ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BÁO CÁO MÔN HỌC
NHẬP MÔN KHOA HỌC DỮ LIỆU

Đề tài

Phân tích thị trường việc làm IT tại Việt Nam

Sinh viên: Nguyễn Xuân Thành – 20204692
Lưu Tiến Ngọc – 20204595
Nguyễn Bá Tuấn – 20204699
Đào Tường Vinh – 20204705
Phạm Đức Hảo – 20200200
GVHD: TS. Trần Việt Trung

Hà Nội, Ngày 18 tháng 12 năm 2023

MỞ ĐẦU

Trong thời đại số hóa ngày càng phát triển, lĩnh vực Công nghệ thông tin (IT) tại Việt Nam đang trở thành một trong những điểm nóng thu hút sự quan tâm của người lao động và doanh nghiệp. Việc áp dụng dữ liệu và phân tích trong việc đánh giá thị trường việc làm trong ngành IT là một phần quan trọng, giúp chúng ta hiểu rõ hơn về xu hướng tuyển dụng, yêu cầu nghề nghiệp, và các khía cạnh khác của ngành này.

Báo cáo này tập trung vào việc phân tích thị trường việc làm IT tại Việt Nam, bao gồm các khía cạnh quan trọng như dữ liệu được sử dụng, các chức năng chính của hệ thống phân tích, cấu trúc mã nguồn, cũng như những vấn đề phổ biến mà chúng ta có thể gặp phải trong quá trình nghiên cứu và phân tích.

Bằng việc nắm vững thông tin và hiểu biết sâu sắc về thị trường việc làm IT tại Việt Nam, chúng ta hy vọng có thể cung cấp thông tin hữu ích cho sinh viên, nhà nghiên cứu, và cộng đồng quan tâm đến lĩnh vực Công nghệ thông tin.

Trong báo cáo này, nhóm xin được gửi lời cảm ơn tới giảng viên hướng dẫn, TS. Trần Việt Trung với những giờ giảng dạy đầy nhiệt huyết và những góp ý sát sao cho nhóm trong việc thực hiện đề tài.

MỤC LỤC

MỞ ĐẦU

DANH MỤC HÌNH VẼ **1**

DANH MỤC BẢNG BIỂU **2**

1. Giới thiệu đề tài **3**

2. Dữ liệu và phương pháp sử dụng **4**

2.1 Các công nghệ, thư viện sử dụng sử dụng 4

2.2 Thu thập và tiền xử lý dữ liệu 6

2.3 Phương pháp sử dụng 7

3. Các chức năng chính của hệ thống **10**

3.1 Trực quan hóa dữ liệu (Visualization Page) 10

3.2 Dự đoán công việc phù hợp dựa trên các kỹ năng của ứng viên (Model Page) 19

4. Cấu trúc mã nguồn **20**

4.1 Cấu trúc tổng quan 20

4.2 Các file mã nguồn quan trọng 20

5. Các vấn đề gặp phải **22**

5.1 Thu thập và xử lý dữ liệu các việc làm IT từ trang web 22

5.2 Mô hình hóa và thực hiện bài toán 22

6. Kết luận **23**

KẾT LUẬN **23**

TÀI LIỆU THAM KHẢO **25**

DANH MỤC HÌNH VẼ

Hình 2.1 Kết quả của MiniLM 9
Hình 2.2 Kết quả của MPNet 9
Hình 3.1 Page 10
Hình 3.2 Fields 10
Hình 3.3 Biểu đồ số jobs ở mỗi page 11
Hình 3.4 Biểu đồ số jobs ở mỗi location 12
Hình 3.5 Biểu đồ số jobs mỗi location của mỗi page 12
Hình 3.6 Biểu đồ số jobs mỗi keyword 13
Hình 3.7 Biểu đồ số jobs mỗi keyword tại Hà Nội 14
Hình 3.8 Biểu đồ số jobs mỗi keyword tại Hồ Chí Minh 15
Hình 3.9 Biểu đồ số jobs mỗi ngôn ngữ 16
Hình 3.10 Biểu đồ top 10 công ty có số jobs cao nhất 17
Hình 3.11 Biểu đồ công ty có jobs ở 2 location 18
Hình 3.12 Biểu đồ requirement với các job của jobs365 18
Hình 3.13 Top 5 công việc phù hợp nhất với ứng viên 19
Hình 4.1 Biểu đồ gói 20
Hình 5.1 Lỗi lặp dữ liệu ở 1 mẫu trong dataset 22

DANH MỤC BẢNG BIỂU

1. Giới thiệu đề tài

Lĩnh vực Công nghệ thông tin tại Việt Nam đã trải qua một giai đoạn phát triển vượt bậc trong những năm gần đây, đồng thời tạo ra nhiều cơ hội việc làm cho người lao động có kỹ năng chuyên môn trong ngành này. Điều này đặt ra một yêu cầu cấp thiết trong việc hiểu rõ hơn về cơ cấu việc làm, nhu cầu tuyển dụng và xu hướng thị trường trong ngành Công nghệ thông tin tại Việt Nam.

Mục tiêu chính của đề tài phân tích chi tiết về thị trường việc làm IT tại Việt Nam thông qua việc sử dụng dữ liệu và các phương pháp phân tích dữ liệu hiện đại. Đồng thời đưa ra hệ thống có khả năng lựa chọn công việc phù hợp dựa trên các kỹ năng của người dùng.

Trong báo cáo, nhóm sẽ trình bày các công việc cụ thể trong quá trình thực hiện đề tài này, bao gồm:

- Giới thiệu đề tài
- Dữ liệu và phương pháp sử dụng
- Cấu trúc mã nguồn
- Các vấn đề gặp phải
- Kết luận

2. Dữ liệu và phương pháp sử dụng

2.1 Các công nghệ, thư viện sử dụng sử dụng

2.1.1 Scrapy

Scrapy là một framework mã nguồn mở được sử dụng để xây dựng và triển khai các ứng dụng web crawling (cào dữ liệu trên web) hiệu quả.

Thành phần của Scrapy:

- Scrapy Engine: Điều khiển luồng giữa tất cả các thành phần
- Scheduler: Nhận yêu cầu từ engine và đưa vào hàng đợi để xử lý
- Downloader: Tải trang web và đưa về cho engine
- Spiders: Bóc tách các trả lời gồm dữ liệu và các đường dẫn yêu cầu mới cần phải truy cập tới
- Item Pipeline: Xử lý dữ liệu sau khi được bóc bởi spider
- Downloader Middlewares: Xử lý các yêu cầu khi chúng đi từ engine tới downloader và ngược lại
- Spider Middlewares: Nằm giữa Engine và Spiders, xử lý các trả lời (responses) và các mục dữ liệu (items) và yêu cầu (requests)

2.1.2 MongoDB

MongoDB là một hệ thống quản lý cơ sở dữ liệu NoSQL phổ biến, thiết kế để lưu trữ và xử lý dữ liệu dưới dạng tài liệu.

Đặc điểm của MongoDB:

- Cơ sở dữ liệu NoSQL: MongoDB sử dụng mô hình cơ sở dữ liệu không quan hệ, cho phép lưu trữ dữ liệu dưới dạng tài liệu JSON-like.
- Tài liệu (Document): Dữ liệu trong MongoDB được tổ chức thành các tài liệu, là đối tượng JSON có thể chứa mọi thông tin cần thiết.
- Các bộ chỉ mục linh hoạt: MongoDB hỗ trợ việc tạo các bộ chỉ mục trên các trường của tài liệu để tối ưu hóa truy vấn.
- Replication: MongoDB cung cấp khả năng sao chép dữ liệu giữa các máy chủ để đảm bảo sự an toàn và sẵn sàng.

- Sharding: MongoDB hỗ trợ sharding, cho phép phân phối dữ liệu trên nhiều máy chủ để xử lý khối lượng dữ liệu lớn.
- Query Language: Truy vấn trong MongoDB sử dụng một ngôn ngữ mạnh mẽ và linh hoạt.

Các thành phần chính của MongoDB:

- MongoDB Server: Quản lý và lưu trữ dữ liệu, xử lý các yêu cầu từ ứng dụng.
- MongoDB Shell: Giao diện dòng lệnh cho phép tương tác với cơ sở dữ liệu MongoDB.
- Driver: Thư viện phần mềm cung cấp giao diện giữa ứng dụng và cơ sở dữ liệu MongoDB.

2.1.3 TorchServe

TorchServe là một framework triển khai mã nguồn mở được kết quả từ sự hợp tác giữa AWS và Facebook (nay là Meta)

Các thành phần chính của TorchServe:

- TorchServe Core: Thành phần này cung cấp các chức năng cơ bản của TorchServe, bao gồm: tải mô hình từ bộ lưu trữ, thực hiện yêu cầu,trả về kết quả
- TorchServe Endpoints: Thành phần này cung cấp các điểm cuối (endpoint) cho các mô hình. Mỗi điểm cuối bao gồm một mô tả điểm cuối và một mô hình.
- TorchServe Middlewares: Thành phần này cung cấp các chức năng xử lý trung gian cho các yêu cầu và kết quả. Các middlewares có thể được sử dụng để thêm các chức năng tùy chỉnh vào TorchServe.

2.1.4 Docker

Docker là một nền tảng mã nguồn mở giúp đơn giản hóa việc triển khai ứng dụng, đóng gói chúng và chạy ở môi trường cô lập gọi là container. *Các thành phần chính của :*

- Containerization: Là khái niệm cốt lõi, cho phép đóng gói ứng dụng và tất cả các phụ thuộc của nó thành một container.
- Docker Image: Được tạo ra từ Dockerfile, là một bản sao không thay đổi của container, chứa mọi thứ cần thiết để chạy ứng dụng.

- Dockerfile: Là một tệp cấu hình văn bản mô tả các bước để xây dựng một Docker image.
- Docker Hub: Dịch vụ lưu trữ trực tuyến cho phép chia sẻ và tìm kiếm các Docker images.
- Docker Engine: Quản lý việc chạy và quản lý container, bao gồm một daemon chạy ở nền và CLI để tương tác.

Ứng dụng của Docker:

- Triển khai ứng dụng: Đơn giản hóa việc triển khai và di chuyển ứng dụng giữa các môi trường khác nhau.
- Phát triển và kiểm thử: Giúp đảm bảo tính nhất quán giữa môi trường phát triển và môi trường sản xuất.
- Tạo môi trường đồng nhất: Đảm bảo ứng dụng chạy đồng nhất trên mọi máy tính và môi trường.

2.2 Thu thập và tiền xử lý dữ liệu

2.2.1 Thu thập dữ liệu

Chúng em đã thực hiện việc thu thập dữ liệu về công việc từ hai trang web phổ biến là trang web Indeed và Job365. Những thông tin thu thập bao gồm các công việc liên quan đến ngành IT ở hai thành phố trọng điểm là thành phố Hà Nội và thành phố Hồ Chí Minh.

Sau đó, chúng em sử dụng ScrapeOps để crawl 25 công việc bao gồm: back-end developer, front-end developer, full-stack developer, mobile developer, game developer, embedded engineer, product manager, product owner, business analyst, project management, IT lead, IT consultant, designer, tester, QA-QC, system engineer, system admin, devOps engineer, data engineer, data architect, data scientist, data analyst, AI engineer, ERP engineer, solution architect.

Sau khi crawl xong dữ liệu, nhóm em tiếp tục tiền xử lý dữ liệu.

2.2.2 Tiền xử lý dữ liệu

Sau khi thu thập xong dữ liệu, chúng em bắt đầu lọc các dữ liệu bị trùng lặp từ các nguồn đã thu thập.

Tiếp theo, nhóm tiếp tục xử lý trường jobDescription là một đoạn html chứa các thẻ liên quan đến requirement, salary, offer,... Chúng em đã lọc các keyword có trong

jobDescription và sau đó đưa về dạng dữ liệu đã được làm sạch và có thể dùng để trực quan hóa bao gồm các trường sau: id, keyword, jobname, requirement, salary, offer, location, company, from.

2.3 Phương pháp sử dụng

Từ những dữ liệu đã thu thập được, nhóm đã phát triển nên 1 hệ thống xác định với những yêu cầu kỹ năng của người dùng thì sẽ phù hợp với ngành nghề gì trong những nghành nghề IT này.

Từ mục đích của hệ thống đấy, nhóm sử dụng những kỹ thuật trong trí tuệ nhân tạo để áp dụng vào cải thiện chất lượng đầu ra của hệ thống

Hệ thống sẽ nhận đầu vào cho phép người dùng nhập Input là những kỹ năng của người dùng. Sau đó hệ thống sẽ tính toán trả về cho người dùng top những chuyên môn phù hợp với những kỹ năng đó

Từ những ý tưởng đó, nhóm quyết định sử dụng các phương pháp trong bài toán Information Retrieval. Một trong những bài toán phổ biến và quan trọng nhất hiện tại trong lĩnh vực xử lí ngôn ngữ tự nhiên

2.3.1 Information Retrieve

Bài toán Information Retrieval (IR) trong lĩnh vực Xử lý Ngôn ngữ Tự nhiên (NLP) liên quan đến việc truy xuất thông tin từ tập dữ liệu văn bản để đáp ứng yêu cầu của người dùng. Dưới đây là mô tả chi tiết về bài toán này:

Mục Tiêu:

- Mục tiêu chính của bài toán là tìm kiếm và trả về các tài liệu (documents) từ một tập dữ liệu lớn mà có liên quan đến nhu cầu thông tin của người dùng..

Dữ Liệu Đầu Vào:

- Tập dữ liệu văn bản: Có thể là một bộ sưu tập văn bản lớn, chẳng hạn như các trang web, bài báo, sách, tài liệu kỹ thuật, v.v.
- Yêu cầu Tìm Kiếm: Người dùng cung cấp một truy vấn (query) hoặc mô tả về thông tin mà họ đang tìm kiếm.

Phương Pháp Truy Xuất:

- Vectorization: Biểu diễn các tài liệu và truy vấn dưới dạng vectơ để thực hiện tính toán độ tương đồng.

- TF-IDF (Term Frequency-Inverse Document Frequency): Phương pháp tính trọng số của các từ trong tài liệu và truy vấn để xác định độ quan trọng của chúng.
- Embedding Models: Sử dụng mô hình nhúng văn bản như Word Embeddings hoặc các mô hình nhúng văn bản nâng cao như BERT để biểu diễn văn bản.

Độ Tương Đồng:

- Sử dụng các độ đo tương đồng như Cosine Similarity, Jaccard Similarity để đo lường mức độ liên quan giữa tài liệu và truy vấn.

Thuật Toán Tìm Kiếm:

- Boolean Retrieval: Sử dụng các toán tử logic (AND, OR, NOT) để truy xuất tài liệu theo điều kiện cụ thể.
- Ranking Algorithms: Sắp xếp các tài liệu theo độ tương đồng và trả về kết quả tìm kiếm theo thứ tự ưu tiên.

Đầu Ra:

- Các tài liệu được trả về dưới dạng danh sách hoặc theo thứ tự độ tương đồng.
- Sử dụng các độ đo như Precision, Recall, F1 Score để đánh giá hiệu suất của hệ thống IR so với kết quả tìm kiếm mong đợi.
- Hệ thống tìm kiếm trên các công cụ tìm kiếm web, thư viện truyện số, hệ thống hỗ trợ quyết định, v.v.

2.3.2 Model

- Nhóm sử dụng 2 baseline được public trên buggingface là:

- all-MiniLM-L6-v2
- all-mpnet-base-v2

- 2 Model này đều đã được distil từ 2 model khác đã được train rất nhiều data trên bộ MSMACRO. Kết quả có độ chính xác cao và dung lượng vừa đủ để có thể deploy lên hệ thống.

- Bộ dữ liệu finetune được lấy từ tập dữ liệu crawl. Xử lí lại những dữ liệu lỗi, nhiễu.

- Từ những dữ liệu đó, nhóm đã tạo nên những dữ liệu negative để có thể tăng mức độ chính xác của hệ thống lên.

- Có 2 version data:

- 5k samples. Mỗi positive sẽ lấy thêm 5 negative
- 22k samples. Mỗi positive sẽ lấy thêm 20 negative

- Dưới đây chính là kết quả đánh giá chất lượng model của nhóm

Baseline	Model	Pre@1	Pre@3	Pre@5	Pre@10	MRR@1	MRR@3	MRR@5	MRR@10
all-MiniLM-L6-v2	Base	0.1603	0.1147	0.082	0.0556	0.1603	0.2429	0.2580	0.2776
	5K-sample	0.2594	0.1619	0.1245	0.0830	0.2594	0.3584	0.3896	0.4174
	22k-sample	0.2924	0.1839	0.1358	0.0853	0.2924	0.4048	0.4334	0.4559

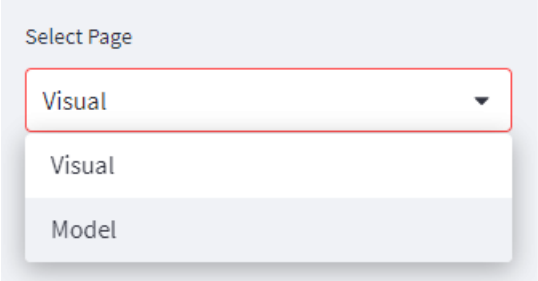
Hình 2.1 Kết quả của MiniLM

all-mpnet-base-v2	Base	0.1415	0.0911	0.0811	0.0542	0.1415	0.2004	0.2299	0.2477
	5k-sample	0.2594	0.1635	0.1198	0.0830	0.2594	0.3647	0.3888	0.4195
	22k-sample	**0.3396**	**0.1839**	**0.1396**	**0.0853**	**0.3396**	**0.4339**	**0.4674**	**0.4881**

Hình 2.2 Kết quả của MPNet

3. Các chức năng chính của hệ thống

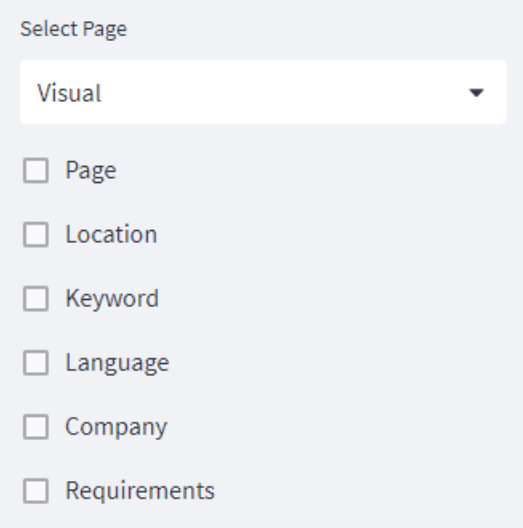
Hệ thống sẽ thể hiện hai công việc chính của project là trực quan hóa dữ liệu **(Visualization Page)** và sử dụng mô hình để dự đoán các công việc phù hợp dựa trên kỹ năng của ứng viên **(Model Page)**.



Hình 3.1 Page

3.1 Trực quan hóa dữ liệu (Visualization Page)

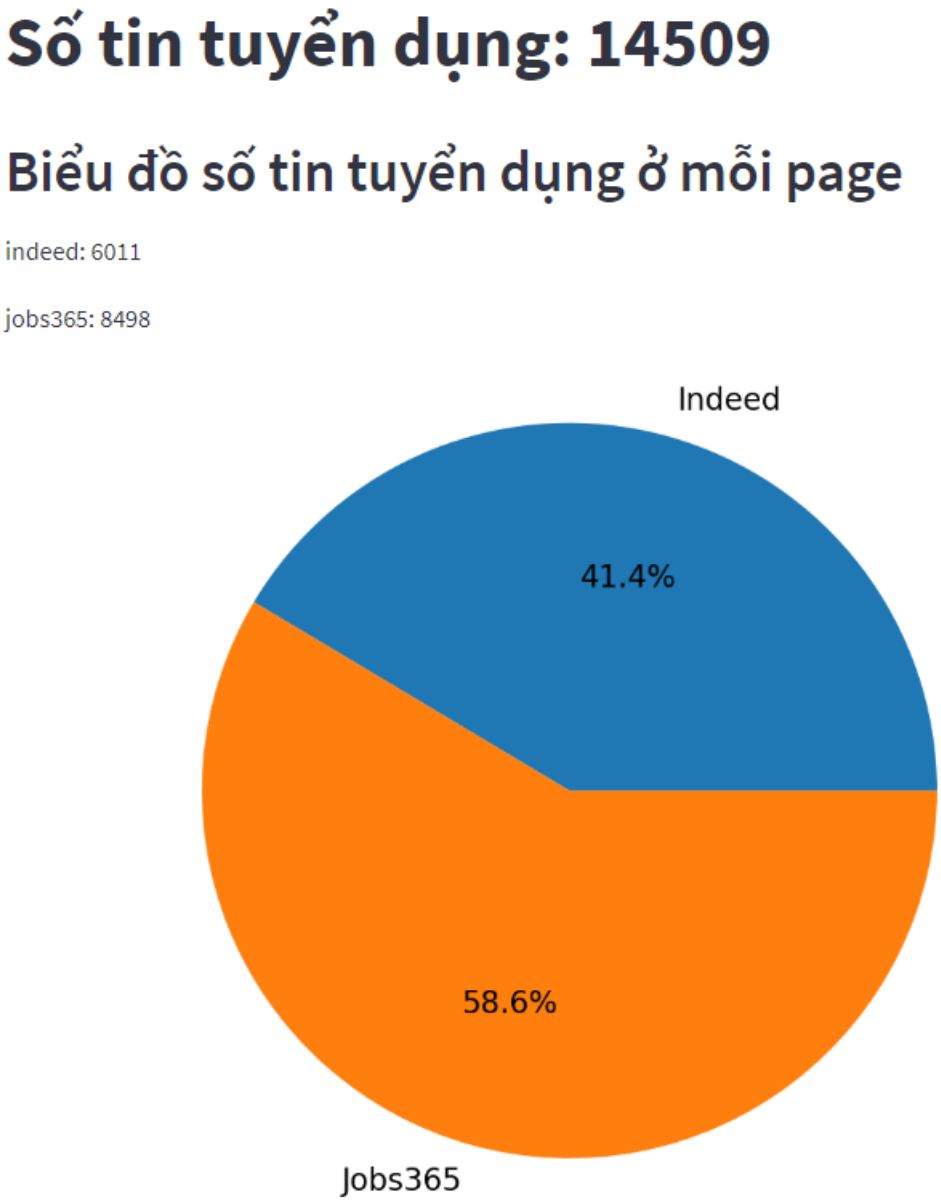
Phần trực quan hóa dữ liệu cung cấp biểu đồ phân tích dựa trên 6 trường sau: Page, Location, Keyword, Language, Company, Requirements.



Hình 3.2 Fields

3.1.1 Page

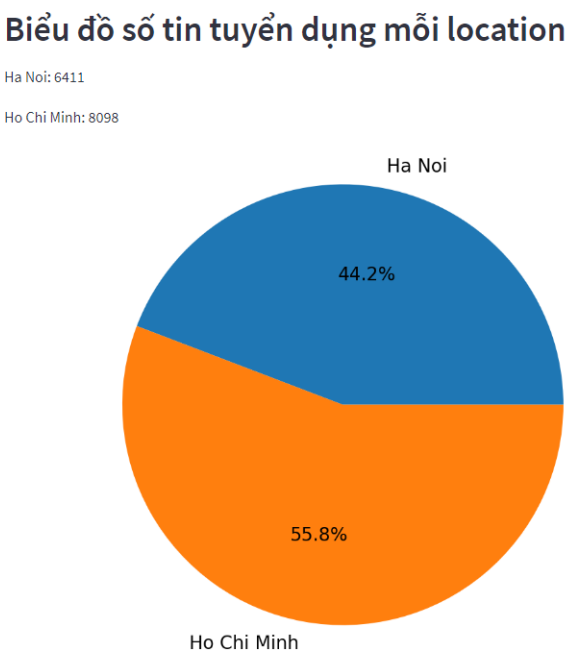
Dữ liệu được sử dụng trong project gồm 14509 jobs được crawl từ 2 page là indeed và jobs365. Trong đó có 6011 jobs đến từ page indeed chiếm 41,4% và 8498 jobs đến từ page jobs365 chiếm 58,6%.



Hình 3.3 Biểu đồ số jobs ở mỗi page

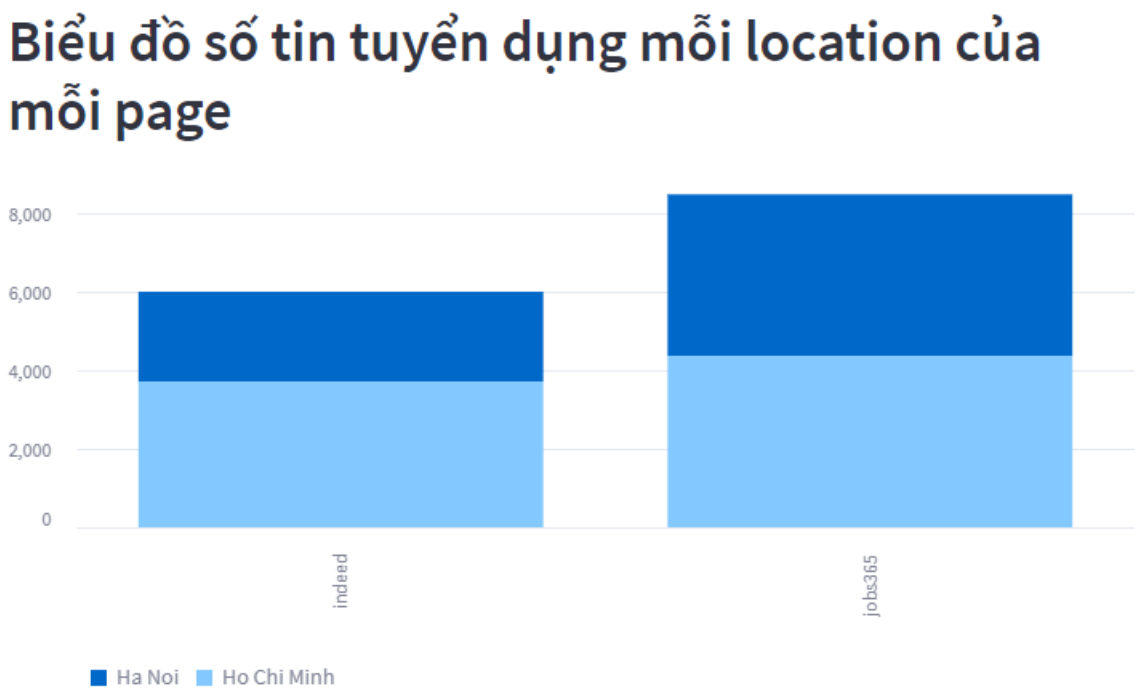
3.1.2 Location

Dữ liệu được crawl từ 2 location là Hà Nội và Hồ Chí Minh. Trong đó có 6411 jobs có location tại Hà Nội chiếm 44.2% và 8098 jobs có location tại Hồ Chính Minh chiếm 55,8%.



Hình 3.4 Biểu đồ số jobs ở mỗi location

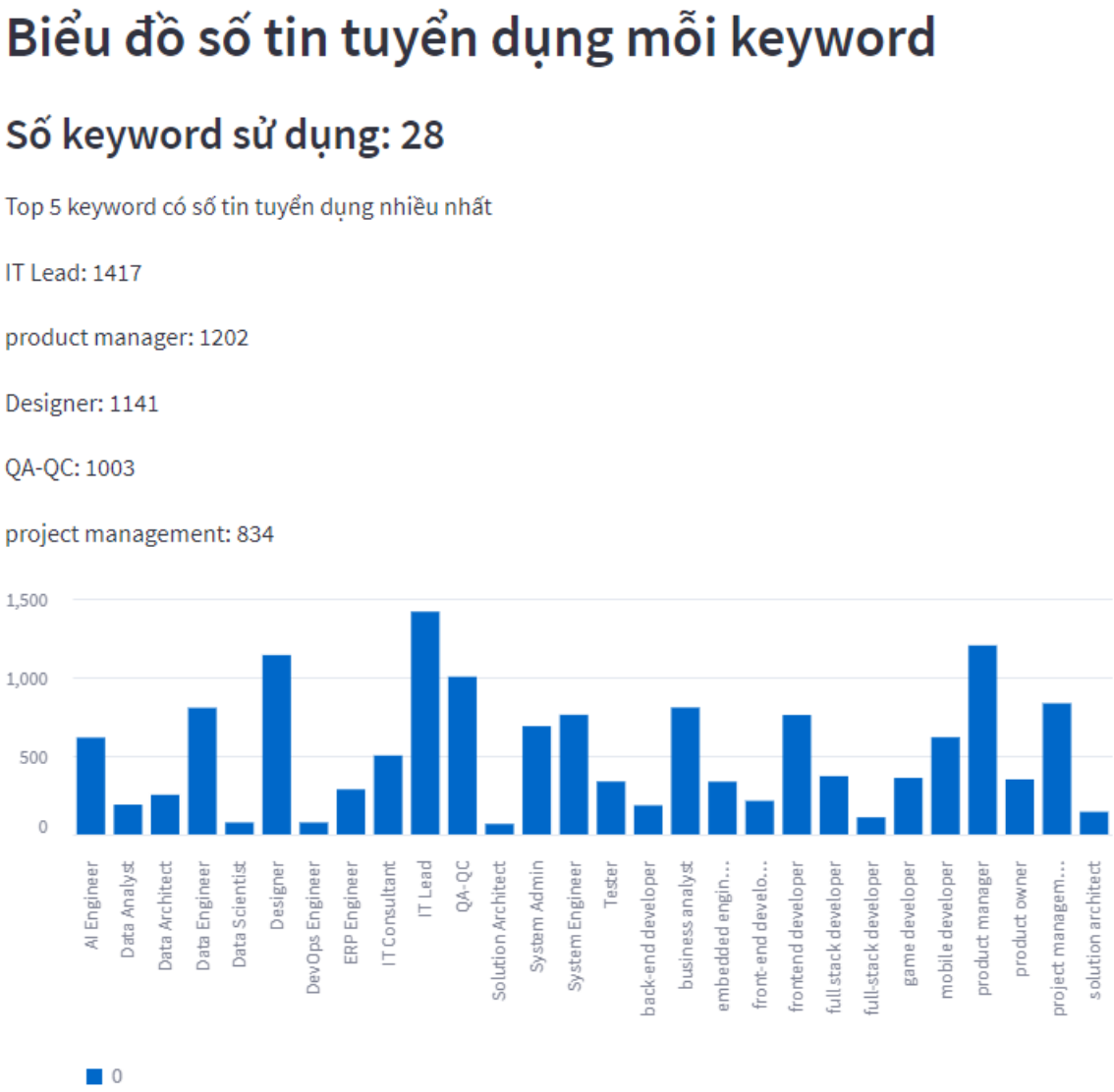
Trong page indeed có 2288 jobs tại Hà Nội, 3723 jobs tại Hồ Chí Minh và page jobs365 có 4123 jobs tại Hà Nội, 4375 jobs tại Hồ Chí Minh.



Hình 3.5 Biểu đồ số jobs mỗi location của mỗi page

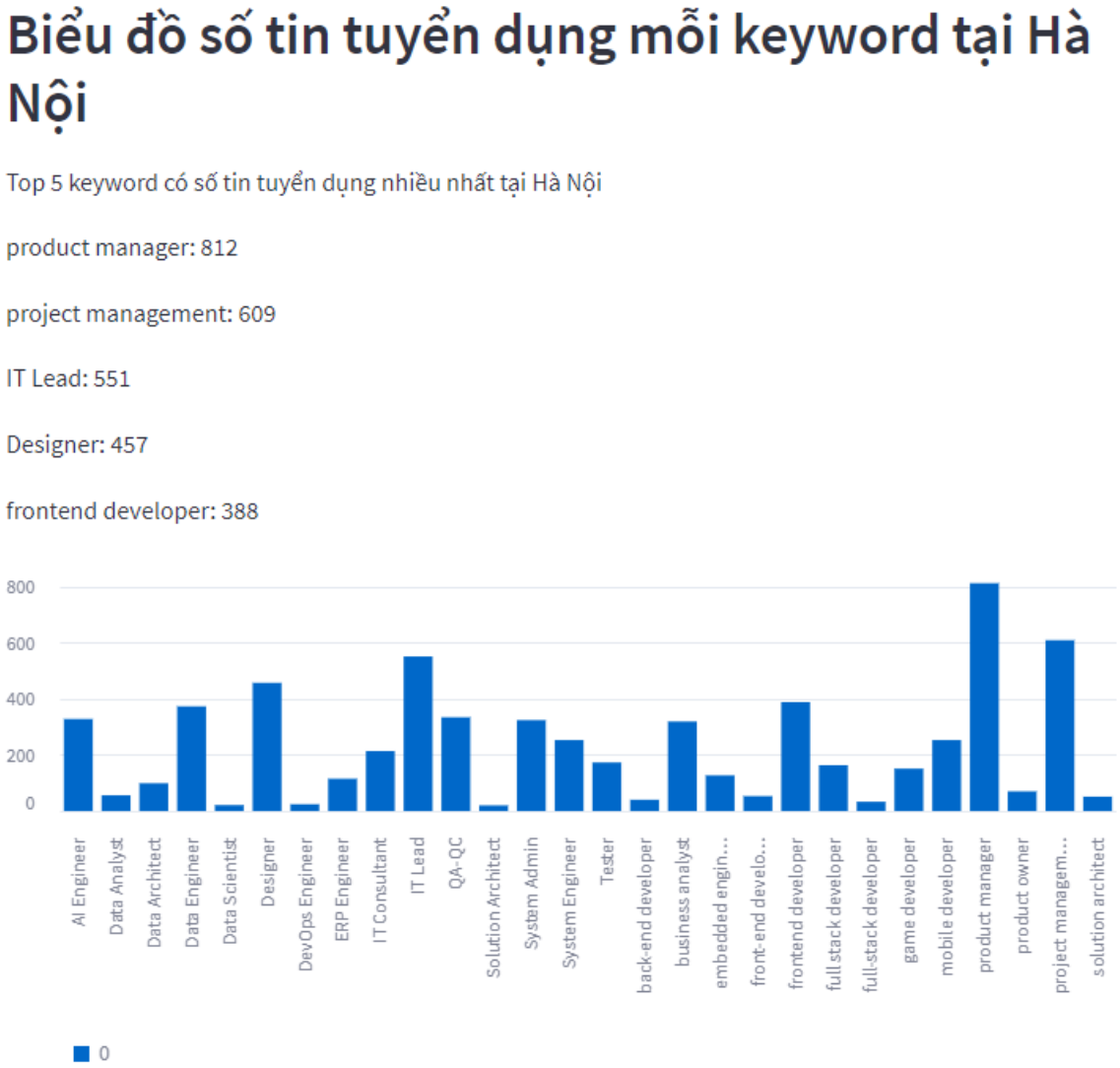
3.1.3 Keyword

Dữ liệu được crawl với 28 keyword: back-end developer, front-end developer, full-stack developer, mobile developer, game developer, embedded engineer, product manager, product owner, business analyst, project management, IT Lead, IT Consultant, Designer, Tester, QA-QC, System Engineer, System Admin, DevOps Engineer, Data Engineer, Data Architect, Data Scientist, Data Analyst, AI Engineer, ERP Engineer, Solution Architect, frontend developer, full stack developer, solution architect.



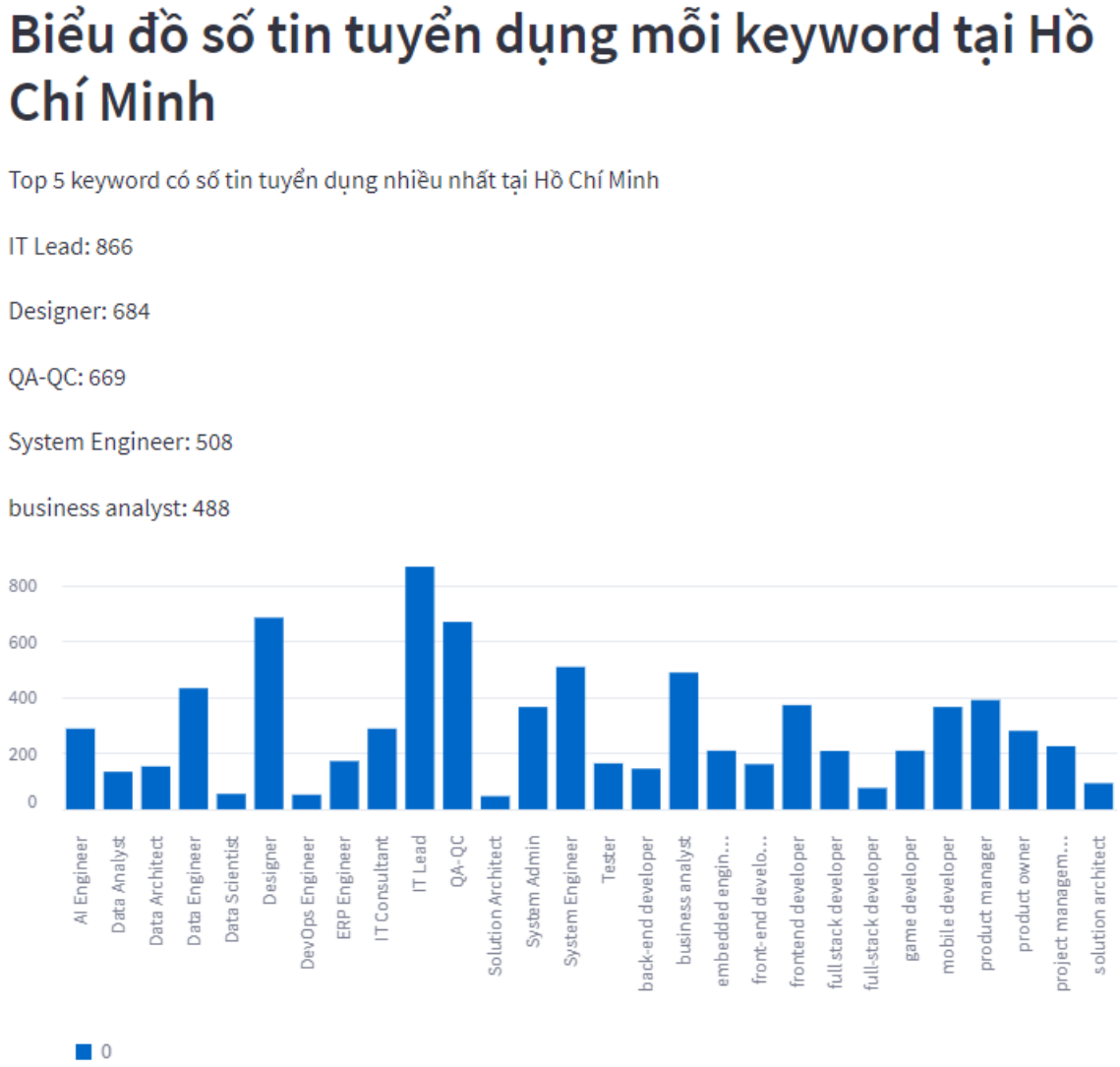
Hình 3.6 Biểu đồ số jobs mỗi keyword

Tại Hà Nội, 5 keyword có số jobs cao nhất là: product manager(812), project management(609), IT Lead(551), Designer(457), frontend developer(388).



Hình 3.7 Biểu đồ số jobs mỗi keyword tại Hà Nội

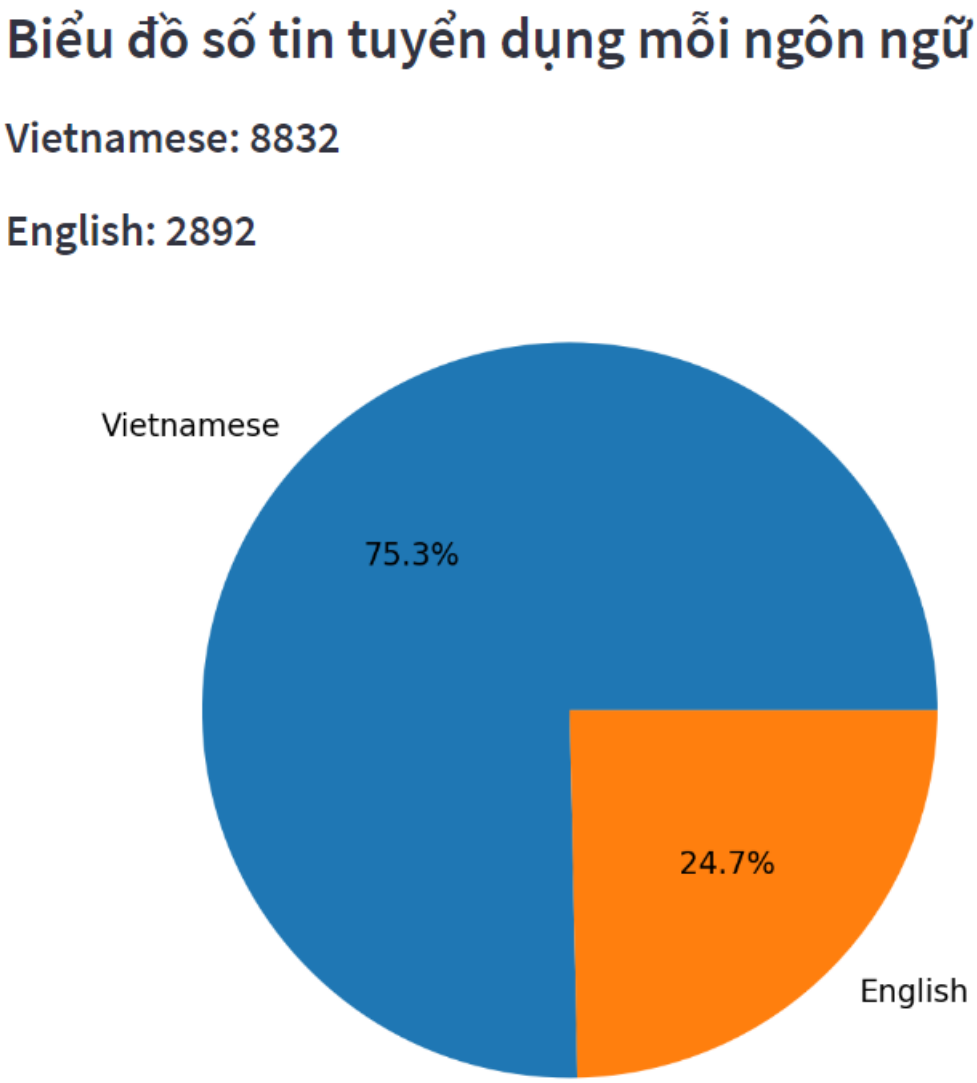
Tại Hồ Chí Minh, 5 keyword có số jobs cao nhất là: IT Lead(866), Designer(684), QA-QC(669), System Engineer(508), business analyst(488).



Hình 3.8 Biểu đồ số jobs mỗi keyword tại Hồ Chí Minh

3.1.4 Language

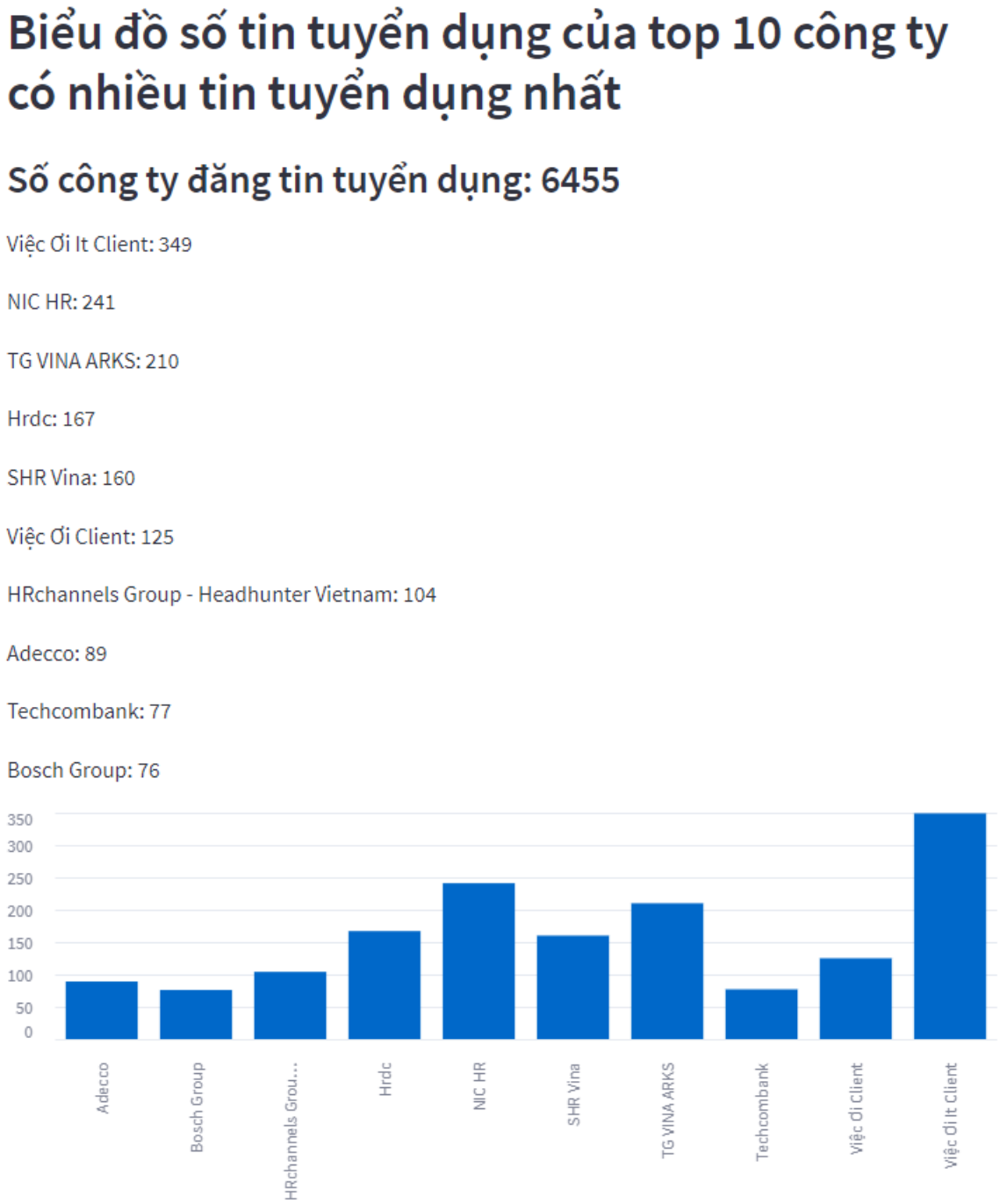
Qua detect nhóm em thấy rằng các jobs được viết bằng 2 ngôn ngữ là Vietnamese và English. Trong đó có 8832 jobs Vietnamese chiếm 75,3% và 2892 jobs English chiếm 24,7%.



Hình 3.9 Biểu đồ số jobs mỗi ngôn ngữ

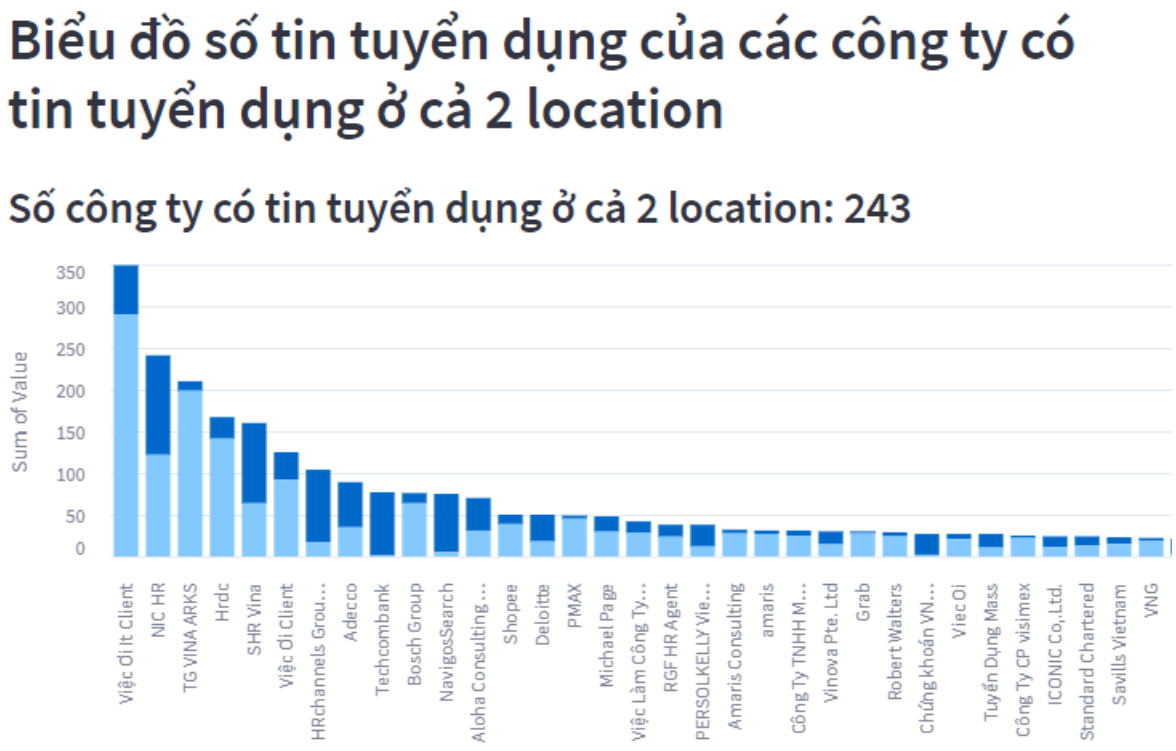
3.1.5 Company

Trong bộ dữ liệu có 6544 công ty đăng bài tuyển dụng, trong đó công ty Việc Ơi IT Client đăng nhiều nhất với 349 tin tuyển dụng.



Hình 3.10 Biểu đồ top 10 công ty có số jobs cao nhất

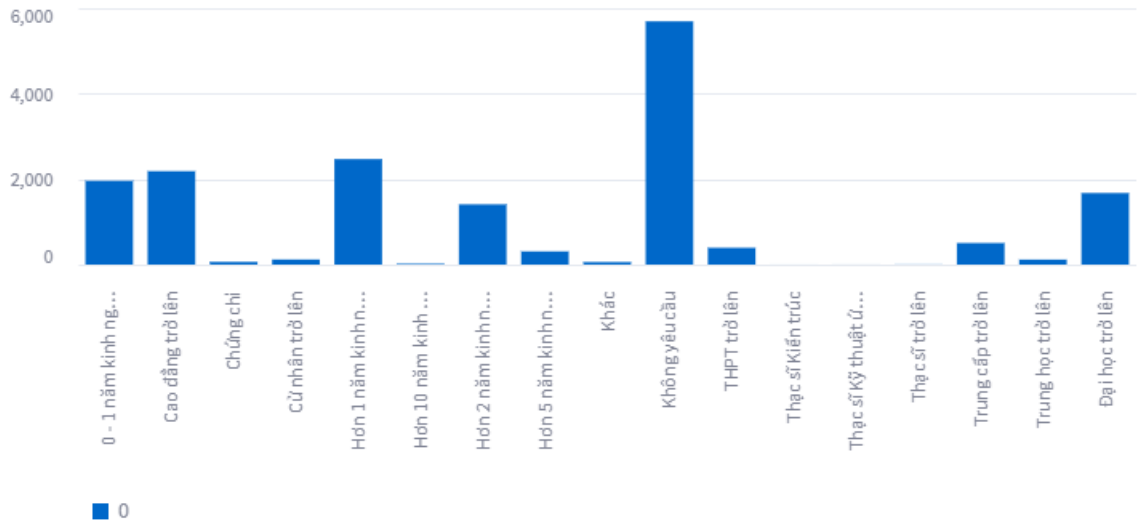
Có 243 công ty đăng tin tuyển dụng tại cả 2 location Hà Nội và Hồ Chí Minh.



Hình 3.11 Biểu đồ công ty có jobs ở 2 location

3.1.6 Requirements

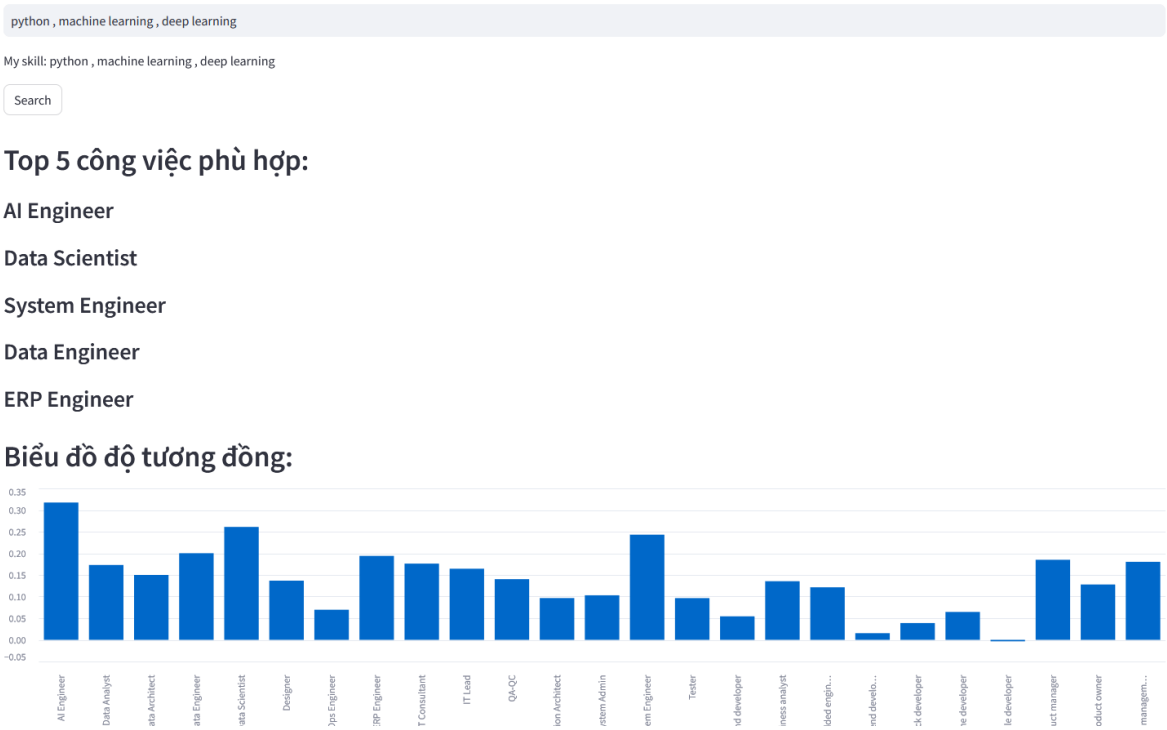
Đối với phần Requirements, chúng em chỉ thực hiện Visualization với các jobs của page jobs365 vì chúng được viết theo cấu trúc. Qua khảo sát có tổng cộng 17 requirements.



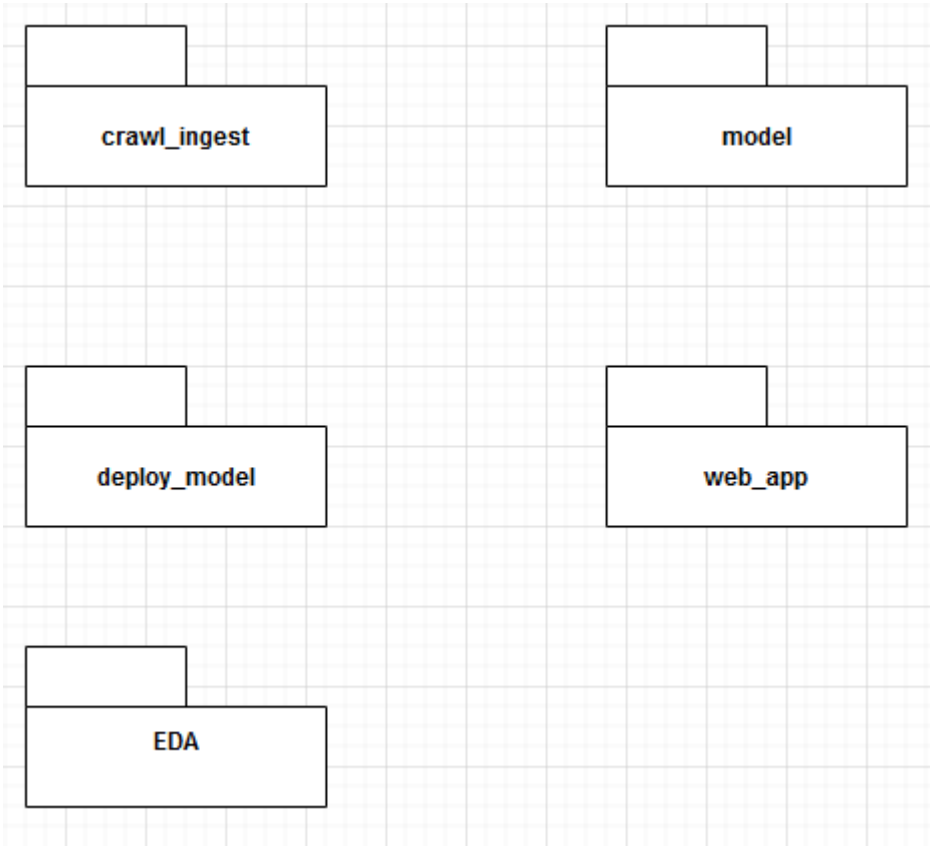
Hình 3.12 Biểu đồ requirement với các job của jobs365

3.2 Dự đoán công việc phù hợp dựa trên các kỹ năng của ứng viên (Model Page)

Trong phần dự đoán công việc phù hợp với ứng viên, nhóm em thực hiện so sánh các kỹ năng của ứng viên với các requirement của công việc từ đó trả ra top 5 công việc có mức phù hợp cao nhất. Trong đó kỹ năng ứng viên được ứng viên nhập vào từ bàn phím.



Hình 3.13 Top 5 công việc phù hợp nhất với ứng viên



Hình 4.1 Biểu đồ gói

4. Cấu trúc mã nguồn

4.1 Cấu trúc tổng quan

- Thư mục crawl_ingest chứa mã nguồn thu thập dữ liệu được viết bằng framwork scrapy , mã nguồn tiền xử lý và tích hợp dữ liệu, lưu vào MongoDB.
- Thư mục model chứa mã nguồn huấn luyện và đánh giá mô hình.
- Thư mục deploy_model chứa mã nguồn deploy mô hình sử dụng TorchServe.
- Thư mục web_app chứa mã nguồn web app .
- Thư mục EDA chứa mã nguồn khám phá và phân tích dữ liệu

4.2 Các file mã nguồn quan trọng

- /web_app/deploy.py : file chạy demo web app , lưu ý chọn đường dẫn đúng tới file dữ liệu .
- /deploy_model/deploy.ipynb : mã nguồn deploy model lên TorchServe và chạy trên Docker

- /EDA/deduplicate_data.ipynb: mã nguồn lọc dữ liệu lặp từ các trang sử dụng thuật toán Local sensitive hashing

5. Các vấn đề gặp phải

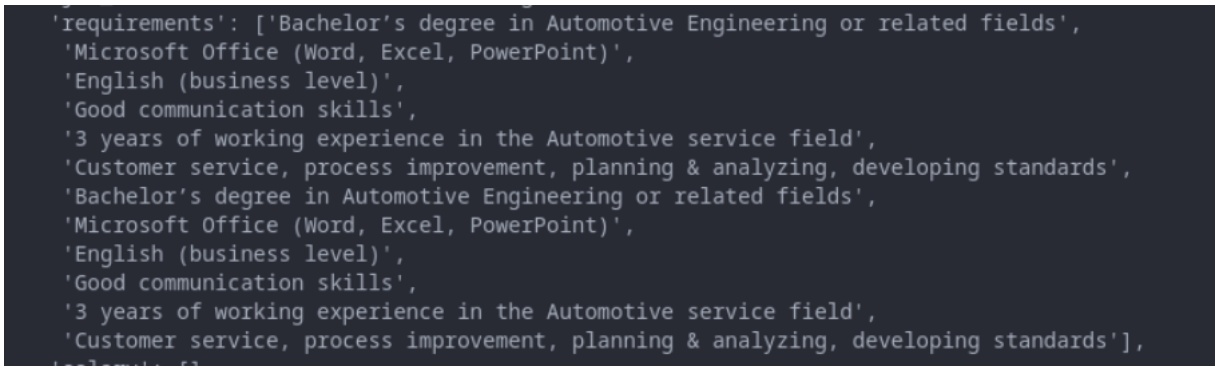
5.1 Thu thập và xử lý dữ liệu các việc làm IT từ trang web

Sử dụng framework Scrapy, tiến hành thu thập dữ liệu từ các trang web:

- Các trang web lớn như TopCV, Linkedin phần lớn chặn truy cập không cho truy cập để bóc tách dữ liệu
- Sau khi có dữ liệu thô, tiến hành tiền xử lý dữ liệu khá phức tạp, phải bóc tách từng dòng để trích xuất các đặc trưng của việc làm
- Xử lý vấn để lặp các việc làm ở các trang sử dụng Mini hash Local sensitive hashing

5.2 Mô hình hóa và thực hiện bài toán

- Việc xác định input, output cho bài toán mất khá nhiều thời gian vì các đặc trưng bị nhiễu rất nhiều
- Sau khi xác định được mô hình bài toán, nhận thấy được các đặc trưng trong một vài dữ liệu bị lặp lại nên phải tiến hành tiền xử lý lại



Hình 5.1 Lỗi lặp dữ liệu ở 1 mẫu trong dataset

6. Kết luận

KẾT LUẬN

Qua báo cáo này nhóm đã trình bày toán bộ công việc trong quá trình thực hiện đề tài về Khoa học dữ liệu để ứng dụng vào bài toán đề ra. Đây là một bài toán rất thực tế và đã phần nào được giải quyết với các phương pháp nhóm đề xuất, cho kết quả tốt và hiệu năng khả thi cho các hệ thống tìm kiếm thời gian thực.

Trong quá trình thực hiện đề tài này, nhóm đã học được thêm nhiều về quy trình đặt vấn đề và giải quyết một bài toán thực tế. Kết quả đạt được là minh chứng cho khả ứng dụng các mô hình học máy, học sâu vào giải quyết một bài toán thực tế cũng như tối ưu hóa về mặt xử lý dữ liệu. Trong thời gian tới, nếu có khả năng, nhóm mong muốn được nghiên cứu thêm về bài toán này, thử nghiệm các cách tiếp cận khác để tối ưu hơn nữa kết quả đạt được.

Trong quá trình làm việc, nhóm đã cố gắng thu thập thông tin và học hỏi, thử nghiệm các giải pháp cho vấn đề, tuy nhiên do tính khó của bài toán cũng như sự đa dạng trong việc thực hành và ứng dụng mà nhóm còn chưa có nhiều kinh nghiệm nên kết quả không thể tránh khỏi những thiếu sót, hạn chế. Nhóm rất mong nhận được đóng góp để có thể hoàn thiện tốt hơn nữa.

Một lần nữa nhóm xin gửi lời cảm ơn chân thành tới thầy Trần Việt Trung đã tận tình giúp đỡ nhóm trong quá trình nghiên cứu, tìm hiểu và thực hiện đề tài đã lựa chọn.

TÀI LIỆU THAM KHẢO

[1] Neil Houlsby, Andrei Giurgiu, Stanisław Jastrze¸bski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. *ICML*, 1902.00751, 2019.

[2] Kenton Lee Kristina Toutanova Jacob Devlin, Ming-Wei Chang. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL*, 1810.04805, 2018.

[3] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A human generated machine reading comprehension dataset. *CoRR*, abs/1611.09268, 2016.

[4] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084, 2019.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[6] Ruofei Zhang Wenhao Lu, Jian Jiao. Twinbert: Distilling knowledge to twin-structured bert models for efficient retrieval. 2002.06275, 2020.