

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÀI TẬP LỚN

MÔN HỌC: NHẬP MÔN KHOA HỌC DỮ LIỆU

Tên đề tài: Phân tích và gợi ý việc làm trên các trang tuyển dụng

Giảng viên: Vũ Hoài Nam

Sinh viên: Đỗ Công Ban

Mã sinh viên: B19DCCN057

Nhóm học phần: 03

Nhóm bài tập lớn: 07

Hà Nội, Tháng 11 Năm 2022

Mục lục

Mở đầu	3
I. Mô tả	4
1. Giới thiệu đề tài	4
2. Phân tích yêu cầu chức năng của đề tài	4
3. Công nghệ sử dụng	4
II. Phân tích các chức năng chính	4
1. Thu nhập dữ liệu	4
2. Gợi ý việc làm	5
3. Thao tác với giao diện người dùng	7
III. Ưu nhược điểm	9
1. Ưu điểm	9
2. Nhược điểm	9
IV. Khắc phục	9
IV. Kết luận	11

Mở đầu

Trong thời buổi hiện đại ngày nay, công nghệ thông tin cũng như những ứng dụng của nó không ngừng phát triển, lượng thông tin và cơ sở dữ liệu được thu thập và lưu trữ cũng tích lũy ngày một nhiều lên. Con người cũng vì thế mà cần có thông tin với tốc độ nhanh nhất để đưa ra quyết định dựa trên lượng dữ liệu khổng lồ đã có. Vì thế cần phải có những phương pháp để thu thập, phân tích, khai phá dữ liệu một cách nhanh chóng và chính xác nhất.

Hiện nay vấn đề tìm việc của những sinh viên mới ra trường hay những người muốn tìm một công việc mới phù hợp hơn đang ngày càng tăng do đó bắt kịp được nhu cầu đó em quyết định thực hiện đề tài “**Phân tích và gợi ý việc làm trên các trang tuyển dụng**”. Đề tài trên là kết quả của quá trình học tập tích lũy và vận dụng những kiến thức mà em tiếp thu và tìm hiểu.

Mặc dù đã cố gắng trong quá trình xây dựng làm đề tài nhưng do còn nhiều hạn chế về thời gian và kiến thức nên sản phẩm của em không tránh khỏi những thiếu sót, những vấn đề chưa được giải quyết hoàn chỉnh. Vì vậy em rất mong nhận được những ý kiến đóng góp của thầy và những bạn trong lớp để có thể hoàn thiện và phát triển hơn.

Em xin chân thành cảm ơn!

Đỗ Công Ban

I. Mô tả

1. Giới thiệu đề tài

- Đề tài được tạo ra nhằm giải quyết vấn đề tìm việc làm của sinh viên mới ra trường hay những người muốn tìm việc mới phù hợp hơn với bản thân. Cụ thể trong đề tài này sẽ phân tích và gợi ý việc làm trên những trang tuyển dụng lớn như ITViec, Topcv, Careerbuilder.
- Người dùng chỉ cần nhập vào chủ đề, từ khóa liên quan đến chuyên ngành mà mình muốn tìm và nhập tên thành phố muốn làm việc hệ thống sẽ thống kê số lượng việc làm trên từng trang tuyển dụng và gợi ý một vài việc làm phù hợp

2. Phân tích yêu cầu chức năng của đề tài

Có 3 chức năng chính:

- Thu thập dữ liệu việc làm trên các trang web tuyển dụng
- Nhập những từ khóa liên quan đến công việc, thành phố làm việc hệ thống sẽ thống kê tổng số lượng việc làm trên các trang tuyển dụng
- Nhập những từ khóa liên quan đến công việc, thành phố làm việc hệ thống sẽ gợi ý ra 4 công việc phù hợp nhất với những keyword nhập vào hoặc sẽ hiển thị toàn bộ nếu người dùng chọn xem thêm

3. Công nghệ sử dụng

Sử dụng ngôn ngữ Python với các thư viện hỗ trợ đi kèm như: pyscript, matplotlib, numpy, selenium, ...

II. Phân tích các chức năng chính

1. Thu nhập dữ liệu

Ví dụ với trang tuyển dụng ITViec

- Sử dụng thư viện selenium thực hiện các tương tác với trang web

```
1 driver.get("https://itviec.com/it-jobs?page=1&query=&source=search_job")
```

- Tìm ra tổng số trang để thực hiện lặp lấy ra tất cả công việc

```
1 # total page to crawl
2 listPaginations = driver.find_elements(By.XPATH, "//ul[@class='pagination']/li")
3 totalPages = int(listPaginations[3].text)
```

- Lặp lấy ra tất cả công việc hiện có của trang mỗi công việc bao gồm những thông tin như: tên công việc, chủ đề công việc, thành phố làm việc, logo của công ty, link đến trang chi tiết công việc

```
1 for i in range(totalPages):
2     driver.get(f'https://itviec.com/it-jobs?page={i+1}&query=&source=search_job')
3     driver.implicitly_wait(20)
4     listTitles = driver.find_elements(By.XPATH, "//div[@class='details']/h3/a")
5     listTags = driver.find_elements(By.XPATH, "//div[@class='job-bottom']/div[@class='tag-list']")
6     listCities = driver.find_elements(By.XPATH, "//div[@class='job_content']/div[@class='city']/div[@class='address']")
7     listLogoes = driver.find_elements(By.XPATH, "//div[@class='logo']/div/a/picture/img")
8     # print(listTags[1].text)
9     for index, title in enumerate(listTitles):
10        count+=1
11        print(count, '-', title.text, '-', title.get_attribute('href'))
12        data.append(['itviec', title.text, listCities[index].text, listTags[index].text, title.get_attribute('href'), listLogoes[index].get_attribute('data-src')])
13    driver.implicitly_wait(20)
```

- Ghi tất cả những dữ liệu thu thập được vào 1 file csv với các cột tương ứng như: Trang tuyển dụng, tên công việc, thành phố làm việc, chủ đề công việc, link chi tiết công việc, logo công ty

```
1 # Create DataFrame
2 df = pd.DataFrame(data, columns=['Page', 'Title', 'City', 'Tag', 'Link', 'Logo'])
3 df.to_csv("./itviec.csv")
```

2. Gợi ý việc làm

- Sử dụng thư viện pytextdist để thực hiện so sánh dữ liệu nhập vào với dữ liệu được thu thập về sau đó sẽ trả ra 4 công việc có tỉ lệ giống cao nhất

```

1 def searchTop4(df,city, keyword):
2
3     listIndex = []
4     dict = {}
5     count = 0
6     for index in df.index:
7         if no_accent_vietnamese(city.lower()) in no_accent_vietnamese(df['City'][index].lower()):
8             if no_accent_vietnamese(keyword.lower()) in no_accent_vietnamese(df['Tag'][index].lower()):
9                 dict[f"{index}"] = lcs_similarity(no_accent_vietnamese(keyword.lower()),no_accent_vietnamese(df['Tag'][index].lower()))
10
11     for i in sorted(dict, key=dict.get, reverse=True):
12         count+=1
13         listIndex.append(int(i))
14         if count == 4:
15             break
16
17     return listIndex

```

- Nếu người dùng muốn xem tất cả thì hệ thống sẽ trả ra toàn bộ công việc được sắp xếp theo độ giống nhau nhất của dữ liệu nhập vào

```

1 def searchAll(df,city, keyword):
2
3     listIndex = []
4     dict = {}
5     for index in df.index:
6         if no_accent_vietnamese(city.lower()) in no_accent_vietnamese(df['City'][index].lower()):
7             if no_accent_vietnamese(keyword.lower()) in no_accent_vietnamese(df['Tag'][index].lower()):
8                 dict[f"{index}"] = lcs_similarity(no_accent_vietnamese(keyword.lower()),no_accent_vietnamese(df['Tag'][index].lower()))
9
10     for i in sorted(dict, key=dict.get, reverse=True):
11         listIndex.append(int(i))
12
13     return listIndex

```

- Tất cả những dữ liệu nhập vào, dữ liệu thu thập khi so sánh đều được loại bỏ những kí tự Tiếng Việt giúp cho quá trình so sánh có kết quả chính xác nhất

```

1 def no_accent_vietnamese(s):
2     s = re.sub('[\àáâãäåăąǎǻǿǻǿǻǿ]', 'a', s)
3     s = re.sub('[\ÀÁÂÃÄÅĂǺǺǺǺǺ]', 'A', s)
4     s = re.sub('[\éeèěēēēēēēēēē]', 'e', s)
5     s = re.sub('[\ÊËĚĚĚĚĚĚĚĚĚĚ]', 'E', s)
6     s = re.sub('[\óôõöøðōōōōōōōōōōō]', 'o', s)
7     s = re.sub('[\ÒÓÔÕÖØÐŌŌŌŌŌŌŌŌŌŌŌ]', 'O', s)
8     s = re.sub('[\íîïĩĩĩĩ]', 'i', s)
9     s = re.sub('[\ÍÎÏĨĨĨĨ]', 'I', s)
10    s = re.sub('[\úûũũũũũũũũũũũ]', 'u', s)
11    s = re.sub('[\ÙÚÛÜÜÜÜÜÜÜÜÜÜÜ]', 'U', s)
12    s = re.sub('[\ýÿýÿýÿýÿý]', 'y', s)
13    s = re.sub('[\ÝŲÝŲÝŲÝŲÝŲÝŲÝŲ]', 'Y', s)
14    s = re.sub('[\đ', 'd', s)
15    s = re.sub('[\Đ', 'D', s)
16    return s

```

3. Thao tác với giao diện người dùng

- 2 ô input để nhập thông tin, từ khóa liên quan đến công việc, thành phố muốn làm việc, nút thống kê để thực hiện vẽ ra biểu đồ tổng quát cho tổng số lượng công việc theo tìm kiếm trên từng trang và nút Search để thực hiện tìm kiếm công việc

Tra cứu việc làm

Nhập từ khóa liên quan đến công việc

VD: reactjs, angular, java,...

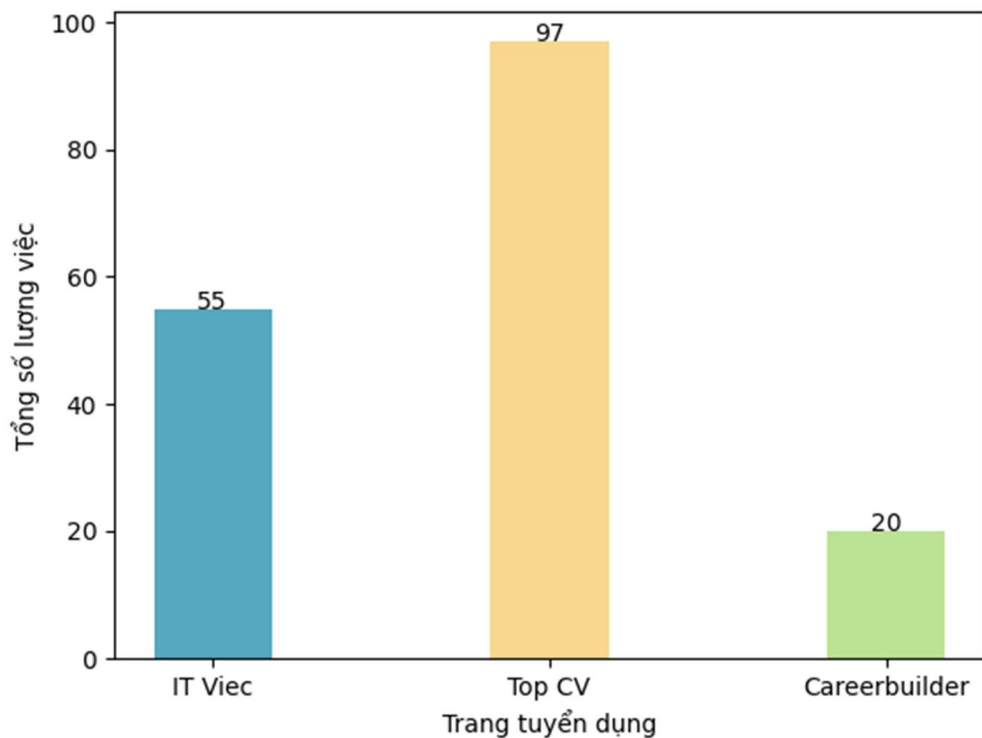
Nhập thành phố

VD: Hà Nội, Hồ Chí Minh, ...





Thống kê

Search

- Ví dụ với keyword nhập vào là: reactjs và thành phố là: Hà Nội



ITviec có 55 việc phù hợp với bạn

 Full Stack Dev (.NET Core C# /React/Eng) .NET ReactJS C# Ha Noi	 Senior .NET Dev (C#, SQL, React, ~\$2000) .NET ReactJS C# Ha Noi
 Software engineer/Team Leader(.NET,iOS) .NET iOS ReactJS Ha Noi Ho Chi Minh	 Jr Web Dev (ReactJS/HTML/CSS) Up to 30M ReactJS CSS HTML5 Ha Noi

[Xem thêm](#)

- Người dùng có thể bấm vào việc sẽ thực hiện chuyển đến trang chi tiết công việc. Nếu muốn hiển thị nhiều hơn chọn Xem thêm

III. Ưu nhược điểm

1. Ưu điểm

- Dễ dàng tra cứu, tìm kiếm thông tin công việc phù hợp với giao diện trực quan
- Dữ liệu được thu thập về nên khi xử lý dữ liệu sẽ cho ra kết quả nhanh chóng

2. Nhược điểm

- Do dữ liệu được thu thập về nên yêu cầu phải crawl dữ liệu về liên tục để dữ liệu được mới nhất

IV. Khắc phục

Để khắc phục nhược điểm trên em đã phát triển 1 hệ thống tìm kiếm trực tiếp. Ưu điểm của cách này là dữ liệu công việc luôn được mới nhất, do dữ liệu được thu thập trực tiếp nên thời gian thu thập và phân tích phụ thuộc khá nhiều vào tốc độ đường truyền mạng

- Nhận 2 input đầu vào của người dùng là: thành phố làm việc và từ khóa của việc làm

```
1 print("Thành phố:", end=" ")
2 province = input()
3 print("Từ khóa:", end=" ")
4 jobInput = input()
```

- Sau đó 2 keyword sẽ được truyền vào các hàm xử lý tương ứng với mỗi trang web tuyển dụng

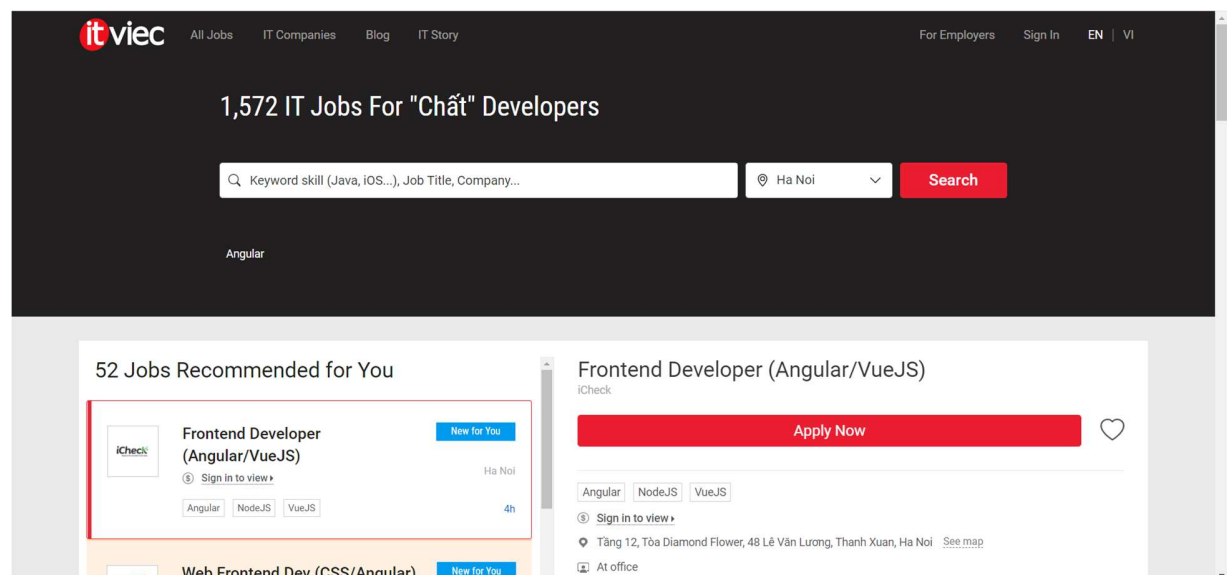
```
1 ITViec.recommend4Job(keyword, province)
2 topcv.recommend4Job(keyword, province)
3 careerbuilder.recommend4Job(keyword, province)
```

- Tại các hàm sẽ thực hiện truy cập vào trang web, thực hiện tự động truyền các keyword và tìm kiếm

Ví dụ với trang tuyển dụng itviec.com

- Thực hiện truy cập trang web

```
1 driver.maximize_window()
2 driver.get("https://itviec.com/?job_selected=c-net-angular-developer-upto-2000-usol-vietnam-4410")
3 driver.implicitly_wait(20)
```



- Sau đó sẽ tự động tìm tất cả công việc dựa vào thành phố và keyword được truyền vào

```

1   for i in range(totalPages):
2       driver.get(f'https://itviec.com/it-jobs/{keyword}/{province}?page={i+1}&source=search_job')
3       driver.implicitly_wait(20)
4       listTitles = driver.find_elements(By.XPATH, "//*[@class='details']/h3/a")
5       for index,title in enumerate(listTitles):
6           listdata.append(f"{title.text}---{title.get_attribute('href')}")
7       driver.implicitly_wait(20)
8
9   for i,data in enumerate(listdata):
10      x = data.split('---')[0]
11      if no_accnt_vietnamese(keyword.lower()) in no_accnt_vietnamese(x.lower()):
12          dict[i] = lcs_similarity(no_accnt_vietnamese(keyword.lower()),no_accnt_vietnamese(x.lower()))
13  for i in sorted(dict, key=dict.get, reverse=True):
14      count+=1
15      listIndex.append(int(i))
16      if count == 4:
17          break

```

- Hệ thống sẽ trả ra 4 công việc phù hợp nhất với những yêu cầu mà người sử dụng với thuật toán pytextdist

```

-----ITViec-----
1 - ReactJS/ React Native---https://itviec.com/it-jobs/reactjs-react-native-lifetek-3113?lab_feature=preview_jd_page
2 - 06 Developers (NodeJS, ReactJS)---https://itviec.com/it-jobs/06-developers-nodejs-reactjs-ssi-securities-corporation-3415?lab_feature=preview_jd_page
3 - NodeJS, Typescript, ReactJS ~ $2500---https://itviec.com/it-jobs/nodejs-typescript-reactjs-2500-gmo-z-com-vietnamlab-center-1501?lab_feature=preview_jd_page
4 - Frontend Engineer (ReactJS)---https://itviec.com/it-jobs/frontend-engineer-reactjs-urbox-0601?lab_feature=preview_jd_page
-----Topcv-----
1 - Frontend Reactjs---https://www.topcv.vn/viec-lam/frontend-reactjs/841039.html?ta_source=JobSearchList
2 - Reactjs Developer---https://www.topcv.vn/viec-lam/reactjs-developer/471249.html?ta_source=JobSearchList
3 - Reactjs Developer---https://www.topcv.vn/viec-lam/reactjs-developer/495884.html?ta_source=JobSearchList
4 - Reactjs Developer---https://www.topcv.vn/viec-lam/reactjs-developer/497645.html?ta_source=JobSearchList
[15412:8548:1107/224332.731:ERROR:util.cc(129)] Can't create base directory: C:\Program Files\Google\GoogleUpdater
[12428:6080:1107/224336.062:ERROR:util.cc(129)] Can't create base directory: C:\Program Files\Google\GoogleUpdater
[19028:5220:1107/224338.326:ERROR:util.cc(129)] Can't create base directory: C:\Program Files\Google\GoogleUpdater
-----Careerbuilder-----
1 - Senior ReactJS | T9122---https://careerbuilder.vn/vi/tim-viec-lam/senior-reactjs-t9122.35BA8DB8.html
2 - Reactjs Developer---https://careerbuilder.vn/vi/tim-viec-lam/reactjs-developer.35BA345F.html
3 - REACTJS DEVELOPER---https://careerbuilder.vn/vi/tim-viec-lam/reactjs-developer.35BACE85.html
4 - Lập trình viên ReactJS---https://careerbuilder.vn/vi/tim-viec-lam/lap-trinh-vien-reactjs.35BAB843.html

```

IV. Kết luận

Đề tài trên được hoàn thiện dựa trên những kiến thức em đã được tiếp thu từ trên lớp và những kiến thức thu thập từ bên ngoài. Sản phẩm còn một vài những sai sót, khuyết điểm không tránh khỏi em rất mong sẽ nhận được những ý kiến đóng góp của thầy để đề tài của em được hoàn thiện hơn.

Source code: <https://github.com/docongbai/job-collection>