**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KHOA HỌC MÁY TÍNH**



**BÁO CÁO ĐỒ ÁN**

**MÔN TRUY VẤN THÔNG TIN ĐA PHƯƠNG TIỆN**

**Đề tài: Truy vấn thông tin trong lĩnh vực hẹp : Bài báo bóng đá**

GVHD: Nguyễn Vinh Tiệp

Nhóm sinh viên thực hiện: 22

1. Nguyễn Hoàng Thắng MSSV: 18521394

2. Phan Quang Tấn MSSV: 18521377

□□ Tp. Hồ Chí Minh, 12/2020 □□

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

**Mục Lục**

[**CHƯƠNG 1 : LÝ DO CHỌN ĐỀ TÀI , HIỆN TRẠNG** 4](#_Toc61717455)

[**1.** **Đề tài :** 4](#_Toc61717456)

[**2.** **Lý do chọn đề tài :** 4](#_Toc61717457)

[**3.** **Hiện trạng :** 4](#_Toc61717458)

[**CHƯƠNG 2 : MÔ TẢ BỘ DỮ LIỆU** 5](#_Toc61717459)

[**CHƯƠNG 3 : CÁC KỸ THUẬT DÙNG TRONG BÀI TOÁN** 8](#_Toc61717460)

[**1.** **Crawl data với BeatifulSoup** 8](#_Toc61717461)

[**2.** **Word\_tokenize của underthesea** 8](#_Toc61717462)

[**3.** **Ma trận Tern – Document với TF-IDF** 10](#_Toc61717463)

[**4.** **Độ tương đồng Cosine** 12](#_Toc61717464)

[**5.** **Chi tiết source** 12](#_Toc61717465)

[**CHƯƠNG 4 : THỰC NGHIỆM VÀ ĐÁNH GIÁ** 13](#_Toc61717466)

[**1.** **Thực nghiệm** 13](#_Toc61717467)

[**2.** **Đánh giá mô hình bằng MAP** 15](#_Toc61717468)

[**3.** **Kết luận** 15](#_Toc61717469)

[**CHƯƠNG 5 : TÀI LIỆU THAM KHẢO** 16](#_Toc61717470)

**BẢNG PHÂN CÔNG CÔNG VIỆC**

|  |  |  |
| --- | --- | --- |
| Họ và Tên | Mssv | Công việc |
| Nguyễn Hoàng Thắng | 18521394 | Xử lý dữ liệu  Xây dựng mô hình truy vấn  File báo cáo |
| Phan Quang Tấn | 18521377 | Crawl dữ liệu  Giao diện  Sile thuyết trình |

# **CHƯƠNG 1 : LÝ DO CHỌN ĐỀ TÀI , HIỆN TRẠNG**

1. **Đề tài :**

Truy vấn thông tin liên quan đến bài báo bóng đá

1. **Lý do chọn đề tài :**

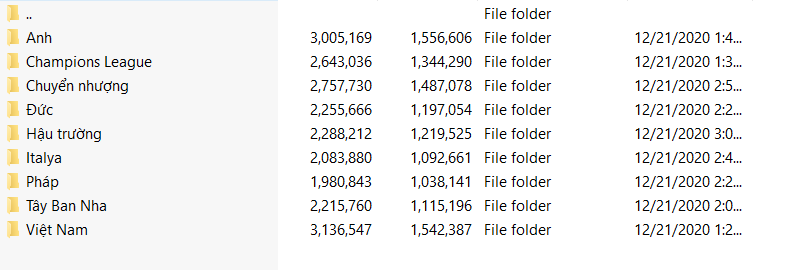
Ngày nay, với việc bùng nổ của thời đại công nghệ thông tin kéo theo đó là việc nhu cầu tìm kiếm thông tin cũng ngày càng phát triển. Trong đó, người dùng thường tìm kiếm các thông tin để tìm hiểu , cập nhật tin tức ,… của một lĩnh vực nhất định mà họ đang quan tâm. Việc bóng đá là môn thể thao “ Vua ” đã được mọi người công nhận từ lâu , chính vì thế mà việc truy vấn các thông tin liên quan đến nó cũng không nằm trong trường hợp ngoại lệ. Tạo ra một công cụ giúp người dùng có thể tìm kiếm nhanh , chính xác các bài báo liên quan đến các giải đấu trong nước và quốc tế là mục tiêu của đề tài này.

1. **Hiện trạng :**

Hiện nay, hầu hết các trang web tin tức đều cập nhật các bài báo liên quan đến thể thao nói chung và bóng đá nói riêng. Và cũng có nhiều trang web chuyên về lĩnh vực bóng đá như : bongdaplus.vn , bongda.com.vn,… hay thậm chí bạn không cần vào bất kì web nào chỉ cần tìm kiếm thông tin trên Google thì kết quả tìm kiếm của nó cũng có thể đáp ứng được các nhu cầu của bạn. Ở đề tài này , chúng tôi không xây dựng một hệ thống tìm kiếm có thể vượt trên các trang web bóng đá hay trên cả Google , mà chỉ dừng lại ở việc đuổi kịp thời gian tìm kiếm hay độ chính xác của các trang web về bóng đá.

# **CHƯƠNG 2 : MÔ TẢ BỘ DỮ LIỆU**

Dữ liệu gồm có 11937 bài viết là các giải vô định của các khu vực trên thế giới cùng với các bài báo có liên quan đến bóng đá được lấy từ trang bongda.com.vn



Trong đó:

+ Anh : 1549 bài viết

+ Champion league : 1319 bài viết

+ Đức : 1318 bài viết

+ Italy : 1164 bài viết

+ Pháp : 1137 bài viết

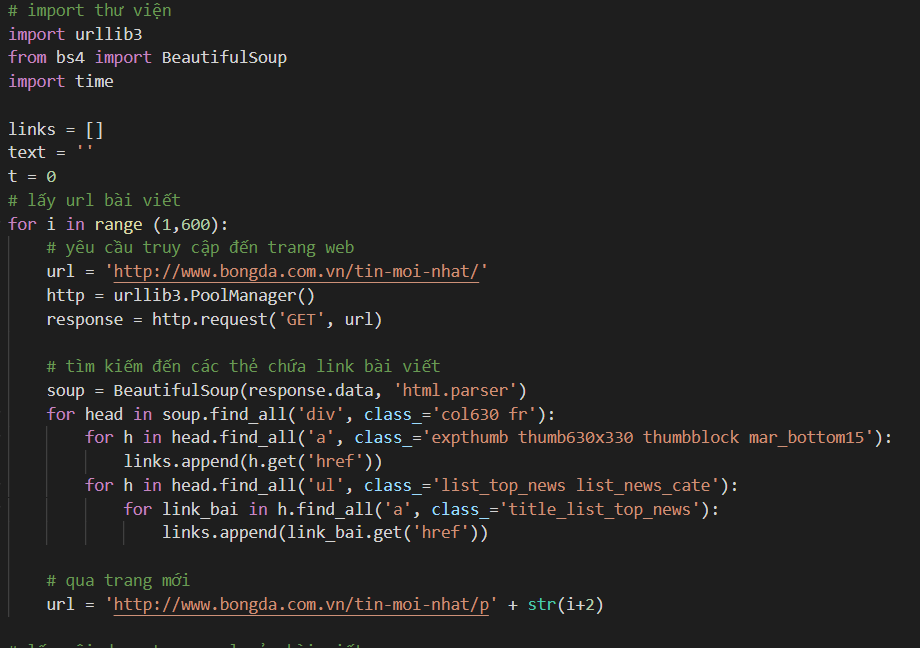
+ Tây Ban Nha : 1047 bài viết

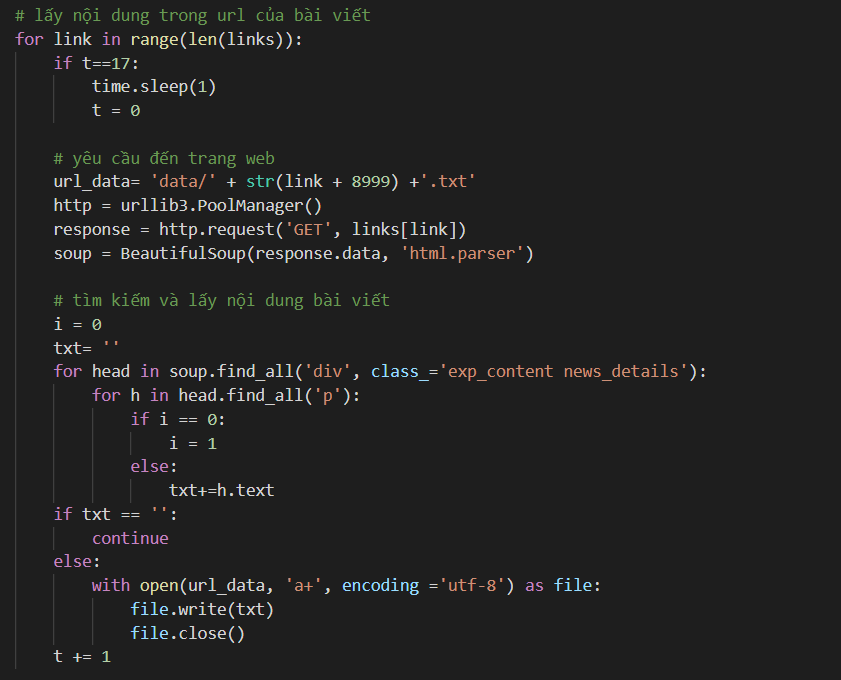
+ Việt Nam : 1480 bài viết

+ Chuyển nhượng : 1566 bài viết

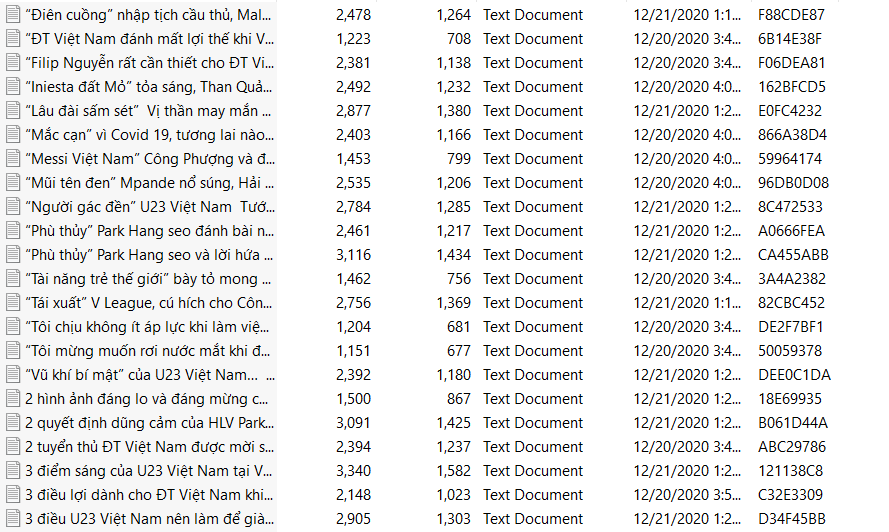
+ Hậu trường : 1357 bài viết

Các bài báo được lấy về thông qua chương trình crawler một cách tự động





Sau khi các bài viết được “ crawl “ từ trang web xuống sẽ được lưu dướng dạng file txt và được đặt tên file là tiêu đề của bài viết.

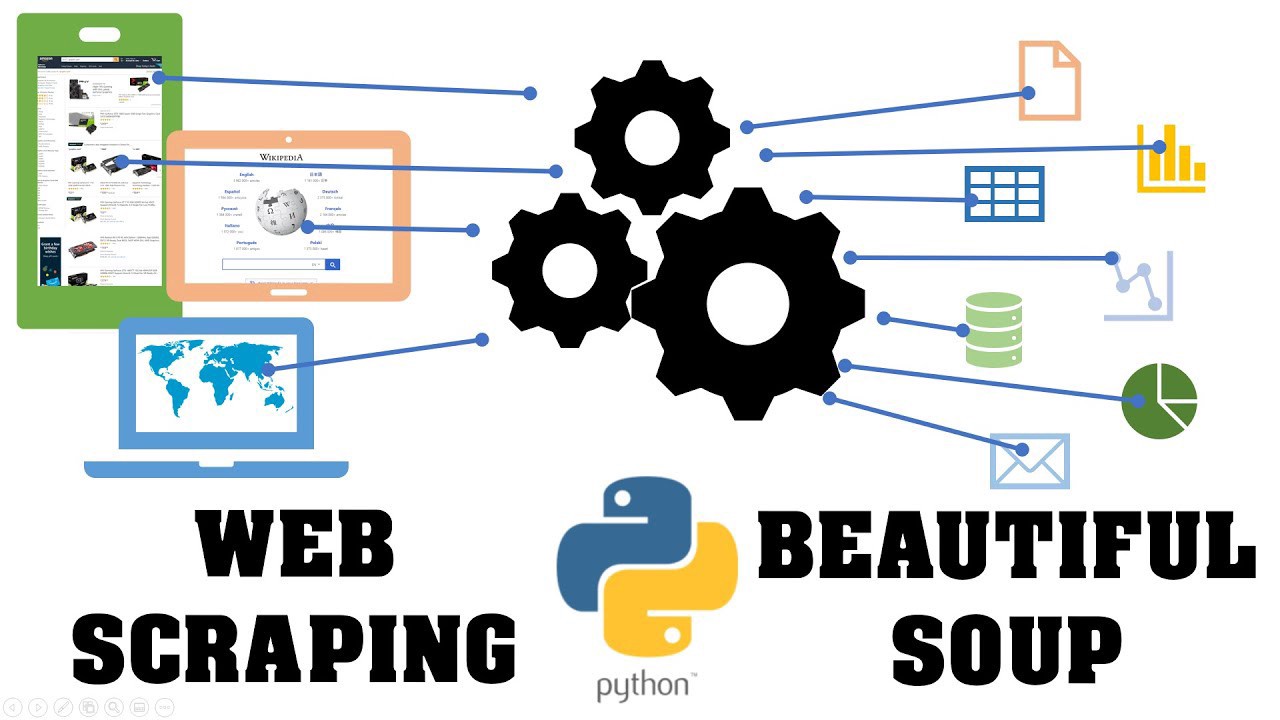


Sau cùng dữ liệu sẽ được nén thành file zip và được lưu trữ trên github

# **CHƯƠNG 3 : CÁC KỸ THUẬT DÙNG TRONG BÀI TOÁN**

1. **Crawl data với BeatifulSoup**

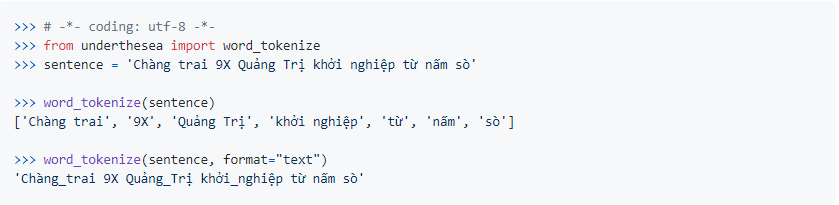
BeautifulSoup là một thư viện Python dùng để lấy dữ liệu ra khỏi các file HTML và XML. Nó hoạt động cùng với các parser (trình phân tích cú pháp) cung cấp cho bạn các cách để điều hướng, tìm kiếm và chỉnh sửa trong parse tree (cây phân tích được tạo từ parser). Nhờ các parser này nó đã giúp các lập trình viên tiết kiệm được nhiều giờ làm việc.

****

1. **Word\_tokenize của underthesea**

- Underthesea là một bộ mô-đun với mã nguồn mở sử dụng Python,nó bao gồm các bộ dữ liệu và các hướng dẫn dùng để hỗ trợ nghiên cứu và phát triển trong Xử lý ngôn ngữ tự nhiên trong Tiếng Việt. 

- Word\_tokenize là một trong các hàm của underthesea , dùng để tách các tern có nghĩa trong tiếng việt.



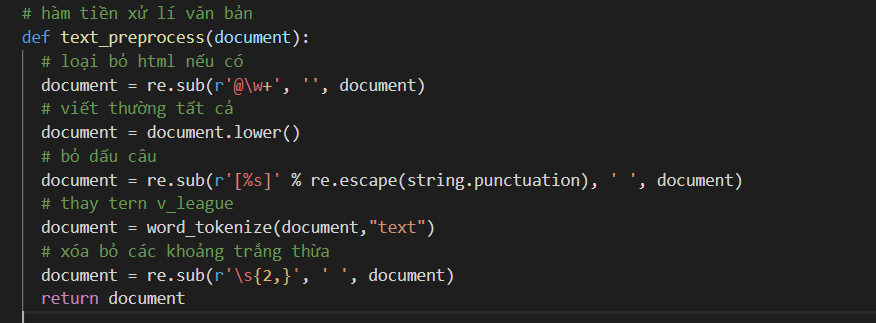
- Word\_tokenize có 2 tham số :

+ sentence : chứa nội dung dạng string mà bạn muốn tách các tern trong đó.

+ format :

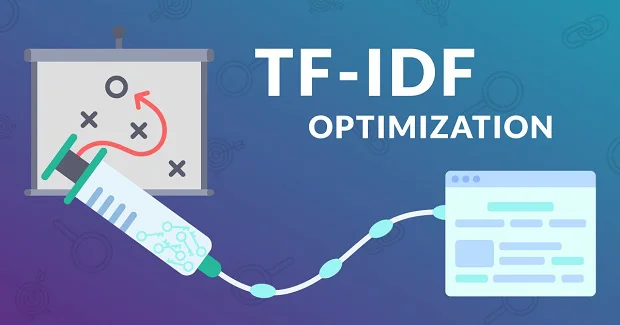
* None : các tern trả về được phân cách nhau bằng dấu cách
* Text : các tern trả về được phân cách nhau bằng dấu gạch dưới

- Trong bài toán : word\_tokenize được sử dụng cùng với regular expression để tiền sử lí văn bản trước khi chúng được tính TF-IDF



1. **Ma trận Tern – Document với TF-IDF**

TF-IDF (Term Frequency – Inverse Document Frequency) là 1 kĩ thuật sử dụng trong khai phá dữ liệu văn bản. Trọng số này được sử dụng để đánh giá tầm quan trọng của một từ trong một văn bản. Giá trị cao thể hiện độ quan trọng cao và nó phụ thuộc vào số lần từ xuất hiện trong văn bản nhưng bù lại bởi tần suất của từ đó trong tập dữ liệu. Một vài biến thể của tf-idf thường được sử dụng trong các hệ thống tìm kiếm như một công cụ chính để đánh giá và sắp xếp văn bản dựa vào truy vấn của người dùng. Tf-idf cũng được sử dụng để lọc những từ stopwords trong các bài toán như tóm tắt văn bản và phân loại văn bản.

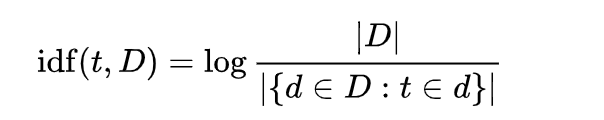


TF: Term Frequency(Tần suất xuất hiện của từ) là số lần từ xuất hiện trong văn bản. Vì các văn bản có thể có độ dài ngắn khác nhau nên một số từ có thể xuất hiện nhiều lần trong một văn bản dài hơn là một văn bản ngắn. Như vậy, term frequency thường được chia cho độ dài văn bản( tổng số từ trong một văn bản).

Trong đó:

* tf(t, d): tần suất xuất hiện của từ t trong văn bản d
* f(t, d): Số lần xuất hiện của từ t trong văn bản d
* max({f(w, d) : w ∈ d}): Số lần xuất hiện của từ có số lần xuất hiện nhiều nhất trong văn bản d

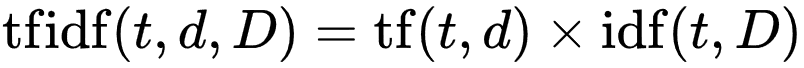
**IDF**: Inverse Document Frequency(Nghịch đảo tần suất của văn bản), giúp đánh giá tầm quan trọng của một từ . Khi tính toán TF , tất cả các từ được coi như có độ quan trọng bằng nhau. Nhưng  một số từ như “is”, “of” và “that” thường xuất hiện rất nhiều lần nhưng độ quan trọng là không cao. Như thế chúng ta cần giảm độ quan trọng của những từ này xuống.



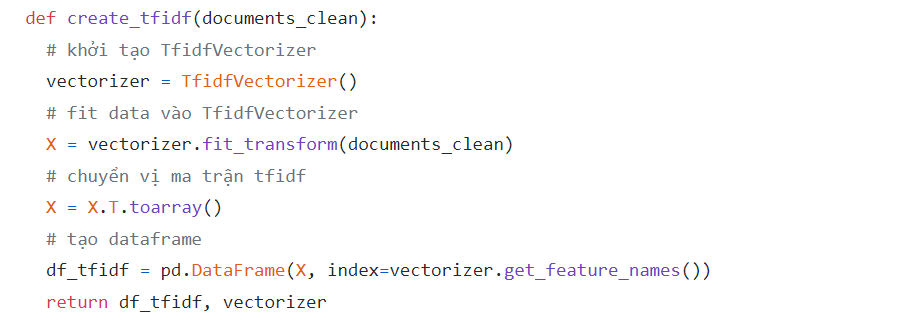
Trong đó:

* idf(t, D): giá trị idf của từ t trong tập văn bản
* |D|: Tổng số văn bản trong tập D
* |{d ∈ D : t ∈ d}|: thể hiện số văn bản trong tập D có chứa từ t.

Tính TF-IDF thông qua công thức :

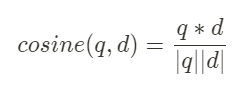


- Bài toàn sử dụng hàm TfidfVectorizer() của thư viện sklearn để tính ma trận TF-IDF của từng tern với bài viết.



1. **Độ tương đồng Cosine**

Sử dụng độ tương đồng cosine để tìm ra văn bản nào chứa nội dung gần giống với câu truy vấn nhất thông qua vector của câu query và vector của bài viết.



Sử dụng thư viện numpy để hỗ trợ tính độ tương đồng cosin:



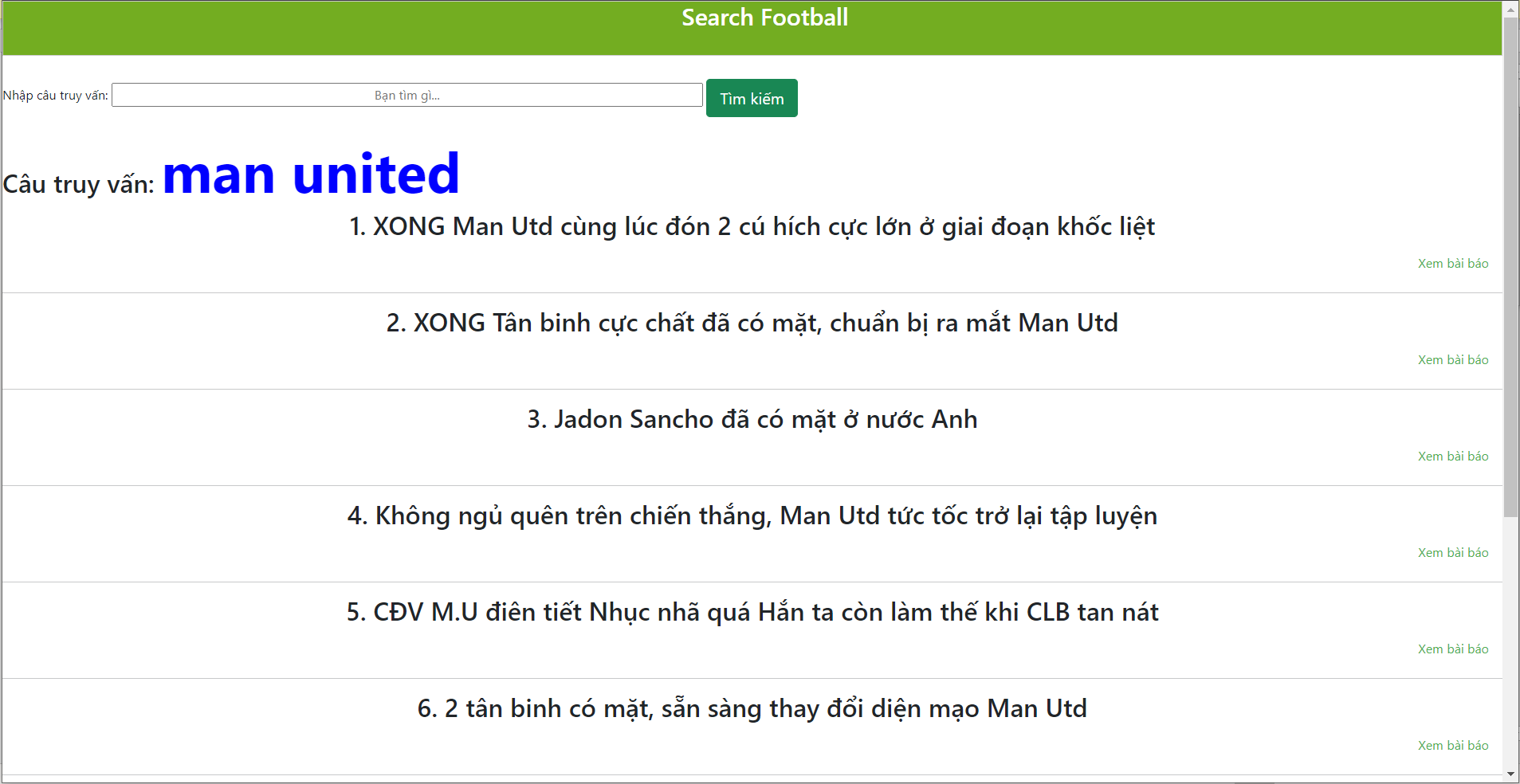
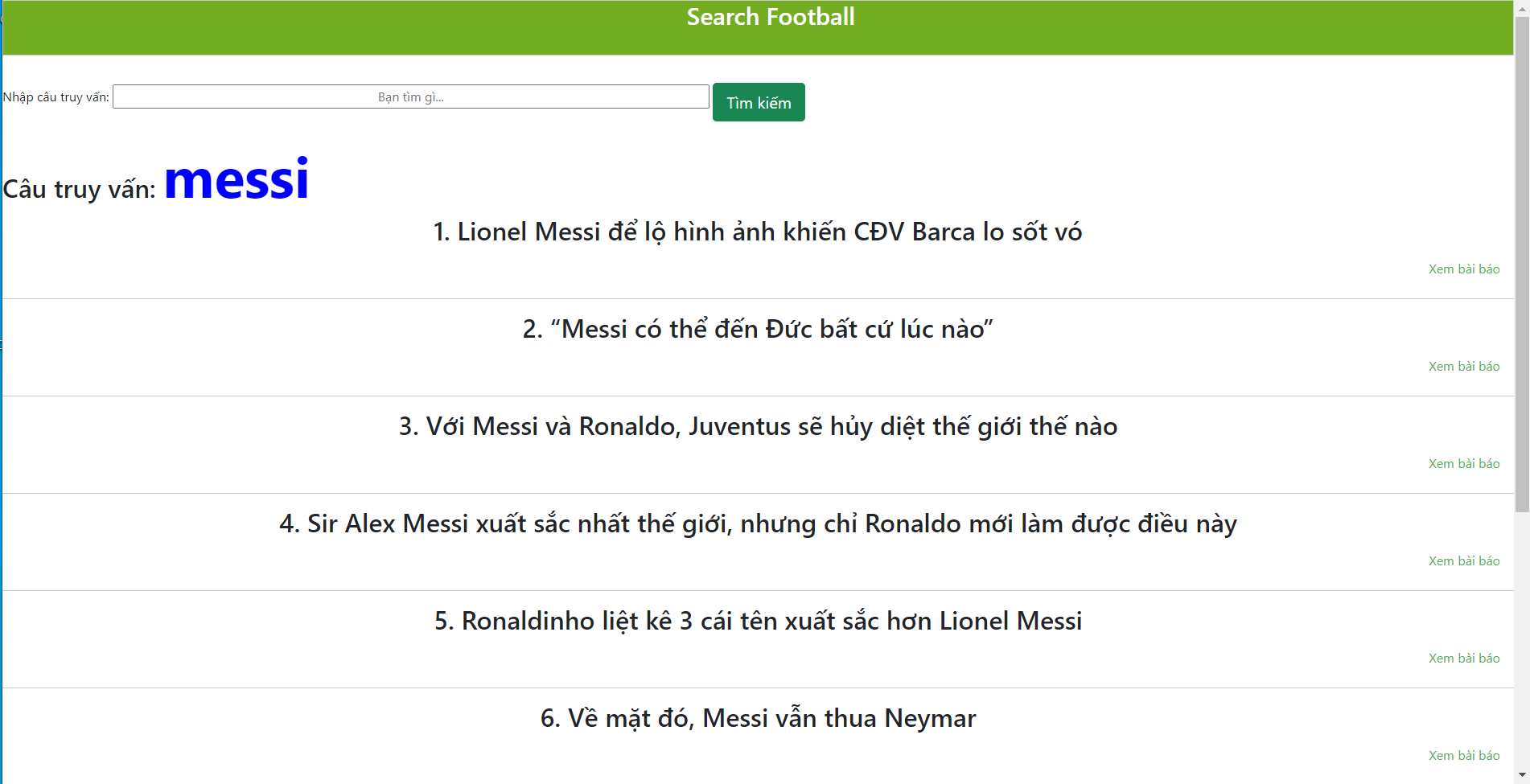
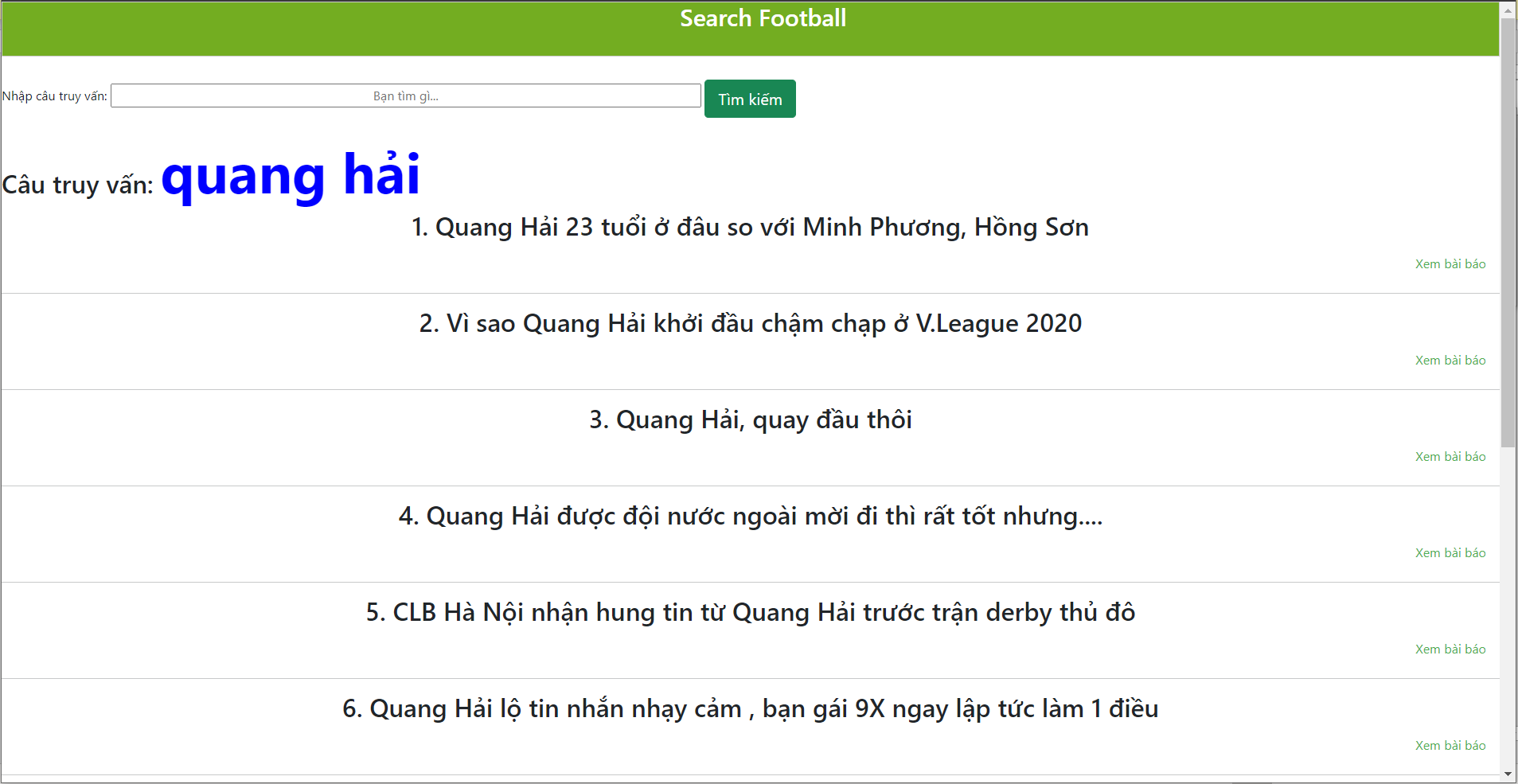
1. **Chi tiết source**

Link github : https://github.com/Hoangthang017/CS336.L11/tree/master/Do\_an\_cuoi\_ki

# **CHƯƠNG 4 : THỰC NGHIỆM VÀ ĐÁNH GIÁ**

1. **Thực nghiệm**

Tiến hành thực nghiệm trên 3 câu truy vấn ngẫu nhiên:



Có thể thấy đối với 3 câu truy vấn đầu không chứa các từ đồng nghỉ thì kết quả trả về cực kì chính xác đúng với nội dung chúng ta cần tìm.

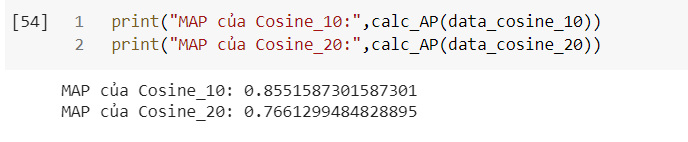
Còn đối với câu truy vấn cuối kết quả trả về vẫn chưa được chính xác. Lý do:

+ Do trong câu chứa các từ đồng nghĩa trong lĩnh vực truy vấn bóng đá nên khi dùng TF-IDF thì bài toán không thể nhận biết được

=> ***Giải pháp :*** có thể sử dụng mô hình máy học để nhận biết các từ đồng nghĩa trong các bài báo

1. **Đánh giá mô hình bằng MAP**

Đánh giá mô hinh thông qua câu truy vấn : Viettel vô địch V-league



Thông qua việc đánh giá ta có thể thấy đối với 10 hay 20 kết quả đầu đều trả về cho kết quả chính xác đương tối cao 86% và 77%.

1. **Kết luận**

Mô hình truy vấn cho kết quả khá tốt . Tuy nhiên cần phải phát triển thêm để mô hình có thể chính xác hơn , hiểu được các từ đồng nghĩa trong tiếng Việt.

# **CHƯƠNG 5 : TÀI LIỆU THAM KHẢO**

- <https://towardsdatascience.com/create-a-simple-search-engine-using-python-412587619ff5>

- <https://www.geeksforgeeks.org/implementing-web-scraping-python-beautiful-soup/>

- <https://www.freecodecamp.org/news/how-to-build-a-web-application-using-flask-and-deploy-it-to-the-cloud-3551c985e492/>