**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP CUỐI KÌ**

**LẬP TRÌNH MẠNG NÂNG CAO**

**ĐỀ TÀI :**

**TRÍCH XUẤT CHỦ ĐỀ TỪ BÀI BÁO TIẾNG VIỆT**

**SINH VIÊN : Nguyễn Vĩnh Đạt**

**MÃ SINH VIÊN : 102170010**

**LỚP : 17T1**

**SINH VIÊN : Trương Quang Hùng**

**MÃ SINH VIÊN : 102170022**

**LỚP : 17T1**

**CBHD : TS. Phạm Minh Tuấn**

**Đà Nẵng, 03/2021**

Mục Lục

[**I.** **Giới thiệu đề tài:** 3](#_Toc66348823)

[**II.** **Cách thức triển khai:** 3](#_Toc66348824)

[**1.** **Thu thập dữ liệu:** 3](#_Toc66348825)

[**2.** **Tiền xử lý dữ liệu:** 3](#_Toc66348826)

[**3.** **Cách xây dựng mô hình huấn luyện:** 3](#_Toc66348827)

[**4.** **Kiểm thử và tỉ lệ nhận dạng:** 4](#_Toc66348828)

[**III.** **Kết quả:** 5](#_Toc66348829)

[**1.** **Các bước chạy chương trình:** 5](#_Toc66348830)

[**2.** **Giao diện Web JSP/Servlet:** 6](#_Toc66348831)

[**IV.** **Tài liệu tham khảo:** 6](#_Toc66348832)

1. **Giới thiệu đề tài:**

Website được viết bằng JSP/Servlet giúp đoán nhận chủ đề của một bài báo bất kì thông qua link dẫn đến bài báo đó bằng cách sử dụng mô hình học máy được xây dựng bằng ngôn ngữ Python.

Sinh viên thực hiện: Nguyễn Vĩnh Đạt 17T1

Trương Quang Hùng 17T1

1. **Cách thức triển khai:**
2. **Thu thập dữ liệu:**

Dữ liệu được thu thập bằng cách Crawl 10000 bài báo thuộc 10 chủ đề khác nhau trên trang vnexpress.net bằng thư viện BeautifulSoup và Newspaper của Python. Việc sử dụng thêm thư viện Newspaper giúp đễ dàng trong việc lấy nội dung của bài báo mà không phải tìm xem nội dung được chứa trong thẻ nào, thuận tiện cho việc lấy nội dung từ nhiều trang đọc báo khác nhau.

1. **Tiền xử lý dữ liệu:**

Nội dung báo sau khi được tải về sẽ trải qua các bước tiền xử lý sau:

* + Chuẩn hóa unicode: các trang báo thường viết bằng nhiều loại unicode khác nhau như Unicode dựng sẵn hay Unicode tổ hợp vì vậy việc đưa tất cả về Unicode tổ hợp dễ dàng cho việc học dữ liệu.
  + Chuẩn hóa cách gõ dấu tiếng việt: đưa hết các cách gõ dấu về dạng cơ bản (ví dụ như “hoá” về lại dạng “hóa”).
  + Tách từ: sử dụng hàm word\_tokenize của thư viện Underthesea giúp tách câu thành các từ, cụm từ có nghĩa thay vì để riêng 1 cách vô nghĩa (ví dụ “so\_sánh” thay vì “so” và “sánh”).
  + Đưa hết về chữ thường: tăng tính đồng bộ dữ liệu.
  + Xóa các ký tự, khoảng trắng thừa: xóa các ký tự như dấu câu, ngoặc câu,…
* Nội dung sau khi được chuẩn hóa sẽ lưu lại vào file “data.txt” cùng với chủ đề để làm dữ liệu train học máy.

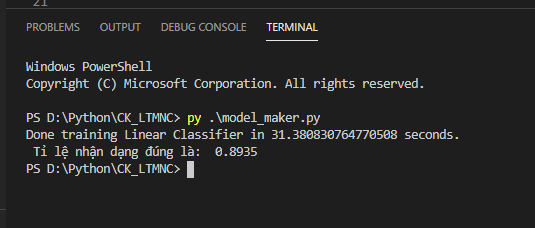
1. **Cách xây dựng mô hình huấn luyện:**

* Bước 1: Dữ liệu qua tiền xử lý được tổng hợp lại qua một bộ đếm, từ đó chọn ra 100 từ có tần số xuất hiện nhiều nhất trên mọi chủ đề để làm stopwords. Lọc bỏ các stopwords ra khỏi dữ liệu giúp học máy đem lại hiệu quả hơn.
* Bước 2: Dữ liệu sau khi lọc stopwords sẽ được tách thành 2 phần: nội dung sẽ được vector hóa bằng hàm TfidfTransformer() của thư viện sklearn, chủ đề sẽ được mã hóa thành nhãn riêng cho từng bộ dữ liệu.
* Bước 3: Sau khi mã hóa dữ liệu sẽ được chi thành 2 bộ: 1 bộ để train học máy và 1 bộ dùng để test mô hình học máy sau xây dựng bằng hàm train\_test\_split của thư viện sklearn.
* Bước 4: Bộ dữ liệu train sau khi tách sẽ được dùng cho mô hình học máy. Ở đây sử dụng mô hình tuyến tính (Linear model) và hàm để xây dựng mô hình là hàm Logistic Regression.
* Bước 5: Mô hình học máy sau khi đã học với bộ dữ liệu train xong sẽ được lưu lại để tiện sau này sử dụng cũng như test lại với bộ dữ liệu test.

1. **Kiểm thử và tỉ lệ nhận dạng:**

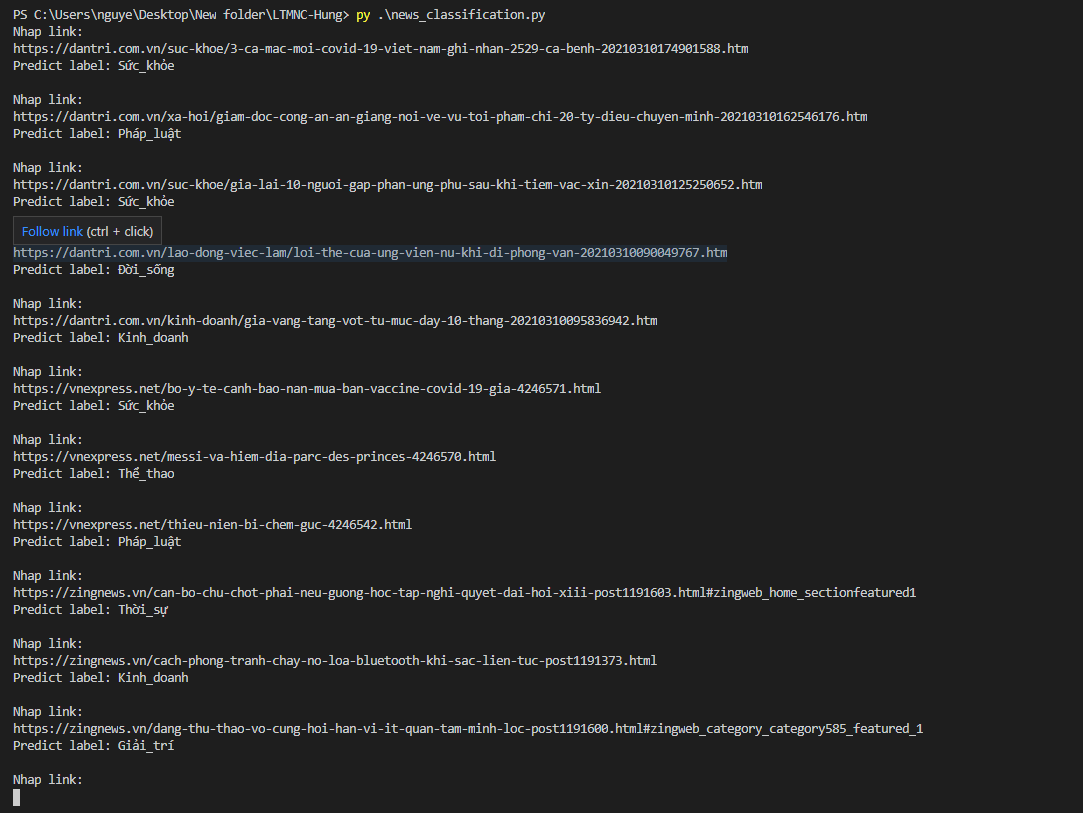
Như đã nói ở trên bộ dữ liệu sau khi xử lý sẽ được chia thành 2 bộ 1 dùng để train và 1 dùng để test. Ở đây ta chia bộ dữ liệu với tỉ lệ test\_percent=0.2 tức với 10000 bộ dữ liệu thuộc 10 nhãn ta sẽ chọn ra 8000 bộ để train và còn lại 2000 bộ để test.

Sau khi mô hình học máy hoàn thành, áp dụng mô hình lên bộ dữ liệu test để nhận dạng ta thu về được tỉ lệ nhận dạng vào khoảng 89.35% và tốn khoảng 30s để xây dựng mô hình (Hình 1).



*Hình 1: Kết quả thời gian xây dựng mô hình và tỉ lệ nhận dạng*

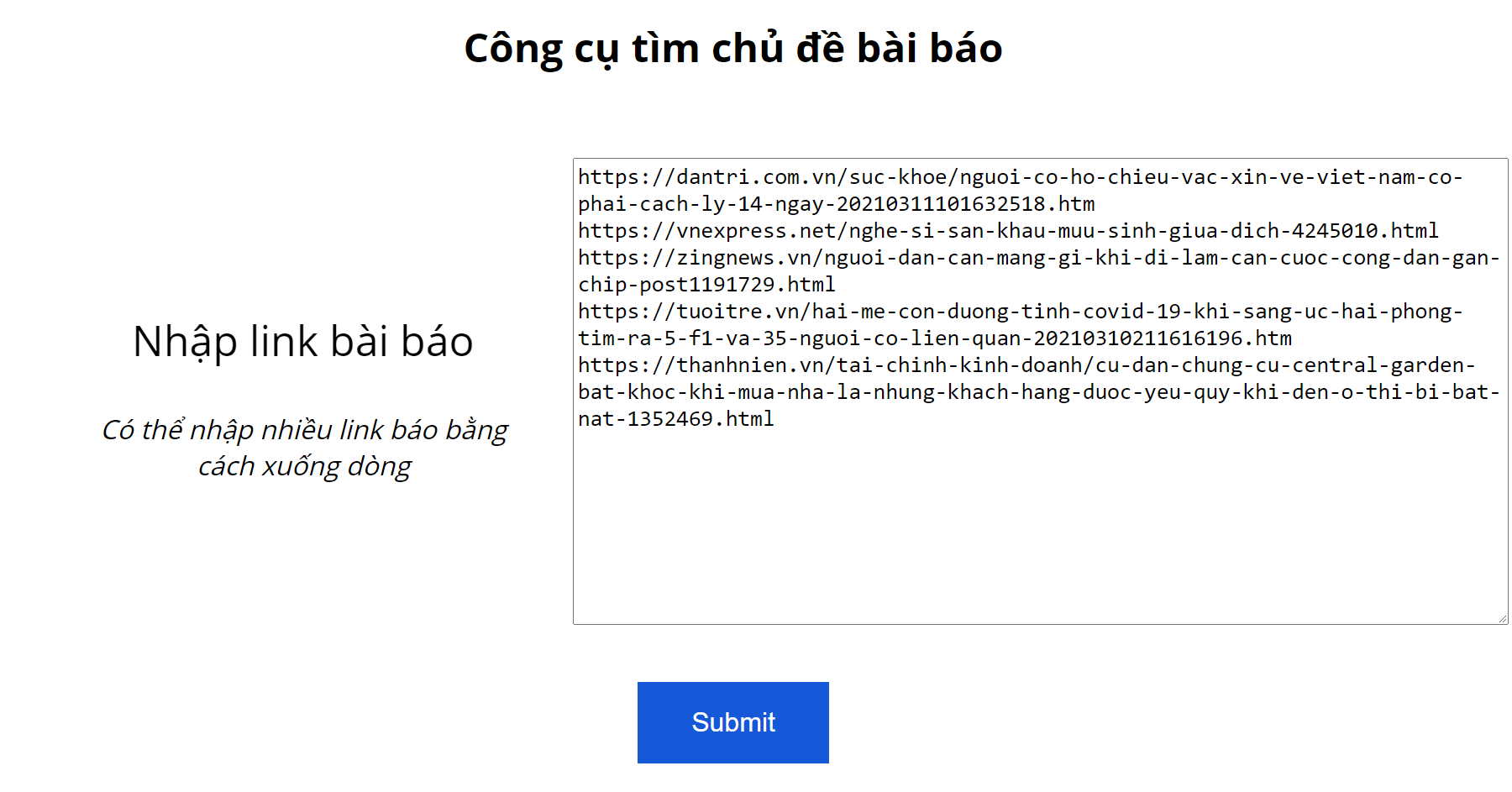
Việc kiểm thử được thực hiện trên nhiều trang báo khác nhau không chỉ với mỗi vnexpress.net đều đem lại kết quả chính xác. Một số bài báo thuộc chủ đề không nằm trong bộ nhãn sẽ trả về kết quả có chủ đề gần giống nhất (Hình 2).

*Hình 2: Kiểm thử trên nhiều trang báo khác nhau*

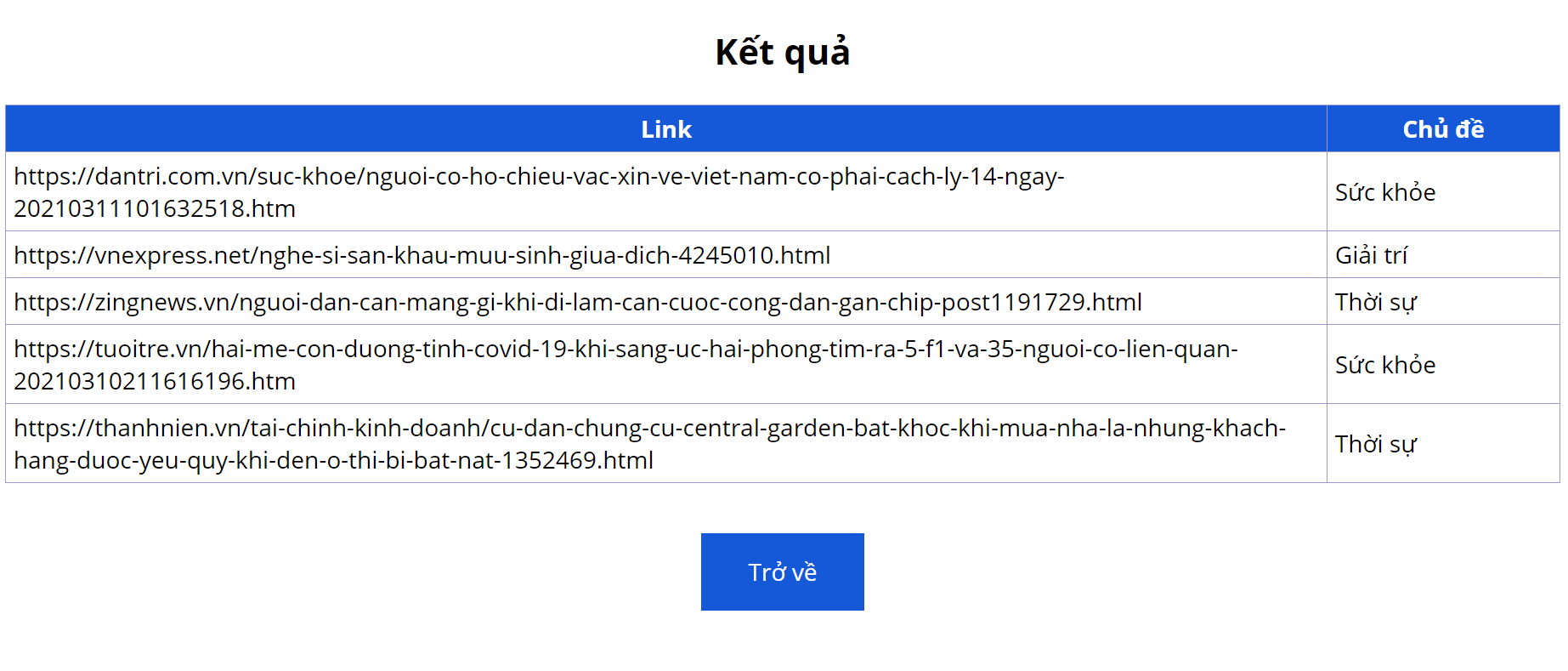
1. **Kết quả:**
2. **Các bước chạy chương trình:**

* Bước 1: Chạy file “data\_crawler.py” để crawl dữ liệu từ mạng về tiền xử lý và lưu vào file “data.txt”.
* Bước 2: Chạy file “stopword\_maker.py” để tạo 1 file “stopwords.txt” chứa bộ từ stopwords.
* Bước 3: Chạy file “remove\_stopword.py” để loại bỏ stopwords khỏi bộ dữ liệu, lưu dữ liệu mới này vào file “data.prep” để dùng cho học máy.
* Bước 4: Chạy file “model\_maker.py” để tạo mô hình học máy được lưu lại để sau này sử dụng và file “labels.txt” chứa chủ đề của các nhãn.
* Bước 5: File “news\_classification.py” sẽ sử dụng model vừa xây dựng ở bước trên để nhận dạng chủ đề bài báo từ link nhận được từ Web JSP/Servlet.

1. **Giao diện Web JSP/Servlet:**

****

*Hình 3: Giao diện nhập đường link bài báo*

****

*Hình 4: Giao diện hiển thị kết quả*

1. **Tài liệu tham khảo:**

Thực hành phân loại văn bản sử dụng Machine learning:

<https://colab.research.google.com/drive/1d7UZqDkssSEtOml_iRnFKM9pRBtLwCew?usp=sharing>

Bộ các thự viện hỗ trợ của Python: <https://pypi.org/>