python còn hỗ trợ rất nhiều framework phổ biến như Django, Flask, Pyramid, cho phép bạn dễ dàng xây dựng các ứng dụng web và API.

Python cũng được sử dụng trong lĩnh vực khoa học dữ liệu và trí tuệ nhân tạo. Các thư viện như Numpy, Pandas, Scikit-learn, TensorFlow và PyTorch giúp người lập trình có thể dễ dàng xử lý, phân tích và khai thác dữ liệu, cũng như xây dựng các mô hình học máy và mạng nơ-ron.

Ngoài ra, Python còn là ngôn ngữ được sử dụng trong nhiều lĩnh vực khác như game, đồ họa, IoT, robot, hacking, v.v.

Với các đặc tính trên, Python đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới và được sử dụng bởi nhiều nhà phát triển, công ty và tổ chức trên toàn cầu.

### 2.2. Giới thiệu công cụ Google Colab.

Google Colaboratory (hay Google Colab) là một nền tảng đám mây miễn phí được cung cấp bởi Google cho phép người dùng tạo và chạy các tệp notebook Jupyter để phát triển và thực thi mã Python. Nó cho phép người dùng sử dụng các tài nguyên tính toán như CPU, GPU và bộ nhớ trong đám mây của Google mà không cần phải cài đặt và cấu hình môi trường phát triển trên máy tính của mình. Nó cung cấp cho người dùng một nền tảng thuận tiện và dễ dàng sử dụng để phát triển và chia sẻ các dự án Python của mình.



**Hình 2.1:** Công cụ Google Colab

## **2.3. Giới thiệu các thư viện của python**

### **2.3.1. Thư viện Numpy**

NumPy là một thư viện Python mã nguồn mở được sử dụng để làm việc với mảng nhiều chiều và ma trận. Nó cung cấp các hàm tính toán khoa học, bao gồm các phép tính toán trên ma trận, đại số tuyến tính, thống kê và phép biến đổi Fourier. NumPy được sử dụng rộng rãi trong các ứng dụng khoa học dữ liệu và tính toán số như phân tích dữ liệu, xử lý ảnh và âm thanh, và học máy.

Các ưu điểm của NumPy bao gồm hiệu suất tính toán cao, hỗ trợ nhiều phép tính toán trên ma trận và các hàm tính toán khoa học, và tính tương thích với các thư viện và công cụ phát triển khác như SciPy, Pandas và Matplotlib. NumPy là một trong những thư viện quan trọng nhất trong cộng đồng khoa học dữ liệu và tính toán số của Python.

### **2.3.2. Thư viện pandas**

Pandas là một thư viện phổ biến cho ngôn ngữ lập trình Python, được sử dụng rộng rãi trong việc xử lý và phân tích dữ liệu. Pandas cung cấp các công cụ để làm việc với các tập dữ liệu có cấu trúc, bao gồm các hàm để đọc và ghi các tệp dữ liệu từ các định dạng phổ biến như CSV, Excel và SQL, và các hàm để chọn, sắp xếp, và tập hợp các dữ liệu.

Pandas cũng cung cấp các công cụ để thực hiện các phép tính toán và phân tích thống kê trên dữ liệu, bao gồm các hàm tính toán tổng hợp, phân tích phân tích biến thể, và hồi quy tuyến tính. Nó cũng hỗ trợ việc tạo các biểu đồ và đồ thị để trực quan hóa dữ liệu.

Pandas là một trong những thư viện quan trọng nhất trong lĩnh vực phân tích dữ liệu và khoa học dữ liệu của Python. Nó được sử dụng rộng rãi trong các ứng dụng phân tích dữ liệu, kinh doanh và tài chính để xử lý và phân tích dữ liệu.

### **2.3.3. Thư viện Requests**

Thư viện Requests là một thư viện Python được sử dụng để tạo và quản lý các yêu cầu HTTP. Đây là một trong những thư viện quan trọng nhất trong lĩnh vực lập trình web của Python.

Requests được sử dụng để tương tác với các API và lấy dữ liệu từ các trang web khác. Nó cung cấp một API đơn giản để gửi các yêu cầu HTTP và lấy các phản hồi tương ứng. Requests hỗ trợ các phương thức HTTP như GET, POST, PUT, DELETE, HEAD, OPTIONS và PATCH. Nó cũng hỗ trợ các phương thức xác thực như Basic, Digest và OAuth.

Requests cung cấp các tính năng như gửi các yêu cầu có đính kèm dữ liệu, xử lý các truy vấn tham số và quản lý các tiêu đề yêu cầu. Điều này giúp cho việc thực hiện các yêu cầu HTTP trở nên dễ dàng và thuận tiện hơn.

### **2.3.4. Thư viện BeautifulSoup**

BeautifulSoup là một thư viện Python được sử dụng để phân tích cú pháp HTML và XML. Thư viện này cho phép bạn trích xuất dữ liệu từ các trang web và tài liệu XML theo cách dễ dàng và thuận tiện.

BeautifulSoup cung cấp các phương thức để tìm kiếm và truy xuất các phần tử HTML và XML dựa trên các thuộc tính của chúng. Nó cũng cho phép bạn tìm kiếm và trích xuất các thông tin từ các thẻ HTML và XML, như các văn bản, đường dẫn, hình ảnh, danh sách, bảng và các phần tử khác.

Với BeautifulSoup, chúng ta có thể trích xuất dữ liệu từ các trang web và tài liệu XML một cách dễ dàng và hiệu quả hơn. Thư viện này được sử dụng rộng rãi trong lĩnh vực web scraping và phân tích dữ liệu, giúp cho việc tự động trích xuất thông tin từ các trang web và tài liệu XML trở nên dễ dàng và thuận tiện hơn.

### **2.3.5. Thư viện regular expression**

Thư viện regular expression (re) là một thư viện Python được sử dụng để xử lý các biểu thức chính quy (regular expressions). Biểu thức chính quy là một chuỗi ký tự đặc biệt được sử dụng để mô tả một mẫu tìm kiếm.

Thư viện re cung cấp các phương thức để tìm kiếm, thay thế và chia tách chuỗi dựa trên các biểu thức chính quy. Nó cho phép bạn xử lý và trích xuất dữ liệu từ các chuỗi một cách nhanh chóng và dễ dàng.

Với thư viện re, ta có thể:

- Tìm kiếm chuỗi trong một văn bản.
- Tách chuỗi thành các thành phần khác nhau dựa trên các định dạng khác nhau.
- Thay thế các chuỗi bằng các chuỗi khác dựa trên các mẫu tìm kiếm.
- Kiểm tra xem một chuỗi có khớp với một biểu thức chính quy hay không.

Thư viện re được sử dụng rộng rãi trong các ứng dụng web và phân tích dữ liệu, giúp cho việc xử lý các chuỗi dữ liệu trở nên dễ dàng và thuận tiện hơn.

### **2.3.6. Thư viện *Regex***

Thư viện regex là một thư viện Python cung cấp các công cụ để xử lý các biểu thức chính quy (regular expressions) một cách hiệu quả và nhanh chóng. Thư viện này là một phần mở rộng của thư viện re cơ bản của Python, nhưng có các tính năng bổ sung và cải tiến để hỗ trợ cho các biểu thức chính quy phức tạp và các dạng dữ liệu khác nhau.

Regex hỗ trợ cho việc so khớp các biểu thức chính quy phức tạp, bao gồm các biểu thức chính quy có thể dùng để tìm kiếm và trích xuất thông tin từ các chuỗi dữ liệu phức tạp như các địa chỉ email, địa chỉ IP, số điện thoại, địa chỉ URL, v.v. Thư viện regex cũng hỗ trợ cho các biểu thức chính quy Unicode và các ký tự đa byte, giúp cho việc xử lý các dữ liệu phi-ASCII dễ dàng hơn.

Các tính năng khác của regex bao gồm các phương thức để thực hiện các tác vụ như: tìm kiếm, thay thế, tách chuỗi và so sánh các chuỗi dựa trên các biểu thức chính quy khác nhau. Regex cũng hỗ trợ cho các biểu thức chính quy động, cho phép bạn xây dựng các biểu thức chính quy dựa trên các biến và các điều kiện khác nhau.

Regex là một thư viện quan trọng và được sử dụng rộng rãi trong các ứng dụng web, phân tích dữ liệu và xử lý ngôn ngữ tự nhiên.

### **2.3.7. Thư viện *Underthesea***

Thư viện underthesea là một thư viện xử lý ngôn ngữ tự nhiên cho tiếng Việt được viết bằng Python. Thư viện này cung cấp các chức năng cho việc phân tích cú pháp (parsing), phân loại từ loại (part-of-speech tagging), tách từ (word segmentation), gán

thể ngữ nghĩa (named entity recognition), phân tích cảm xúc (sentiment analysis), và nhiều chức năng khác để xử lý văn bản tiếng Việt.

### **2.3.8. *TfidfVectorizer* class**

`TfidfVectorizer` là một class trong thư viện Scikit-learn của Python được sử dụng để chuyển đổi một tập hợp các văn bản thô thành một ma trận các đặc trưng TF-IDF.

TF-IDF là viết tắt của thuật ngữ term frequency-inverse document frequency, nó được sử dụng để đánh giá độ quan trọng của một từ trong một tài liệu. TF-IDF tính toán trọng số của mỗi từ trong một tài liệu bằng cách nhân tần suất xuất hiện của từ đó (term frequency) với đảo ngược tần suất xuất hiện của từ đó trong tất cả các tài liệu (inverse document frequency).

`TfidfVectorizer` sẽ trích xuất các từ trong các văn bản và tính toán ma trận các đặc trưng TF-IDF tương ứng với các từ đó. Ma trận đặc trưng này có thể được sử dụng để huấn luyện các mô hình học máy hoặc thực hiện các tác vụ khác như phân loại văn bản, phân cụm văn bản, và tìm kiếm thông tin.

### **2.3.9. Thư viện *NLTK***

NLTK là viết tắt của Natural Language Toolkit, là một thư viện mã nguồn mở được viết bằng ngôn ngữ Python để xử lý và phân tích ngôn ngữ tự nhiên. Nó cung cấp các công cụ và thư viện cho các tác vụ như xử lý văn bản, tách từ, phân loại văn bản, phân tích ngữ nghĩa và cú pháp, và nhiều hơn nữa.

NLTK có thể được sử dụng để xử lý dữ liệu văn bản trong nhiều ứng dụng khác nhau, bao gồm khoa học dữ liệu, phân tích cảm xúc, phân tích dữ liệu xã hội và nhiều lĩnh vực khác.

NLTK cũng có thể được sử dụng để phát triển các ứng dụng tự động hóa, chẳng hạn như chatbot và các hệ thống hỗ trợ quyết định.

### **2.3.10. Thư viện *Matplotlib***

Matplotlib là một thư viện Python được sử dụng rộng rãi để tạo ra các biểu đồ và đồ thị đẹp mắt, đa dạng về kiểu dáng và chất lượng.

Cung cấp cho người dùng một công cụ mạnh mẽ để trực quan hóa dữ liệu, phân tích và hiển thị kết quả một cách dễ dàng.

Matplotlib có thể được sử dụng để vẽ các biểu đồ đường, biểu đồ cột, biểu đồ tròn, biểu đồ phân tán, biểu đồ 3D và nhiều loại biểu đồ khác.

Là một trong những thư viện trực quan hóa dữ liệu phổ biến nhất trong cộng đồng khoa học dữ liệu và máy học Python.

### ***2.3.11. Giải thuật Cosine Similartiry***

Là một giải thuật đo độ tương đồng giữa 2 vector dựa trên góc giữa chúng trong không gian n chiều. Nó được sử dụng phổ biến trong các bài toán liên quan đến xử lý ngôn ngữ tự nhiên.

Để tính cosine giữa 2 vector  $x$  và  $y$ :

$$\text{Cosine\_similartiry}(x,y) = \text{dot}(x,y)/\|x\| * \|y\|$$

Trong đó:

**Dot(x,y)** là tích vô hướng (dot product) của 2 vector  $x$  và  $y$ , được tính bằng cách lấy tổng tích của từng phần tử tương ứng của 2 vector.

**$\|x\|$  và  $\|y\|$**  là độ dài của vector  $x$  và  $y$ , được tính bằng căn bậc hai của tổng bình phương của từng phần tử trong vector.

Kết quả của Cosine Similarity nằm trong khoảng  $[-1, 1]$ , trong đó giá trị 1 thể hiện hai vector hoàn toàn giống nhau, giá trị 0 thể hiện hai vector không có tương đồng và giá trị -1 thể hiện hai vector hoàn toàn khác nhau (tương đương với việc chúng nằm ở hai phía đối lập của không gian vector).

Trong bài toán tìm kiếm, cosine similarity được sử dụng để tính độ tương đồng giữa cụm từ tìm kiếm và các văn bản trong tập dữ liệu, và xác định những văn bản có độ tương đồng cao nhất với cụm từ đó.

## CHƯƠNG 3. MÔ HÌNH BÀI TOÁN

### 3.1. Mô hình bài toán



**Hình 3.1:** Mô hình bài toán

### 3.2. Giải thích các bước

Thu thập dữ liệu: Crawl 1000 tin tức từ trang web Vietnamnet.vn về bằng thư viện requests với các nội dung: Tiêu đề, Thẻ loại, Mô tả, Nội dung.

Xuất File Excel: lưu các dữ liệu vừa thu thập được thành một file excel bằng thư viện pandas với các trường: Tiêu đề, Thẻ loại, Mô tả, Nội dung.

Xoá tab HTML, xoá số: tiến hành xử lý dữ liệu, xoá các tab HTML và các số trong dữ liệu bằng cách sử dụng thư viện re.

Chuyển văn bản thành chữ thường: sử dụng hàm lower để chuyển dữ liệu về dạng chữ thường.

Chuẩn hoá kiểu gõ dấu tiếng Việt: Chuẩn hoá bảng mã sau đó chuẩn hoá kiểu gõ về kiểu gõ cũ. Ví dụ: Hoà – Hòa

Xoá Stopword: tiến hành xoá các từ dừng như: “thế”, “thì”, “là”,...

Chuyển văn bản sang dạng ngữ nghĩa: sử dụng thư viện underthesea để tách văn bản thành dạng ngữ nghĩa.

Chuyển văn bản sang dạng vector: sử dụng class TfidfVectorizer của thư viện Scikit-learn để chuyển dữ liệu thành dạng vector.



Tìm kiếm và so khớp: sử dụng thuật toán tìm kiếm mẫu trong một chuỗi văn bản dựa vào thuật toán Knuth-Morris-Pratt. Tìm kiếm

## CHƯƠNG 4. THỰC NGHIỆM

### 4.1. Các bước trong mô hình

#### 4.1.1. Code thêm thư viện

```
import numpy as np
import requests as rq
from bs4 import BeautifulSoup as bs
import pandas as pd
import re
import regex as re
import os
import sys
# from Logger import LogEventSourcing
from datetime import datetime
import dateutil.parser
import traceback
import time
import requests
```

Giải thích: Đoạn mã trên đầu tiên import các thư viện và modules cần thiết cho chương trình:

- **numpy** để xử lý mảng nhiều chiều.
- **requests** để tạo và gửi HTTP requests đến các URL.
- **BeautifulSoup** từ thư viện bs4 để phân tích cú pháp của HTML và XML.
- **pandas** để làm việc với dữ liệu dưới dạng bảng.
- **re** và **regex** để sử dụng biểu thức chính quy.
- **os** và **sys** để tương tác với hệ điều hành và các thông tin hệ thống.
- **datetime** để xử lý các giá trị ngày tháng.
- **traceback** để hiển thị các lỗi và thông tin debug.
- **time** để đo thời gian thực thi của chương trình.

#### 4.1.2. Code lấy thông tin trang web

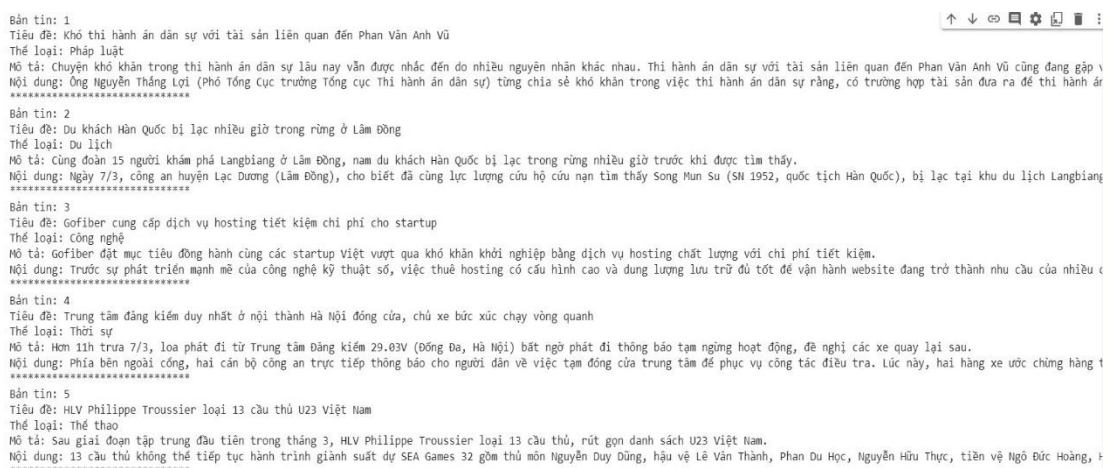
```
page = 1
i = 0
#Requests trang web
r = rq.get("https://vietnamnet.vn/tin-tuc-24h")
#Chuẩn hoá định dạng HTML
soup = bs(r.content, "html.parser")
#Lấy title bảng tin mới
titles = soup.find_all('h2', class_="feature-box__content--title vnn-
title")
while (len(titles)<100):
    rr = rq.get("https://vietnamnet.vn/tin-tuc-24h" + "-p" + str(page))
    soup = bs(rr.content, "html.parser")
    titless = titles + soup.find_all('h2', class_="feature-box__content--
title vnn-title")
    titles += titless
    page += 1
links = [link.find('a').attrs["href"] for link in titless]
tieudes, motas, theloais, noidungs = [], [], [], []
for link in links:
    i = i + 1
    news = rq.get(link)
    soup = bs(news.content, "html.parser")
    try:
        tieude = soup.find("h2", class_="vnn-title").text.strip()
        mota = soup.find("div", class_="newFeature__main-
textBold").text.strip()
        theloai = soup.find("a", class_="leading-30").text.strip()
        noidung = soup.find("div", class_="maincontent ").text
        noidung = " ".join(noidung.split())
    except:
        pass
    print(30*"*)
```

```

tieudes.append(tieude)
motas.append(mota)
theloais.append(theloai)
noidungs.append(noidung)
print("Tiêu đề:", tieude)
print("Mô tả:", mota)
print("Thể loại:", theloai)
print("Nội dung:", noidung)
print("Bản tin:", str(i))

# Tạo một dataframe
data = { 'Tiêu đề': tieudes,
        'Mô tả': motas,
        'Thể loại': theloais,
        'Nội dung': noidungs,
        }
df = pd.DataFrame(data)
# Lưu dataframe thành một file excel
df.to_excel('./content/drive/MyDrive/File Excel/VietNamNet1.xlsx', i
ndex=False)

```



Bản tin: 1  
Tiêu đề: Khó thi hành án dân sự với tài sản liên quan đến Phan Văn Anh Vũ  
Thể loại: Pháp luật  
Mô tả: Chuyện khó khăn trong thi hành án dân sự lâu nay vẫn được nhắc đến do nhiều nguyên nhân khác nhau. Thi hành án dân sự với tài sản liên quan đến Phan Văn Anh Vũ cũng đang gặp \n  
Nội dung: Ông Nguyễn Thắng Lợi (Phó Tổng Cục trưởng Tổng cục Thi hành án dân sự) từng chia sẻ khó khăn trong việc thi hành án dân sự rằng, có trường hợp tài sản đưa ra để thi hành án \n  
\*\*\*\*\*

Bản tin: 2  
Tiêu đề: Du khách Hàn Quốc bị lạc nhiều giờ trong rừng ở Lâm Đồng  
Thể loại: Du lịch  
Mô tả: Cùng đoàn 15 người khám phá Langbiang ở Lâm Đồng, nam du khách Hàn Quốc bị lạc trong rừng nhiều giờ trước khi được tìm thấy.  
Nội dung: Ngày 7/3, công an huyện Lạc Dương (Lâm Đồng), cho biết đã cùng lực lượng cứu hộ cứu nạn tìm thấy Song Mun Su (SN 1952, quốc tịch Hàn Quốc), bị lạc tại khu du lịch Langbiang \n  
\*\*\*\*\*

Bản tin: 3  
Tiêu đề: Gofiber cung cấp dịch vụ hosting tiết kiệm chi phí cho startup  
Thể loại: Công nghệ  
Mô tả: gofiber đặt mục tiêu đồng hành cùng các startup Việt vượt qua khó khăn khởi nghiệp bằng dịch vụ hosting chất lượng với chi phí tiết kiệm.  
Nội dung: Trước sự phát triển mạnh mẽ của công nghệ kỹ thuật số, việc thuê hosting có cấu hình cao và dung lượng lưu trữ đủ tốt để vận hành website đang trở thành nhu cầu của nhiều c \n  
\*\*\*\*\*

Bản tin: 4  
Tiêu đề: Trung tâm đăng kiểm duy nhất ở nội thành Hà Nội đóng cửa, chủ xe bức xúc chạy vòng quanh  
Thể loại: Thời sự  
Mô tả: Hơn 11h trưa 7/3, loa phát đi từ Trung tâm Đăng kiểm 29.03V (Đống Đa, Hà Nội) bắt ngờ phát đi thông báo tạm ngừng hoạt động, đề nghị các xe quay lại sau.  
Nội dung: Phía bên ngoài cổng, hai cán bộ công an trực tiếp thông báo cho người dân về việc tạm đóng cửa trung tâm để phục vụ công tác điều tra. Lúc này, hai hàng xe ước chừng hàng t \n  
\*\*\*\*\*

Bản tin: 5  
Tiêu đề: HLV Philippe Troussier loại 13 cầu thủ U23 Việt Nam  
Thể loại: Thể thao  
Mô tả: Sau giải đấu tập trung đầu tiên trong tháng 3, HLV Philippe Troussier loại 13 cầu thủ, rút gọn danh sách U23 Việt Nam.  
Nội dung: 13 cầu thủ không thể tiếp tục hành trình giành suất dự SEA Games 32 gồm thủ môn Nguyễn Duy Dũng, hậu vệ Lê Văn Thành, Phan Du Học, Nguyễn Hữu Thực, tiền vệ Ngô Đức Hoàng, t \n  
\*\*\*\*\*

**Hình 4.1:** Kết quả lấy thông tin từ website

Giải thích: đoạn mã trên sử dụng thư viện requests và BeautifulSoup để thực hiện các thao tác sau:

- Truy cập vào trang web <https://vietnamnet.vn/tin-tuc-24h> để lấy thông tin về các bài báo.
- Lấy tất cả các tiêu đề bài báo và link của chúng thông qua việc tìm các thẻ h2 có class là "feature-box\_\_content--title vnn-title".
- Lặp lại việc truy cập trang web và lấy các tiêu đề bài báo và link của chúng trên các trang tiếp theo bằng cách thêm số trang vào đường dẫn url.
- Dùng link của mỗi bài báo để truy cập vào trang bài báo và lấy thông tin về tiêu đề, mô tả, thể loại và nội dung bài báo.
- Lưu thông tin về các bài báo vào một dataframe và lưu dataframe đó thành một file excel.

#### 4.1.3. Code load dữ liệu và xóa số

```
# load dữ liệu từ excel
data = pd.read_excel('../content/drive/MyDrive/File Excel/VietNamNet1.xlsx')
data.head(1000)
# Xóa Số
data['Tiêu đề'] = data['Tiêu đề'].str.replace('\d', "", regex = True)
data['Mô tả'] = data['Mô tả'].str.replace('\d', "", regex = True)
data['Nội dung'] = data['Nội dung'].str.replace('\d', "", regex = True)
data.head(1000)
```

	Tiêu đề	Thể loại	Mô tả	Nội dung
0	'Đừng làm mẹ cầu' bất ngờ tặng số tập	Giải trí	Đại diện VFC cho biết 'Đừng làm mẹ cầu' không ...	Bộ phim Đừng làm mẹ cầu đang dẫn đến hồi kết v...
1	Ngành chăn nuôi và bài toán vượt khó trong quý...	Kinh Doanh	Bất chấp những thách thức ảnh hưởng đến chuỗi ...	Được kỳ vọng sẽ tiếp tục tăng trưởng trong năm...
2	Bất ngờ người đưa HLV Philippe Troussier trở l...	Thể thao	Đề xuất mời HLV Philippe Troussier dẫn dắt tuy...	Ngày khi HLV Park Hang Seo thông báo không gia...
3	Vợ tiếp tục đưa NSND Công Lý sang Nhật Bản chứ...	Giải trí	Trên trang cá nhân, Ngọc Hà đăng ảnh chụp ở Nh...	Sao Việt ngày 23/2. "Khi Hà muốn hôm nay phải ...
4	Tiến sĩ Bùi Sỹ Lợi: Giải tỏa ngay máy móc 'đắp...	Sức khỏe	Tiến sĩ Lợi cho rằng các sai phạm đều do con n...	Tại buổi tọa đàm "Ngành y vượt khó" ngày 23/2,...
...	...	...	...	...
995	Kết quả bóng đá hôm nay 24/2: MU loại Barca, R...	Thể thao	Cập nhật nhanh kết quả bóng đá hôm nay 24/2/20...	Ngày giờ Cập đầu Trực tiếp UEFA EUROPA LEAGUE ...
996	Link xem trực tiếp bóng đá MU vs Barcelona: Vô...	Thể thao	Cập nhật kênh phát sóng và link xem trực tiếp ...	Trực tiếp MU vs Barcelona: Đội hình ra sân mạn...
997	Bộ trưởng Quốc phòng Nga tuyên bố cứng rắn trư...	Thể giới	Bộ trưởng Quốc phòng Nga Sergei Shoigu cho răn...	"Một lần nữa, chúng ta đang gặp nguy hiểm. Phư...
998	Đương kim Hoa hậu Hoàn vũ đến Việt Nam	Giải trí	Tối 23/2, siêu mẫu Lan Khuê có mặt tại sân bay...	Hai mỹ nhân Hoa hậu Hoàn vũ đến Việt Nam lần n...
999	Văn Khang mắc lỗi, U20 Việt Nam thua Dubai Cit...	Thể thao	Ở bài test cuối cùng trước thêm VCK U20 châu Á...	HLV Hoàng Anh Tuấn sử dụng đội hình xuất phát ...

1000 rows x 4 columns

**Hình 4.2:** Kết quả load dữ liệu từ excel

	Tiêu đề	Thể loại	Mô tả	Nội dung
0	'Đừng làm mẹ cầu' bất ngờ tặng số tập	Giải trí	Đại diện VFC cho biết 'Đừng làm mẹ cầu' không ...	Bộ phim Đừng làm mẹ cầu đang dẫn đến hồi kết v...
1	Ngành chăn nuôi và bài toán vượt khó trong quý /	Kinh Doanh	Bất chấp những thách thức ảnh hưởng đến chuỗi ...	Được kỳ vọng sẽ tiếp tục tăng trưởng trong năm...
2	Bất ngờ người đưa HLV Philippe Troussier trở l...	Thể thao	Đề xuất mời HLV Philippe Troussier dẫn dắt tuy...	Ngày khi HLV Park Hang Seo thông báo không gia...
3	Vợ tiếp tục đưa NSND Công Lý sang Nhật Bản chứ...	Giải trí	Trên trang cá nhân, Ngọc Hà đăng ảnh chụp ở Nh...	Sao Việt ngày / . "Khi Hà muốn hôm nay phải làm...
4	Tiến sĩ Bùi Sỹ Lợi: Giải tỏa ngay máy móc 'đắp...	Sức khỏe	Tiến sĩ Lợi cho rằng các sai phạm đều do con n...	Tại buổi tọa đàm "Ngành y vượt khó" ngày / , lã...
...	...	...	...	...
995	Kết quả bóng đá hôm nay /: MU loại Barca, Roma...	Thể thao	Cập nhật nhanh kết quả bóng đá hôm nay //, kết...	Ngày giờ Cập đầu Trực tiếp UEFA EUROPA LEAGUE ...
996	Link xem trực tiếp bóng đá MU vs Barcelona: Vô...	Thể thao	Cập nhật kênh phát sóng và link xem trực tiếp ...	Trực tiếp MU vs Barcelona: Đội hình ra sân mạn...
997	Bộ trưởng Quốc phòng Nga tuyên bố cứng rắn trư...	Thể giới	Bộ trưởng Quốc phòng Nga Sergei Shoigu cho răn...	"Một lần nữa, chúng ta đang gặp nguy hiểm. Phư...
998	Đương kim Hoa hậu Hoàn vũ đến Việt Nam	Giải trí	Tối /, siêu mẫu Lan Khuê có mặt tại sân bay để...	Hai mỹ nhân Hoa hậu Hoàn vũ đến Việt Nam lần n...
999	Văn Khang mắc lỗi, U Việt Nam thua Dubai City ...	Thể thao	Ở bài test cuối cùng trước thêm VCK U châu Á ,...	HLV Hoàng Anh Tuấn sử dụng đội hình xuất phát ...

1000 rows x 4 columns

**Hình 4.3:** Kết quả đoạn mã xóa số

Giải thích: Đoạn mã trên có chức năng load dữ liệu từ file Excel "VietNamNet1.xlsx" vào một dataframe (gọi là "data"). Sau đó, nó sử dụng phương thức head() để hiển thị 1000 hàng đầu tiên của dataframe.

Tiếp theo, đoạn mã sử dụng phương thức str.replace() để xóa các ký tự số từ cột "Tiêu đề", "Mô tả" và "Nội dung" của dataframe.

Cuối cùng, đoạn mã sử dụng phương thức head() để hiển thị 1000 hàng đầu tiên của dataframe sau khi xóa các ký tự số.

#### 4.1.4. Code xóa tab html

```
# Xóa tab HTML
clear = re.compile('<.*?>')
def XoaHTML(str):
    id = 0
    for e in data[str]:
        data[str][id] = re.sub(clear, "", e)
        id+=1
XoaHTML("Tiêu đề")
XoaHTML("Mô tả")
XoaHTML("Nội dung")
data.to_excel("../content/vietnamnet.xlsx")
data.head(1000)
```



	Tiêu đề	Thể loại	Mô tả	Nội dung
0	'Đừng làm mẹ cầu' bất ngờ tăng số tập	Giải trí	Dại diện VFC cho biết 'Đừng làm mẹ cầu' không ...	Bộ phim Đừng làm mẹ cầu đang dần đến hồi kết v...
1	Ngành chăn nuôi và bài toán vượt khó trong quý II	Kinh Doanh	Bất chấp những thách thức ảnh hưởng đến chuỗi ...	Được kỳ vọng sẽ tiếp tục tăng trưởng trong năm...
2	Bất ngờ người đưa HLV Philippe Troussier trở l...	Thể thao	Đề xuất mời HLV Philippe Troussier dẫn dắt tuy...	Ngay khi HLV Park Hang Seo thông báo không gia...
3	Vợ tiếp tục đưa NSND Công Lý sang Nhật Bản chữ...	Giải trí	Trên trang cá nhân, Ngọc Hà đăng ảnh chụp ở Nh...	Sao Việt ngày /. "Khi Hà muốn hôm nay phải làm...
4	Tiến sĩ Bùi Sỹ Lợi: Giải tòa ngay máy móc 'đáp...	Sức khỏe	Tiến sĩ Lợi cho rằng các sai phạm đều do con n...	Tại buổi toa đàm "Ngành y vượt khó" ngày /, là...
...	...	...	...	...
995	Kết quả bóng đá hôm nay /: MU loại Barca, Roma...	Thể thao	Cập nhật nhanh kết quả bóng đá hôm nay //, kết...	Ngày giờ Cập đầu Trực tiếp UEFA EUROPA LEAGUE ...
996	Link xem trực tiếp bóng đá MU vs Barcelona: Vô...	Thể thao	Cập nhật kênh phát sóng và link xem trực tiếp ...	Trực tiếp MU vs Barcelona: Đội hình ra sân mạn...
997	Bộ trưởng Quốc phòng Nga tuyên bố cứng rắn trư...	Thể giới	Bộ trưởng Quốc phòng Nga Sergei Shoigu cho rằn...	"Một lần nữa, chúng ta đang gặp nguy hiểm. Phư...
998	Đường kim Hoa hậu Hoàn vũ đến Việt Nam	Giải trí	Tối /, siêu mẫu Lan Khuê có mặt tại sân bay để...	Hai mỹ nhân Hoa hậu Hoàn vũ đến Việt Nam lần n...
999	Văn Khang mắc lỗi, U Việt Nam thua Dubai City ...	Thể thao	Ở bài test cuối cùng trước thêm VCK U châu Á ,...	HLV Hoàng Anh Tuấn sử dụng đội hình xuất phát ...

1000 rows x 4 columns

**Hình 4.4:** Kết quả đoạn mã xóa tab html

Giải thích các bước thực hiện:

- Khai báo biến clear là một biểu thức chính quy được sử dụng để tìm và loại bỏ các thẻ HTML. Biểu thức chính quy này sẽ tìm các chuỗi bắt đầu bằng ký tự '<', theo sau bởi bất kỳ số lượng ký tự nào và kết thúc bằng ký tự '>', tức là các thẻ HTML.
- Sử dụng một biến đếm id để duyệt qua từng phần tử của ba cột "Tiêu đề", "Mô tả" và "Nội dung" trong dataframe data.
- Sử dụng phương thức re.sub() để thay thế các chuỗi khớp với biểu thức chính quy clear bằng chuỗi rỗng, tức là loại bỏ các thẻ HTML.

- Tăng biến đếm id lên 1 để duyệt đến phần tử tiếp theo.
- Sau khi xử lý hết các phần tử của ba cột, dataframe data được xuất ra file excel tại đường dẫn "../content/VietNamNet1.xlsx" và in ra 1000 phần tử đầu tiên của dataframe.

#### 4.1.5. Code chuyển văn bản thành chữ thường

```
# Chuyển văn bản thành chữ thường
data['Tiêu đề'] = data['Tiêu đề'].str.lower()
data['Thể loại'] = data['Thể loại'].str.lower()
data['Mô tả'] = data['Mô tả'].str.lower()
data['Nội dung'] = data['Nội dung'].str.lower()
data.head(1000)
```

	Tiêu đề	Thể loại	Mô tả	Nội dung
0	'đừng làm mẹ cầu' bất ngờ tăng số tập	giải trí	dại diện vfc cho biết 'đừng làm mẹ cầu' không ...	bộ phim đừng làm mẹ cầu đang dần đến hồi kết v...
1	ngành chăn nuôi và bài toán vượt khó trong quý i/	kinh doanh	bất chấp những thách thức ảnh hưởng đến chuỗi ...	được kỳ vọng sẽ tiếp tục tăng trưởng trong năm...
2	bất ngờ người đưa hiv philippe troussier trở l...	thể thao	đề xuất mời hiv philippe troussier dẫn dắt tuy...	ngay khi hiv park hang seo thông báo không gia...
3	vợ tiếp tục đưa nsnd công lý sang nhật bản chữ...	giải trí	trên trang cá nhân, ngọc hà đăng ảnh chụp ở nh...	sao việt ngày /. "khi hà muốn hôm nay phải làm...
4	tiền sĩ bùi sỹ lợi: giải tỏa ngay mây móc 'đáp...	sức khỏe	tiền sĩ lợi cho rằng các sai phạm đều do con n...	tại buổi tọa đàm "ngành y vượt khó" ngày /, lã...
...	...	...	...	...
995	kết quả bóng đá hôm nay /: mu loại barca, roma...	thể thao	cập nhật nhanh kết quả bóng đá hôm nay //, kết...	ngày giờ cập đầu trực tiếp uefa europa league ...
996	link xem trực tiếp bóng đá mu vs barcelona: vò...	thể thao	cập nhật kênh phát sóng và link xem trực tiếp ...	trực tiếp mu vs barcelona: đội hình ra sân mạn...
997	bộ trưởng quốc phòng nga tuyên bố cứng rắn trư...	thế giới	bộ trưởng quốc phòng nga sergei shoigu cho rằn...	"một lần nữa, chúng ta đang gặp nguy hiểm. phư...
998	đương kim hoa hậu hoàn vũ đến việt nam	giải trí	tôi /, siêu mẫu lan khuê có mặt tại sân bay để...	hai mỹ nhân hoa hậu hoàn vũ đến việt nam lần n...
999	văn khương mắc lỗi, u việt nam thua dubai city ...	thể thao	ở bài test cuối cùng trước thêm vck u châu á ,...	hiv hoàng anh tuấn sử dụng đội hình xuất phát ...

1000 rows x 4 columns

**Hình 4.5:** Kết quả đoạn mã chuyển văn bản thành chữ thường

Giải thích: Đoạn mã trên được sử dụng để chuyển đổi văn bản trong các cột của dataframe data thành chữ thường. Cụ thể, các cột Tiêu đề, Thể loại, Mô tả và Nội dung sẽ được chuyển đổi sang chữ thường bằng cách sử dụng phương thức str.lower().

- Sử dụng phương thức str.lower() để chuyển đổi văn bản trong cột "Tiêu đề" sang chữ thường, và gán kết quả trở lại cột "Tiêu đề".
- Sử dụng phương thức str.lower() để chuyển đổi văn bản trong cột "Thể loại" sang chữ thường, và gán kết quả trở lại cột "Thể loại".
- Sử dụng phương thức str.lower() để chuyển đổi văn bản trong cột "Mô tả" sang chữ thường, và gán kết quả trở lại cột "Mô tả".
- Sử dụng phương thức str.lower() để chuyển đổi văn bản trong cột "Nội dung" sang chữ thường, và gán kết quả trở lại cột "Nội dung".





[illegible]

#### Hình 4.6: Kết quả đoạn mã chuẩn hóa bảng mã

#### 4.1.7. Code chuẩn hóa kiểu gõ

```
# Chuẩn hóa kiểu gõ
bang_nguyen_am = [['a', 'à', 'á', 'ả', 'ã', 'ạ', 'a'],
                  ['ă', 'ằ', 'ắ', 'ẳ', 'ẵ', 'ặ', 'aw'],
                  ['â', 'ầ', 'ấ', 'ẩ', 'ẫ', 'ậ', 'aa'],
                  ['e', 'è', 'é', 'ẻ', 'ẽ', 'ẹ', 'e'],
                  ['ê', 'ề', 'ế', 'ể', 'ễ', 'ệ', 'ee'],
                  ['i', 'ì', 'í', 'ỉ', 'ĩ', 'ị', 'i'],
                  ['o', 'ò', 'ó', 'ỏ', 'õ', 'ộ', 'o'],
                  ['ô', 'ồ', 'ố', 'ỗ', 'ỗ', 'ộ', 'oo'],
                  ['ơ', 'ờ', 'ớ', 'ở', 'ỡ', 'ợ', 'ow'],
                  ['u', 'ù', 'ú', 'ủ', 'ũ', 'ụ', 'u'],
                  ['ư', 'ừ', 'ứ', 'ử', 'ữ', 'ự', 'uw'],
                  ['y', 'ỳ', 'ý', 'ỷ', 'ỹ', 'ỵ', 'y']]
bang_ky_tu_dau = [' ', 'f', 's', 'r', 'x', 'j']

nguyen_am_to_ids = {}

for i in range(len(bang_nguyen_am)):
    for j in range(len(bang_nguyen_am[i]) - 1):
        nguyen_am_to_ids[bang_nguyen_am[i][j]] = (i, j)

def vn_word_to_tex_type(word):
    dau_cau = 0
    new_word = ""
    for char in word:
        x, y = nguyen_am_to_ids.get(char, (-1, -1))
        if x == -1:
            new_word += char
            continue
        if y != 0:
            dau_cau = y
        new_word += bang_nguyen_am[x][-1]
        new_word += bang_ky_tu_dau[dau_cau]
    return new_word
```

```

def vn_sentence_to_telex_type(sentence):
    """
    Chuyển câu tiếng việt có dấu về kiểu gõ telex.
    :param sentence:
    :return:
    """

    words = sentence.split()
    for index, word in enumerate(words):
        words[index] = vn_word_to_telex_type(word)
    return ''.join(words)
"""

    End section: Chuyển câu văn về kiểu gõ telex khi không bật Unikey
"""

def chuan_hoa_dau_tu_tiang_viet(word):
    if not is_valid_vietnam_word(word):
        return word
    chars = list(word)
    dau_cau = 0
    nguyen_am_index = []
    qu_or_gi = False
    for index, char in enumerate(chars):
        x, y = nguyen_am_to_ids.get(char, (-1, -1))
        if x == -1:
            continue
        elif x == 9: # check qu
            if index != 0 and chars[index - 1] == 'q':
                chars[index] = 'u'
                qu_or_gi = True
        elif x == 5: # check gi
            if index != 0 and chars[index - 1] == 'g':
                chars[index] = 'i'
                qu_or_gi = True
        if y != 0:
            dau_cau = y
            chars[index] = bang_nguyen_am[x][0]

```

```

if not qu_or_gi or index != 1:
    nguyen_am_index.append(index)
if len(nguyen_am_index) < 2:
    if qu_or_gi:
        if len(chars) == 2:
            x, y = nguyen_am_to_ids.get(chars[1])
            chars[1] = bang_nguyen_am[x][dau_cau]
        else:
            x, y = nguyen_am_to_ids.get(chars[2], (-1, -1))
            if x != -1:
                chars[2] = bang_nguyen_am[x][dau_cau]
            else:
                chars[1] = bang_nguyen_am[5][dau_cau] if chars[1] == 'i'
else bang_nguyen_am[9][dau_cau]
    return ".join(chars)
return word

for index in nguyen_am_index:
    x, y = nguyen_am_to_ids[chars[index]]
    if x == 4 or x == 8: # ê, ô
        chars[index] = bang_nguyen_am[x][dau_cau]
    return ".join(chars)
if len(nguyen_am_index) == 2:
    if nguyen_am_index[-1] == len(chars) - 1:
        x, y = nguyen_am_to_ids[chars[nguyen_am_index[0]]]
        chars[nguyen_am_index[0]] = bang_nguyen_am[x][dau_cau]
    else:
        x, y = nguyen_am_to_ids[chars[nguyen_am_index[1]]]
        chars[nguyen_am_index[1]] = bang_nguyen_am[x][dau_cau]
    else:
        x, y = nguyen_am_to_ids[chars[nguyen_am_index[1]]]
        chars[nguyen_am_index[1]] = bang_nguyen_am[x][dau_cau]
return ".join(chars)

```

```

def is_valid_vietnam_word(word):
    chars = list(word)
    nguyen_am_index = -1
    for index, char in enumerate(chars):
        x, y = nguyen_am_to_ids.get(char, (-1, -1))
        if x != -1:
            if nguyen_am_index == -1:
                nguyen_am_index = index
            else:
                if index - nguyen_am_index != 1:
                    return False
                nguyen_am_index = index
    return True

def chuan_hoa_dau_cau_tiang_viet(sentence):
    """
    Chuyển câu tiếng việt về chuẩn gõ dấu kiểu cũ.
    :param sentence:
    :return:
    """
    sentence = sentence.lower()
    words = sentence.split()
    for index, word in enumerate(words):
        cw = re.sub(r'(^p{P}*)(p{L}.)*p{L}+(\p{P}*$)', r'\1/\2/\3', word).split('/')
        # print(cw)
        if len(cw) == 3:
            cw[1] = chuan_hoa_dau_tu_tiang_viet(cw[1])
        words[index] = ''.join(cw)
    return ''.join(words)

```

```
def Duyet(str):
    id = 0
    for e in data[str]:
        if __name__ == '__main__':
            data[str][id] = chuan_hoa_dau_cau_tiemg_viet(data[str][id])
            id+=1
    Duyet('Tiêu đề')
    Duyet('Thể loại')
    Duyet('Mô tả')
    Duyet('Nội dung')
    data.to_excel("../content/VietNamNet1.xlsx")
    data.head(1000)
```

	Tiêu đề	Thể loại	Mô tả	Nội dung
0	'đừng làm mẹ cầu' bất ngờ tăng số tập	giải trí	đại diện vlc cho biết 'đừng làm mẹ cầu' không ...	bộ phim đừng làm mẹ cầu đang dần đến hồi kết v...
1	ngành chăn nuôi và bài toán vượt khó trong quý i	kinh doanh	bất chấp những thách thức ảnh hưởng đến chuỗi ...	được kỳ vọng sẽ tiếp tục tăng trưởng trong năm...
2	bất ngờ người đưa hiv philippe troussier trở l...	thể thao	đề xuất mời hiv philippe troussier dẫn dắt tuy...	ngay khi hiv park hang seo thông báo không gia...
3	vợ tiếp tục đưa nsnd công lý sang nhật bản chữ...	giải trí	trên trang cá nhân, ngọc hà đăng ảnh chụp ở nh...	sao việt ngày . "khi hà muốn hôm nay phải làm ...
4	tiến sĩ bùi sỹ lợi: giải tỏa ngay máy móc 'đáp...	sức khỏe	tiến sĩ lợi cho rằng các sai phạm đều do con n...	tại buổi tọa đàm "ngành y vượt khó" ngày , lần...
...	...	...	...	...
995	kết quả bóng đá hôm nay : mu loại barca, roma ...	thể thao	cập nhật nhanh kết quả bóng đá hôm nay , kết q...	ngày giờ cặp đấu trực tiếp uefa europa league ...
996	link xem trực tiếp bóng đá mu vs barcelona: vô...	thể thao	cập nhật kênh phát sóng và link xem trực tiếp ...	trực tiếp mu vs barcelona: đội hình ra sân mạn...
997	bộ trưởng quốc phòng nga tuyên bố cứng rắn trư...	thế giới	bộ trưởng quốc phòng nga sergei shoigu cho răn...	"một lần nữa, chúng ta đang gặp nguy hiểm. phư...
998	đương kim hoa hậu hoàn vũ đến việt nam	giải trí	tôi , siêu mẫu lan khuê có mặt tại sân bay để ...	hai mỹ nhân hoa hậu hoàn vũ đến việt nam lần n...
999	văn khang mắc lỗi, u việt nam thua dubai city ...	thể thao	ở bài test cuối cùng trước thêm vlc u châu á , ...	hiv hoàng anh tuấn sử dụng đội hình xuất phát ...

1000 rows x 4 columns

**Hình 4.7:** Kết quả đoạn mã chuẩn hóa kiểu gõ

#### 4.1.8. Code xóa stopwords

## # Xóa Stopword

```
def Xoa_stop_words(text):
    tmp = text.split(' ')
    for stop_word in stop_words:
        temp = stop_word.strip()
        for chu in tmp:
            if chu == temp: tmp.remove(chu)
    return " ".join(tmp)

with open('../content/drive/MyDrive/File Excel/vietnamese-
stopwords.txt', 'r', encoding='utf-8') as file:
    stop_words = file.readlines()

def TuDung (str):
    id = 0
    for e in data[str]:
        data[str][id]= Xoa_stop_words(data[str][id])
        id += 1
    TuDung('Tiêu đề')
    TuDung('Thể loại')
    TuDung('Mô tả')
    TuDung('Nội dung')
    data.head(1000)
```

	Tiêu đề	Thể loại	Mô tả	Nội dung
0	'đừng mẹ cầu' bất ngờ tập	giải trí	đại diện vtc 'đừng mẹ cầu' đừng tập dự kiến ba...	phim đừng mẹ cầu dẫn hồi kết đầu tỷ lệ truyền ...
1	ngành chăn nuôi toàn quý i	kinh doanh	bất chấp thách thức ảnh hưởng chuỗi cung ứng t...	kỳ vọng tiếp tục trưởng , liệu khăn đối doanh ...
2	bất ngờ hiv philippe troussier trở việt nam	thể thao	đề xuất mời hiv philippe troussier dắt tuyển v...	hiv park hang seo thông báo gia hạn hợp đồng ...
3	vợ tiếp tục nsnd công lý nhật chữa bệnh	giải trí	trang cá nhân, ngọc hà đăng ảnh chụp nhật bản...	việt . "khi hà hôm hiệu quả, thúc đẩy hà ơn ỉ...
4	tiền sĩ bùi sỹ lợi: giải tỏa máy móc 'đáp chiế...	sức khỏe	tiền sĩ lợi sai phạm người, máy móc sai thao g...	toạ đàm "ngành y khó" , lãnh đạo bệnh viện tuy...
...	...	...	...	...
995	kết bóng đá hôm : mu barca, roma "rủ" juventus...	thể thao	cập nhật kết bóng đá hôm , kết lượt vòng play-...	cập đầu trực tiếp uefa europa league - lượt v...
996	link trực tiếp bóng đá mu vs barcelona: vòng p...	thể thao	cập nhật kênh phát sóng link trực tiếp trận đá...	trực tiếp mu vs barcelona: đội hình sân nhấtr...
997	trưởng quốc phòng nga tuyên bố cứng rắn thêm c...	giới	trưởng quốc phòng nga sergei shoigu phương tây...	"một nửa, ta nguy hiểm. phương tây lợi dụng uk...
998	đương kim hoa hậu hoàn vũ việt nam	giải trí	tối , siêu mẫu lan khuê mặt sân bay đón đẹp ho...	hai mỹ nhân hoa hậu hoàn vũ việt nam đương kim...
999	ván khang mặc lỗi, u việt nam thua dubai city ...	thể thao	test thêm vck u châu , u việt nam thua dubai c...	hiv hoàng tuần sử dụng đội hình xuất phát quan...

1000 rows × 4 columns

**Hình 4.8:** Kết quả đoạn mã xóa stopwords



Giải thích:

Hàm **Xoa\_stop\_words(text)** nhận đầu vào là một văn bản **text**, được phân tách thành các từ bằng dấu cách. Đối với mỗi từ trong văn bản, nếu từ đó có trong danh sách **stop\_words** (được đọc từ file **vietnamese-stopwords.txt**), thì từ đó sẽ bị xóa bỏ. Cuối cùng, văn bản được ghép lại từ các từ còn lại bằng dấu cách và được trả về.

Hàm **TuDung(str)** nhận đầu vào là tên cột trong DataFrame data. Hàm sử dụng hàm **Xoa\_stop\_words(text)** để xóa bỏ các stopwords trong mỗi phần tử của cột đó. Cuối cùng, DataFrame data được cập nhật lại các phần tử trong cột tương ứng với các từ đã được xóa stopwords.

#### 4.1.9. Code chuyển sang dạng ngữ nghĩa

```
!pip install underthesea
import underthesea
rd = pd.read_excel('../content/drive/MyDrive/File Excel/VietNamNet1
.xlsx')
dt = pd.DataFrame(rd, columns = ['Tiêu đề', 'Thể loại', 'Mô tả', 'Nội du
ng'] )
Tieude = rd['Tiêu đề']
Theloai =rd['Thể loại']
Mota = rd['Mô tả']
Noidung = rd['Nội dung']

def ngunghia(str):
    ngunghia =[]
    for x in str:
        text = x
        word_segmented_text = underthesea.word_tokenize(text)
        ngunghia.append(word_segmented_text)
    print(ngunghia)
    ngunghia(Tieude)
    ngunghia(Theloai)
    ngunghia(Mota)
    ngunghia(Noidung)
```

```

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: underthesea in /usr/local/lib/python3.8/dist-packages (6.1.4)
Requirement already satisfied: nltk in /usr/local/lib/python3.8/dist-packages (from underthesea) (3.7)
Requirement already satisfied: python-crfsuite>0.9.6 in /usr/local/lib/python3.8/dist-packages (from underthesea) (0.9.9)
Requirement already satisfied: joblib in /usr/local/lib/python3.8/dist-packages (from underthesea) (1.2.0)
Requirement already satisfied: requests in /usr/local/lib/python3.8/dist-packages (from underthesea) (2.25.1)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.8/dist-packages (from underthesea) (1.0.2)
Requirement already satisfied: Click>=6.0 in /usr/local/lib/python3.8/dist-packages (from underthesea) (8.1.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.8/dist-packages (from underthesea) (4.64.1)
Requirement already satisfied: underthesea-core==1.0.0 in /usr/local/lib/python3.8/dist-packages (from underthesea) (1.0.0)
Requirement already satisfied: PyYAML in /usr/local/lib/python3.8/dist-packages (from underthesea) (6.0)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.8/dist-packages (from nltk->underthesea) (2022.6.2)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.8/dist-packages (from requests->underthesea) (2.10)
Requirement already satisfied: chardet<5,>=3.0.2 in /usr/local/lib/python3.8/dist-packages (from requests->underthesea) (4.0.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.8/dist-packages (from requests->underthesea) (2022.12.7)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.8/dist-packages (from requests->underthesea) (1.26.14)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.8/dist-packages (from scikit-learn->underthesea) (3.1.0)
Requirement already satisfied: scipy>=1.1.0 in /usr/local/lib/python3.8/dist-packages (from scikit-learn->underthesea) (1.7.3)
Requirement already satisfied: numpy>=1.14.6 in /usr/local/lib/python3.8/dist-packages (from scikit-learn->underthesea) (1.22.4)
[[['', 'Đừng', 'lắm', 'me', 'cầu', '', 'bất ngờ', 'tàng', 'số', 'tập'], ['Ngành', 'chăn nuôi', 'và', 'bài toán', 'vượt', 'khó', 'trong', 'quý', 'I', '/', '2023'], ['bất ngờ', 'người',
['Giải trí'], ['Kinh Doanh'], ['Thể thao'], ['Giải trí'], ['Sức khỏe'], ['Thể thao'], ['Sức khỏe'], ['Pháp luật'], ['Thời sự'], ['Kinh Doanh'], ['Pháp luật'], ['Giải trí'], ['Thời sự']
['Đại diện', 'VFC', 'cho', 'biết', 'Đừng', 'lắm', 'me', 'cầu', '', 'không', 'đừng', 'lại', 'ò', 'tập', '24', 'như', 'dự kiến', 'ban đầu', '.'], ['Bất chấp', 'những', 'thách thức',
IOPub data rate exceeded.
The notebook server will temporarily stop sending output
to the client in order to avoid crashing it.
To change this limit, set the config variable
`--NotebookApp.iopub_data_rate_limit`.

Current values:
NotebookApp.iopub_data_rate_limit=1000000.0 (bytes/sec)
NotebookApp.rate_limit_window=3.0 (secs)

```

Activate Windows

## Hình 4.9: Kết quả đoạn mã chuyển sang dạng ngữ nghĩa

Giải thích:

- Đầu tiên, thư viện underthesea được cài đặt bằng lệnh `!pip install underthesea`.
- Sau đó, đọc dữ liệu từ tệp `VietNamNet1.xlsx` và lưu vào dataframe `dt`. Các cột của dataframe được chọn là Tiêu đề, Thể loại, Mô tả và Nội dung.
- Sau đó định nghĩa hàm `ngunghia(str)` để phân tích ngữ nghĩa của các câu trong các cột của DataFrame `dt`.
- Trong hàm `ngunghia(str)`, đầu tiên khai báo một danh sách rỗng `ngunghia` để lưu trữ kết quả ngữ nghĩa của các câu. Sau đó, lặp qua các phần tử trong đầu vào `str`, với mỗi phần tử là một câu, tiến hành phân tích ngữ nghĩa bằng cách sử dụng hàm `word_tokenize` của thư viện Underthesea, kết quả được lưu vào biến `word_segmented_text`. Cuối cùng, danh sách `ngunghia` được cập nhật bằng kết quả ngữ nghĩa của câu hiện tại. Cuối cùng, hàm `ngunghia` sẽ in ra các danh sách chứa kết quả ngữ nghĩa của các cột trong DataFrame `dt` được truyền vào thông qua đối số `str`.

### 4.1.10. Code chuyển thành vector

```

# Vector
from sklearn.feature_extraction.text import TfidfVectorizer
rd = pd.read_excel('./content/drive/MyDrive/File Excel/vietnamnet.xlsx')
dt = pd.DataFrame(rd, columns = ['Tiêu đề', 'Thể loại', 'Mô tả', 'Nội dung'])
Tieude = dt['Tiêu đề']
Theloai = dt['Thể loại']
Mota = dt['Mô tả']
Noidung = dt['Nội dung']
vectorizer = TfidfVectorizer()
def vector(str):
    a_vector = []
    for doc in str:
        vector = vectorizer.fit_transform([doc]).toarray()
        a_vector.append(vector)
    print(a_vector)
vector(Tieude)
vector(Theloai)
vector(Mota)
vector(Noidung)

```

```

0.24770694, 0.04954139, 0.11889933, 0.00990828, 0.00990828,
0.00990828, 0.00990828, 0.00990828, 0.00990828, 0.00990828,
0.01981656, 0.03963311, 0.00990828, 0.01981656, 0.01981656,
0.00990828, 0.06935794, 0.01981656, 0.01981656, 0.00990828,
0.00990828, 0.02972483, 0.02972483, 0.01981656, 0.00990828,
0.05944967, 0.00990828, 0.0891745, 0.02972483, 0.00990828,
0.02972483, 0.01981656, 0.01981656, 0.07926622, 0.00990828,
0.00990828, 0.23779866, 0.00990828, 0.13871589, 0.03963311,
0.00990828, 0.01981656, 0.00990828, 0.00990828, 0.02972483,
0.00990828, 0.01981656, 0.05944967, 0.00990828, 0.00990828,
0.00990828, 0.03963311, 0.00990828, 0.04954139, 0.00990828,
0.03963311, 0.00990828, 0.00990828, 0.00990828, 0.00990828,
0.10899105, 0.04954139, 0.04954139, 0.06935794, 0.00990828,
0.04954139, 0.03963311, 0.00990828, 0.01981656, 0.04954139,
0.01981656, 0.01981656, 0.01981656, 0.05944967, 0.00990828,
0.01981656, 0.00990828, 0.00990828, 0.00990828, 0.03963311,
0.00990828, 0.02972483, 0.05944967, 0.05944967, 0.24770694,
0.05944967, 0.03963311, 0.01981656, 0.01981656, 0.04954139,
0.02972483, 0.01981656, 0.04954139, 0.02972483, 0.00990828,
0.01981656, 0.00990828, 0.02972483, 0.00990828, 0.01981656,
0.00990828, 0.00990828, 0.03963311, 0.04954139, 0.02972483,
0.16844072, 0.07926622, 0.00990828, 0.04954139, 0.00990828,
0.04954139, 0.02972483, 0.00990828, 0.00990828, 0.00990828,
0.00990828, 0.05944967, 0.04954139, 0.03963311, 0.00990828,
0.0990828, 0.00990828, 0.03963311, 0.03963311, 0.00990828,
0.03963311, 0.13871589, 0.00990828, 0.07926622, 0.00990828,
0.00990828, 0.10899105, 0.01981656, 0.00990828, 0.07926622,
0.00990828, 0.16844072, 0.02972483, 0.00990828, 0.00990828,
0.00990828, 0.00990828, 0.12880761, 0.00990828, 0.01981656,
0.02972483, 0.00990828, 0.02972483, 0.04954139, 0.03963311,
0.02972483, 0.02972483, 0.00990828, 0.01981656, 0.00990828,
0.00990828, 0.07926622, 0.00990828, 0.01981656, 0.03963311,
0.11889933, 0.00990828, 0.04954139, 0.07926622, 0.01981656,
0.0891745, 0.01981656, 0.03963311, 0.01981656, 0.05944967,
0.00990828, 0.02972483, 0.00990828, 0.03963311, 0.02972483,
0.00990828, 0.0891745, 0.00990828, 0.00990828, 0.01981656,
0.00990828, 0.00990828, 0.00990828, 0.00990828]], array([[0.01857274, 0.05571821, 0.01857274, 0.01857274, 0.01857274,
0.01857274, 0.01857274, 0.01857274, 0.01857274, 0.01857274,

```

**Hình 4.10:** Kết quả đoạn mã chuyển sang vector

*Giải thích:*

Đầu tiên, mã đọc dữ liệu từ file excel và lưu vào DataFrame dt với các cột là Tiêu đề, Thể loại, Mô tả và Nội dung. Sau đó, định nghĩa hàm vector(str) để tạo vector TF-IDF cho các câu trong các cột của DataFrame dt.

Trong hàm vector(str), đầu tiên khai báo một danh sách rỗng a\_vector để lưu trữ kết quả vector của các câu. Sau đó, lặp qua các phần tử trong đầu vào str, với mỗi phần tử là một câu, tiến hành tạo vector TF-IDF cho câu hiện tại bằng cách sử dụng hàm fit\_transform của vectorizer, sau đó chuyển đổi kết quả sang mảng và lưu vào biến vector. Cuối cùng, danh sách a\_vector được cập nhật bằng kết quả vector của câu hiện tại.

Và cuối cùng, hàm vector sẽ in ra các danh sách chứa kết quả vector TF-IDF của các cột trong DataFrame dt được truyền vào thông qua đối số str.

#### **4.1.11. Code tìm bài báo giống cụm từ**

##### **❖ Thực nghiệm với One Hot Vector**

```
df = pd.read_excel('../content/drive/MyDrive/Nhóm 5 - Cuối Kỳ/VietNamNet2Nhom5.xlsx')
data = df["Tiêu đề"]
# Tách các từ trong câu thành các token riêng lẻ
#tokens = [nltk.word_tokenize(sentence) for sentence in data]
tokens = [nltk.word_tokenize(sentence) for sentence in data]

# Chuyển đổi danh sách các token thành one-hot vector
mlb = MultiLabelBinarizer()
one_hot = mlb.fit_transform(tokens)

# Nhập cụm từ
#Nhập cụm từ tìm kiếm
query = input("Nhập từ cần tìm: ")

# Xóa số
query_non_num = re.sub('\d+', '', query)
#print(query_non_num)
```

# Xoá ký tự đặc biệt

```
qery_non_num_char= remove_special_characters_vn(query_non_num)
#print(qery_non_num_char)
```

#Chuyển thành chữ thường

```
query_lower = qery_non_num_char.lower()
#print(query_lower)
```

#Chuẩn hoá TV

```
if __name__ == '__main__':
    query_plus = chuan_hoa_dau_cau_tiang_viet(query_lower)
#print(query_plus)
```

#Xoá stopwords

```
query_non_sw = remove_stop_words(query_plus)
```

# Tách cụm từ thành các token riêng lẻ

```
query_tokens = nltk.word_tokenize(query_non_sw)
```

# Tìm kiếm khớp với one-hot vector

```
query_vector = mlb.transform([query_tokens])
matches = np.dot(one_hot, query_vector.T)
matching_indices = np.nonzero(matches)
```

# Xuất kết quả

```
results = []
for i in range(matches.shape[0]):
    if matches[i] > 0:
        match_accuracy = int((matches[i] / len(query_tokens)) * 100)
        match_string = data[i]
        results.append({'title': match_string, 'accuracy': match_accuracy})
```

```

#Sắp xếp 5 tin có độ khớp cao nhất
for i in range(matches.shape[0]):
    if matches[i] > 0:
        match_accuracy = int((matches[i] / len(query_tokens))*100)
        match_string = data[i]
        print(f"Tiêu đề: {match_string} - Độ chính xác: {match_accuracy}
%)
        print(30*'*')

```

```

Nhập từ cần tìm: không bị kháng cáo nếu bị trục xuất khỏi
Tiêu đề: man city kháng cáo trục xuất premier league - Độ chính xác: 100%
*****
Tiêu đề: vụ nữ hướng viên homestay hoàng su phi hiệp dân hại kháng cáo - Độ chính xác: 50%
*****
Tiêu đề: newcastle liệu karius đấu mu kháng cáo thẻ đỏ nick pope - Độ chính xác: 50%
*****
Tiêu đề: bảng xếp hạng premier league man city trục xuất - Độ chính xác: 50%
*****
Tiêu đề: xây dựng đề xuất huy động tỷ đồng xây triệu xã hội - Độ chính xác: 25%
*****

```

**Hình 4.11:** Kết quả của đoạn mã tìm bài báo giống cụm từ

Giải thích: Đầu tiên, dòng lệnh đầu tiên sử dụng thư viện pandas để đọc tệp Excel và lấy dữ liệu từ cột "Tiêu đề". Dòng lệnh thứ hai tách các câu trong cột "Tiêu đề" thành các token riêng lẻ bằng cách sử dụng thư viện Natural Language Toolkit (nltk).

Dòng lệnh tiếp theo sử dụng MultiLabelBinarizer để chuyển đổi danh sách các token thành one-hot vector.

Sau đó, người dùng sẽ nhập cụm từ cần tìm kiếm. Đoạn mã tiếp theo loại bỏ các số, ký tự đặc biệt, chuyển về chữ thường và chuẩn hóa tiếng Việt bằng các hàm được định nghĩa trước đó.

Sau đó, các stopwords được loại bỏ và cụm từ cần tìm kiếm được tách thành các token.

Tiếp theo, đoạn mã sử dụng hàm transform để chuyển đổi các token của câu tìm kiếm thành one-hot vector và tìm các vị trí khớp của câu tìm kiếm với các token trong tệp Excel. Các vị trí khớp được lưu trữ trong biến `matching_indices`.

Cuối cùng, đoạn mã in ra các tiêu đề có chứa cụm từ tìm kiếm với độ chính xác được tính toán bằng cách chia tổng số khớp của câu tìm kiếm cho độ dài của câu tìm kiếm và nhân với 100.

### ❖ Thực nghiệm với TF – IDF Vector

```
# Đọc dữ liệu từ file Excel
rd = pd.read_excel('/content/drive/MyDrive/Colab Notebooks/viet
namnet_non_sw.xlsx')

# Tạo một DataFrame với cột 'Nội dung'
df = pd.DataFrame(rd, columns=['Tiêu đề'])

# Tạo một đối tượng TfidfVectorizer
vectorizer = TfidfVectorizer()
vectorizer.fit(df['Tiêu đề'])

# Chuyển đổi dữ liệu thành ma trận tính năng TF-IDF
data = vectorizer.transform(df['Tiêu đề'])

# Chuyển đổi ma trận thưa thành ma trận dày đặc
data = data.toarray()

#Nhập cụm từ tìm kiếm
query = input("Nhập từ cần tìm: ")

# Xoá số
query_non_num = re.sub('\d+', '', query)
#print(query_non_num)

# Xoá ký tự đặc biệt
query_non_num_char = remove_special_characters_vn(query_non_num)
#print(query_non_num_char)

#Chuyển thành chữ thường
query_lower = query_non_num_char.lower()
#print(query_lower)

#Chuẩn hoá TV
if __name__ == '__main__':
    query_plus = chuan_hoa_dau_cau_tiang_viet(query_lower)
    #print(query_plus)

#Xoá stopwords
query_non_sw = remove_stop_words(query_plus)
```

```

# Chuyển đổi cụm từ tìm kiếm thành đối tượng TfidfVectorizer
query_vector = vectorizer.transform([query_non_sw])

# Chuyển đổi ma trận thưa thành ma trận dày đặc
query_vector = query_vector.toarray()

# In hình dạng của vector truy vấn
print('hình dạng của vector truy vấn:', query_vector.shape)

from sklearn.metrics.pairwise import cosine_similarity

def search(query_vector, data):
    similarity_scores = cosine_similarity(query_vector, data)

    sorted_documents = sorted(range(len(similarity_scores[0])),
                              key=lambda i: similarity_scores[0][i], reverse=True)

    results = set()
    for i in range(min(5, len(sorted_documents))):
        if similarity_scores[0][sorted_documents[i]] > 0 and df
        .iloc[sorted_documents[i]]['Tiêu đề'] not in results:
            print('Tiêu đề {}: vector: {}'.format(i+1, similarity_scores[0][sorted_documents[i]]))
            print(df.iloc[sorted_documents[i]]['Tiêu đề'])
            print('Tỉ lệ phần trăm tương đồng: {:.2f}%\n'.format(cosine_similarity(query_vector, data[sorted_documents[i]].reshape(1,-1))[0][0] * 100))
            results.add(df.iloc[sorted_documents[i]]['Tiêu đề'])
    )

    if len(results) >= 5:
        break

search(query_vector, data)

```



Nhập từ cần tìm: Man City không được kháng cáo nếu bị trục xuất khỏi Premier League  
hình dạng của vector truy vấn: (1, 2008)  
Tiêu đề 1: vector: 1.0  
man city kháng cáo trục xuất premier league  
Tỉ lệ phần trăm tương đồng: 100.00%

Tiêu đề 2: vector: 0.6749784414464572  
bảng xếp hạng premier league man city trục xuất  
Tỉ lệ phần trăm tương đồng: 67.50%

Tiêu đề 3: vector: 0.40844512960446366  
bất ngờ cửa vô địch premier league man city arsenal mu xếp  
Tỉ lệ phần trăm tương đồng: 40.84%

Tiêu đề 4: vector: 0.2261408012821815  
man city hạ gục arsenal đầu bảng  
Tỉ lệ phần trăm tương đồng: 22.61%

Tiêu đề 5: vector: 0.21102799655611398  
newcastle liệu karius đấu mu kháng cáo thẻ đỏ nick pope  
Tỉ lệ phần trăm tương đồng: 21.10%

#### **Hình 4.12:** Kết quả của đoạn mã tìm bài báo giống cụm từ

*Đoạn code này được giải thích như sau:*

Dòng lệnh đầu tiên sử dụng thư viện Pandas để đọc dữ liệu từ tệp Excel và tạo một DataFrame với cột 'Tiêu đề'.

Sau đó khởi tạo một đối tượng TfidfVectorizer và sử dụng phương thức fit để học từ dữ liệu tiêu đề. Sử dụng phương thức transform để chuyển đổi dữ liệu thành ma trận tính năng TF-IDF và chuyển đổi ma trận thưa thành ma trận dày đặc.

Nhập cụm từ tìm kiếm từ người dùng và chuẩn hoá dữ liệu nhập vào bằng các hàm đã được xây dựng ở phía trên.

Sử dụng cosine\_similarity từ thư viện sklearn.metrics.pairwise để tính toán độ tương đồng giữa vector truy vấn và các vector tiêu đề và lấy ra top 5 tiêu đề có độ tương đồng cao nhất. Cuối cùng dùng phương thức iloc để lấy tiêu đề và in kết quả tìm kiếm cho người dùng.

#### **❖ So sánh 2 phương pháp tìm kiếm**

Cả hai đoạn code này đều dùng để tìm kiếm khớp từ trong một bộ dữ liệu văn bản, nhưng có sự khác biệt về phương pháp biểu diễn văn bản và thuật toán tìm kiếm.

- Đoạn code thứ nhất sử dụng TfidfVectorizer để biểu diễn văn bản dưới dạng ma trận TF-IDF. TfidfVectorizer sẽ tính toán giá trị TF-IDF cho mỗi từ trong tập dữ liệu, và sử dụng chúng để biểu diễn văn bản dưới dạng ma trận

số. Sau đó, thuật toán tìm kiếm sử dụng độ đo cosine similarity để tìm kiếm các văn bản tương tự nhất với câu truy vấn.

- Đoạn code thứ hai sử dụng MultiLabelBinarizer để biểu diễn văn bản dưới dạng one-hot vector. Mỗi từ trong tập dữ liệu sẽ được biểu diễn dưới dạng một vector nhị phân với đúng một phần tử có giá trị bằng 1, tương ứng với từ đó. Sau đó, thuật toán tìm kiếm sử dụng tích vô hướng (dot product) giữa one-hot vector của các văn bản trong tập dữ liệu và one-hot vector của câu truy vấn để tìm kiếm các văn bản có chứa các từ trong câu truy vấn.

Về độ phức tạp của 2 đoạn code trên:

- Đoạn code này có 2 vòng lặp, mỗi vòng lặp chạy từ 0 đến một giá trị cố định. Do đó, độ phức tạp của đoạn code này là  $O(n * m)$ , trong đó  $n$  và  $m$  là kích thước của các vòng lặp.
- Đoạn code này triển khai thuật toán tìm kiếm nhị phân trên một mảng đã sắp xếp. Độ phức tạp của thuật toán tìm kiếm nhị phân là  $O(\log n)$ , trong đó  $n$  là số lượng phần tử trong mảng. Do đó, độ phức tạp của đoạn code này cũng là  $O(\log n)$ .

Vậy, độ phức tạp của đoạn code thứ nhất là  $O(n * m)$ , độ phức tạp của đoạn code thứ hai là  $O(\log n)$ .

## KẾT LUẬN

### 1. Kết quả đạt được

- Thực hiện được các bước cơ bản trong quá trình phân tích dữ liệu.
- Crawl thành công 1000 tin tức từ trang web đã chọn trước đó và xuất ra file excel để lưu trữ.
- Nắm được quy trình và cách thức để xử lý dữ liệu, như:
  - Xoá tab HTML, xoá số.
  - Chuyển dữ liệu sang chữ thường.
  - Chuẩn hoá kiểu gõ dấu trong tiếng Việt.
  - Xoá Stopword.
  - Chuyển văn bản thành dạng ngữ nghĩa.
  - Chuyển văn bản thành vector.
  - Tìm kiếm và so khớp một chuỗi trong một danh sách.

### 2. Hướng phát triển của đề tài

Đề tài phân tích dữ liệu trang Web Vietnamnet.vn có thể được áp dụng để phân tích các dữ liệu lớn hơn, từ các trang web lớn của Việt Nam cũng như của quốc tế.

Việc phân tích dữ liệu sẽ được áp dụng vào các bài toán phân tích nhu cầu khách hàng, phân tích xu hướng xã hội, thống kê các dữ liệu cần thiết.

Đề tài phân tích dữ liệu có thể tiến hành phân tích các yếu tố của thị trường, phân tích vi mô, vĩ mô để có thể đưa ra các quyết định đúng đắn trong các lĩnh vực kinh doanh cũng như các lĩnh vực khác.

## TÀI LIỆU THAM KHẢO

### Tiếng Việt

1. Trần Đăng Hưng (2020), Giáo trình lập trình python, NXB Đại Học Sư Phạm.
2. Lê Cảnh Trung (2022), Python dành cho người bắt đầu, NXB Thanh Niên.
3. Nguyễn Quốc Huy, Nguyễn Tất Bảo Thiện (2022), Kỹ thuật lập trình nâng cao, NXB Thanh Niên.

### Website

4. <https://www.w3schools.com/python/default.asp> .Ngày truy cập 02/02/2023.
5. <https://toidicode.com/python-co-ban>. Ngày truy cập 29/01/2023.
6. <https://howkteam.vn/course/lap-trinh-python-co-ban-37>. Ngày truy cập 29/01/2023.