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
<u>Đề tài:</u> 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

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

2. **Lý do chọn đề tài :**

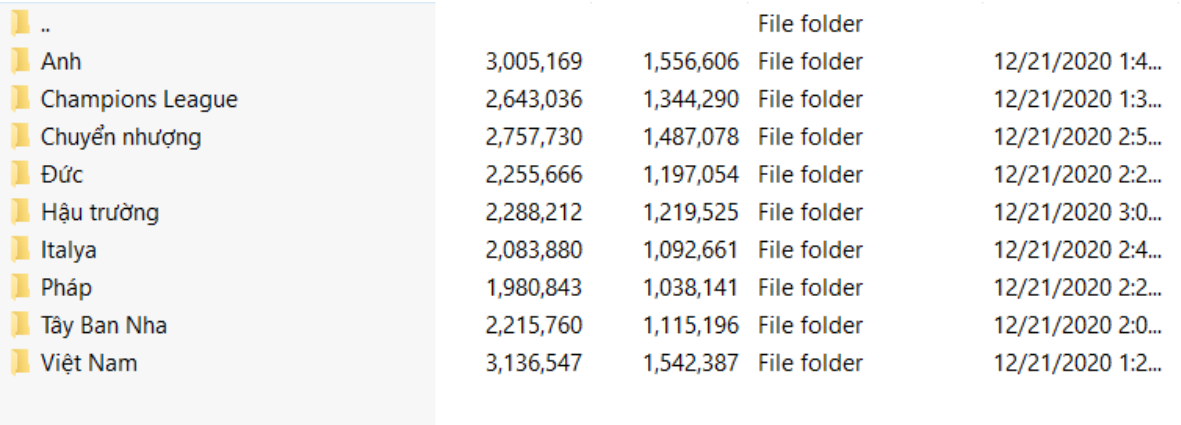
   Ngày nay, với việc bùng nổ của thời đại công nghệ thông tin kéo theo đó là việc nhu cầu tìm kiếm thông tin cũng ngày càng phát triển. Trong đó, người dùng thường tìm kiếm các thông tin để tìm hiểu , cập nhật tin tức ,… của một lĩnh vực nhất định mà họ đang quan tâm. Việc bóng đá là môn thể thao " Vua " 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.

3. **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



..			File folder	
Anh	3,005,169	1,556,606	File folder	12/21/2020 1:4...
Champions League	2,643,036	1,344,290	File folder	12/21/2020 1:3...
Chuyển nhượng	2,757,730	1,487,078	File folder	12/21/2020 2:5...
Đức	2,255,666	1,197,054	File folder	12/21/2020 2:2...
Hậu trường	2,288,212	1,219,525	File folder	12/21/2020 3:0...
Italya	2,083,880	1,092,661	File folder	12/21/2020 2:4...
Pháp	1,980,843	1,038,141	File folder	12/21/2020 2:2...
Tây Ban Nha	2,215,760	1,115,196	File folder	12/21/2020 2:0...
Việt Nam	3,136,547	1,542,387	File folder	12/21/2020 1:2...

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

```python
# import thư viện
import urllib3
from bs4 import BeautifulSoup
import time

links = []
text = ''
t = 0
# lấy url bài viết
for i in range (1,600):
    # yêu cầu truy cập đến trang web
    url = 'http://www.bongda.com.vn/tin-moi-nhat/'
    http = urllib3.PoolManager()
    response = http.request('GET', url)

    # tìm kiếm đến các thẻ chứa link bài viết
    soup = BeautifulSoup(response.data, 'html.parser')
    for head in soup.find_all('div', class_='col630 fr'):
        for h in head.find_all('a', class_='expthumb thumb630x330 thumbblock mar_bottom15'):
            links.append(h.get('href'))
        for h in head.find_all('ul', class_='list_top_news list_news_cate'):
            for link_bai in h.find_all('a', class_='title_list_top_news'):
                links.append(link_bai.get('href'))

    # qua trang mới
    url = 'http://www.bongda.com.vn/tin-moi-nhat/p' + str(i+2)
```

```python
# lấy nội dung trong url của bài viết
for link in range(len(links)):
    if t==17:
        time.sleep(1)
        t = 0

    # yêu cầu đến trang web
    url_data= 'data/' + str(link + 8999) +'.txt'
    http = urllib3.PoolManager()
    response = http.request('GET', links[link])
    soup = BeautifulSoup(response.data, 'html.parser')

    # tìm kiếm và lấy nội dung bài viết
    i = 0
    txt= ''
    for head in soup.find_all('div', class_='exp_content news_details'):
        for h in head.find_all('p'):
            if i == 0:
                i = 1
            else:
                txt+=h.text
    if txt == '':
        continue
    else:
        with open(url_data, 'a+', encoding ='utf-8') as file:
            file.write(txt)
            file.close()
    t += 1
```

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.

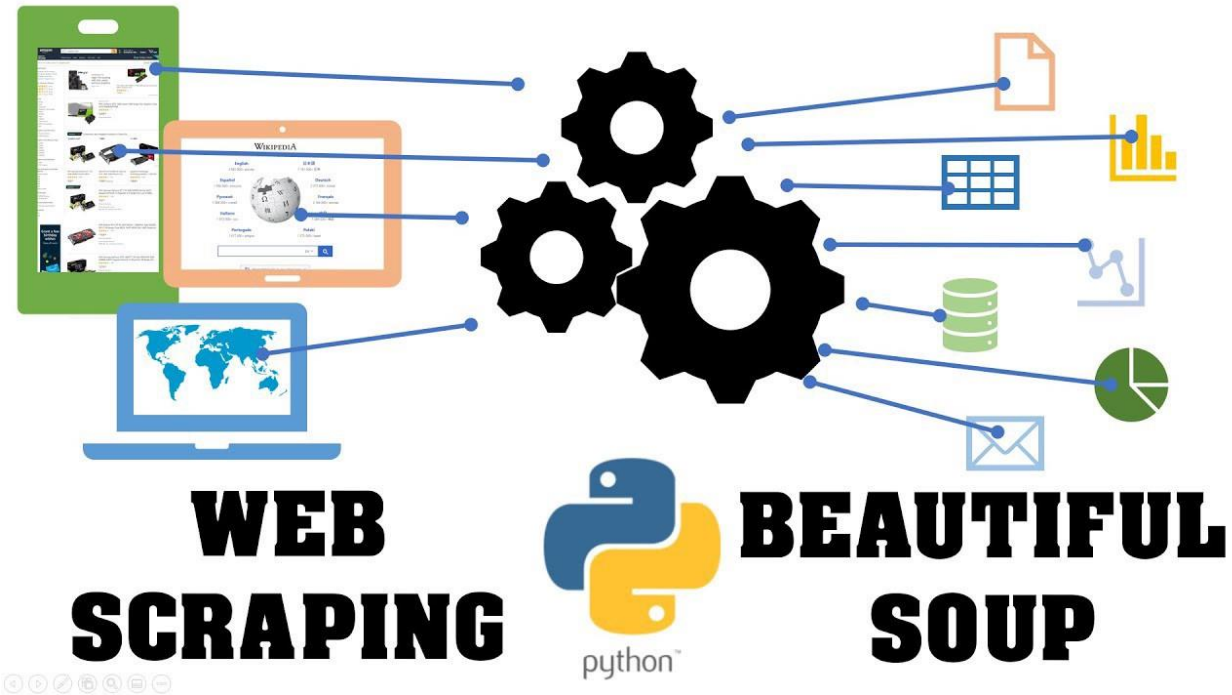
"Điên cuồng" nhập tịch cầu thủ, Mal...	2,478	1,264	Text Document	12/21/2020 1:1...	F88CDE87
"ĐT Việt Nam đánh mất lợi thế khi V...	1,223	708	Text Document	12/20/2020 3:4...	6B14E38F
"Filip Nguyễn rất cần thiết cho ĐT Vi...	2,381	1,138	Text Document	12/20/2020 3:4...	F06DEA81
"Iniesta đất Mỏ" tỏa sáng, Than Quả...	2,492	1,232	Text Document	12/20/2020 4:0...	162BFCD5
"Lâu đài sấm sét" Vị thần may mắn ...	2,877	1,380	Text Document	12/21/2020 1:2...	E0FC4232
"Mắc cạn" vì Covid 19, tương lai nào...	2,403	1,166	Text Document	12/20/2020 4:0...	866A38D4
"Messi Việt Nam" Công Phượng và đ...	1,453	799	Text Document	12/20/2020 4:0...	59964174
"Mũi tên đen" Mpande nổ súng, Hải ...	2,535	1,206	Text Document	12/20/2020 4:0...	96DB0D08
"Người gác đền" U23 Việt Nam Tướ...	2,784	1,285	Text Document	12/21/2020 1:2...	8C472533
"Phù thủy" Park Hang seo đánh bài n...	2,461	1,217	Text Document	12/21/2020 1:2...	A0666FEA
"Phù thủy" Park Hang seo và lời hứa ...	3,116	1,434	Text Document	12/21/2020 1:2...	CA455ABB
"Tài năng trẻ thế giới" bày tỏ mong ...	1,462	756	Text Document	12/20/2020 3:4...	3A4A2382
"Tái xuất" V League, cú hích cho Côn...	2,756	1,369	Text Document	12/21/2020 1:1...	82CBC452
"Tôi chịu không ít áp lực khi làm việ...	1,204	681	Text Document	12/20/2020 3:4...	DE2F7BF1
"Tôi mừng muốn rơi nước mắt khi đ...	1,151	677	Text Document	12/20/2020 3:4...	50059378
"Vũ khí bí mật" của U23 Việt Nam... ...	2,392	1,180	Text Document	12/21/2020 1:2...	DEE0C1DA
2 hình ảnh đáng lo và đáng mừng c...	1,500	867	Text Document	12/21/2020 1:2...	18E69935
2 quyết định dũng cảm của HLV Park...	3,091	1,425	Text Document	12/21/2020 1:2...	B061D44A
2 tuyển thủ ĐT Việt Nam được mời s...	2,394	1,237	Text Document	12/20/2020 3:4...	ABC29786
3 điểm sáng của U23 Việt Nam tại V...	3,340	1,582	Text Document	12/21/2020 1:2...	121138C8
3 điều lợi dành cho ĐT Việt Nam khi...	2,148	1,023	Text Document	12/20/2020 3:5...	C32E3309
3 điều U23 Việt Nam nên làm để già...	2,905	1,303	Text Document	12/21/2020 1:2...	D34F45BB

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.



2. 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.

```
>>> # -*- coding: utf-8 -*-
>>> from underthesea import word_tokenize
>>> sentence = 'Chàng trai 9X Quảng Trị khởi nghiệp từ nấm sò'

>>> word_tokenize(sentence)
['Chàng trai', '9X', 'Quảng Trị', 'khởi nghiệp', 'từ', 'nấm', 'sò']

>>> word_tokenize(sentence, format="text")
'Chàng_trai 9X Quảng_Trị khởi_nghiệp từ nấm sò'
```

- 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

```python
# hàm tiền xử lí văn bản
def text_preprocess(document):
  # loại bỏ html nếu có
  document = re.sub(r'@\w+', '', document)
  # viết thường tất cả
  document = document.lower()
  # bỏ dấu câu
  document = re.sub(r'[%s]' % re.escape(string.punctuation), ' ', document)
  # thay tern v_league
  document = word_tokenize(document,"text")
  # xóa bỏ các khoảng trắng thừa
  document = re.sub(r'\s{2,}', ' ', document)
  return document
```

3. 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.

$$\mathrm{idf}(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

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 :

$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \times \mathrm{idf}(t, D)$$

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

```python
def create_tfidf(documents_clean):
  # khởi tạo TfidfVectorizer
  vectorizer = TfidfVectorizer()
  # fit data vào TfidfVectorizer
  X = vectorizer.fit_transform(documents_clean)
  # chuyển vị ma trận tfidf
  X = X.T.toarray()
  # tạo dataframe
  df_tfidf = pd.DataFrame(X, index=vectorizer.get_feature_names())
  return df_tfidf, vectorizer
```

4. **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.

$$cosine(q,d) = \frac{q * d}{|q||d|}$$

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

```python
sim[i] = np.dot(df.loc[:, str(i)].values, q_vec) / np.linalg.norm(df.loc[:, str(i)]) * np.linalg.norm(q_vec)
```

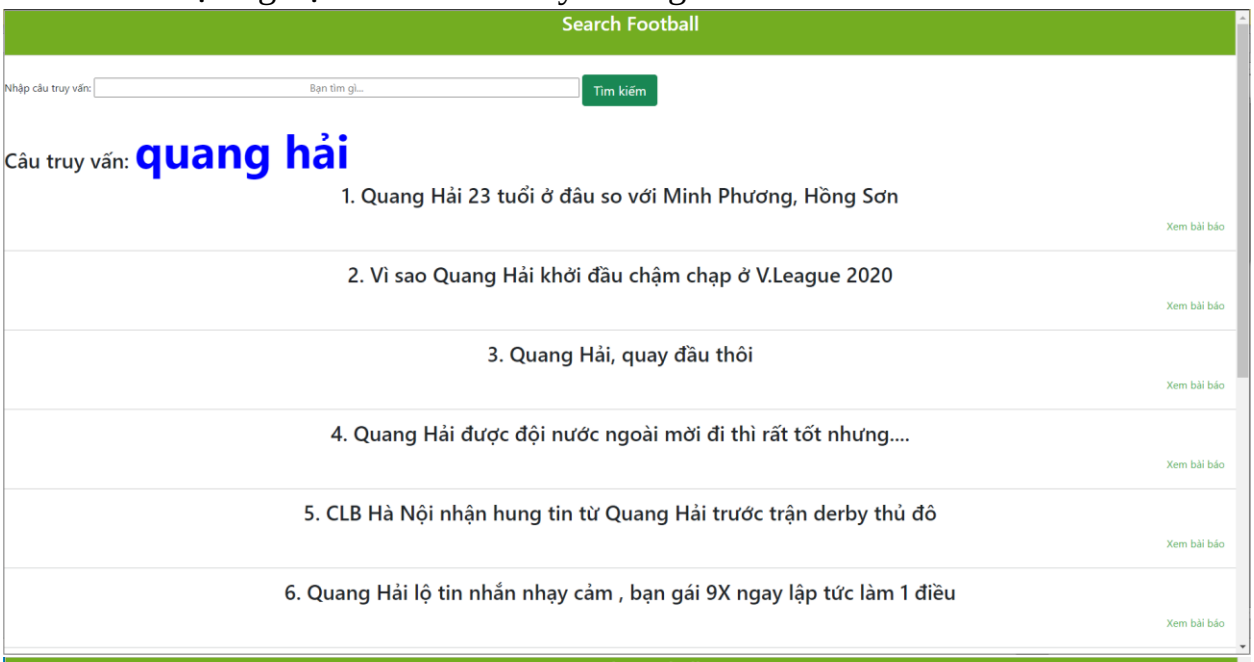
5. **Chi tiết source**

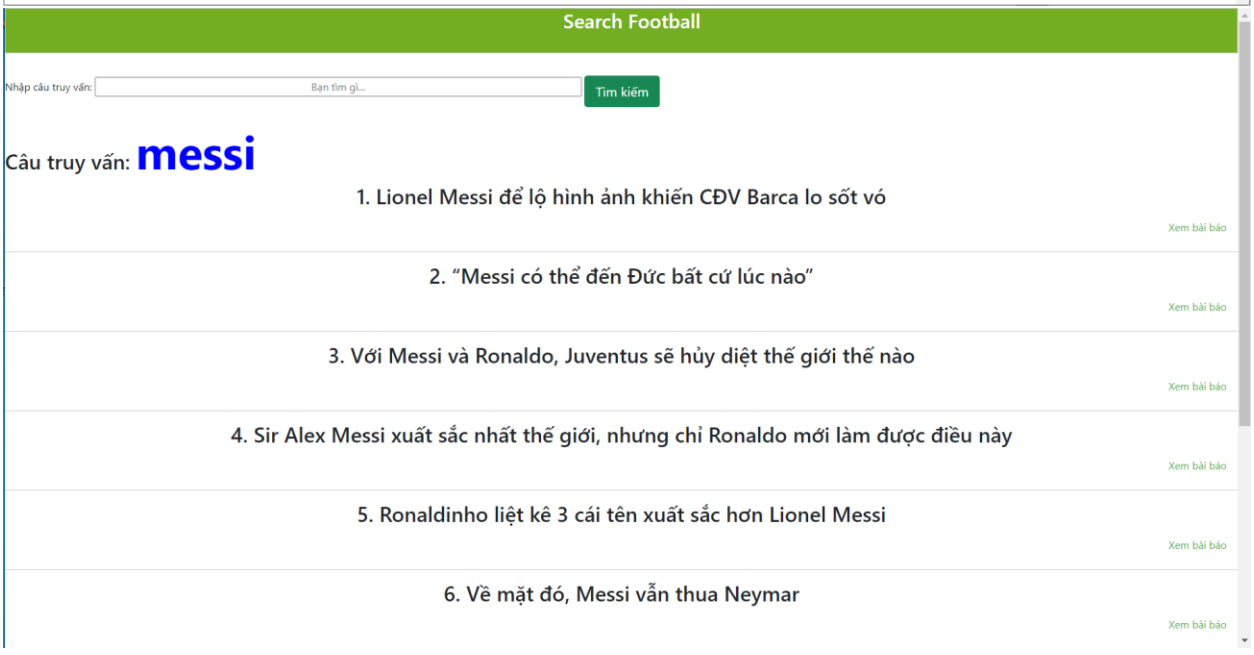
   Link github :
   https://github.com/Hoangthang017/CS336.L11/tree/master/Do_an_cuoi_ki

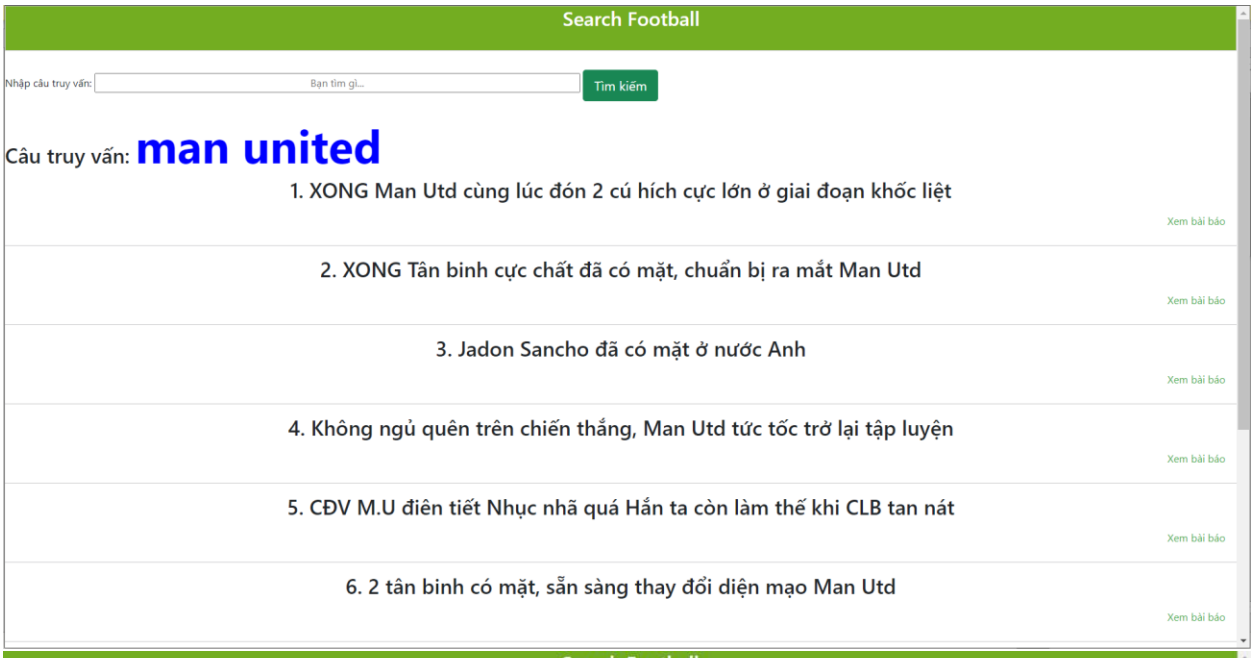
CHƯƠNG 4 : THỰC NGHIỆM VÀ ĐÁNH GIÁ

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

2. **Đánh giá mô hình bằng MAP**

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

```
[54]  1  print("MAP của Cosine_10:",calc_AP(data_cosine_10))
      2  print("MAP của Cosine_20:",calc_AP(data_cosine_20))
```

```
MAP của Cosine_10: 0.8551587301587301
MAP của Cosine_20: 0.7661299484828895
```

   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%.

3. **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/