HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN 1

_____ o0o _____

BÁO CÁO BÀI TẬP LỚN

Đề tài: Xếp hạng độ phù hợp của ứng viên với vị trí công việc trên Linkedln

Môn học: Khai phá dữ liệu
Số thứ tự nhóm: 10
Lớp: 04

Nguyễn Minh Thắng	**MSV: B21DCCN668**
Nguyễn Đức Quỳnh	**MSV: B21DCCN646**
Nguyễn Thu Hà	**MSV: B21DCCN041**
Phạm Văn Tiến	**MSV: B21DCCN708**
Đồng Thị Hiền	**MSV: B21DCCN046**

Giảng viên hướng dẫn: Nguyễn Quỳnh Chi

HÀ NỘI, 2025

Mục lục

Danh sách hình vẽ

Danh sách bảng biểu

1

Giới thiệu chung

1.1 Giới thiệu bài toán

Trong bối cảnh thị trường lao động ngày càng cạnh tranh và số lượng ứng viên, công việc đăng tuyển liên tục gia tăng, việc lựa chọn ứng viên phù hợp với từng vị trí công việc trở nên khó khăn và tốn nhiều thời gian cho nhà tuyển dụng.

Bài toán đặt ra là xây dựng một hệ thống có khả năng xếp hạng mức độ phù hợp giữa các ứng viên với các công việc đang tuyển dụng, sử dụng các thuật toán học máy trong lĩnh vực xếp hạng (ranking) và hệ thống gợi ý (recommendation systems).

Mục tiêu của dự án bao gồm:

- Tự động đánh giá và xếp hạng độ phù hợp giữa từng ứng viên với các công việc cụ thể.
- Hỗ trợ nhà tuyển dụng lọc nhanh danh sách ứng viên tiềm năng nhất cho mỗi vị trí đăng tuyển.
- Nâng cao trải nghiệm người dùng trên các nền tảng tuyển dụng thông qua kết quả gợi ý chính xác và cá nhân hóa.

Lợi ích đem lại của dự án:

- Tiết kiệm thời gian và chi phí cho cả nhà tuyển dụng và ứng viên.
- Tăng tỷ lệ tuyển dụng thành công nhờ vào việc kết nối đúng người - đúng việc.
- Giảm tỷ lệ nghỉ việc sớm do sự không phù hợp giữa ứng viên và môi trường làm việc.
- Nâng cao năng lực cạnh tranh cho các nền tảng tuyển dụng thông minh.

Đầu vào và đầu ra mong muốn của dự án:
Đầu vào:

- **Dữ liệu ứng viên:**
  - Hồ sơ cá nhân (học vấn, kinh nghiệm, kỹ năng, nguyện vọng, mô tả bản thân,...)
  - Hành vi tương tác trên hệ thống (các công việc đã xem, ứng tuyển,...)
- **Dữ liệu công việc:**
  - Mô tả công việc, yêu cầu kỹ năng, mức lương, vị trí địa lý, tính chất công việc,...
  - Thông tin từ nhà tuyển dụng

Đầu ra:

- Một **danh sách xếp hạng các ứng viên phù hợp** cho mỗi vị trí công việc.
- Điểm số đánh giá mức độ phù hợp giữa từng cặp (ứng viên, công việc).

1.2 Tổng quan nghiên cứu

Trong thời gian gần đây, nhiều nghiên cứu đã được triển khai nhằm tự động hóa quá trình sàng lọc CV và đánh giá mức độ phù hợp với công việc. Các hướng tiếp cận chính có thể chia thành ba nhóm:

1. **Hệ thống lọc dựa trên nội dung (Content-Based Filtering)**
   Các hệ thống này sử dụng kỹ thuật so khớp từ khóa, độ tương đồng cosine, TF-IDF để đo mức độ giống nhau giữa văn bản CV và mô tả công việc. Ví dụ:
   - **Daryani et al. (2020)** sử dụng kết hợp NLP và cosine similarity, cho kết quả cải thiện tốc độ và độ chính xác, nhưng vẫn bị hạn chế về khả năng hiểu ngữ nghĩa sâu.
   - **Tejaswini et al. (2022)** đạt độ chính xác 92% với cosine similarity và k-NN, tuy nhiên cần xử lý trước mạnh tay với dữ liệu không cấu trúc.

   Hạn chế: Phụ thuộc nhiều vào từ khóa, khó xử lý các ngữ nghĩa ngầm, biểu hiện từ vựng đa dạng.
2. **Mô hình học máy và học sâu (ML/DL Models)**

Các mô hình học máy truyền thống như SVM, k-NN đã được sử dụng cho phân loại CV theo danh mục. Tiếp đó là các mô hình học sâu như CNN, Bi-GRU và CRF để khai thác ngữ cảnh tốt hơn.

- **Ali et al. (2022):** SVM đạt 96% độ chính xác trong bài toán phân loại CV.
- **Priyanka et al. (2024):** Đề xuất mô hình DeepSkillNER kết hợp CNN + Bi-GRU, đạt độ chính xác 99.3% trên tập dữ liệu công khai.

Hạn chế: Cần dữ liệu gán nhãn lớn, tiêu tốn tài nguyên tính toán, thiếu khả năng giải thích.

3. **Ứng dụng mô hình ngôn ngữ lớn (LLMs)**
   Các mô hình như BERT, GPT đã mở ra hướng mới cho xử lý CV dạng văn bản tự do. Chúng được dùng để:

   - Tóm tắt và đánh giá CV (**Gan et al., 2024**: dùng GPT-3.5, đạt F1 = 87.73%).
   - Trích xuất thông tin bằng NER và gán nhãn kỹ năng (**Priyanka et al., 2024**).
   - Sinh embedding giàu ngữ nghĩa dùng trong so khớp ngữ cảnh.

   Hạn chế: Chi phí cao, yêu cầu xử lý quyền riêng tư, cần cơ chế giải thích rõ ràng (XAI).

Từ các công trình đã khảo sát, nhóm nghiên cứu có thể xác định một số hướng đi khả thi cho dự án trước khi tiến hành phân tích dữ liệu có được:

- **Kết hợp LLM và KGAT (Knowledge Graph Attention Network):**
  - Dùng LLM để trích xuất và biểu diễn ngữ nghĩa các phần tử trong CV và JD.
  - Xây dựng đồ thị tri thức gồm các thực thể như: kỹ năng, ngành nghề, vị trí, chứng chỉ, học vấn,... với các mối quan hệ liên kết giữa chúng.
  - Sử dụng KGAT để học embedding ngữ nghĩa cho từng thực thể trong đồ thị, với cơ chế attention giúp mô hình hóa tầm quan trọng của từng liên kết trong việc đánh giá độ phù hợp.
  - Đầu ra là điểm số tương thích giữa ứng viên và công việc, tính trên toàn đồ thị tri thức.
- **Tăng cường khả năng giải thích (XAI):**

  - Áp dụng các kỹ thuật như attention scores từ KGAT, SHAP hoặc LIME để phân tích nguyên nhân xếp hạng.

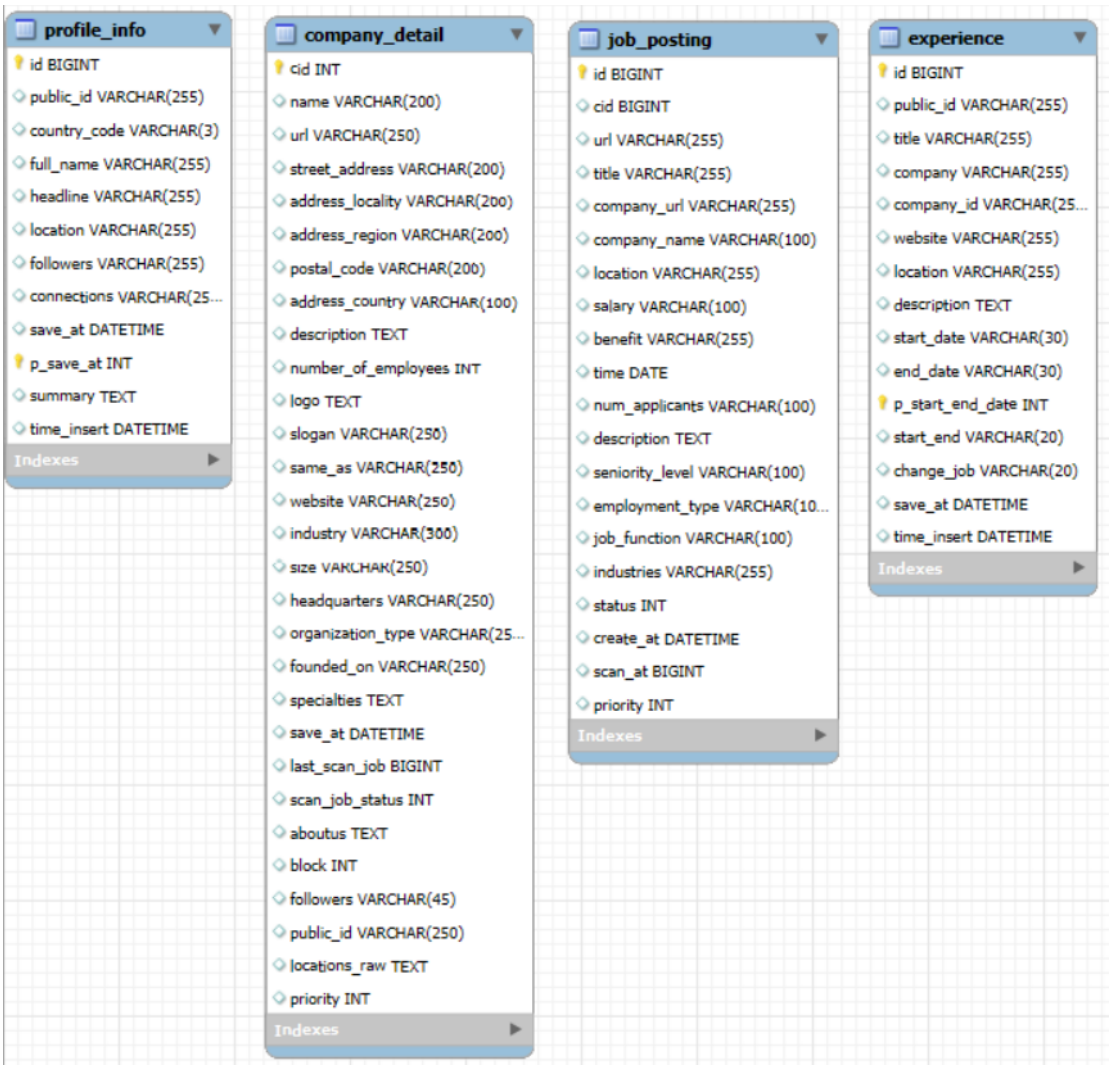
- **Xử lý CV đa định dạng:**
  - Phát triển pipeline trích xuất cấu trúc từ nhiều định dạng CV (PDF, DOCX).

- **Thực hiện bài toán theo hướng hồi quy thay vì phân loại:**
  - Dự đoán điểm số phù hợp liên tục (0–100), dễ dàng phản ánh mức độ phù hợp hơn so với chỉ phân loại nhị phân.

2

Bộ dữ liệu

2.1 Nguồn dữ liệu



Hình 2.1: *Cơ sở dữ liệu Linkedln.*

Bộ dữ liệu được lưu trữ trong hệ quản trị cơ sở dữ liệu MySQL, bao gồm bốn bảng chính: profile_info, experience, company_detail và job_posting. Mỗi bảng thể hiện một khía cạnh riêng biệt trong hệ sinh thái nghề nghiệp — từ cá nhân, tổ chức đến hoạt động tuyển dụng:

- profile_info: chứa thông tin tổng quan về từng cá nhân, được định danh bởi public_id. Đây là bảng đại diện cho hồ sơ người dùng, bao gồm các trường như tên đầy đủ, tiêu đề nghề nghiệp, vị trí địa lý, số lượng người theo dõi, số kết nối và phần tóm tắt giới thiệu bản thân. Dữ liệu trong bảng này phản ánh mức độ hiện diện và vị trí nghề nghiệp của người dùng trên nền tảng mạng xã hội chuyên ngành.
- experience: lưu trữ chi tiết các vị trí công việc mà người dùng từng đảm nhiệm. Mỗi dòng đại diện cho một trải nghiệm làm việc cụ thể, với thông tin như chức danh, tên công ty, thời gian làm việc và mô tả công việc.
- company_detail: chứa thông tin mô tả chi tiết về các công ty, định danh bởi cid khi lưu trong cơ sở dữ liệu. Các trường dữ liệu bao gồm tên công ty, ngành nghề hoạt động chính, quy mô nhân sự, địa chỉ, năm thành lập, mô tả hoạt động, vị trí địa lý,... Đây là nguồn thông tin quan trọng giúp nhận diện tổ chức nơi người dùng từng làm việc hoặc nơi phát hành các tin tuyển dụng.
- job_posting: đại diện cho các bài đăng tuyển dụng được tạo bởi các công ty, được liên kết với cid. Mỗi dòng trong bảng mô tả một cơ hội việc làm cụ thể, bao gồm tiêu đề công việc, mô tả chi tiết, cấp bậc, loại hình việc làm, ngành nghề, vị trí địa lý, số lượng ứng viên và thời gian đăng bài.

Mặc dù các bảng không được thiết kế với các ràng buộc khóa ngoại rõ ràng khi lưu trong cơ sở dữ liệu, chúng vẫn thể hiện mối quan hệ logic và nhất quán về mặt dữ liệu. Các liên kết ngầm giữa các bảng có thể được khai thác để xây dựng mô hình dữ liệu quan hệ phục vụ cho các tác vụ phân tích hoặc học máy. Cụ thể:

- Trường public_id trong bảng profile_info liên kết với public_id trong bảng experience, thể hiện mối quan hệ một–nhiều giữa hồ sơ người dùng và kinh nghiệm làm việc.
- Trường cid trong bảng company_detail liên kết với cid trong bảng job_posting, thể hiện mối quan hệ một–nhiều giữa công ty và các bài đăng tuyển dụng.
- Trường company_id trong bảng experience liên kết với public_id của bảng company_detail, từ đó xác định một công ty có thể xuất hiện trong nhiều trải nghiệm làm việc của người dùng.

2.2 Phân tích dữ liệu

2.2.1 Bảng company_detail

Bảng này gồm 500.000 bản ghi với các trường có thể phân tích:

cid

Mã định danh cho mỗi công ty lưu trong cơ sở dữ liệu, là số nguyên như 1000, 1001,... Trường này có đầy đủ 500.000 giá trị.

name

Tên công ty, là dạng chuỗi ký tự, trường này có 349.756 bản ghi chứa giá trị (chiếm 69,95%)

url

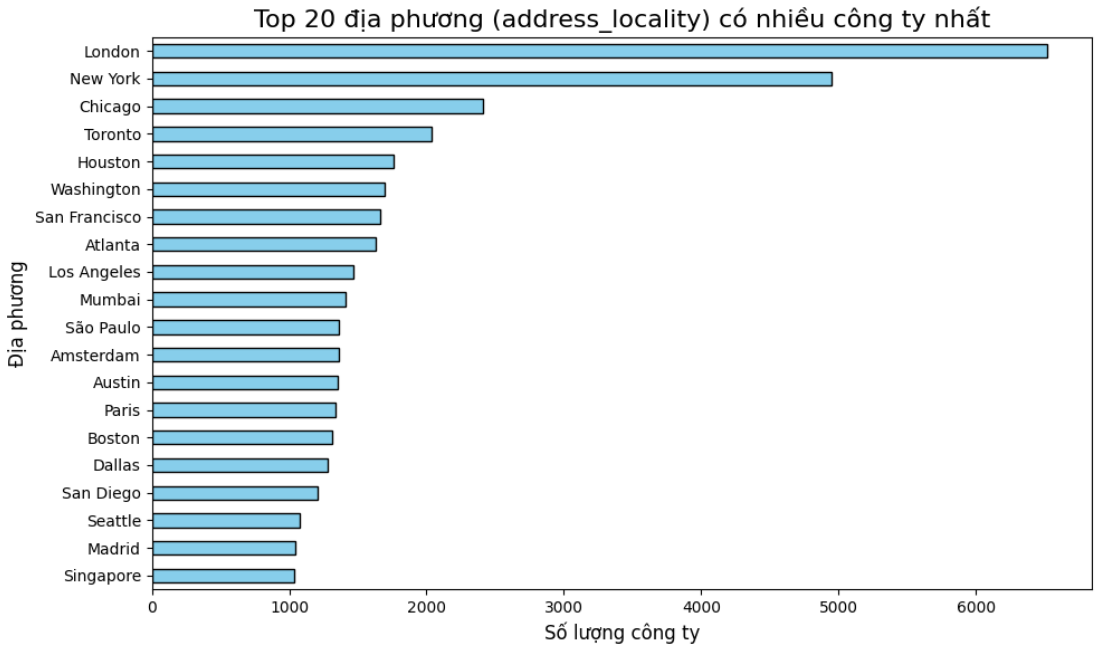
Đường dẫn đến trang linkedln của công ty, là dạng chuỗi ký tự, trường này có 347.055 bản ghi chứa giá trị (chiếm 69,41%).

street_address

Địa chỉ đường phố của trụ sở hoặc văn phòng công ty, là dạng chuỗi ký tự, trường này chỉ có 224.497 bản ghi chứa giá trị (chiếm 44,9%).

address_locality

Thành phố, thị trấn hoặc khu vực hành chính cấp địa phương nơi công ty đặt trụ sở, là dạng chuỗi ký tự, trường này chỉ có 237.805 bản ghi chứa giá trị (chiếm 47,56%).



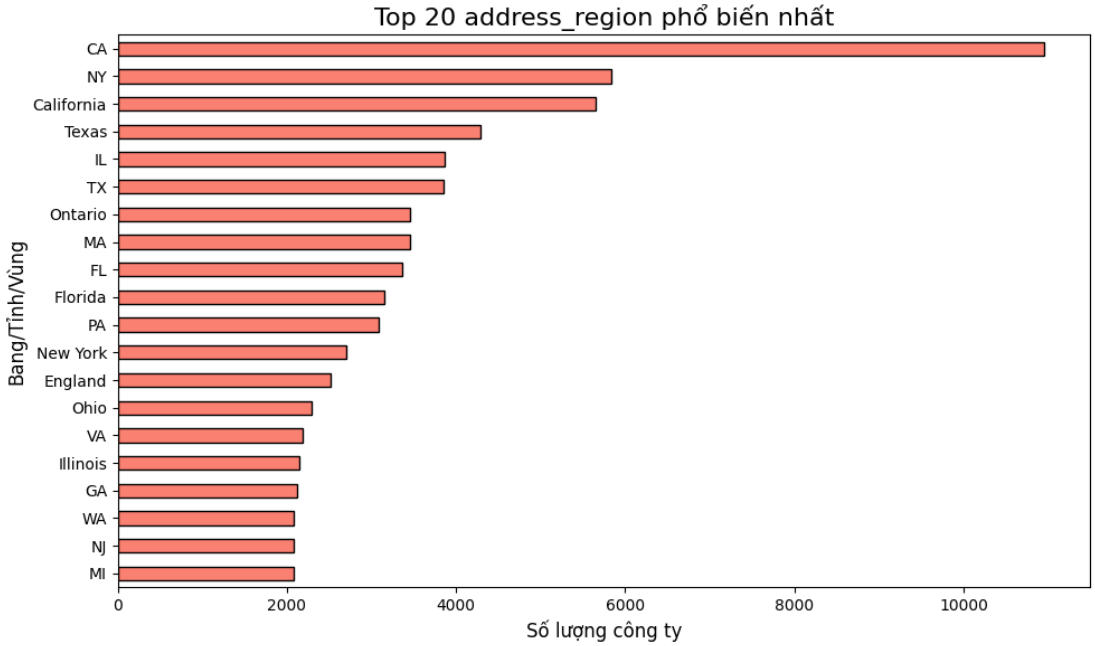
Hình 2.2: *Top 20 địa phương nhiều công ty nhất.*

Dựa vào biểu đồ trên, có thể thấy các công ty tập trung chủ yếu tại các thành phố lớn như London, New York, Chicago, Toronto, Houston, v.v. Điều này phản ánh xu hướng phát triển kinh tế tập trung ở các đô thị lớn, nơi có nhiều cơ hội kinh doanh và nguồn lực. Các thành phố này là trung tâm kinh tế lớn của thế giới, thu hút nhiều doanh nghiệp đặt trụ sở và hoạt động.

address_region

Bang, tỉnh hoặc vùng hành chính lớn hơn thành phố, thường gặp nhất ở các quốc gia như Mỹ, Canada và Úc, là dạng chuỗi ký tự, trường này chỉ có 195.601 bản ghi chứa giá trị (chiếm 39,12%).

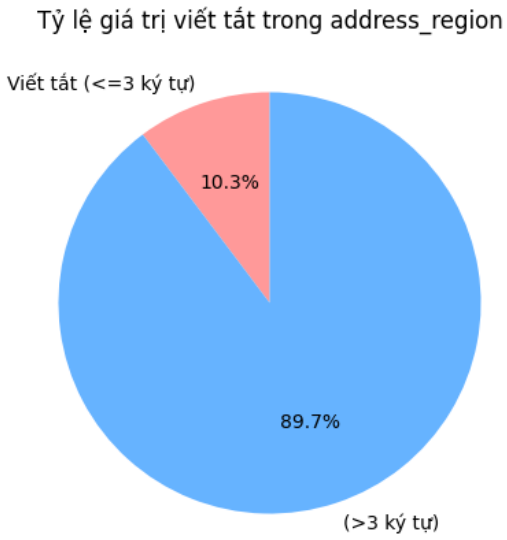
Trường này bị thiếu khá nhiều do nhiều quốc gia không sử dụng phân cấp này và dữ liệu cũng không đầy đủ.



Hình 2.3: *Top 20 address_region phổ biến nhất.*

Biểu đồ trên cho thấy các giá trị xuất hiện nhiều nhất trong cột address_region chủ yếu là tên các bang, tỉnh hoặc vùng hành chính lớn, với tần suất cao nhất thuộc về các khu vực như CA, NY, California, Texas, v.v. Tuy nhiên, phân phối này chưa phản ánh đúng thực tế vì dữ liệu cột này bị lẫn cả dạng viết tắt (ví dụ: ”CA”, ”NY”, ”TX”) và dạng viết đầy đủ (ví dụ: ”California”, ”New York”, ”Texas”). Việc này khiến số lượng công ty thực tế ở một khu vực bị chia nhỏ ra nhiều giá trị khác nhau, làm sai lệch kết quả thống kê.

Nếu giả sử các giá trị viết tắt là những giá trị có độ dài nhỏ hơn hoặc bằng 3 ký tự thì chúng chiếm khoảng 10,27% trên tổng số 7461 giá trị khác nhau của trường address_region.



Hình 2.4: *Số address_region có số lượng kí tự nhỏ.*

Bên cạnh đó, dữ liệu còn tồn tại nhiều giá trị bất thường như các chuỗi số (ví dụ: ”3482”, ”6140”, ”3000”,...), các giá trị trong ngoặc (ví dụ: ”(MI)”, ”(FE)”, ”(RE)”), hoặc các ký hiệu khó xác định ý nghĩa. Điều này cho thấy dữ liệu trường này khá là không đồng nhất.

```
'USVI', '(BL)', '浦东新区', 'LIma', 'FBiH', 'Gard', 'EMEA', '(BZ)', 'Chui', 'Lugo', 'DOHA',
'(SP)', '9410', 'M.P.', 'Jaén', 'Киев', 'IASI', ' 上海市', 'Vic.', 'EAST', 'Mayo', 'vaud',
'LEON', 'cdmx', 'Colo', '8011', '1000', 'WORC', 'Soho', 'SP -', 'IDF ', 'Down', 'caba',
'(MB)', 'CEBU', 'Perú', '1205', 'PARA', 'W.B.', 'KENT', 'Beja', 'BARI', '1351', 'osun',
'Auck', '(BE)', 'Hull', '(SI)', '(Gn)', 'LUGO', 'Faro', 'N.A.', 'giza', 'Chur', 'W.A.',
'A.P.', 'Gifu', 'Metn', '4128', 'wien', 'JAFZ', 'Lara', ' DK ', 'D.N.', 'Ill.', 'DMCC',
'Lodi', 'Baar', 'mass', '3046', 'Guam', 'QLD.', '(TI)', 'Gyzi', 'Mahe', 'BSAS', 'Yoro',
'Guj.', '2340', '3420', 'Buca', 'PISA', '(LT)', 'FRL.', '(RA)', 'PHP ', 'Gorj', 'rome',
'1070', 'Mza.', 'nono', '(CN)', 'Ilia', 'Sfax', '6221', 'CHER', 'Acre', 'ROMA', 'BALI',
'Evry', 'Teso', 'Utan', 'Alba', 'Cook', 'KIEV', 'Skye', 'Matn', 'Qld ', 'Vigo', '2000',
'Ohip', 'R.M.', 'Aley', 'Qro.', 'Mich', 'oslo', 'QLD ', '(TA)', 'Léón', '(Co)', 'D.f.',
'E.P.', 'NSW.', 'Gent', '(PN)', 'KYIV', 'LRMP', 'Mali', 'Gto.', 'NORD', 'Iran', '(RM)',
'USCA', '2064', 'Cape', 'O VL', 'test', '(Ra)', 'Riau', 'Yafo', 'Coah', 'Utr.', 'Gozo',
'D.F,', 'ROME', '(CS)', 'East', 'Fafe', '1504', '1260', '(VR)', 'GERS']
```

Hình 2.5: *Một số giá trị trong address_region.*

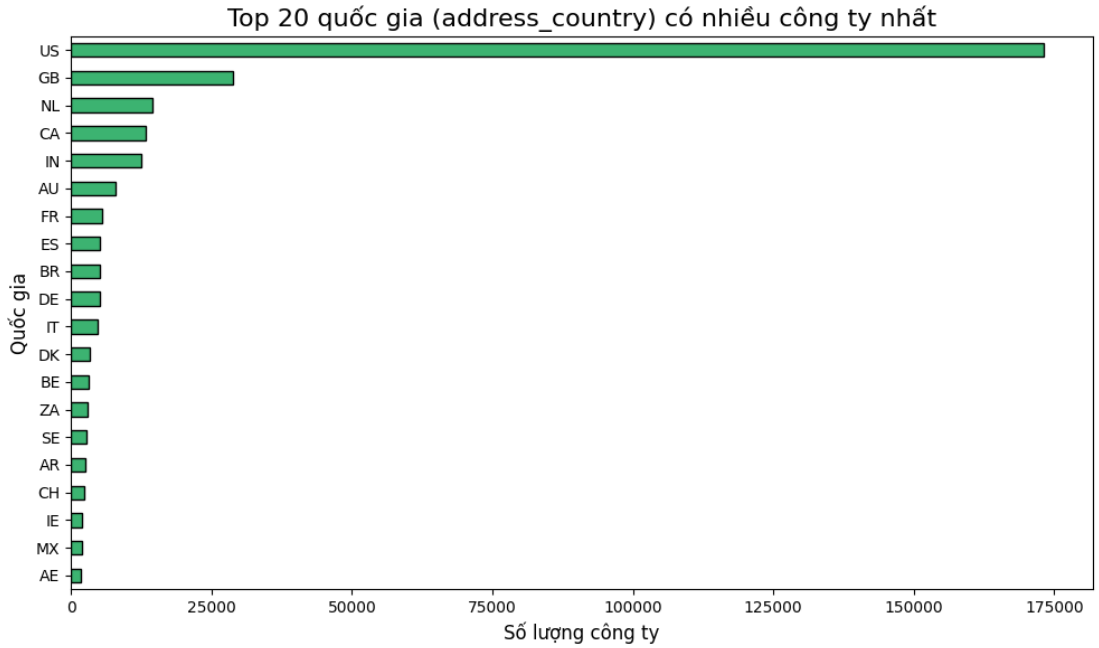
postal_code

Mã bưu chính (ZIP Code) tương ứng địa điểm, là dạng chuỗi ký tự số (như 10001, 91360,...), trường này có 283.140 bản ghi chứa giá trị (chiếm 56,63%).

address_country

Mã quốc gia 2 ký tự theo chuẩn ISO 3166-1 alpha-2, là dạng chuỗi ký tự (như US - Mỹ, CA - Canada,...), trường này có 337.887 bản ghi chứa giá trị (chiếm 67,58%).

Với 246 mã quốc gia và dễ thấy dữ liệu chủ yếu tập trung ở một số quốc gia

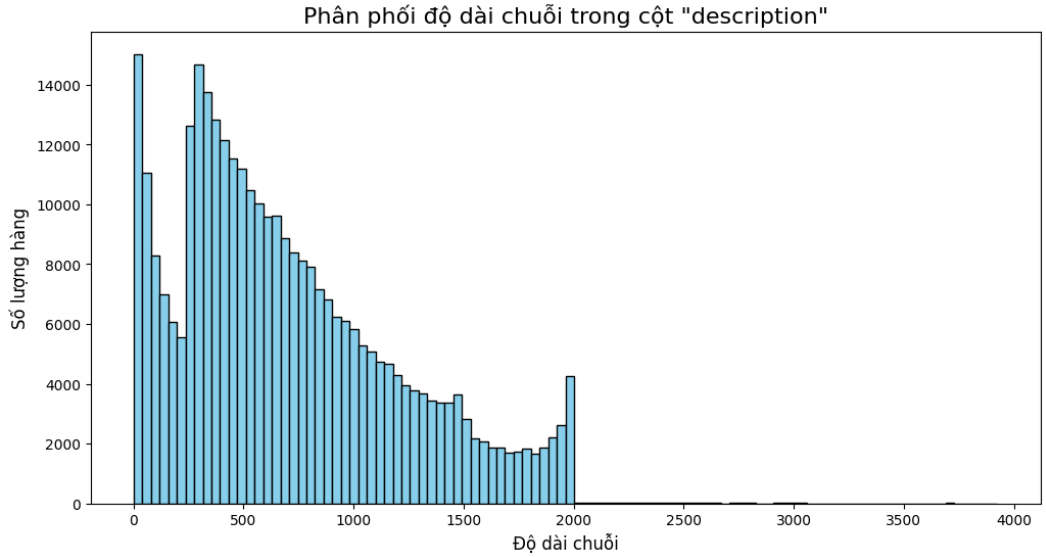


Hình 2.6: *Top 20 quốc gia nhiều công ty nhất.*

như Mỹ (US), Anh (GB), Hà Lan (NL), Canada (CA), Ấn Độ (IN), Úc (AU),... với Mỹ chiếm áp đảo so với các nước còn lại. Các giá trị trong cột này không xuất hiện giá trị bất thường hoặc sai định dạng.

description

Mô tả tổng quan về công ty, bao gồm nhiều thông tin như hoạt động, sứ mệnh, sản phẩm/dịch vụ,..., là dạng chuỗi ký tự , trường này có 325.150 bản ghi chứa giá trị (chiếm 65,03%).



Hình 2.7: *Độ dài description các công ty.*

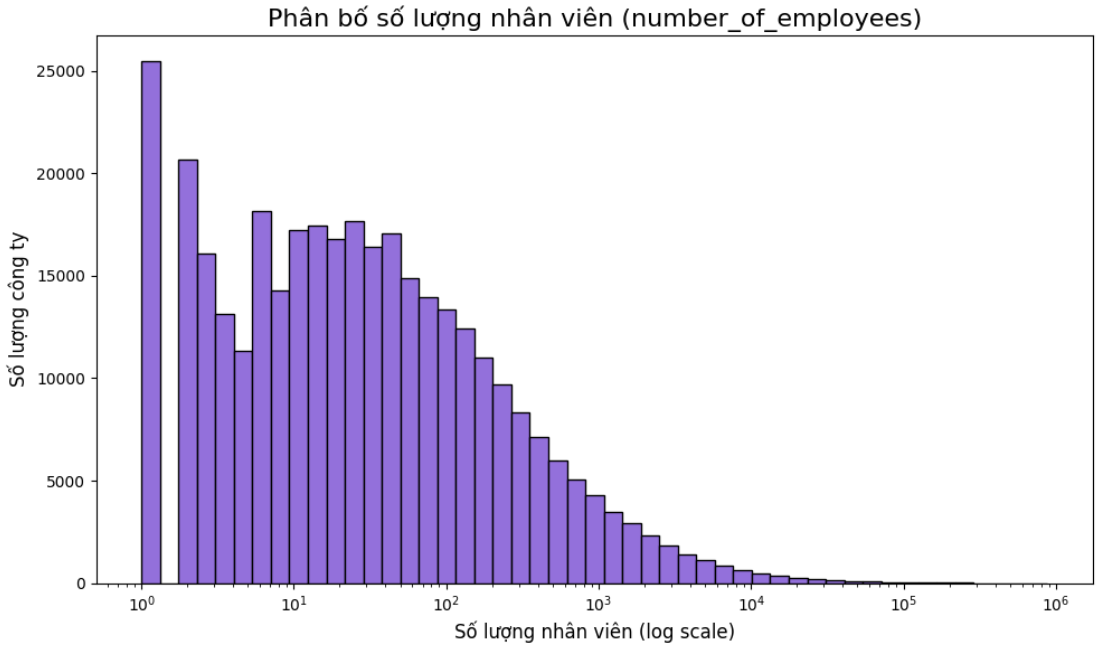
Các giá trị trong cột description đều có độ dài nhỏ hơn 2000 ký tự, phù hợp với mục đích mô tả ngắn gọn về công ty. Dữ liệu ở cột này là dạng văn bản tự do nên thường chứa nhiều ký tự đặc biệt, dấu xuống dòng, hoặc các định dạng khác nhau. Ngoài ra, một số trường hợp chỉ chứa mỗi đường dẫn URL đến trang web của công ty thay vì mô tả nội dung, cho thấy dữ liệu chưa được chuẩn hóa hoàn toàn. Đặc biệt, mô tả có thể xuất hiện bằng nhiều ngôn ngữ khác nhau, không chỉ tiếng Anh, phản ánh tính đa dạng về nguồn gốc công ty trong tập dữ liệu.

```
Row 67 contains non-English text in de: Standard Federal Bank
Row 84 contains non-English text in nl: De Postbank is nu ING Bank.

De ING is een Nederlands onderdeel van de ING Groep, een wereldwijde financiële instelling die
diensten levert op het gebied van bankieren, beleggen, levensverzekeringen en pensioenen. ING
heeft wereldwijd meer dan 85 miljoen klanten in Europa, de Verenigde Staten, Canada, Latijns-
Amerika, Azië en Australië.

Meer informatie over ING Groep kunt u vinden op www.ing.com.
Row 2 contains a URL: https://onlinecareer360.com/
```

Hình 2.8: *Một số ví dụ description công ty.*

number_of_employees

Số nhân viên hiện tại của công ty, là dạng số nguyên, trường này có đầy đủ giá trị.



Hình 2.9: *Phân bố số lượng nhân viên công ty.*

Có 156.011 công ty có số nhân viên bằng 0, chiếm tới 31,2% tổng số bản ghi, cho thấy rất nhiều công ty không thu được thông tin về số lượng nhân viên. Nếu chỉ xét các công ty có số liệu nhân viên, đồ thị trên cho thấy phân bố số lượng

nhân viên trải rộng trên nhiều bậc giá trị, tập trung chủ yếu ở nhóm công ty nhỏ (dưới 100 nhân viên). Số lượng công ty giảm dần khi quy mô nhân sự tăng lên, rất ít công ty có số lượng nhân viên lớn (trên 1.000 hoặc 10.000), phản ánh phần lớn doanh nghiệp trong tập dữ liệu là doanh nghiệp vừa, nhỏ hoặc siêu nhỏ.

logo

Đường dẫn đến hình ảnh logo công ty được lưu trữ trên máy chủ của Linkedin.

slogan

Khẩu hiệu của công ty, là dạng chuỗi ký tự , trường này chỉ có 151.052 bản ghi chứa giá trị (chiếm 30,21%).

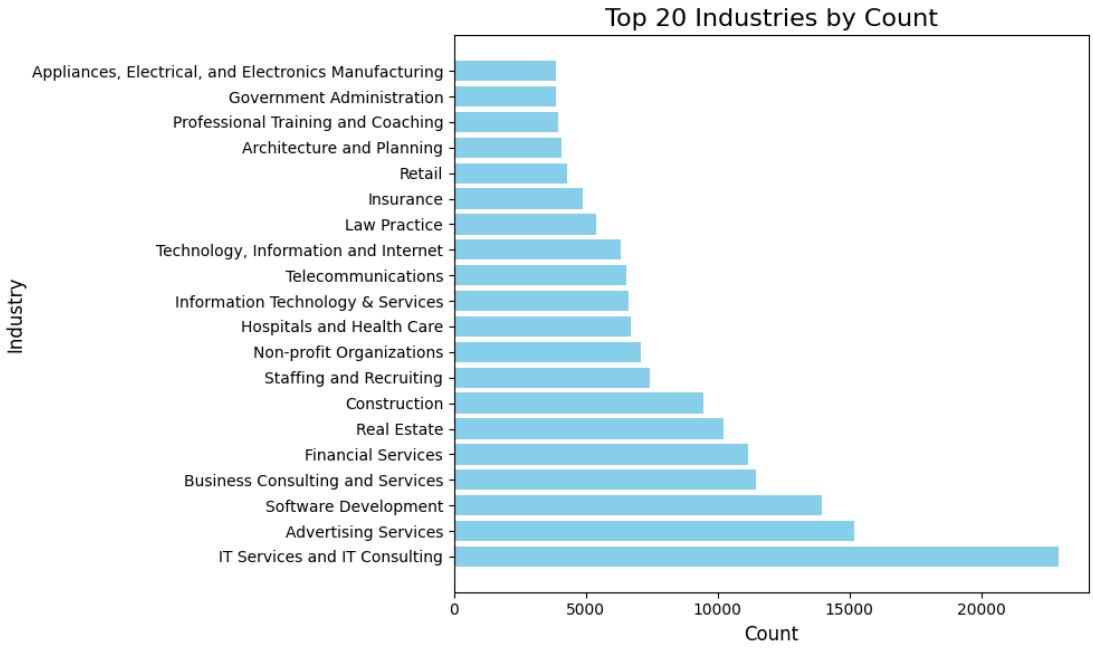
Các giá trị trong cột này là dạng văn bản tự do nên cũng có cùng vấn đề như cột description.

website/ same_as

Hai cột này y hệt nhau và đều là đường dẫn URL đến trang web công ty, là dạng chuỗi ký tự, trường này có 313.738 bản ghi chứa giá trị (chiếm 62,75%).

industry

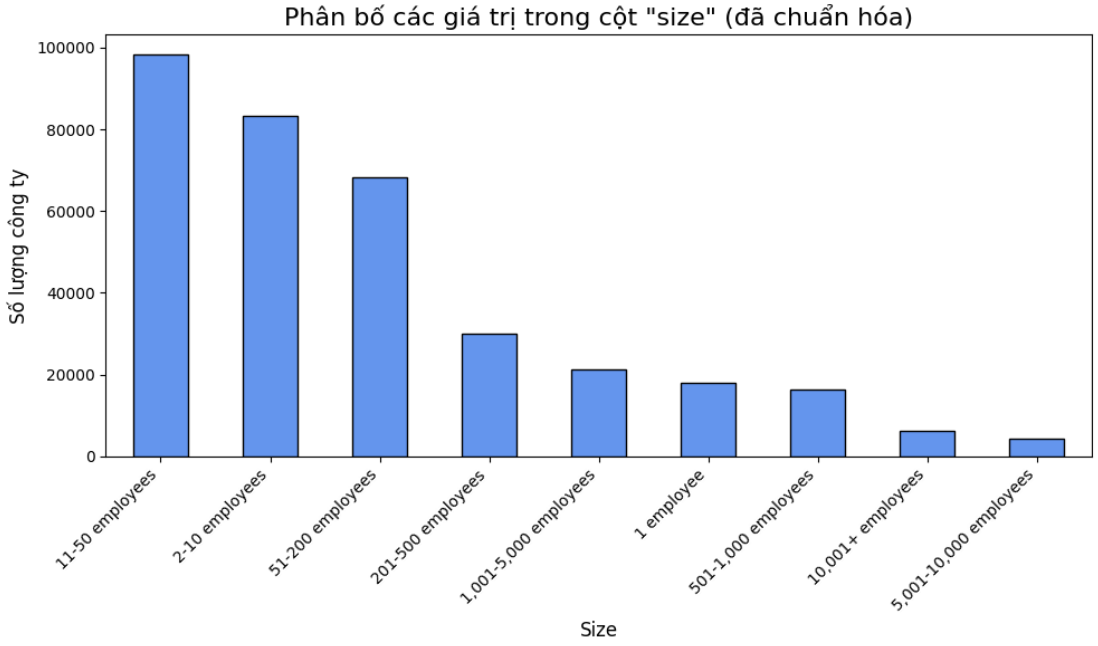
Lĩnh vực hoạt động hoặc ngành nghề kinh doanh chính của từng công ty, được lưu dưới dạng chuỗi ký tự tiếng Anh. Trường này có 346.384 bản ghi chứa giá trị (chiếm 69,28%). Các ngành nghề trong cột này phân bố không đồng đều giữa các công ty.



Hình 2.10: *Top 20 ngành có số công ty nhiều nhất.*

size

Quy mô nhân sự công ty, được lưu dưới dạng chuỗi ký tự. Trường này có 346.109 bản ghi chứa giá trị (chiếm 69,22%).



Hình 2.11: *Phân bố quy mô nhân sự công ty.*

Phần lớn công ty thuộc nhóm quy mô nhỏ và vừa, tập trung ở các mức 11-50, 2-10 và 51-200 nhân viên. Số lượng công ty giảm dần khi quy mô nhân sự tăng lên, rất ít công ty có quy mô trên 1.000 nhân viên. Điều này phản ánh đa số doanh nghiệp trong tập dữ liệu là doanh nghiệp nhỏ và vừa.

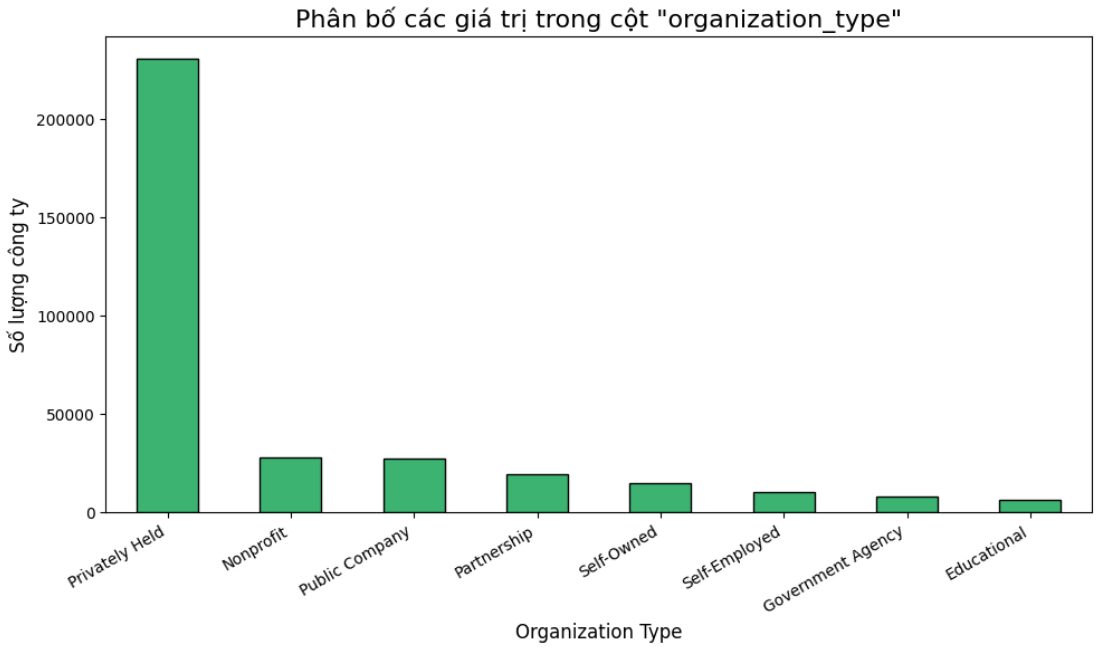
headquarters

Địa điểm trụ sở chính của công ty (thường là dạng thành phố, bang + quốc gia), được lưu dưới dạng chuỗi ký tự, trường này có 238.184 bản ghi chứa giá trị (chiếm 47,64%).

organization_type

Loại hình tổ chức của công ty, được lưu dưới dạng chuỗi ký tự, trường này có 346.296 bản ghi chứa giá trị (chiếm 69,26%).

Phần lớn công ty thuộc loại Privately Held (công ty tư nhân), chiếm ưu thế vượt trội so với các loại hình khác. Các loại hình như Nonprofit (phi lợi nhuận), Public Company (công ty đại chúng), Partnership (hợp danh), Self-Owned, Self-Employed, Government Agency (cơ quan nhà nước) và Educational (giáo dục) có số lượng ít hơn nhiều.



Hình 2.12: *Phân bố loại hình tổ chức công ty.*

founded_on

Năm thành lập công ty, được lưu dưới dạng số, trường này có 225.110 bản ghi chứa giá trị (chiếm 45,02%).

specialties

Các lĩnh vực chuyên môn hoặc dịch vụ cung cấp của công ty, được lưu dưới dạng chuỗi ký tự, trường này chỉ có 218,854 bản ghi chứa giá trị (chiếm 43,77%).

Có thể xem specialties là phần mở rộng, chi tiết hơn cho cột industry, tuy nhiên dữ liệu ở đây là văn bản tự do tương tự như description. Thông thường, các lĩnh vực được liệt kê và phân tách bằng dấu phẩy, nhưng vẫn tồn tại nhiều thứ tiếng hoặc chỉ chứa URL.

	specialties	industry
93095	Home Loan / Mortgages, Insurance, Planning / B...	Financial Services
165196	Electric cars, Operational Lease, Tesla Model ...	Renewable Energy Semiconductor Manufacturing
161700	Relocation Management, Household Goods, Domest...	Truck Transportation
198208	Armored Truck for CIT (Cash in transit) and Ta...	Motor Vehicle Manufacturing
449910	Agricultural Products, Commercial Carriers, El...	Wholesale Building Materials
107346	Anti-idiotype Antibodies, High-Throughput Flow...	Biotechnology Research
324298	Empreendimentos Imobiliários, Imóveis Residenc...	Construction
235044	Naming, Brand design, Brand strategy, Brand id...	Advertising Services
419373	Payroll Software and Payroll Outsourcing	IT Services and IT Consulting
114803	Fiber-Optic Sensor Products	Business Consulting and Services

Hình 2.13: *Một số giá trị specialties tương ứng industry.*

followers

Số người theo dõi linkedln công ty, được lưu dưới dạng chuỗi ký tự (như ”664,569 followers”, ”189 followers”,...), trường này chỉ có 197.896 bản ghi chứa giá trị (chiếm 39,58%).

public_id

Định danh công ty trên linkedln, được lưu dưới dạng chuỗi ký tự (như ”symphony-health-solutions”, ”avinci”,...), trường này có 494.692 bản ghi chứa giá trị (chiếm 98,94%).

locations_raw

Danh sách chi tiết các địa điểm tương ứng với công ty này (có thể là các chi nhánh, trụ sở khác) dưới dạng chuỗi (thường cách nhau bởi dấu ;), trường này có 259.120 bản ghi chứa giá trị (chiếm 51,82%).

2.2.2 Bảng job_posting

Bảng này gồm 500.000 bản ghi với các trường có thể phân tích:

id

Mã định danh cho mỗi bài đăng tuyển công lưu trong cơ sở dữ liệu, là số nguyên, trường này có đầy đủ 500.000 giá trị.

cid

Mã định danh cho mỗi công ty lưu trong cơ sở dữ liệu, là số nguyên như 1000, 1001,... Trường này có đầy đủ 500.000 giá trị. Cột này liên kết với cid của bảng company_detail.

url

Đường dẫn đến bài đăng tuyển dụng trên linkedln, là dạng chuỗi ký tự, trường này có đầy đủ 500.000 giá trị.

company_url

Đường dẫn đến trang linkedln của công ty, là dạng chuỗi ký tự, trường này có đầy đủ 500.000 giá trị.

company_name

Tên công ty, là dạng chuỗi ký tự, trường này có đầy đủ 500.000 giá trị.

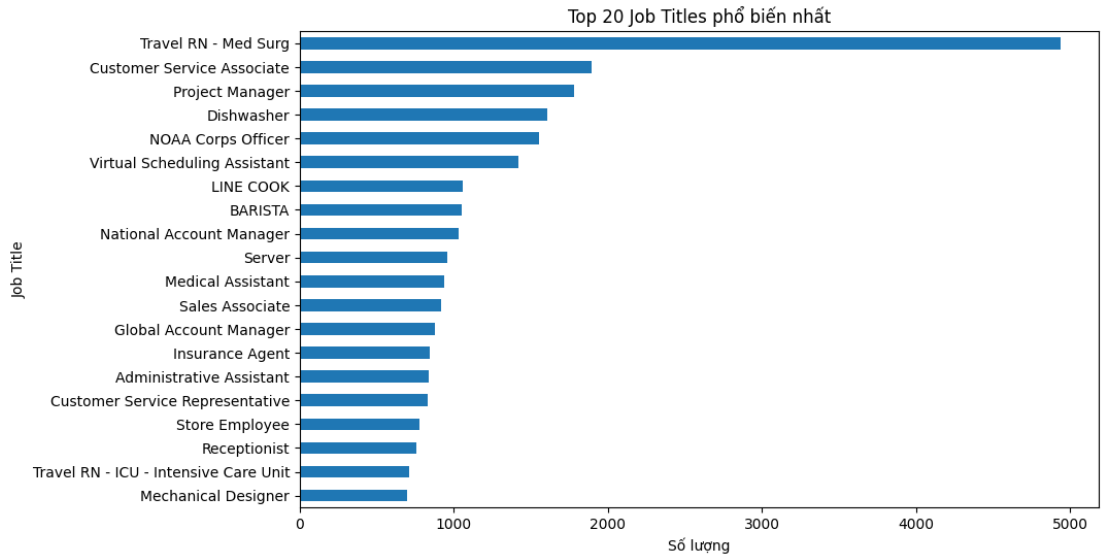
title

Tên vị trí công việc được đăng tuyển, là dạng chuỗi ký tự, trường này có đầy đủ 500.000 giá trị.

Cột này có 201.107 giá trị khác nhau, số lượng khá nhiều bởi vì mỗi công ty sẽ có một cách viết tên vị trí công việc khác nhau, kể cả khi ý nghĩa tổng thể theo tên là giống nhau.

```
Travel RN - Med Surg
4938
Customer Service Associate
1893
Project Manager
1783
Dishwasher
1609
NOAA Corps Officer
1550

...
-Get Hired and earn a for our Caregivers and Homemakers Position
(Non CNA or Experience Required)       1
Lead Outpatient Services Counselor- LCASA/A - Flexible Schedule
1
Medical Assistant for Family Care Office
1
Adjunct Instructor- Art of China
1
Project Engineer - Co-op Program
1
```

Hình 2.14: *Một số giá trị job_title tương ứng.*



Hình 2.15: *Top 20 vị trí công việc đăng tuyển nhiều.*

Biểu đồ trên thể hiện top 20 vị trí công việc được đăng tuyển nhiều nhất trong dữ liệu. Vị trí ”Travel RN - Med Surg” có số lượng vượt trội, cho thấy nhu cầu rất lớn về điều dưỡng viên đi công tác trong lĩnh vực y tế. Các vị trí tiếp theo như ”Customer Service Associate”, ”Project Manager”, ”Dishwasher”, ”NOAA

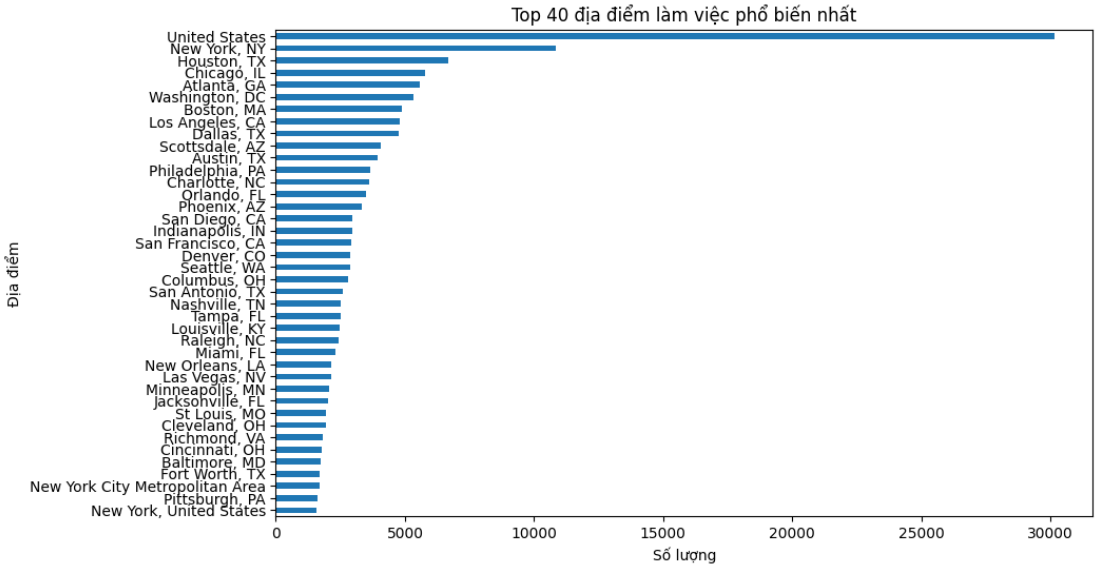
Corps Officer", "Virtual Scheduling Assistant" cũng có số lượng đăng tuyển cao, phản ánh nhu cầu lớn trong các ngành dịch vụ khách hàng, quản lý dự án, nhà hàng, tổ chức chính phủ và hỗ trợ từ xa.

Ngoài ra, các vị trí như "BARISTA", "LINE COOK", "Server" cho thấy nhu cầu lao động phổ thông trong ngành dịch vụ ăn uống cũng rất lớn. Các vị trí như "Medical Assistant", "Sales Associate", "Insurance Agent", "Administrative Assistant" là những công việc phổ biến, phù hợp với nhiều đối tượng lao động khác nhau.

Sự đa dạng của các job title cho thấy dữ liệu bao phủ nhiều lĩnh vực: y tế, dịch vụ, hành chính, kỹ thuật, bán lẻ, ... Một số vị trí có tên gọi gần giống nhau hoặc khác biệt nhỏ.

location

Địa điểm làm việc theo bài tuyển dụng, được lưu dưới dạng chuỗi ký tự, trường này có 499.998 bản ghi chứa giá trị (chiếm 99,99%).



Hình 2.16: *Top 40 địa điểm làm việc phổ biến.*

Dữ liệu trong cột location có nhiều dạng khác nhau, phổ biến nhất gồm:

- Tên quốc gia: Ví dụ "United States". Đây là trường hợp chỉ ghi tên quốc gia mà không có thông tin chi tiết về bang hoặc thành phố.
- Tên thành phố, mã bang: Ví dụ "New York, NY", "Houston, TX", "Chicago, IL". Đây là dạng chuẩn nhất, gồm tên thành phố và mã bang (2 ký tự viết hoa theo chuẩn Mỹ), giúp xác định rõ địa điểm làm việc.
- Tên bang, United States: Ví dụ "California, United States", "Maryland, United States". Dạng này chỉ rõ tên bang nhưng không ghi thành phố cụ thể.

- Tên vùng đô thị/khu vực: Ví dụ ”Dallas-Fort Worth Metroplex”, ”Greater Orlando”, ”Iowa City-Cedar Rapids Area”. Đây là các vùng đô thị lớn hoặc khu vực liên thành phố, không phải một thành phố riêng lẻ.

Như vậy, dữ liệu location vừa có thể rất cụ thể (tới thành phố, bang), vừa có thể chung chung (chỉ bang hoặc quốc gia), hoặc mô tả theo vùng/khu vực. Ngoài ra có thể có nhiều biến thể tên địa điểm cho cùng một nơi.

salary

Mức lương được đề xuất cho vị trí tuyển dụng, là dạng chuỗi ký tự, trường này chỉ có 30.321 bản ghi chứa giá trị (chiếm 6,06%).

Hình 2.17: *Tỉ lệ giá trị salary rỗng.*

Trường salary gần như toàn bộ là giá trị rỗng có thể do phần lớn nhà tuyển dụng không công khai mức lương cụ thể trên tin tuyển dụng, mà chỉ trao đổi khi phỏng vấn hoặc thương lượng trực tiếp với ứng viên hoặc có thể do lỗi trong quá trình thu thập, trích xuất dữ liệu từ website gốc.

num_applicants

Số lượng ứng viên đã ứng tuyển vào vị trí , là dạng chuỗi ký tự, trường này có 499.993 bản ghi chứa giá trị (chiếm 99,99%).

Các giá trị trong cột được biểu diễn dưới dạng: Dưới 25 ứng viên (Be among the first 25 applicants), các giá trị cụ thể số ứng viên từ 25 đến 200 (25 applicants,...) và trên 200 ứng viên (Over 200 applicants)

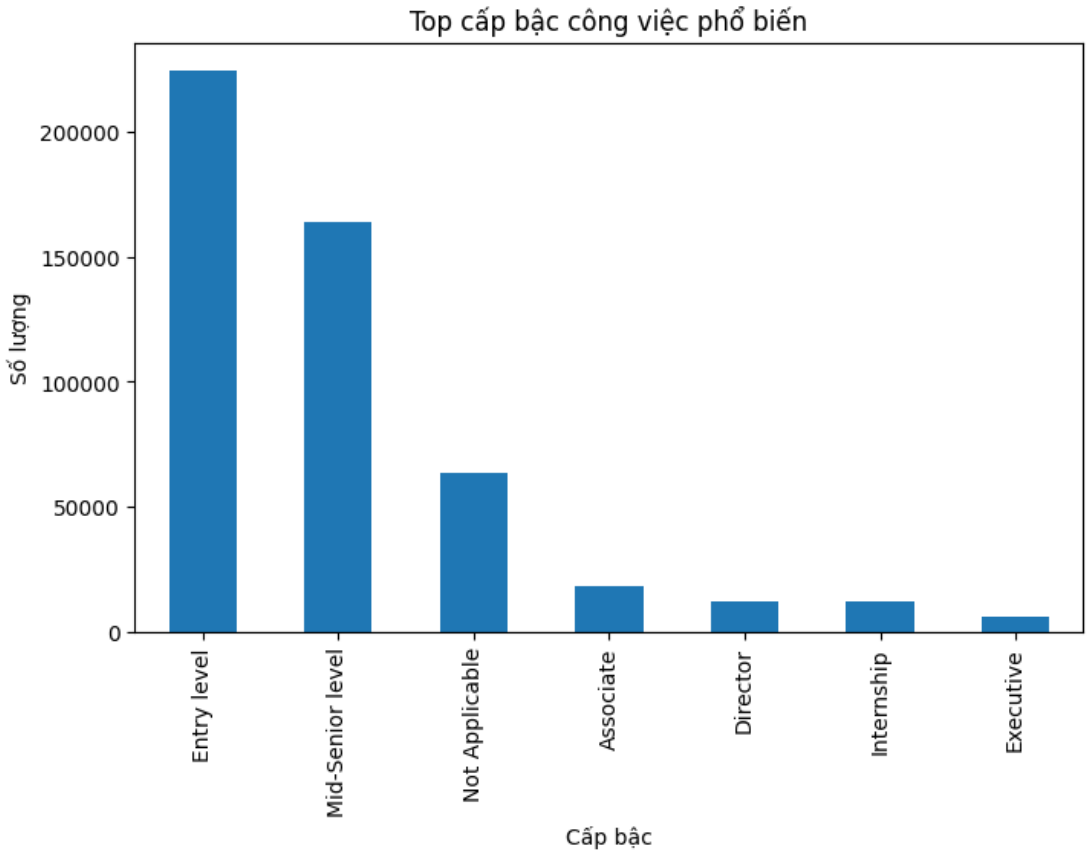
description

Mô tả chi tiết về công việc, là dạng chuỗi ký tự, trường này có 499.993 bản ghi chứa giá trị (chiếm 99,99%).

Ngoài các vấn đề tương tự description của bảng company_detail, gần như tất cả dữ liệu vẫn còn chứa các thẻ HTML khi tiến hành thu thập dữ liệu về, đòi hỏi cần xử lý thêm nếu muốn sử dụng.

seniority_level

Cấp bậc vị trí tuyển dụng, là dạng chuỗi ký tự, trường này có 499.993 bản ghi chứa giá trị (chiếm 99,99%).



Hình 2.18: *Các cấp bậc công việc.*

Phần lớn các vị trí tuyển dụng thuộc cấp bậc ”Entry level” (mới vào nghề) và ”Mid-Senior level” (trung cấp đến cao cấp), chiếm số lượng áp đảo so với các cấp bậc khác. Điều này phản ánh nhu cầu tuyển dụng lớn đối với các vị trí dành cho người mới đi làm hoặc có một vài năm kinh nghiệm.

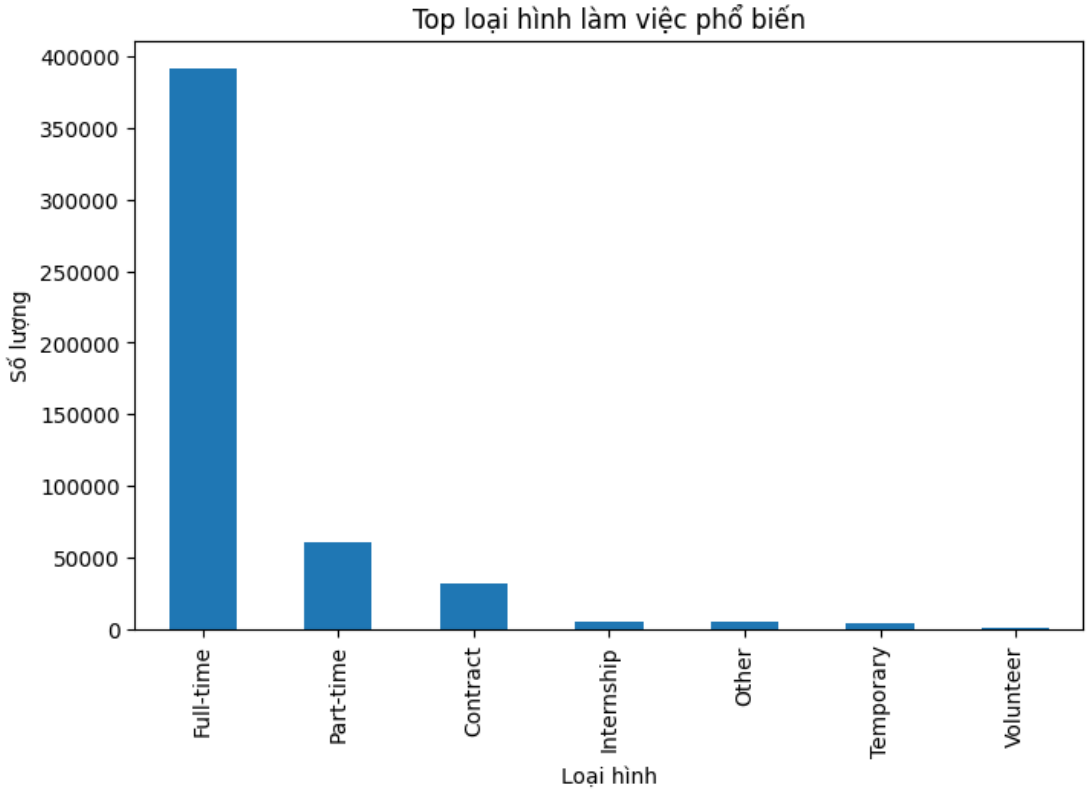
Cấp bậc ”Not Applicable” cũng chiếm tỷ trọng đáng kể, cho thấy nhiều tin tuyển dụng không chỉ rõ cấp bậc hoặc không áp dụng phân loại này.

Các cấp bậc như ”Associate”, ”Director”, ”Internship” và ”Executive” có số lượng thấp hơn nhiều. Đặc biệt, số lượng vị trí thực tập (Internship) và quản lý

cấp cao (Executive) khá ít, phù hợp với thực tế là các vị trí này thường ít được tuyển dụng hơn so với các vị trí phổ thông hoặc trung cấp.

employment_type

Loại hình làm việc, là dạng chuỗi ký tự, trường này có 499.993 bản ghi chứa giá trị (chiếm 99,99%).



Hình 2.19: *Các loại hình làm việc.*

Loại hình làm việc phổ biến nhất là ”Full-time” (toàn thời gian), chiếm áp đảo so với các loại hình khác. Các loại hình ”Part-time” (bán thời gian) và ”Contract” (hợp đồng) cũng có số lượng đáng kể.

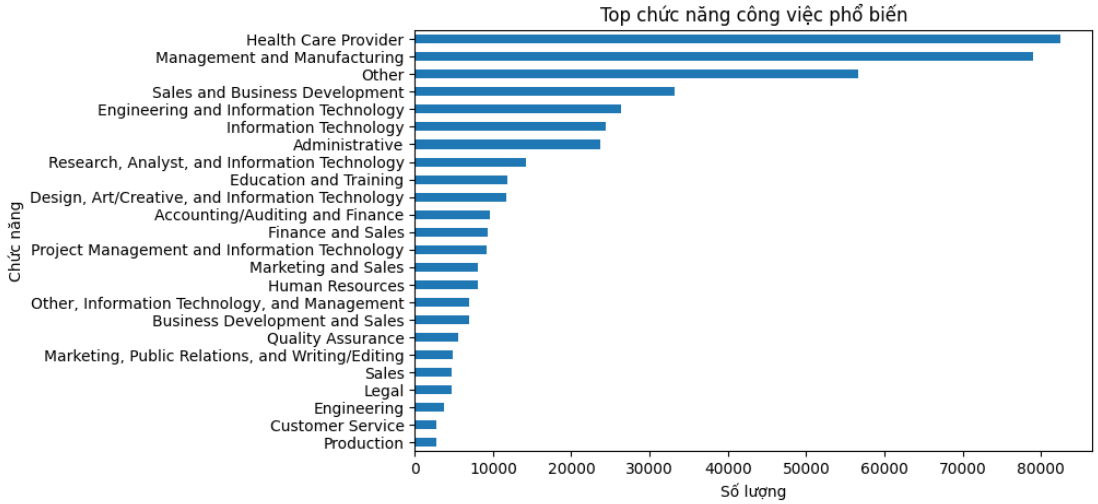
Các loại hình khác như ”Internship” (thực tập), ”Other”, ”Temporary” (thời vụ), và ”Volunteer” (tình nguyện) có số lượng thấp hơn nhiều. Điều này phù hợp với thực tế là các vị trí thực tập, thời vụ hoặc tình nguyện thường chỉ chiếm một phần nhỏ trong tổng số cơ hội việc làm.

job_function

Lĩnh vực/ chức năng chính của công việc, là dạng chuỗi ký tự, trường này có 499.164 bản ghi chứa giá trị (chiếm 99,83%).

Có 4342 giá trị khác nhau, mỗi giá trị có thể là một chức năng cụ thể (ví dụ: ”Health Care Provider”) hoặc kết hợp nhiều chức năng cùng lúc ngăn cách

nhau bởi dấu phẩy (ví dụ: ”Information Technology, Art/Creative, and Project Management”).



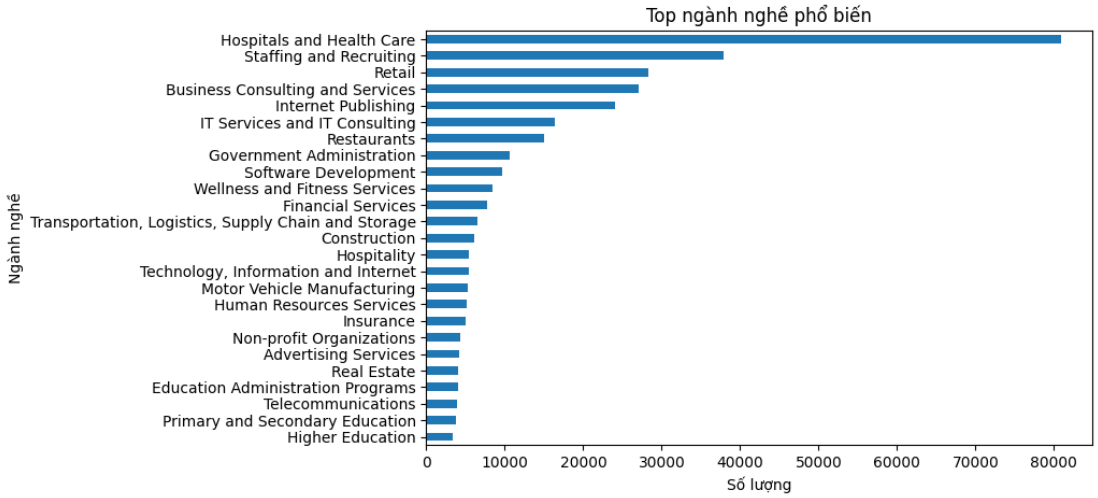
Hình 2.20: *Top chức năng công việc phổ biến.*

Sự xuất hiện của giá trị ”Other” với số lượng lớn cho thấy có nhiều vị trí không thuộc các nhóm chức năng cụ thể hoặc được phân loại chung chung.

industries

Ngành nghề liên quan đến vị trí làm việc, là dạng chuỗi ký tự, trường này có 499.940 bản ghi chứa giá trị (chiếm 99,99%).

Có 6209 giá trị khác nhau, mỗi giá trị có thể là một ngành nghề cụ thể (ví dụ: ”Hospitals and Health Care”) hoặc là tổ hợp nhiều ngành nghề, phân tách bằng dấu phẩy (ví dụ: ”Banking, Leasing Non-residential Real Estate, and Real Estate”).



Hình 2.21: *Top ngành nghề công việc phổ biến.*

2.2.3 Bảng profile_info

Bảng này gồm 510.000 bản ghi với các trường có thể phân tích:

public_id

Định danh người dùng trên linkedln, được lưu dưới dạng chuỗi ký tự (như ”jeremypecina24”, ”xun-jia-8602655”,...), Trường này có đầy đủ 510.000 giá trị.

country_code

Mã quốc gia (2 ký tự) của người dùng, được lưu dưới dạng chuỗi ký tự, Trường này có đầy đủ 510.000 giá trị.

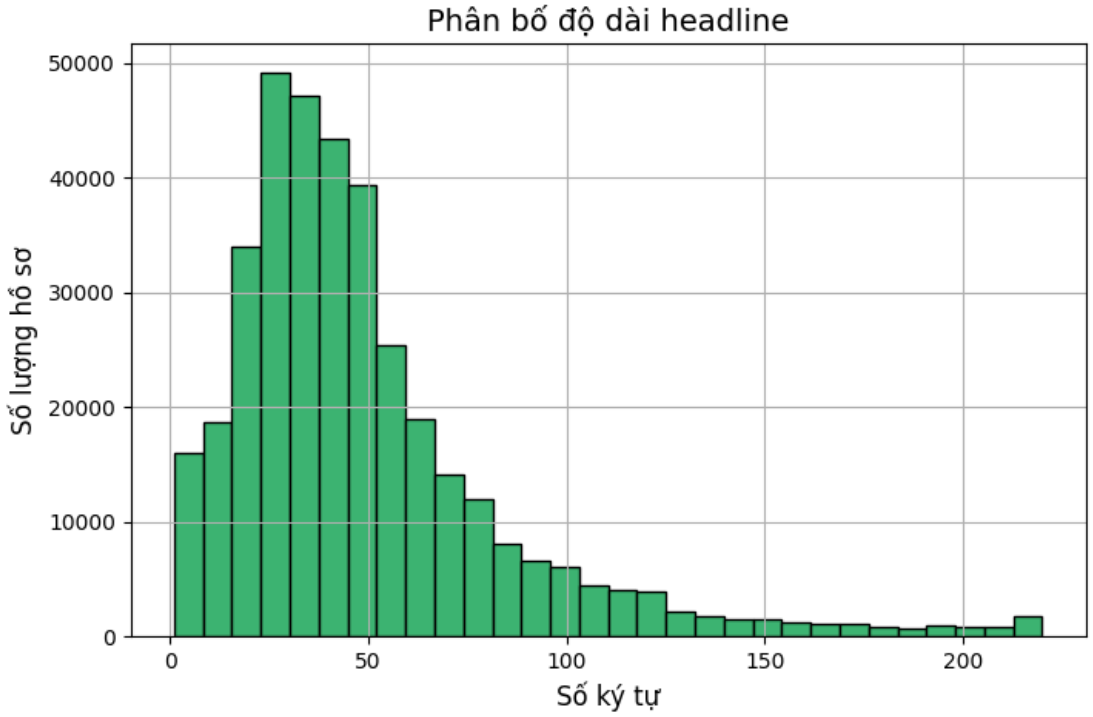
Ngoài các giá trị phổ biến như uk, br, in,... cột này có tới 190.561 bản ghi giá trị www không phải mã của quốc gia nào, đây có thể là dữ liệu lỗi, dữ liệu bị nhập sai hoặc bị trích xuất nhầm từ các trường khác.

full_name

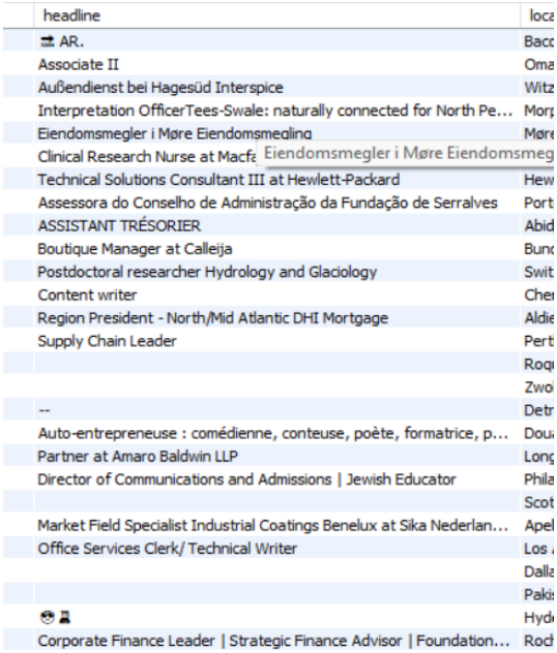
Họ tên đầy đủ của người dùng, được lưu dưới dạng chuỗi ký tự, trường này có đầy đủ 510.000 giá trị.

headline

Tiêu đề mô tả ngắn về bản thân hoặc vị trí công việc hiện tại của người dùng (thường xuất hiện dưới tên trên LinkedIn), được lưu dưới dạng chuỗi ký tự, trường này có 367.636 bản ghi chứa giá trị (chiếm 72,08%).



Hình 2.22: *Phân bố độ dài headline.*



headline	loca
AR.	Bacc
Associate II	Oma
Außendienst bei Hagesüd Interspice	Witz
Interpretation OfficerTees-Swale: naturally connected for North Pe...	Morp
Eiendomsmegler i Møre Eiendomsmegling	Møre
Clinical Research Nurse at Macfa	Eiendomsmegler i Møre Eiendomsmeg
Technical Solutions Consultant III at Hewlett-Packard	Hew
Assessora do Conselho de Administração da Fundação de Serralves	Port
ASSISTANT TRÉSORIER	Abid
Boutique Manager at Calleija	Bunc
Postdoctoral researcher Hydrology and Glaciology	Swit
Content writer	Cher
Region President - North/Mid Atlantic DHI Mortgage	Aldie
Supply Chain Leader	Pert
	Rogu
	Zwo
--	Detr
Auto-entrepreneuse : comédienne, conteuse, poète, formatrice, p...	Doua
Partner at Amaro Baldwin LLP	Long
Director of Communications and Admissions | Jewish Educator	Phila
	Scot
Market Field Specialist Industrial Coatings Benelux at Sika Nederlan...	Apel
Office Services Clerk/ Technical Writer	Los
	Dalla
	Paki
	Hyd
Corporate Finance Leader | Strategic Finance Advisor | Foundation...	Roch

Hình 2.23: *Một số giá trị headline.*

Các giá trị trong cột này có các đặc điểm sau:

- Giá trị phổ biến: Chủ yếu là các chức danh công việc chuẩn, ví dụ: Project Manager, Software Engineer, Registered Nurse, Director, Consultant, ... hoặc trình trạng nghỉ việc (Retired)
- Giá trị đặc biệt/ngắn: Có nhiều giá trị không hợp lệ hoặc không mang ý nghĩa, ví dụ: –, ., -, at, hoặc chỉ là ký tự đặc biệt, icon, dấu gạch ngang.
- Giá trị dài: Một số headline rất dài, có thể liệt kê nhiều chức danh, giải thưởng, mô tả chi tiết hoặc chứa nhiều ngôn ngữ khác nhau.
- Ngôn ngữ: Chủ yếu là tiếng Anh, nhưng cũng có thể xuất hiện các headline bằng ngôn ngữ khác.

location

Địa điểm sinh sống/làm việc của người dùng (có thể bao gồm thành phố, bang, quốc gia), được lưu dưới dạng chuỗi ký tự, trường này có 509.938 bản ghi chứa giá trị (chiếm 99,99%).

Các giá trị trong cột chủ yếu có thành phố, bang/vùng, quốc gia, tùy bản ghi có thể không đầy đủ 3 thông tin trên. Ngoài ra, có một số giá trị là tên tổ chức, trường học, công ty (ví dụ: HTWG Hochschule Konstanz – Technik, Wirtschaft und Gestaltung, Segue Financial Services), không phải địa danh thực sự.

summary

Phần mô tả về bản thân do người dùng tự viết, có thể bao gồm kinh nghiệm,

học vấn, kĩ năng,... của người dùng, được lưu dưới dạng chuỗi ký tự, trường này có 300.785 bản ghi chứa giá trị (chiếm 58,98%).

Là văn bản viết tự do nên cũng có các vấn đề tương tự description của bảng company_detail như chứa nhiều ký tự đặc biệt, dấu xuống dòng, nhiều ngôn ngữ ngoài tiếng Anh hay chỉ chứa đường dẫn URL đến 1 trang web khác mà người dùng tự tạo.

2.2.4 Bảng experience

Bảng này gồm 2.036.181 bản ghi với các trường có thể phân tích:

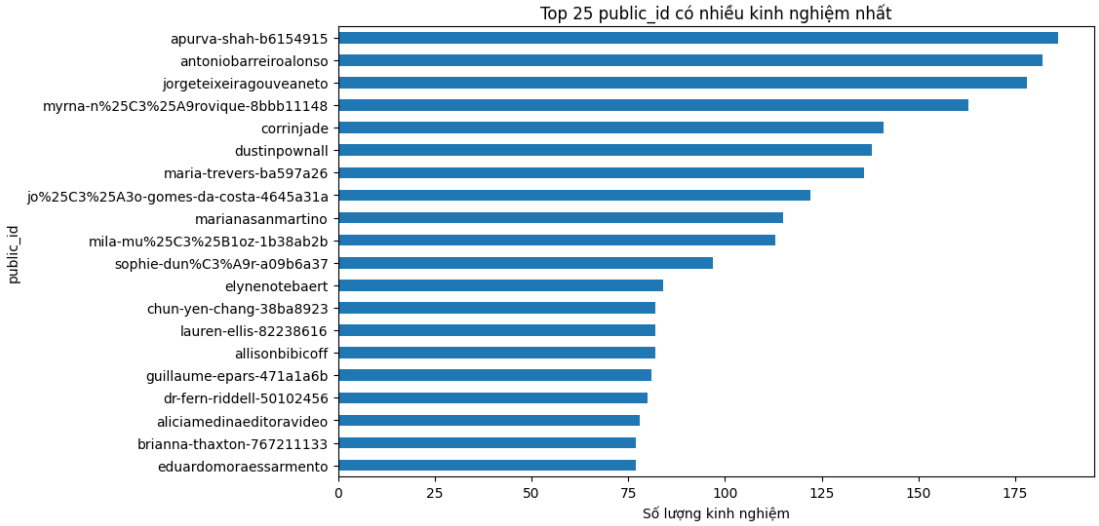
id

Mã định danh cho mỗi kinh nghiệm làm việc của một người lưu trong cơ sở dữ liệu, là số nguyên, trường này có đầy đủ giá trị.

public_id

Định danh người dùng trên linkedln, được lưu dưới dạng chuỗi ký tự (như ”jeremypecina24”, ”xun-jia-8602655”,...), trường này có đầy đủ 2.036.181 giá trị. Cột này liên kết với public_id của bảng profile_info.

Dữ liệu có 341.607 giá trị mã định danh người dùng khác nhau, với top 25 người dùng có số lượng bản ghi kinh nghiệm nhiều nhất trong dữ liệu:



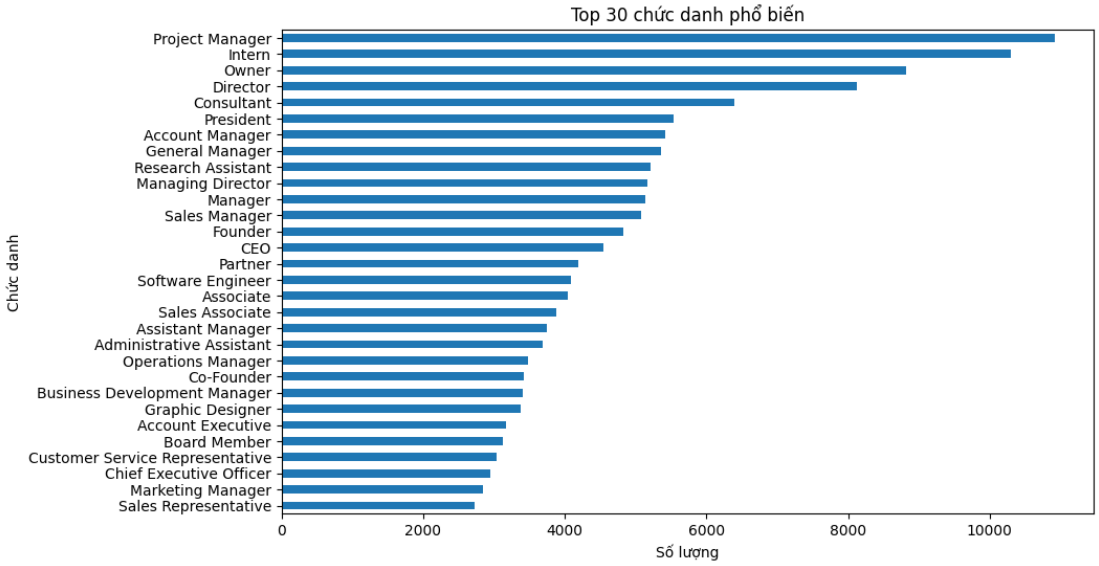
Hình 2.24: *Top 25 người dùng có số lượng bản ghi kinh nghiệm nhiều nhất.*

title

Tên vị trí/ chức danh công việc mà người dùng từng/ đang đảm nhận, là dạng chuỗi ký tự, trường này có 2.035.927 giá trị (chiếm 99,99%).

Cột này có 882.877 giá trị khác nhau, số lượng khá nhiều bởi vì mỗi người

dùng sẽ có một cách viết tên vị trí công việc khác nhau, có thể đề cập cả chức danh và vị trí của người đó (ví dụ ”UK Accounting Manager (#1)”, ”CS111 Exam Grader & Proctor”,...) hoặc viết theo nhiều thứ tiếng.



Hình 2.25: *Top 30 chức danh phổ biến trong kinh nghiệm làm việc.*

company

Tên công ty nơi người dùng từng làm việc, là dạng chuỗi ký tự, trường này có 2.035.686 bản ghi chứa giá trị (chiếm 99,97%)

company_id

Định danh công ty trên linkedln, được lưu dưới dạng chuỗi ký tự, trường này có 1.419.018 bản ghi chứa giá trị (chiếm 69,69%)

website

Đường dẫn URL đến trang web công ty, là dạng chuỗi ký tự, trường này có 1.309.901 bản ghi chứa giá trị (chiếm 64,33%).

location

Địa điểm làm việc, được lưu dưới dạng chuỗi ký tự, trường này chỉ có 380.515 bản ghi chứa giá trị (chiếm 18,69%).

Tương tự như location trong bảng job_posting, dữ liệu trong cột cũng có nhiều dạng khác nhau, có thể chứa tên thành phố, mã bang, vùng/ bang, quốc gia,... Có thể có nhiều biến thể tên địa điểm cho cùng 1 nơi.

description

Mô tả chi tiết về công việc đã làm, là dạng chuỗi ký tự, trường này có 1.056.736 bản ghi chứa giá trị (chiếm 51,9%).

Mô tả dưới dạng văn bản tự do nên cũng có vấn đề tương tự description của

bảng company_detail như chứa ký tự đặc biệt, nhiều thứ tiếng, chứa đường dẫn URL,...

start_date và end_date

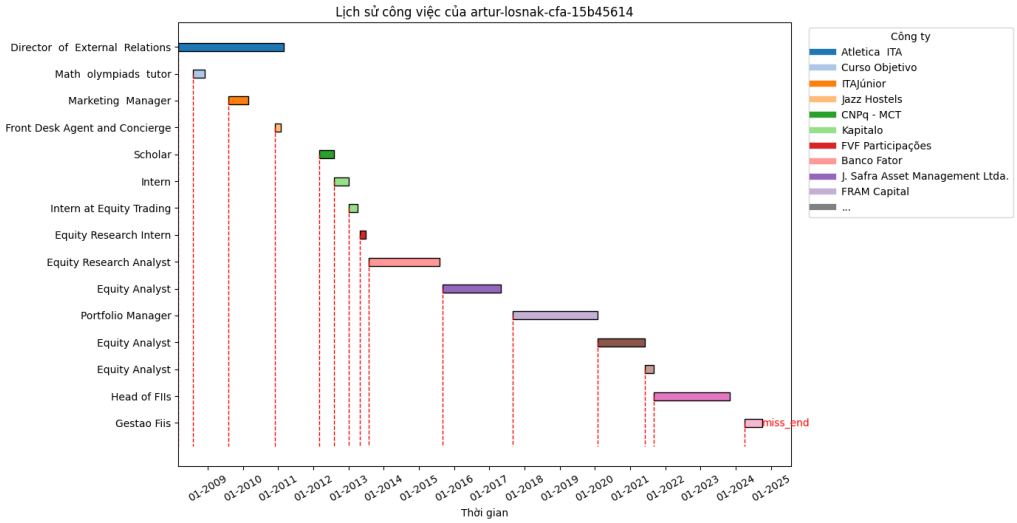
Đây là 2 trường thời gian cho biết thời gian bắt đầu và kết thúc một công việc, hay một kinh nghiệm của một người dùng, là dạng chuỗi ký tự thời gian, trường start_date có đầy đủ bản ghi chứa giá trị nhưng end_date chỉ có 1.566.495 bản ghi chứa giá trị (chiếm 76,93%), phản ánh một điều là có thể người dùng không thường xuyên cập nhật thời gian kết thúc một công việc lên profile linkedln, hoặc có thể vẫn đang làm công việc đó.

Các giá trị trên hai trường này có thể rơi vào các trường hợp sau. Ta gán nhãn và minh họa bằng đồ thị thời gian cho một số người dùng:

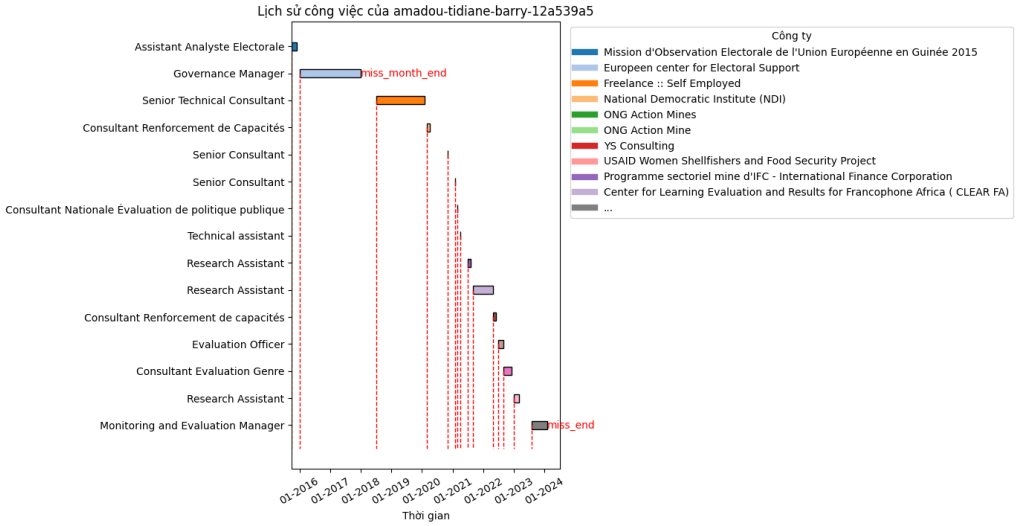
- Trường hợp `start_date` và `end_date` đều có định dạng đầy đủ (YYYY-MM-DD) hoặc chỉ thiếu ngày (YYYY-MM): Không gán nhãn đặc biệt, vẽ bình thường.
- Trường hợp `end_date` bị thiếu (null) và `start_date` chỉ có năm (YYYY): Gán `start_date` là tháng 1 của năm đó, `end_date` là ngày lớn nhất trong các ngày bắt đầu/ kết thúc kinh nghiệm người dùng +6 tháng. Gán nhãn `miss_end_month_start`.
- Trường hợp `end_date` bị thiếu (null) và `start_date` đủ tháng-năm trở lên: Gán `end_date` là ngày lớn nhất trong các ngày bắt đầu/ kết thúc kinh nghiệm người dùng +6 tháng. Gán nhãn `miss_end`.
- Trường hợp `start_date` đủ tháng-năm trở lên nhưng `end_date` chỉ có năm (YYYY): Gán `end_date` là tháng 12 của năm đó. Gán nhãn `miss_month_end`.
- Trường hợp `end_date` đủ tháng-năm trở lên nhưng `start_date` chỉ có năm (YYYY): Gán `start_date` là tháng 1 của năm đó. Gán nhãn `miss_month_start`.
- Trường hợp cả hai trường chỉ có năm (YYYY): Gán `start_date` là tháng 1, `end_date` là tháng 12 của năm tương ứng. Gán nhãn `miss_month_start_end`.

Qua phân tích chung về dữ liệu và một số biểu đồ, có thể thấy:

- Dữ liệu về thời gian làm việc không đồng nhất, không phải lúc nào cũng ở dạng chuẩn ngày-tháng-năm. Có nhiều trường hợp chỉ có năm hoặc chỉ có tháng-năm, thậm chí có trường hợp thiếu hoàn toàn ngày kết thúc.
- Quan sát trên biểu đồ, có thể thấy tại cùng một thời điểm, một người có thể đảm nhận nhiều vị trí công việc khác nhau ở nhiều công ty. Điều này thể hiện tính đa dạng trong quá trình làm việc, đồng thời cũng đặt ra yêu cầu phải xử lý chồng lấn thời gian khi phân tích dữ liệu kinh nghiệm.



Hình 2.26: *Biểu đồ thời gian làm việc của người dùng 1.*



Hình 2.27: *Biểu đồ thời gian làm việc của người dùng 2.*

Hình 2.28: *Biểu đồ thời gian làm việc của người dùng 3.*

2.3 Lựa chọn thuộc tính

Sau khi phân tích dữ liệu gốc, nhóm chỉ giữ lại các trường sau từ hai bảng chính để phục vụ cho toàn bộ pipeline xử lý và phân tích về sau:

Bảng experience

- id: Mã định danh từng kinh nghiệm, dùng để quản lý và truy xuất dữ liệu.
- public_id: Mã định danh ứng viên, dùng để liên kết với bảng thông tin ứng viên.
- title: Chức danh/vị trí công việc mà ứng viên đã đảm nhận. Trường này là cơ sở để xác định vai trò nghề nghiệp, phục vụ trích xuất thông tin như job_role hoặc so khớp với yêu cầu công việc.
- description: Mô tả chi tiết công việc. Đây là nguồn dữ liệu chính để trích xuất các trường như skill, experience, job_role bằng các mô hình ngôn ngữ lớn (LLM) trong các bước tiếp theo.
- start_date, end_date: Thời gian bắt đầu và kết thúc công việc, dùng để tính toán số năm kinh nghiệm, kiểm tra tính liên tục, xác định mức độ cập nhật của kinh nghiệm.

Bảng profile_info

- public_id: Mã định danh ứng viên, liên kết với bảng experience.
- headline: Tiêu đề nghề nghiệp (headline LinkedIn), thường tóm tắt vai trò hoặc định hướng nghề nghiệp của ứng viên.
- summary: Tóm tắt hồ sơ cá nhân, chứa thông tin tổng quan về kỹ năng, kinh nghiệm, mục tiêu nghề nghiệp.

Lý do lựa chọn các trường này:

- Các trường title, description, headline, summary là nguồn dữ liệu văn bản tự do, có thể sử dụng các mô hình ngôn ngữ lớn (LLM) để tự động trích xuất thông tin cấu trúc như vai trò công việc (job_role), kỹ năng (skill), kinh nghiệm (experience). Điều này giúp chuẩn hóa dữ liệu và tăng khả năng tự động hóa trong phân tích, xếp hạng ứng viên.
- Các trường về thời gian (start_date, end_date) giúp đánh giá quá trình phát triển nghề nghiệp, tính toán số năm kinh nghiệm, và xác định mức độ cập nhật của ứng viên.
- Trường public_id đảm bảo khả năng liên kết dữ liệu giữa hai bảng, phục vụ cho việc tổng hợp thông tin ứng viên và xây dựng pipeline xử lý về sau.

Việc giữ lại các trường này giúp tập trung vào các thông tin quan trọng nhất, đồng thời đơn giản hóa quá trình tiền xử lý, trích xuất thông tin và xây dựng mô hình xếp hạng độ phù hợp giữa ứng viên và vị trí công việc.

2.4 Định hướng giải quyết đề tài

Dự án hướng tới xây dựng một hệ thống xếp hạng độ phù hợp giữa ứng viên với vị trí công việc trên LinkedIn, dựa trên những nghiên cứu thực tế, phân tích dữ liệu có được và các kỹ thuật hiện đại về xử lý ngôn ngữ tự nhiên, đồ thị tri thức và học sâu. Pipeline giải quyết bài toán gồm các bước chính sau:

1. **Tiền xử lý dữ liệu**

   Dữ liệu thô từ LinkedIn thường tồn tại dưới dạng văn bản tự do, không đồng nhất về ngôn ngữ, định dạng và mức độ chi tiết. Vì vậy, bước đầu tiên là tiền xử lý dữ liệu để đảm bảo chất lượng cho các bước tiếp theo.

   Cụ thể, dữ liệu sẽ được làm sạch bằng cách loại bỏ các ký tự đặc biệt, đường dẫn, chuẩn hóa khoảng trắng và xử lý các trường hợp thiếu thông tin. Nếu phát hiện ngôn ngữ khác tiếng Anh, hệ thống sẽ tự động dịch sang tiếng Anh để đồng nhất dữ liệu đầu vào.

   Các trường ngày tháng như start_date, end_date cũng được chuẩn hóa về cùng một định dạng, đồng thời xử lý các trường hợp thiếu hoặc không hợp lệ. Việc này giúp giảm nhiễu, tránh lỗi khi phân tích và đảm bảo dữ liệu đầu vào đồng nhất, thuận tiện cho các bước xử lý tiếp theo.
2. **Trích xuất thông tin cấu trúc từ văn bản tự do**

   Các trường như title, description, headline, summary đều là văn bản tự do, chứa nhiều thông tin nhưng không có cấu trúc rõ ràng. Để khai thác hiệu quả, hệ thống sử dụng các mô hình ngôn ngữ lớn (LLM) để tự động trích xuất các trường thông tin cấu trúc như vai trò công việc (job_role), kỹ năng (skill), kinh nghiệm (experience).

   Việc chuyển đổi từ dữ liệu không cấu trúc sang dữ liệu có cấu trúc giúp chuẩn hóa thông tin, đồng thời tạo nền tảng cho các bước phân tích tự động và so khớp về sau. Đây là ứng dụng của kỹ thuật Information Extraction trong xử lý ngôn ngữ tự nhiên, giúp rút trích tri thức từ văn bản và biến dữ liệu thô thành các đặc trưng có thể sử dụng cho mô hình hóa.
3. **Xây dựng đồ thị tri thức (Knowledge Graph)**

   Sau khi đã có dữ liệu cấu trúc, toàn bộ thông tin về ứng viên, kinh nghiệm, kỹ năng, vai trò... được biểu diễn dưới dạng các thực thể (node) và

mối quan hệ (edge) trong một đồ thị tri thức.

Đồ thị tri thức là mô hình dữ liệu cho phép lưu trữ và truy vấn các mối liên hệ phức tạp giữa các thực thể, rất phù hợp cho các bài toán gợi ý và xếp hạng đa chiều. Trong đồ thị này, mỗi ứng viên, kỹ năng, vai trò công việc... là một node, và các mối quan hệ như "ứng viên có kỹ năng", "ứng viên từng đảm nhận vai trò", "công việc yêu cầu kỹ năng" được biểu diễn bằng các cạnh nối giữa các node.

Lý thuyết về đồ thị tri thức cho phép tận dụng triệt để các mối liên hệ gián tiếp, tăng khả năng suy luận và mở rộng tri thức, giúp hệ thống hiểu sâu hơn về mối liên hệ giữa ứng viên và công việc.

4. **Sinh embedding cho node và quan hệ**

   Để có thể áp dụng các mô hình học sâu trên đồ thị, mỗi node và mỗi loại quan hệ trong đồ thị tri thức sẽ được ánh xạ thành một vector số (embedding). Việc này sử dụng các mô hình embedding hiện đại như Sentence-Transformer, giúp chuyển đổi thông tin rời rạc thành không gian vector liên tục, nơi mà các thực thể có liên quan sẽ nằm gần nhau.

   Embedding là nền tảng cho việc học biểu diễn dữ liệu phức tạp, giúp mô hình học sâu có thể khai thác triệt để các mối liên hệ trong đồ thị, đồng thời tăng khả năng tổng quát hóa và hiệu quả khi huấn luyện mô hình.

5. **Huấn luyện mô hình xếp hạng trên đồ thị tri thức**

   Trên cơ sở các embedding này, hệ thống sử dụng mô hình học sâu trên đồ thị tri thức, tiêu biểu là KGAT (Knowledge Graph Attention Network). KGAT kết hợp giữa collaborative filtering (lọc cộng tác) và lan truyền thông tin trên đồ thị tri thức, đồng thời học trọng số attention cho từng loại quan hệ.

   Nhờ đó, mô hình không chỉ tận dụng được thông tin tương tác giữa ứng viên và công việc, mà còn khai thác triệt để các mối liên hệ gián tiếp thông qua kỹ năng, vai trò, kinh nghiệm... Kết quả của quá trình huấn luyện là mỗi ứng viên và mỗi công việc đều có một vector biểu diễn tối ưu, và mô hình có thể tính toán điểm độ phù hợp giữa bất kỳ cặp ứng viên – công việc nào.

   Về mặt lý thuyết, KGAT là một trong những mô hình mạnh nhất hiện nay cho bài toán gợi ý dựa trên đồ thị tri thức, tận dụng cả thông tin trực tiếp và gián tiếp trong hệ thống.

6. **Triển khai và đánh giá hệ thống**

   Hệ thống nhận đầu vào là một vị trí công việc mới, sinh embedding cho đối tượng này và tính toán điểm phù hợp với các đối tượng còn lại trong hệ

thống. Kết quả là một danh sách xếp hạng các ứng viên phù hợp nhất cho từng vị trí công việc, giúp doanh nghiệp dễ dàng tìm thấy lựa chọn tốt nhất.

Hiệu quả của hệ thống được đánh giá bằng các chỉ số phổ biến trong bài toán xếp hạng như Precision@K, Recall@K, MRR... Đây là các chỉ số đo lường khả năng đưa ra các kết quả phù hợp nhất với nhu cầu thực tế, đảm bảo hệ thống không chỉ chính xác mà còn hữu ích trong thực tiễn.

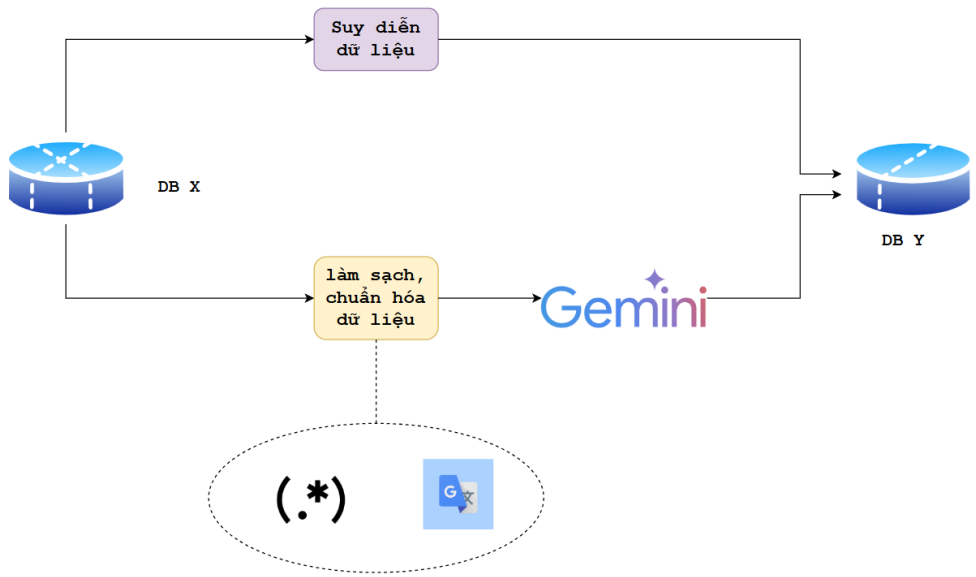
Tóm lại, pipeline giải quyết đề tài gồm các bước: tiền xử lý dữ liệu, trích xuất thông tin cấu trúc, xây dựng đồ thị tri thức, sinh embedding, huấn luyện mô hình học sâu trên đồ thị và triển khai đánh giá hệ thống. Mỗi bước đều dựa trên các lý thuyết hiện đại trong lĩnh vực khai phá dữ liệu, xử lý ngôn ngữ tự nhiên và học sâu, đảm bảo hệ thống vừa hiệu quả vừa có khả năng mở rộng trong thực tế.

3

Tiền xử lý dữ liệu

3.1 Tổng quan

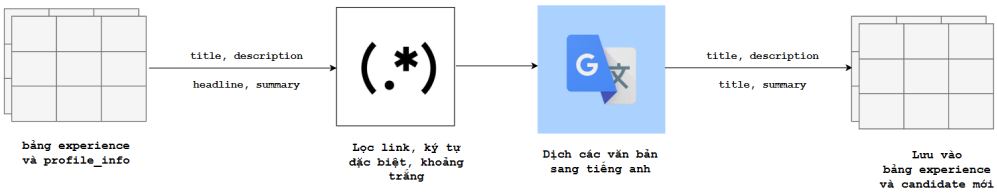
Tiền xử lý dữ liệu là một giai đoạn quan trọng trong quy trình khai phá dữ liệu, đặc biệt khi làm việc với dữ liệu thô từ các nguồn không cấu trúc hoặc bán cấu trúc như LinkedIn. Mục tiêu chính của bước này là biến đổi dữ liệu từ các bảng `profile_info` và `experience` thành một tập dữ liệu sạch, chuẩn hóa, và có cấu trúc, sẵn sàng cho các phân tích sâu hơn và phát triển ứng dụng. Quá trình tiền xử lý bao gồm các bước chính: làm sạch và chuẩn hóa dữ liệu văn bản, tính toán số năm kinh nghiệm, trích xuất đặc trưng bằng mô hình ngôn ngữ lớn (LLM) Gemini, và cập nhật cơ sở dữ liệu với các thông tin mới được làm giàu.



Hình 3.1: *Sơ đồ tổng quan tiền xử lý.*

3.2 Làm sạch và chuẩn hóa dữ liệu văn bản

Dữ liệu văn bản trong các trường như `title`, `description` (bảng `experience`), và `summary`, `headline` (bảng `profile_info`) thường chứa nhiều nhiễu như URL, ký tự đặc biệt, khoảng trắng thừa, hoặc văn bản đa ngôn ngữ. Những yếu tố này có thể làm giảm hiệu suất của các mô hình xử lý ngôn ngữ tự nhiên (NLP) và ảnh hưởng đến chất lượng phân tích. Vì vậy, bước đầu tiên là làm sạch và chuẩn hóa dữ liệu văn bản để đảm bảo tính nhất quán và phù hợp cho các bước tiếp theo.



Hình 3.2: *Sơ đồ làm sạch và chuẩn hóa dữ liệu.*

3.2.1 Loại bỏ nhiễu

Dự án loại bỏ các loại dữ liệu nhiễu sau :

- **Loại bỏ URL**: Các liên kết như “https://on:heterstergej.con/” thường xuất hiện trong trường `description`. Những URL này không mang lại giá trị thông tin cho việc trích xuất đặc trưng và có thể gây nhiễu cho mô hình NLP. Dự án sử dụng biểu thức chính quy (regex) như sau để phát hiện và xóa bỏ chúng:

```
r'http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|[!*\\(\\)
   ,]|(?:%[0-9a-fA-F][0-9a-fA-F]))+'
```

- **Loại bỏ ký tự đặc biệt**: Các ký tự như #, @, *, hoặc các ký tự không phải chữ cái/số (non-alphanumeric) được loại bỏ để đơn giản hóa văn bản. Ví dụ, trong bảng `experience`, một tiêu đề công việc như “UK Accounting Manager (#1)” sẽ được làm sạch thành “UK Accounting Manager”.
- **Loại bỏ khoảng trắng thừa**: Các chuỗi khoảng trắng liên tiếp hoặc khoảng trắng ở đầu/cuối văn bản được chuẩn hóa để đảm bảo định dạng đồng nhất. Ví dụ, “ Data Scientist ” sẽ được chuyển thành “Data Scientist”.

3.2.2 Chuẩn hóa ngôn ngữ

Dữ liệu từ LinkedIn thường chứa văn bản bằng nhiều ngôn ngữ khác nhau, chẳng hạn như tiếng Hà Lan trong trường `description` của bảng `company_detail` ("De Postbank is nu IMG Bank..."). Để đảm bảo tính nhất quán và tương thích với mô hình Gemini (được tối ưu hóa cho tiếng Anh), dự án dịch tất cả văn bản sang tiếng Anh bằng Google Translator API.

Ví dụ cụ thể:

- Văn bản gốc: "De Postbank is nu IMG Bank..."
- Văn bản sau khi dịch: "The Postbank is now IMG Bank..."

Quá trình này không chỉ chuẩn hóa ngôn ngữ mà còn bảo toàn ý nghĩa và ngữ cảnh của văn bản, tạo điều kiện thuận lợi cho việc trích xuất đặc trưng sau này.

3.3 Tính toán số năm kinh nghiệm

Bên cạnh các trường dữ liệu cần được làm sạch và chuẩn hóa thì bộ dữ liệu này tồn tại một số trường dữ liệu không cần phải thực hiện quy trình trên mà vẫn có thể thu được dữ liệu có ích cho các bước sau qua quá trình suy diễn dữ liệu. Dựa trên các trường `start_date` và `end_date` trong bảng `experience` dự án tiến hành tính toán số năm kinh nghiệm cho mỗi bản ghi .

Phương pháp tính toán

- **Công thức tính**: Số năm kinh nghiệm được xác định bằng cách lấy hiệu giữa `end_date` và `start_date`, sau đó quy đổi sang năm:

$$\text{years\_experience} = \frac{(\text{end\_date} - \text{start\_date}).\text{days}}{365}$$

  Kết quả được làm tròn thành số nguyên để đơn giản hóa.
- **Xử lý dữ liệu thiếu**: Nếu `end_date` không có (do ứng viên vẫn đang làm việc tại vị trí đó), chúng tôi sử dụng ngày hiện tại làm giá trị thay thế. Ví dụ, với ngày hiện tại là "2025-05-27":
  - Bản ghi có `start_date` = "2015-01-01", `end_date` = "2020-01-01" → `years_experience` = 5.
  - Bản ghi có `start_date` = "2015-01-01", `end_date` = null → `years_experience` = 10 (tính đến "2025-05-27").

Ý nghĩa

Số năm kinh nghiệm là một chỉ số quan trọng để đánh giá mức độ phù hợp của ứng viên với các vị trí công việc. Việc tính toán này giúp chuẩn hóa dữ liệu thời gian, hỗ trợ các ứng dụng như xếp hạng ứng viên hoặc gợi ý công việc.

3.4 Trích xuất đặc trưng bằng mô hình Gemini

Mô hình ngôn ngữ lớn Gemini được sử dụng để trích xuất thông tin có cấu trúc từ dữ liệu văn bản không cấu trúc trong các bảng `experience` và `profile_info`.

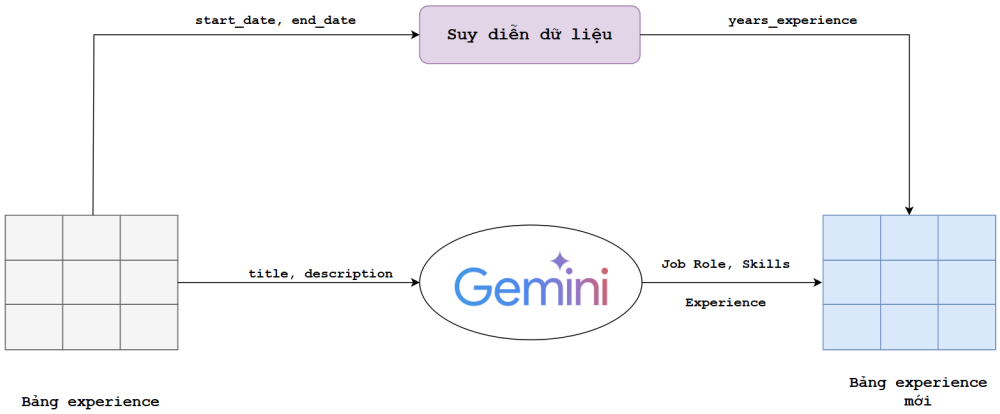
Tổng quan về Gemini

Gemini là một LLM tiên tiến, có khả năng hiểu ngữ cảnh và ngữ nghĩa của văn bản. So với các phương pháp truyền thống dựa trên từ khóa, Gemini linh hoạt hơn và có thể xử lý các biến thể ngôn ngữ phức tạp trong dữ liệu LinkedIn.

Lý do chọn Gemini

- **Khả năng xử lý văn bản tự do**: Gemini không yêu cầu danh sách từ khóa cố định, phù hợp với các tiêu đề công việc đa dạng như "UK Accounting Manager (#1)" hoặc "CS111 Exam Grader & Proctor".
- **Độ chính xác cao**: Gemini giảm thiểu lỗi trong việc trích xuất thông tin từ văn bản phức tạp.
- **Hỗ trợ ứng dụng**: Các đặc trưng được trích xuất là nền tảng cho hệ thống gợi ý công việc và xếp hạng ứng viên.

3.4.1 Trích xuất từ bảng `experience`

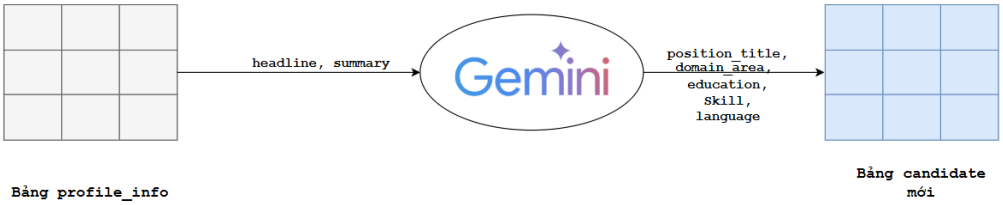


Hình 3.3: *Tiền xử lý bảng experience.*

Từ các trường `title` và `description` đã làm sạch, Gemini trích xuất :

- **Vai trò công việc**: "UK Accounting Manager (#1)" → "Accounting Manager".
- **Kỹ năng**: Từ `description` như "Managed financial reporting and budgeting" → "financial reporting", "budget management".
- **Kinh nghiệm yêu cầu**: "5+ years in accounting" → 5.

3.4.2 Trích xuất từ bảng `profile_info`



Hình 3.4: *Tiền xử lý bảng profile_info.*

Từ các trường `headline` và `summary` trong bảng `profile_info`, qua quá trình lọc và chuẩn hóa sẽ thu được hai trường `title` và `summary` trong bảng `candidate`, thông qua 2 trường này Gemini trích xuất:

- **Chức danh**: "Experienced Data Scientist" → "Data Scientist".
- **Lĩnh vực chuyên môn**: "Specialized in Machine Learning" → "Machine Learning".
- **Học vấn**: "BSc from MIT" → "BSc, MIT".
- **Kỹ năng**: "Proficient in Python, SQL" → "Python", "SQL".
- **Ngôn ngữ**

3.5 Kết quả

	id	candidate_id	title	description	years_experience	job_role	skill	experience
0	3	nicolemesser	Broker	NaN	3.0	Broker	NaN	NaN
1	4	nicolemesser	Broker Associate Commercial Real Estate Sales ...	NaN	3.0	Commercial Real Estate Broker	Real estate sales, Real estate leasing	NaN
2	5	nicolemesser	President	NaN	3.0	President	NaN	NaN
3	6	nicolemesser	Senior Vice President	NaN	0.0	Senior Vice President	NaN	NaN
4	7	nicolemesser	Site Aquisitions and Leasing Manager	NaN	5.0	Site Acquisitions and Leasing Manager	Real Estate, Leasing, Acquisitions	NaN
...	...	...	...	...	...	...	...	...
6189	6996	carlosubedac	Research Assistant	Research about electrolytes for primary Mg bat...	0.0	Research Assistant	Research, Electrolytes, Magnesium batteries, M...	NaN
6190	6997	carlosubedac	Tutor	Tutor of North American engineering students d...	2.0	Tutor	Tutoring, Engineering	NaN
6191	6998	carlosubedac	Power Gas Exchange Market Operation	Operation of the dayahead European Electricity...	1.0	Market Operator	Electricity market operation, gas market opera...	NaN
6192	6999	carlosubedac	Data Interoperability and Market Expert	NaN	1.0	Data Interoperability and Market Expert	Data interoperability, Market analysis	NaN
6193	7000	carlosubedac	Power System Engineer	Regional Security Coordinator RSC of the Weste...	1.0	Power System Engineer	Power systems, Electric power systems, Grid an...	NaN

6194 rows × 8 columns

Hình 3.5: *Bảng experience sau tiền xử lý.*

Quá trình tiền xử lý tạo ra 2 bảng dữ liệu đã được làm sạch, chuẩn hóa, và làm giàu. Trong đó bảng experience thu được 6194 bản ghi và bảng candidate với 987 bản ghi

	candidate_id	title	summary	position_title	domain_area	education	certification	skill	language
0	%25C3%25ASse-karlsen-644903b9	Managing Director and State Authorized Account...	Accountant with high ambitions for efficiency ...	Managing Director	Accounting	NaN	State Authorized Accountant	Accounting, Automation, Customer Service, Effi...	NaN
1	aarti-rawat-85b325203	Airtel	Upgrading and implementing ERP Software soluti...	ERP Software Solutions Implementer	Information Technology	NaN	NaN	ERP Software, Management Information Systems (...	NaN
2	aarti-seth	Transforming Ideas into Impactful Solutions CS...	Im eager to explore diverse technical fields s...	Exploring Diverse Technical Fields	Technology, Business	CS BITS	NaN	IoT, WEB3, Cybersecurity, App Development, Bus...	NaN
3	aarti-seth-0876922a	Head Quality Assurance Secure Meters CMMI V20 ...	With over 20 years of work experience in quali...	Head Quality Assurance	Quality Assurance	NaN	CISP, ISO 31000, ITIL V4 Foundation, ISO 27001...	Quality Assurance, Technical Architecture, ISO...	NaN
4	abdel-gonzalez-a5b06717	Assoc Dir IT Central America Caribbean at MSD	IT professional with more than 15 year of expe...	Associate Director	Information Technology	NaN	NaN	IT leadership, Digital Transformation	NaN
...	...	...	...	...	...	...	...	...	...
982	w%25C3%25AAnia-cristina-da-silva-dos-santos-92...	Tourism Guide Tourism Supervisor Nannai	Bachelor of Tourism Technical Tourism Guide	Tourism Supervisor	Tourism	Bachelor of Tourism	Technical Tourism Guide	Tourism, Guiding	NaN
983	william-caldwell-57221713	Network Systems Engineer at Penn State University	Specialties ITIL army atm cabling cat5 cat6 ci...	Network Systems Engineer	Information Technology	NaN	ITIL	ITIL, cabling, cat5, cat6, cisco, cryptography...	NaN
984	xun-jia-8602655	Professor and Chief of Medical Physics Divisio...	Specialties Numerical and analytical skills C ...	Professor and Chief of Medical Physics Division	Medical Physics, Radiation Oncology	NaN	NaN	Numerical Skills, Analytical Skills, C, C++, F...	NaN
985	y%25C3%25BCcel-demir-kol-8a641b1b4	Forklift Operator at Alfebor	NaN	Forklift Operator	NaN	NaN	NaN	Forklift Operation	NaN
986	yashpal-sharma-1668a1a2	Dy Director Training Placements at Sushant Uni...	NaN	Director of Training Placements	Education	NaN	NaN	Training, Placements, Leadership	NaN

987 rows × 9 columns

Hình 3.6: *Bảng candidate sau tiền xử lý.*

Qua thống kê, số lượng giá trị null của các trường trong bảng experience và candidate được thể hiện chi tiết qua bảng:

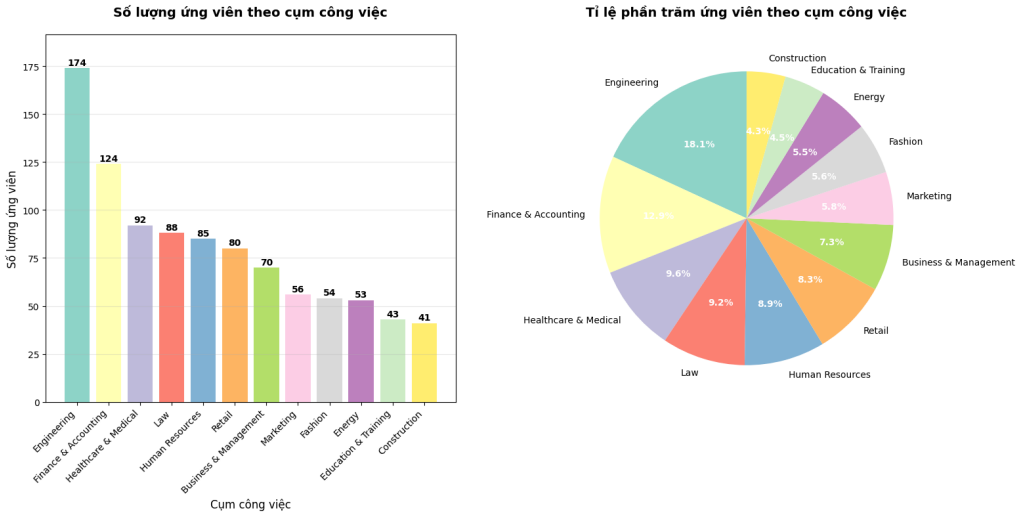
Tên cột	**Số lượng giá trị null**
candidate_id	0
title	0
description	2618
years_experience	0
job_role	0
`requires_skill`	642

Bảng 3.1: *Số lượng giá trị null của các cột trong experience sau tiền xử lý*

Tên cột	**Số lượng giá trị null còn lại**
candidate_id	0
title	13
summary	325
position_title	0
domain_area	0
skill	71

Bảng 3.2: *Số lượng giá trị null bảng candidate sau tiền xủ lý*

Ngoài ra đối với bảng canđiate, dữ thấy sự phân bố không đồng đều về số lượng ứng viên giữa các cụm công việc. Một số lĩnh vực chứa nhiều ứng viên hơn hẳn, trong khi các ngành còn lại có tỷ lệ ứng viên thấp hơn.



Hình 3.7: *Tỉ lệ ứng viên giữa các lĩnh vực.*

4

Xây dựng Knowledge Graph

4.1 Cơ sở lý thuyết

4.1.1 Lý thuyết Knowledge Graph (Đồ thị tri thức)

Đồ thị Knowledge Graph (Đồ thị tri thức) là một cấu trúc dữ liệu mô tả các thực thể (entities) và mối quan hệ giữa chúng dưới dạng đồ thị. Trong đồ thị tri thức, các thực thể được biểu diễn dưới dạng các nút (nodes) và các mối quan hệ được biểu diễn dưới dạng các cạnh (edges) nối các nút này lại với nhau. Mỗi cạnh thường có một kiểu quan hệ và có thể có các thuộc tính bổ sung.

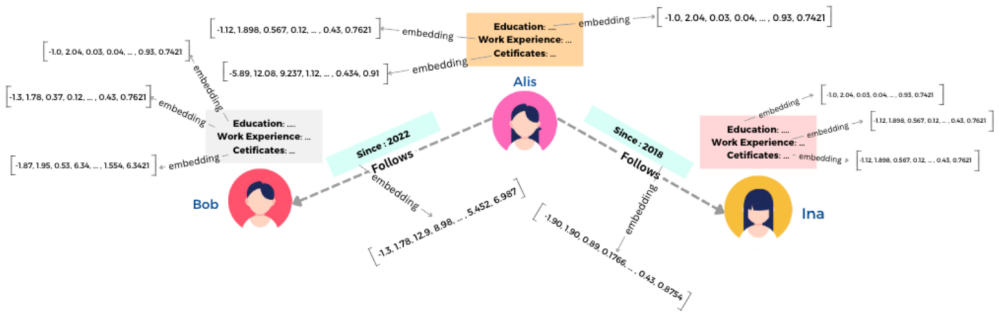
Các thành phần cơ bản của một đồ thị tri thức bao gồm:

- **Nút (Nodes/Entities):** Đại diện cho các đối tượng, khái niệm, con người, địa điểm, sự kiện, v.v., trong thế giới thực hoặc một miền cụ thể. Mỗi nút có một kiểu (type) để phân loại nó (ví dụ: Candidate, Skill, Job, Domain).
- **Cạnh (Edges/Relations):** Đại diện cho mối quan hệ có hướng giữa hai nút. Mỗi cạnh có một kiểu quan hệ (relation type) để mô tả bản chất của mối liên kết đó (ví dụ: HAS_SKILL, HAS_EXPERIENCE, WORK_AT_DOMAIN). Cạnh thường có hướng từ nút ”đầu” (head node) đến nút ”đuôi” (tail node).
- **Thuộc tính (Properties/Attributes):** Thông tin bổ sung gắn liền với nút hoặc cạnh để cung cấp ngữ cảnh chi tiết hơn.

4.1.2 Cách áp dụng vào dự án

Dự án này áp dụng lý thuyết đồ thị tri thức để mô hình hóa và tổ chức dữ liệu từ cơ sở dữ liệu MySQL liên quan đến ứng viên và kinh nghiệm làm việc của họ. Mục tiêu là chuyển đổi dữ liệu dạng bảng truyền thống thành một mạng lưới các

mối quan hệ, giúp khám phá các kết nối tiềm ẩn và hỗ trợ các tác vụ thông minh hơn như hệ thống khuyến nghị.



Hình 4.1: *Mối quan hệ giữa các ứng viên.*

Mapping dữ liệu sang cấu trúc đồ thị

- **Nút:**
  - Mỗi `candidate_id` trong bảng `candidate` được chuyển thành một nút kiểu `candidate`.
  - Mỗi `position_title` của ứng viên được chuyển thành một nút kiểu `position`.
  - Mỗi `domain_area` được phân tách và chuyển thành một nút kiểu `domain`.
  - Mỗi `skill` được phân tách và chuyển thành một nút kiểu `skill`.
  - Mỗi `job_role` từ bảng `experience` được chuyển thành một nút kiểu `job`.
  - Mỗi `title` của `job_role` được chuyển thành một nút kiểu `job_title`.
- **Cạnh:**
  - `HAS_POSITION_CURRENT`: Nối từ nút `candidate` đến nút `position` (từ `position_title` hiện tại của ứng viên).
  - `WORK_AT_DOMAIN`: Nối từ nút `candidate` đến nút `domain` (từ `domain_area` của ứng viên).
  - `HAS_SKILL`: Nối từ nút `candidate` đến nút `skill` (từ `skill` của ứng viên).
  - `HAS_EXPERIENCE`: Nối từ nút `candidate` đến nút `job` (từ `job_role` trong bảng `experience`). Cạnh này có thể mang thuộc tính `years` nếu `years_experience` lớn hơn 0.
  - `REQUIRES_SKILL`: Nối từ nút `job` đến nút `skill` (các kỹ năng được liệt kê trong `job_role`).

- HAS_TITLE: Nối từ nút job đến nút job_title (tiêu đề của job_role).

Các hàm tạo đồ thị trong mã nguồn

Các hàm sau đây được định nghĩa và sử dụng trong mã nguồn để xây dựng đồ thị:

- add_node(node_name, node_type): Chịu trách nhiệm thêm một thực thể mới (nút) vào bảng nodes của cơ sở dữ liệu. Nó đảm bảo tính duy nhất của nút bằng cách kiểm tra sự tồn tại trước khi thêm.
- add_relation(relation_name): Đăng ký một kiểu quan hệ mới vào bảng relations.
- add_edge(head_id, relation_name, tail_id, attributes=None): Tạo một mối quan hệ (cạnh) giữa hai nút đã tồn tại (dựa trên head_id và tail_id) với một kiểu quan hệ cụ thể. Nó cũng cho phép lưu trữ các thuộc tính bổ sung cho cạnh dưới dạng JSON.

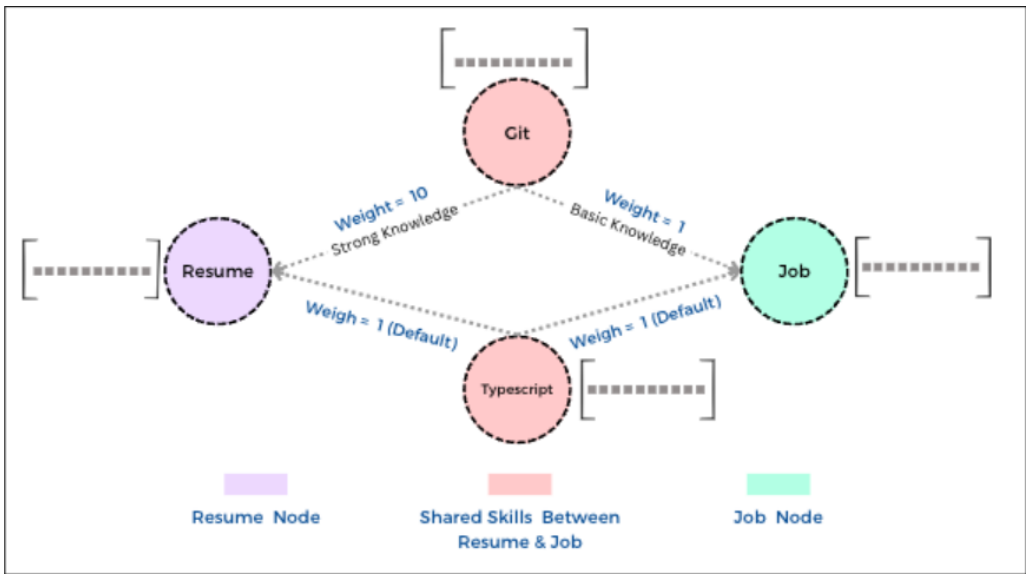
Cấu trúc lại dữ liệu sang dạng đồ thị

Dữ liệu ban đầu được lưu trữ trong cơ sở dữ liệu quan hệ (MySQL) dưới dạng các bảng phẳng. Mặc dù các bảng này hiệu quả cho việc lưu trữ và truy vấn dữ liệu có cấu trúc, chúng lại kém hiệu quả trong việc biểu diễn và truy vấn các mối quan hệ phức tạp, đa chiều giữa các thực thể. Việc cấu trúc lại dữ liệu sang dạng đồ thị mang lại nhiều lợi ích:

- **Biểu diễn tự nhiên và trực quan:** Đồ thị tri thức trực quan hóa các mối quan hệ giữa các thực thể một cách rõ ràng và dễ hiểu hơn nhiều so với các bảng quan hệ. Thay vì JOIN nhiều bảng để hiểu một mối liên hệ, đồ thị cho phép nhìn thấy ngay các kết nối.
- **Khám phá mối quan hệ phức tạp:** Trong cơ sở dữ liệu quan hệ, việc truy vấn các mối quan hệ nhiều cấp (ví dụ: ”ứng viên nào có kỹ năng X và đã từng làm ở công ty mà các nhân viên khác ở đó cũng có kỹ năng Y”) trở nên rất phức tạp với các câu lệnh SQL JOIN dài và hiệu năng kém. Đồ thị cho phép các truy vấn này được thực hiện một cách hiệu quả hơn thông qua việc duyệt đồ thị.
- **Dễ dàng mở rộng và linh hoạt:** Khi có thêm các loại thực thể hoặc mối quan hệ mới, việc mở rộng đồ thị tri thức thường dễ dàng hơn nhiều so với việc thay đổi lược đồ cơ sở dữ liệu quan hệ truyền thống (thêm cột, thêm bảng, sửa khóa ngoại).

- **Hỗ trợ phân tích đồ thị:** Đồ thị cho phép áp dụng các thuật toán đồ thị chuyên biệt (như tìm đường đi ngắn nhất, phân tích cộng đồng, tìm nút trung tâm, phân tích dòng chảy) để khám phá những thông tin ẩn, các mẫu quan hệ và cấu trúc dữ liệu mà không thể thấy được trong các bảng phẳng.

Cấu trúc dạng đồ thị trong bài toán khuyến nghị



Hình 4.2: *Sơ đồ kết nối các nút sơ yếu lý lịch và công việc với các kỹ năng chung.*

Các bài toán khuyến nghị, như đề xuất ứng viên cho vị trí tuyển dụng, đề xuất kỹ năng cho ứng viên, hay đề xuất công việc, được hưởng lợi rất nhiều từ cấu trúc đồ thị vì:

- **Tìm kiếm mối quan hệ ẩn và gián tiếp:** Đồ thị cho phép hệ thống tìm thấy các kết nối ”ẩn” hoặc gián tiếp giữa các thực thể mà không thể dễ dàng phát hiện trong dữ liệu dạng bảng.
- **Cá nhân hóa cao:** Bằng cách phân tích các đường đi và mối quan hệ trong đồ thị xung quanh một thực thể (ví dụ: một ứng viên), hệ thống có thể tạo ra các khuyến nghị rất cá nhân hóa dựa trên sở thích, kinh nghiệm, và các kết nối liên quan của thực thể đó.
- **Giải quyết vấn đề ”cold start”:** Ngay cả khi có ít dữ liệu về một thực thể mới (ví dụ: ứng viên mới không có nhiều kinh nghiệm), đồ thị vẫn có thể tạo ra các khuyến nghị sơ bộ dựa trên các mối quan hệ với các thực thể đã biết. Ví dụ, nếu ứng viên mới có một kỹ năng hiếm, hệ thống có thể tìm các công việc yêu cầu kỹ năng đó.

- **Kết hợp nhiều loại dữ liệu (Heterogeneous Data):** Đồ thị tri thức có khả năng tích hợp nhiều loại thông tin khác nhau (thông tin ứng viên, thông tin công việc, kỹ năng, lĩnh vực, công ty, v.v.) vào một mô hình thống nhất. Điều này giúp tạo ra các khuyến nghị toàn diện và ngữ cảnh hơn, tận dụng tối đa các nguồn dữ liệu đa dạng.
- **Giải thích khuyến nghị (Explainability):** Do tính chất trực quan của đồ thị, dễ dàng giải thích lý do tại sao một khuyến nghị được đưa ra (ví dụ: "Chúng tôi đề xuất công việc này vì bạn có kỹ năng X, là kỹ năng chính mà công việc yêu cầu").

4.2 Xây dựng và trực quan hóa

4.2.1 Triển khai với bộ dữ liệu

Mã Python trong `create_KG.ipynb` thực hiện việc xây dựng đồ thị tri thức bằng cách đọc dữ liệu từ hai bảng MySQL: `candidate` và `experience`. Quy trình triển khai như sau:

1. **Thiết lập kết nối cơ sở dữ liệu:** Mở kết nối đến cơ sở dữ liệu MySQL (`cv`) để tương tác với các bảng `nodes`, `relations`, `edges`.
2. **Định nghĩa các hàm xây dựng đồ thị:** Các hàm `add_node`, `add_relation`, `add_edge` được định nghĩa để thao tác với dữ liệu đồ thị trong MySQL.
3. **Xử lý và chuyển đổi dữ liệu từ bảng `candidate`:**
    - Truy vấn tất cả dữ liệu từ bảng `candidate`.
    - Với mỗi hàng dữ liệu ứng viên:
        - Tạo một nút `candidate` nếu chưa tồn tại.
        - Trích xuất `position_title`, `domain_area`, `skill` của ứng viên.
        - Tạo các nút tương ứng cho `position`, `domain`, `skill` (nếu chúng chưa tồn tại).
        - Tạo các cạnh `HAS_POSITION_CURRENT`, `WORK_AT_DOMAIN`, `HAS_SKILL` từ nút `candidate` đến các nút tương ứng. Các trường `domain_area` và `skill` được xử lý để phân tách các giá trị nếu chúng là chuỗi chứa nhiều mục được phân tách bằng dấu phẩy.
    - **Xử lý và chuyển đổi dữ liệu từ bảng `experience`:**
    - Truy vấn tất cả dữ liệu từ bảng `experience`.
    - Với mỗi hàng dữ liệu kinh nghiệm:

- Tìm nút `candidate` tương ứng (đã được tạo ở bước trước).
- Trích xuất `job_role`, `skill`, `title` từ kinh nghiệm.
- Tạo nút `job` (nếu chưa tồn tại) cho `job_role`.
- Tạo cạnh `HAS_EXPERIENCE` từ nút `candidate` đến nút `job`. Nếu có `years_experience` lớn hơn 0, nó được thêm làm thuộc tính cho cạnh này.
- Trích xuất các kỹ năng liên quan đến `job_role` và tạo các nút `skill` tương ứng.
- Tạo cạnh `REQUIRES_SKILL` từ nút `job` đến các nút `skill` đó.
- Trích xuất `title` của `job_role` và tạo nút `job_title`.
- Tạo cạnh `HAS_TITLE` từ nút `job` đến nút `job_title`.

4.2.2 Trực quan hóa bằng Neo4j

Mã Python hiện tại lưu trữ đồ thị tri thức vào cơ sở dữ liệu MySQL dưới dạng các bảng. Để trực quan hóa đồ thị này bằng Neo4j, bạn cần thực hiện các bước bổ sung để chuyển đổi và nhập dữ liệu từ MySQL sang Neo4j:

(a) **Xuất dữ liệu từ MySQL:** Từ các bảng `nodes`, `relations`, và `edges` trong cơ sở dữ liệu `cv` của MySQL, bạn cần xuất dữ liệu ra các định dạng mà Neo4j có thể nhập (ví dụ: CSV).

(b) **Chuẩn bị dữ liệu cho Neo4j:** Các tệp CSV này cần được xử lý để phù hợp với định dạng nhập của Neo4j. Cụ thể, bạn cần ánh xạ `node_type` thành nhãn nút trong Neo4j và `relation_name` thành kiểu quan hệ. Thuộc tính `attributes` (JSON) cũng cần được phân tích cú pháp để trở thành thuộc tính của cạnh.

(c) **Nhập dữ liệu vào Neo4j** Sử dụng công cụ `cypher-shell` hoặc Neo4j Browser với lệnh `LOAD CSV` để nhập dữ liệu.

(d) **Trực quan hóa và khám phá trong Neo4j Browser**
Sau khi dữ liệu đã được nhập, bạn có thể truy cập Neo4j Browser. Sử dụng các câu lệnh Cypher để khám phá và trực quan hóa đồ thị.
Neo4j Browser sẽ hiển thị các nút và cạnh dưới dạng đồ thị tương tác, cho phép bạn kéo, phóng to, thu nhỏ và khám phá các mối quan hệ một cách trực quan.

4.2.3 Kết quả

Sau khi hoàn thành bước ”Xây dựng và trực quan hóa” (bao gồm cả việc nhập dữ liệu vào Neo4j), output chính sẽ là:

- **Dữ liệu đồ thị tri thức được lưu trữ trong cơ sở dữ liệu Neo4j.** Các nút sẽ có nhãn tương ứng với `node_type` (ví dụ: `candidate`, `skill`, `job`), và các cạnh sẽ có kiểu tương ứng với `relation_name` (ví dụ: `HAS_SKILL`, `HAS_EXPERIENCE`). Các thuộc tính (như `years`) sẽ được gắn vào nút hoặc cạnh nếu có.
- **Trực quan hóa tương tác của đồ thị trong Neo4j Browser.** Đây là hình ảnh trực quan của mạng lưới các ứng viên, kỹ năng, công việc, miền, và các mối quan hệ giữa chúng. Người dùng có thể tương tác với đồ thị để khám phá các kết nối, chạy các truy vấn đồ thị để tìm kiếm thông tin cụ thể, hoặc thực hiện phân tích đồ thị.
- Các truy vấn Cypher sẽ trả về **tập hợp các nút và cạnh** thỏa mãn điều kiện truy vấn, được biểu diễn dưới dạng bảng dữ liệu hoặc trực quan trên giao diện đồ thị trong Neo4j Browser.

5

Xây dựng mô hình KGAT

5.1 Cơ sở lý thuyết

5.1.1 Giới thiệu tổng quan

Mô hình *Knowledge Graph Attention Network (KGAT)* được đề xuất nhằm khai thác hiệu quả các kết nối tiềm ẩn giữa người dùng và mặt hàng trong hệ thống gợi ý. Không giống như các phương pháp truyền thống chỉ xem xét tương tác trực tiếp, KGAT tận dụng cả cấu trúc đồ thị tri thức và cơ chế attention để học được biểu diễn ngữ nghĩa sâu sắc hơn cho các thực thể.

Đồ thị hai phần User-Item: Trong hệ thống gợi ý thông thường, lịch sử tương tác giữa người dùng và mặt hàng (ví dụ: mua hàng, nhấp chuột) được biểu diễn dưới dạng một đồ thị hai phần G_1:

$$G_1 = \{(u, y_{ui}, i) \mid u \in \mathcal{U}, i \in \mathcal{I}\}$$

trong đó $\mathcal{U}$ và $\mathcal{I}$ lần lượt là tập người dùng và mặt hàng. $y_{ui} = 1$ thể hiện rằng người dùng u đã tương tác với mặt hàng i.

Đồ thị tri thức (Knowledge Graph): Thông tin phụ trợ như thuộc tính mặt hàng, thông tin từ cơ sở dữ liệu ngoài (ví dụ: phim, diễn viên, thể loại) được tổ chức dưới dạng đồ thị tri thức G_2:

$$G_2 = \{(h, r, t) \mid h, t \in \mathcal{E}, r \in \mathcal{R}\}$$

Mỗi bộ ba (h, r, t) thể hiện rằng thực thể h có quan hệ r đến thực thể t. Ví dụ: (Hugh Jackman, ActorOf, Logan).

Ngoài ra, tập ánh xạ giữa mặt hàng và thực thể được định nghĩa là:

$$\mathcal{A} = \{(i, e) \mid i \in \mathcal{I}, e \in \mathcal{E}\}$$

Đồ thị tri thức cộng tác (Collaborative Knowledge Graph - CKG): Mô hình KGAT xây dựng một đồ thị thống nhất tích hợp cả tương tác người dùng và tri thức:

$$G = \{(h, r, t) \mid h, t \in \mathcal{E}', r \in \mathcal{R}'\}$$

với $\mathcal{E}' = \mathcal{E} \cup \mathcal{U}$ và $\mathcal{R}' = \mathcal{R} \cup \{\text{Interact}\}$. Trong đó, mỗi tương tác người dùng được biểu diễn dưới dạng $(u, \text{Interact}, i)$ nếu $y_{ui} = 1$.

Mô tả bài toán gợi ý: Với KGAT, bài toán gợi ý được định nghĩa lại như sau:

- **Đầu vào:** Đồ thị tri thức cộng tác G bao gồm thông tin tương tác và tri thức.
- **Đầu ra:** Dự đoán xác suất $\hat{y}_{ui}$ rằng người dùng u sẽ tương tác (mua/chọn) với mặt hàng i.

Khai thác kết nối bậc cao (High-Order Connectivity): Khác với các mô hình truyền thống, KGAT tận dụng kết nối nhiều bước trong KG, chẳng hạn như:

$$e_0 \xrightarrow{r_1} e_1 \xrightarrow{r_2} \cdots \xrightarrow{r_L} e_L$$

Điều này giúp khai thác mối liên hệ gián tiếp giữa người dùng và mặt hàng thông qua các thực thể trung gian.

Ưu điểm nổi bật của KGAT:

- Khả năng học được biểu diễn ngữ nghĩa phong phú nhờ attention trên KG.
- Tận dụng tri thức bên ngoài để bổ sung cho dữ liệu tương tác khan hiếm (sparse).
- Cải thiện đáng kể chất lượng gợi ý trong các tập dữ liệu như MovieLens, Amazon, Last.fm với các chỉ số như Hit@10 và NDCG.

Áp dụng KGAT vào bài toán lọc CV: Bài toán lọc CV cũng là một bài toán gợi ý, trong đó:

- Người dùng u được thay bằng **ứng viên**.
- Mặt hàng i được thay bằng **công việc**.
- Tương tác $(u, \text{Interact}, i)$ tương ứng với việc ứng viên từng làm hoặc phù hợp với công việc đó.
- Các tri thức phụ trợ gồm: kỹ năng, số năm kinh nghiệm, ngành nghề, công nghệ, v.v.

Xây dựng một đồ thị tri thức cộng tác G tích hợp (đã được nêu chi tiết tại chương 4):

- Quan hệ `HAS_SKILL`: *Candidate* → `HAS_SKILL` → *Skill*
- Quan hệ `HAS_POSITION_CURRENT`: *Candidate* → `HAS_POSITION_CURRENT` → *Position Title*
- Quan hệ `WORK_AT_DOMAIN`: *Candidate* → `WORK_AT_DOMAIN` → *Domain*
- Quan hệ `HAS_EXPERIENCE`: *Candidate* → `HAS_EXPERIENCE` → *Job*
- Quan hệ `REQUIRES_SKILL`: *Job* → `REQUIRES_SKILL` → *Skill*
- Quan hệ `HAS_TITLE`: *Job* → `HAS_TITLE` → *Job Title*

Với cấu trúc đồ thị như vậy, KGAT có thể:

- Học được biểu diễn vector cho mỗi ứng viên và công việc.
- Dự đoán xác suất phù hợp giữa ứng viên và công việc mới.
- Khai thác được các kết nối gián tiếp như: "ứng viên A có kỹ năng giống ứng viên B, người từng được tuyển cho công việc C".

So sánh với các phương pháp truyền thống: Các phương pháp collaborative filtering hoặc factorization machine chỉ khai thác tương đồng trực tiếp giữa ứng viên và công việc, trong khi KGAT tận dụng toàn bộ cấu trúc tri thức liên kết các thực thể, từ đó học được các quan hệ tổ hợp sâu như:

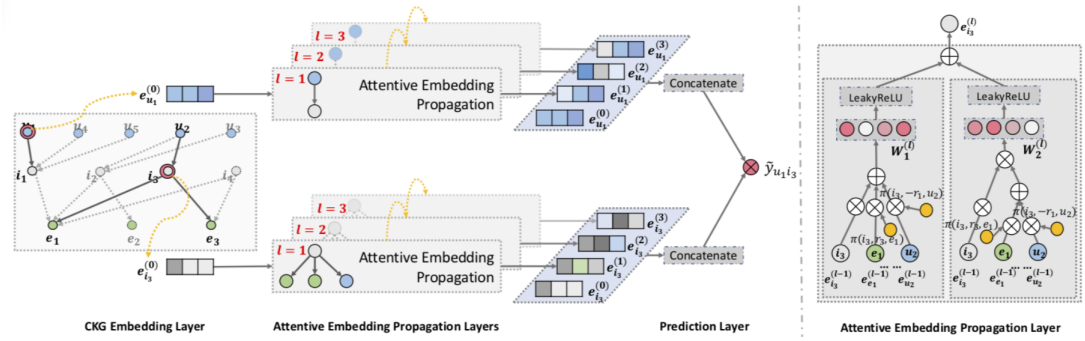
$$job_1 \xrightarrow{r_1} user_1 \xrightarrow{r_2} skill_1 \xrightarrow{r_3} user_2$$

Điều này cho phép mô hình đánh giá cao các ứng viên tiềm năng không chỉ dựa vào khớp trực tiếp, mà còn dựa vào các yếu tố gián tiếp nhưng có tính gợi ý cao.

Kết luận: Với đặc tính cấu trúc đồ thị tri thức phức tạp trong bài toán lọc CV, cùng với yêu cầu cần mô hình có khả năng khai thác mối liên hệ sâu giữa ứng viên và công việc, KGAT là lựa chọn phù hợp và đầy tiềm năng để giải quyết bài toán này.

5.2 Kiến trúc mô hình KGAT cho bài toán lọc CV

Mô hình KGAT (Knowledge Graph Attention Network) được sử dụng trong bài toán lọc CV với mục tiêu tận dụng tri thức ẩn chứa trong đồ thị để học biểu diễn vector cho các thực thể như ứng viên, công việc, kỹ năng, kinh nghiệm, và các yếu tố liên quan. Hình 5.1 minh họa kiến trúc tổng thể của mô hình, bao gồm ba thành phần chính:



Hình 5.1: *Kiến trúc tổng thể của mô hình KGAT được áp dụng cho bài toán lọc CV. Bên trái thể hiện khung kiến trúc tổng thể, trong khi hình bên phải mô tả chi tiết cơ chế lan truyền embedding có trọng số bằng attention.*

- **CKG Embedding Layer – Tầng nhúng đồ thị tri thức cộng tác:** Khởi tạo biểu diễn vector ban đầu cho các thực thể trong hệ thống tuyển dụng như ứng viên, công việc, kỹ năng, và kinh nghiệm. Mục tiêu của tầng này là cung cấp nền tảng biểu diễn cho các tầng kế tiếp khai thác quan hệ giữa các thực thể.
- **Attentive Embedding Propagation Layers – Tầng lan truyền nhúng có trọng số:** Lan truyền thông tin trong đồ thị thông qua cơ chế attention, cho phép mô hình học được sự ảnh hưởng không đồng đều của các thực thể lân cận. Mục tiêu là cập nhật embedding sao cho phản ánh chính xác các mối quan hệ đa bậc và mức độ liên quan giữa ứng viên và công việc.
- **Prediction Layer – Tầng dự đoán:** Sử dụng embedding đã học để tính

điểm phù hợp giữa từng cặp ứng viên và công việc. Mục tiêu là đưa ra danh sách các hồ sơ ứng viên phù hợp nhất cho từng vị trí cần tuyển.

Tổng thể, kiến trúc KGAT cho phép hệ thống lọc CV tận dụng cả thông tin trực tiếp và ngữ nghĩa gián tiếp từ đồ thị tri thức để đưa ra dự đoán chính xác hơn, thay vì chỉ dựa vào các tiêu chí rời rạc như kỹ năng hoặc số năm kinh nghiệm.

5.2.1 Lớp nhúng (Embedding Layer)

Nhúng đồ thị tri thức (knowledge graph embedding) là một phương pháp quan trọng nhằm biểu diễn các thực thể (entities) và quan hệ (relations) trong đồ thị dưới dạng vector trong không gian liên tục, trong khi vẫn duy trì được cấu trúc liên kết ban đầu. Sử dụng mô hình TransR, một phương pháp nhúng đồ thị phổ biến, để học biểu diễn các thành phần trong đồ thị tri thức cộng tác (Collaborative Knowledge Graph - CKG) phục vụ cho bài toán lọc hồ sơ ứng viên (CV screening).

TransR học embedding cho mỗi thực thể và quan hệ bằng cách tối ưu hóa nguyên lý dịch chuyển (translation principle): nếu tồn tại một bộ ba (h, r, t) trong đồ thị, thì biểu diễn vector phải thỏa mãn $\mathbf{W}_r\mathbf{e}_h + \mathbf{e}_r \approx \mathbf{W}_r\mathbf{e}_t$. Trong đó, $\mathbf{e}_h, \mathbf{e}_t \in \mathbb{R}^d$ là vector nhúng của thực thể đầu h và thực thể đuôi t, $\mathbf{e}_r \in \mathbb{R}^k$ là vector nhúng của quan hệ r, và $\mathbf{W}_r \in \mathbb{R}^{k\times d}$ là ma trận biến đổi ánh xạ thực thể sang không gian quan hệ.

Điểm hợp lý (plausibility score) của một bộ ba (h, r, t) được định nghĩa như sau:

$$g(h, r, t) = \|\mathbf{W}_r\mathbf{e}_h + \mathbf{e}_r - \mathbf{W}_r\mathbf{e}_t\|_2^2 . \tag{5.1}$$

Giá trị $g(h, r, t)$ càng nhỏ thì bộ ba càng có khả năng đúng, và ngược lại.

Quá trình huấn luyện sử dụng hàm mất mát xếp hạng cặp (pairwise ranking loss), giúp phân biệt giữa các bộ ba hợp lệ và bộ ba giả (negative samples):

$$\mathcal{L}_{KG} = \sum_{(h,r,t,t')\in\mathcal{T}} -\log\sigma\left(g(h, r, t') - g(h, r, t)\right), \tag{5.2}$$

với $\mathcal{T} = \{(h, r, t, t') \mid (h, r, t) \in \mathcal{G},\ (h, r, t') \notin \mathcal{G}\}$ là tập các cặp bộ ba đúng và sai, và $\sigma(\cdot)$ là hàm sigmoid.

Khác với các phương pháp truyền thống khởi tạo embedding một cách ngẫu nhiên, trong bài toán lọc CV, chúng em khai thác tri thức ngữ nghĩa sẵn có từ các mô hình ngôn ngữ lớn (LLMs) để khởi tạo embedding cho các quan

hệ (ví dụ: `HAS_SKILL`, `HAS_EXPERIENCE`, `REQUIRES_SKILL`, ...). Cụ thể, chúng em sử dụng mô hình `sentence-transformers/all-MiniLM-L12-v2`, một mô hình nhẹ và hiệu quả trong việc sinh embedding cho các cụm từ ngắn. Việc sử dụng LLM giúp đưa thêm ngữ nghĩa ngôn ngữ tự nhiên vào biểu diễn quan hệ, từ đó hỗ trợ mô hình phân biệt tốt hơn các mối liên kết giữa ứng viên, kỹ năng, và công việc — một yếu tố then chốt trong nhiệm vụ gợi ý CV phù hợp.

5.2.2 Lớp Lan Truyền Embedding Có Trọng Số Chú Ý

Được mở rộng từ kiến trúc mạng tích chập đồ thị GCN để thực hiện lan truyền embedding theo các kết nối bậc cao trong đồ thị. Đồng thời, áp dụng cơ chế chú ý từ mạng chú ý đồ thị GAT để gán trọng số cho các kết nối này, phản ánh tầm quan trọng tương đối của chúng.

Lan truyền thông tin. Một thực thể có thể tham gia vào nhiều bộ ba và đóng vai trò là cầu nối giữa các bộ ba khác nhau. Ví dụ, từ hai bộ ba $e_1 \xrightarrow{r_2} user_2 \xrightarrow{-r_1} job_2$ và $e_2 \xrightarrow{r_3} user_2 \xrightarrow{-r_1} job_2$, có thể thấy thực thể $user_2$ có thể tổng hợp thuộc tính từ e_1 và e_2, và từ đó ảnh hưởng đến biểu diễn của job_2. Do đó, định nghĩa tập láng giềng bậc nhất (ego-network) của một thực thể h là $\mathcal{N}_h = \{(h, r, t) \mid (h, r, t) \in \mathcal{G}\}$. Biểu diễn lan truyền từ láng giềng của h được tính như sau:

$$\mathbf{e}_{\mathcal{N}_h} = \sum_{(h,r,t)\in\mathcal{N}_h} \pi(h, r, t)\, \mathbf{e}_t, \tag{5.3}$$

với $\pi(h, r, t)$ là hệ số suy giảm, phản ánh lượng thông tin được truyền từ t đến h theo quan hệ r.

Chú ý có nhận thức tri thức. Trọng số $\pi(h, r, t)$ được tính bằng cơ chế chú ý có điều kiện theo quan hệ:

$$\pi(h, r, t) = (\mathbf{W}_r \mathbf{e}_t)^\top \tanh(\mathbf{W}_r \mathbf{e}_h + \mathbf{e}_r), \tag{5.4}$$

trong đó $\mathbf{W}_r \in \mathbb{R}^{k\times d}$ là ma trận biến đổi quan hệ, $\mathbf{e}_r$ là embedding của quan hệ r, và $\tanh(\cdot)$ là hàm kích hoạt phi tuyến.

Để chuẩn hóa các trọng số này, sử dụng hàm softmax:

$$\pi(h,r,t) = \frac{\exp(\pi(h,r,t))}{\sum_{(h,r',t')\in\mathcal{N}_h} \exp(\pi(h,r',t'))}. \tag{5.5}$$

Trọng số này giúp xác định các nút láng giềng nào nên được chú ý nhiều hơn trong quá trình lan truyền embedding.

Tổng hợp thông tin. Sau khi có embedding từ láng giềng, biểu diễn mới của thực thể h tại lớp đầu tiên được tính như sau:

$$\mathbf{e}_h^{(1)} = f(\mathbf{e}_h, \mathbf{e}_{\mathcal{N}_h}), \tag{5.6}$$

trong đó $f(\cdot)$ là hàm tổng hợp, có thể được hiện thực bằng ba lựa chọn sau:

- **GCN Aggregator kipf2017semi**:

$$f_{\text{GCN}} = \text{LeakyReLU}\left(\mathbf{W}(\mathbf{e}_h + \mathbf{e}_{\mathcal{N}_h})\right), \tag{5.7}$$

- **GraphSage Aggregator hamilton2017inductive**:

$$f_{\text{GraphSage}} = \text{LeakyReLU}\left(\mathbf{W}[\mathbf{e}_h \,\|\, \mathbf{e}_{\mathcal{N}_h}]\right), \tag{5.8}$$

  trong đó $[\cdot\,\|\,\cdot]$ là phép nối vector.
- **Bi-Interaction Aggregator**:

$$f_{\text{Bi-Interaction}} = \text{LeakyReLU}\left(\mathbf{W}_1(\mathbf{e}_h + \mathbf{e}_{\mathcal{N}_h})\right) + \text{LeakyReLU}\left(\mathbf{W}_2(\mathbf{e}_h \odot \mathbf{e}_{\mathcal{N}_h})\right), \tag{5.9}$$

  trong đó $\odot$ là tích từng phần tử (element-wise product).

Lan truyền bậc cao. Để tận dụng các kết nối xa hơn trong đồ thị, ta có thể xếp chồng nhiều lớp lan truyền. Biểu diễn tại lớp thứ l được tính đệ quy:

$$\mathbf{e}_h^{(l)} = f\left(\mathbf{e}_h^{(l-1)}, \mathbf{e}_{\mathcal{N}_h}^{(l-1)}\right), \tag{5.10}$$

trong đó:

$$\mathbf{e}_{\mathcal{N}_h}^{(l-1)} = \sum_{(h,r,t)\in\mathcal{N}_h} \pi(h,r,t)\,\mathbf{e}_t^{(l-1)}. \tag{5.11}$$

Embedding ban đầu $\mathbf{e}_h^{(0)}$ được khởi tạo từ vector nhúng ban đầu $\mathbf{e}_h$. Quá trình

lan truyền qua nhiều lớp cho phép mô hình học được các tương tác gián tiếp giữa thực thể, ví dụ như:

$$u_2 \xrightarrow{r_1} i_2 \xrightarrow{r_2} e_1 \xrightarrow{r_2} i_1 \xrightarrow{r_1} u_1,$$

nhằm đưa các tín hiệu cộng tác dựa trên thuộc tính vào biểu diễn embedding cuối cùng.

Tổng kết. Lớp lan truyền embedding có chú ý này cho phép mô hình khai thác rõ ràng cấu trúc kết nối bậc một và bậc cao trong đồ thị, từ đó liên kết biểu diễn người dùng, mục tiêu và các thực thể tri thức.

5.2.3 Dự đoán từ Mô hình (Model Prediction)

Sau khi lan truyền embedding qua L lớp trong kiến trúc lan truyền đồ thị, thu được các biểu diễn đặc trưng ở nhiều mức cho nút công việc j (job) và ứng viên u (user), được ký hiệu lần lượt là $\{\mathbf{e}_j^{(0)}, \mathbf{e}_j^{(1)}, \cdots, \mathbf{e}_j^{(L)}\}$ và $\{\mathbf{e}_u^{(0)}, \mathbf{e}_u^{(1)}, \cdots, \mathbf{e}_u^{(L)}\}$. Mỗi lớp lan truyền trong mô hình học embedding từ đồ thị góp phần tổng hợp thông tin từ các lân cận bậc cao hơn, phản ánh mối quan hệ gián tiếp giữa ứng viên và công việc thông qua các thực thể trung gian như kỹ năng, kinh nghiệm, hoặc các công việc đã từng đảm nhận. Do đó, embedding từ các lớp khác nhau thể hiện các mức độ kết nối khác nhau trong đồ thị tri thức.

Để tận dụng toàn bộ thông tin thu được, áp dụng cơ chế tổng hợp tầng (layer aggregation) bằng cách nối các biểu diễn của từng lớp thành một vector duy nhất:

$$\mathbf{e}_u^* = \mathbf{e}_u^{(0)} \,\|\, \mathbf{e}_u^{(1)} \,\|\, \cdots \,\|\, \mathbf{e}_u^{(L)}, \quad \mathbf{e}_j^* = \mathbf{e}_j^{(0)} \,\|\, \mathbf{e}_j^{(1)} \,\|\, \cdots \,\|\, \mathbf{e}_j^{(L)}, \tag{5.12}$$

trong đó ký hiệu $\|$ biểu thị phép nối vector.

Việc nối này không chỉ làm giàu embedding ban đầu của từng nút, mà còn cho phép mô hình tận dụng tốt hơn thông tin từ các tầng khác nhau của đồ thị, đồng thời điều chỉnh được mức lan truyền thông qua siêu tham số L.

Cuối cùng, để tính toán điểm phù hợp giữa một ứng viên và một công việc, chúng tôi sử dụng tích vô hướng (inner product) giữa hai vector biểu diễn:

$$\hat{y}(u, j) = (\mathbf{e}_u^*)^\top \mathbf{e}_j^*, \tag{5.13}$$

trong đó $\hat{y}(u, j)$ là điểm số phản ánh mức độ phù hợp giữa ứng viên u và công việc j. Điểm số này được sử dụng để xếp hạng các ứng viên cho một công việc cụ thể, từ đó chọn ra danh sách các CV phù hợp nhất để đề xuất.

5.2.4 Tối ưu hóa mô hình

Để tối ưu mô hình gợi ý ứng viên, sử dụng hàm mất mát BPR (Bayesian Personalized Ranking). BPR giả định rằng mối tương tác quan sát được (ứng viên phù hợp với công việc) nên được gán giá trị dự đoán cao hơn so với những tương tác chưa quan sát (không phù hợp). Hàm mất mát được định nghĩa như sau:

$$\mathcal{L}_{\text{CF}} = - \sum_{(u,i,j) \in O} \ln \sigma \left(\hat{y}_{(u,i)} - \hat{y}_{(u,j)} \right) \tag{5.14}$$

trong đó:

- $O = \{(u, i, j) \mid (u, i) \in \mathcal{R}^+, (u, j) \in \mathcal{R}^-\}$ là tập huấn luyện.
- $\mathcal{R}^+$ là tập các tương tác dương (ứng viên u được đánh giá là phù hợp với công việc i).
- $\mathcal{R}^-$ là tập tương tác âm được lấy mẫu (ứng viên u không phù hợp với công việc j).
- $\sigma(\cdot)$ là hàm sigmoid.
- $\hat{y}_{(u,i)}$ là điểm phù hợp được mô hình dự đoán giữa ứng viên u và công việc i.

Hàm mục tiêu tổng thể của mô hình sẽ kết hợp giữa hàm mất mát tri thức $\mathcal{L}_{\text{KG}}$ và hàm mất mát cộng tác $\mathcal{L}_{\text{CF}}$, cùng với điều chuẩn L2:

$$\mathcal{L}_{\text{KGAT}} = \mathcal{L}_{\text{KG}} + \mathcal{L}_{\text{CF}} + \lambda \|\Theta\|_2^2 \tag{5.15}$$

trong đó:

- $\Theta = \{\mathbf{E}, \mathbf{W}_r, \forall r \in \mathcal{R}, \mathbf{W}_1^{(l)}, \mathbf{W}_2^{(l)}, \forall l \in \{1, \ldots, L\}\}$ là tập các tham số mô hình.
- $\mathbf{E}$ là bảng embedding cho các thực thể và quan hệ trong đồ thị tri thức.
- λ là hệ số điều chuẩn nhằm tránh hiện tượng overfitting.

Huấn luyện

Mô hình được tối ưu bằng cách xen kẽ giữa hai hàm mất mát $\mathcal{L}_{\text{KG}}$ và $\mathcal{L}_{\text{CF}}$. Sử dụng thuật toán Adam với mini-batch để cập nhật embedding và tham số mô hình. Cụ thể:

(a) Với một batch gồm các bộ (h, r, t, t') được chọn ngẫu nhiên, cập nhật embedding cho các thực thể liên quan.
(b) Sau đó, lấy một batch gồm các bộ (u, i, j), truyền qua L lớp lan truyền embedding, và cập nhật tham số mô hình dựa trên gradient từ $\mathcal{L}_{\text{CF}}$.

Phân tích độ phức tạp thời gian

Sử dụng chiến lược tối ưu xen kẽ, chi phí thời gian chủ yếu đến từ hai phần:

- Với phần embedding đồ thị tri thức (Equation 2), độ phức tạp tính toán là $O(|\mathcal{G}^2|d^2)$.
- Với phần lan truyền embedding sử dụng attention, phép nhân ma trận ở lớp thứ l có độ phức tạp là $O(|\mathcal{G}|\, d_l d_{l-1})$.
- Lớp dự đoán cuối cùng chỉ thực hiện tích vô hướng, với chi phí mỗi epoch là $O\left(\sum_{l=1}^{L} |\mathcal{G}|\, d_l\right)$.

5.3 Kết luận

Mô hình *Knowledge Graph Attention Network (KGAT)* thể hiện là một hướng tiếp cận đầy hứa hẹn trong bài toán lọc hồ sơ ứng viên (CV screening), nhờ khả năng khai thác thông tin ngữ nghĩa sâu và các mối liên hệ gián tiếp thông qua cấu trúc đồ thị tri thức. Việc tích hợp giữa dữ liệu tương tác và tri thức miền thông qua đồ thị tri thức cộng tác (Collaborative Knowledge Graph) đã giúp KGAT vượt qua nhiều hạn chế của các phương pháp truyền thống.

Lợi thế của mô hình KGAT

- **Biểu diễn ngữ nghĩa sâu sắc:** Cơ chế attention cho phép mô hình học được mức độ ảnh hưởng khác nhau của các thực thể lân cận, từ đó cải thiện chất lượng embedding.
- **Tận dụng tri thức bổ sung:** Việc kết hợp các quan hệ như `HAS_SKILL`, `HAS_EXPERIENCE`, hay `REQUIRES_SKILL` giúp bổ sung tri thức bên ngoài,

đặc biệt hữu ích khi dữ liệu tương tác giữa ứng viên và công việc bị khan hiếm.
- **Khai thác kết nối bậc cao:** KGAT có khả năng khai thác các chuỗi liên kết nhiều bước giữa các thực thể, hỗ trợ phát hiện các mối quan hệ gián tiếp nhưng có ý nghĩa trong thực tiễn tuyển dụng.
- **Tính linh hoạt cao:** Mô hình có thể dễ dàng mở rộng để tích hợp thêm các loại quan hệ hoặc thực thể khác như học vấn, chứng chỉ, hoặc sở thích nghề nghiệp.

Khó khăn và thách thức

- **Chi phí tính toán lớn:** Việc lan truyền attention trên đồ thị lớn với nhiều lớp embedding đòi hỏi tài nguyên tính toán cao và thời gian huấn luyện dài.
- **Phụ thuộc vào chất lượng đồ thị tri thức:** Hiệu quả của KGAT bị chi phối mạnh bởi độ đầy đủ và chính xác của đồ thị tri thức. Với bộ dữ liệu hiện tại còn nhiều thiếu hụt và rời rạc, đây là một thách thức đáng kể. Để khắc phục, chúng em đã tích hợp thêm cơ chế làm giàu thông tin tự động bằng cách sử dụng API miễn phí của Gemini nhằm bổ sung và chuẩn hóa tri thức từ nguồn dữ liệu.

Tổng kết

Tóm lại, KGAT là một mô hình mạnh mẽ cho bài toán lọc CV nhờ khả năng kết hợp hiệu quả giữa thông tin tương tác và tri thức chuyên ngành. Tuy nhiên, để ứng dụng thành công trong thực tế, cần giải quyết tốt các thách thức về tài nguyên tính toán, chất lượng đồ thị tri thức và quy trình xử lý dữ liệu.

6

Thử nghiệm và đánh giá

6.1 Cài đặt thực nghiệm

6.1.1 Thiết lập dữ liệu

Dữ liệu được xây dựng dựa trên hồ sơ người dùng và lịch sử làm việc thu thập từ LinkedIn, với cấu trúc và cách xây dựng đồ thị tri thức đã được trình bày chi tiết ở Chương 2 và Chương 3.

6.1.2 Chiến lược đánh giá

Sử dụng chiến lược đánh giá theo hướng xếp hạng phổ biến trong hệ thống gợi ý. Cụ thể, đối với mỗi ứng viên trong tập kiểm tra:

- Các công việc chưa từng đảm nhiệm được coi là negative samples.
- Mỗi mô hình dự đoán điểm tương thích giữa ứng viên và các công việc.
- Kết quả được đánh giá bằng hai chỉ số phổ biến: **Recall@K** và **nDCG@K**, với $K = 5$, $K = 10$ và $K = 20$.

Các chỉ số được tính trung bình trên toàn bộ tập kiểm tra để phản ánh hiệu quả tổng thể.

6.2 Cấu hình huấn luyện

Các mô hình được huấn luyện bằng TensorFlow với các thiết lập như sau:

- Kích thước embedding: 384
- Batch size: 1024

- Optimizer: Adam
- Learning rate: 0.0001
- L2 regularization: {1e-5, ..., 1e2}
- Dropout: {0.0, 0.1, ..., 0.8} áp dụng với NFM, GC-MC, và KGAT

Với KGAT, sử dụng:

- Chiều sâu đồ thị: 3 tầng
- Kích thước ẩn các tầng: 64, 32, 16
- Hàm tổng hợp: Bi-Interaction Aggregator

6.3 Kết quả

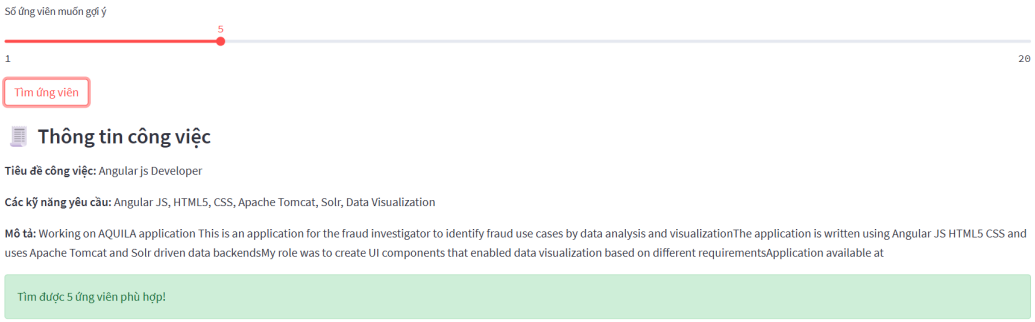
Top-k	Recall@k	nDCG@k
5	0.7449	0.7992
10	0.7449	0.7989
20	0.7449	0.7988

Bảng 6.1: *Kết quả đánh giá mô hình theo các chỉ số Recall và nDCG ở các mức cắt (k)*

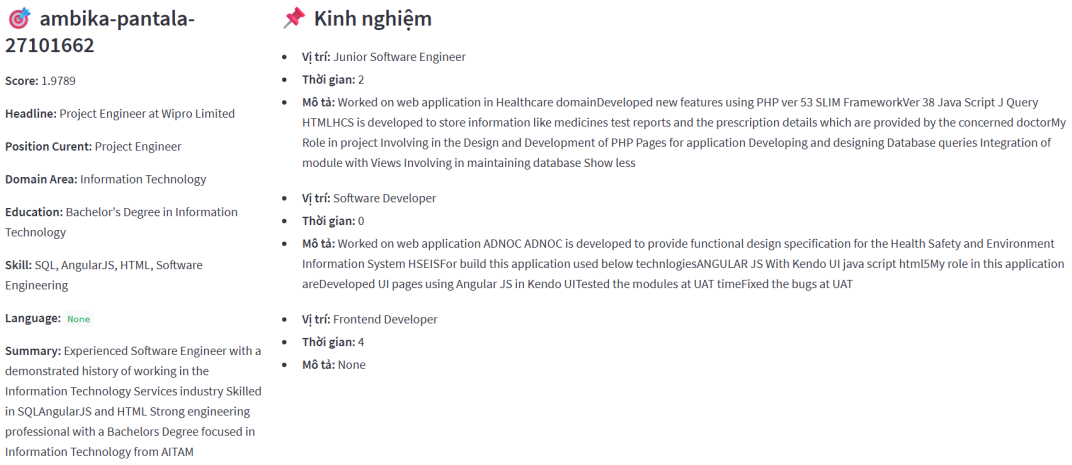
Dựa vào Bảng 6.1, có thể nhận thấy rằng mô hình đạt hiệu suất khá ổn định ở cả ba mức cắt khác nhau (k = 5, 10, 20). Cụ thể, giá trị **Recall@k** duy trì ở mức cao và không thay đổi (0.7449), cho thấy khả năng bao phủ tốt của mô hình – tức là mô hình liên tục đề xuất được các mục liên quan trong top-k.
Bên cạnh đó, các giá trị **nDCG@k** có xu hướng giảm nhẹ khi k tăng, từ 0.7992 ở mức cắt 5 xuống còn 0.7988 tại mức cắt 20. Điều này phản ánh rằng mặc dù các mục liên quan vẫn được đề xuất, nhưng thứ tự sắp xếp của chúng trở nên kém tối ưu hơn khi danh sách đề xuất dài hơn. Tuy nhiên, mức giảm này không đáng kể, cho thấy mô hình vẫn duy trì được chất lượng xếp hạng tương đối tốt ngay cả với danh sách dài.
Tổng thể, kết quả cho thấy mô hình đề xuất hoạt động hiệu quả và nhất quán, đặc biệt là trong việc đảm bảo sự hiện diện của các mục liên quan trong danh sách đề xuất.

7

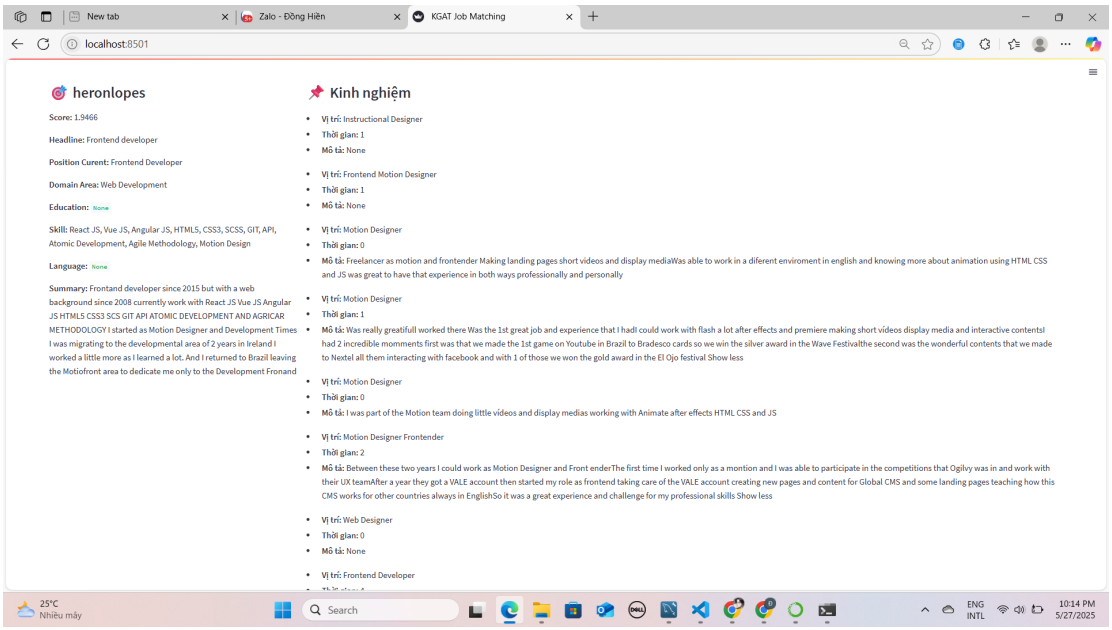
Demo giao diện người dùng



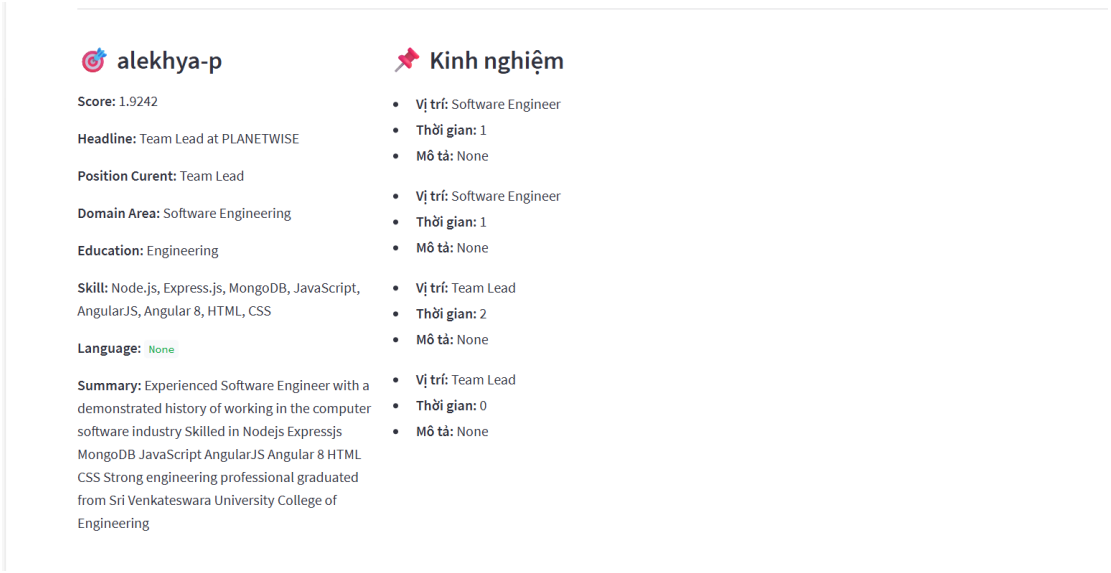
Hình 7.1: *Công việc cần được gợi ý ứng viên.*



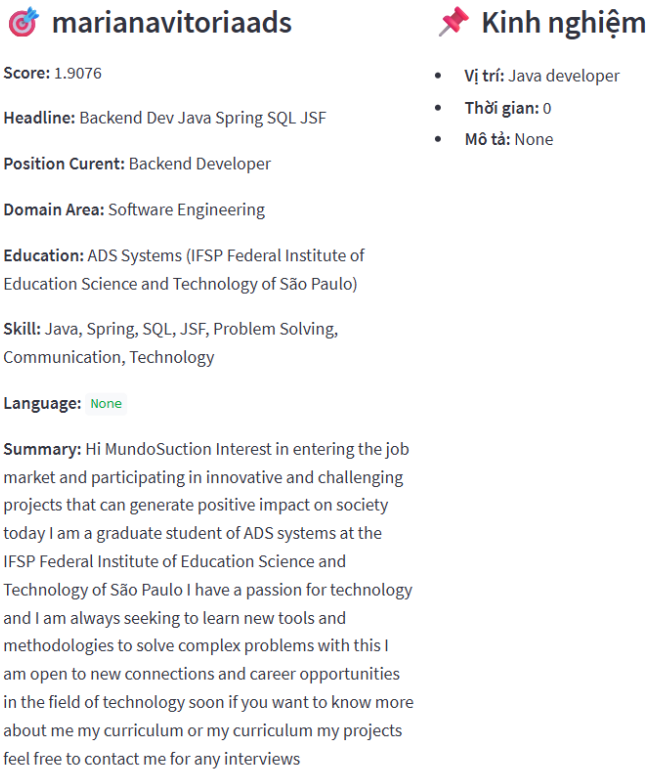
Hình 7.2: *Ứng viên số 1.*



Hình 7.3: *Ứng viên số 2.*



Hình 7.4: *Ứng viên số 3.*



Hình 7.5: *Ứng viên số 4.*

8

Kết luận

Dự án này đã thành công trong việc xây dựng một hệ thống xếp hạng độ phù hợp của ứng viên với các vị trí công việc trên LinkedIn, tận dụng các thuật toán học máy tiên tiến và đồ thị tri thức. Qua các chương trước, chúng tôi đã trình bày toàn bộ quy trình thực hiện, từ việc phân tích dữ liệu ban đầu, tiền xử lý, xây dựng đồ thị tri thức, phát triển mô hình KGAT (Knowledge Graph Attention Network), đến thử nghiệm và đánh giá hiệu quả của hệ thống.

8.1 Thành tựu chính

(a) **Phân tích dữ liệu**
Dự án đã tiến hành phân tích sâu bộ dữ liệu LinkedIn, bao gồm các bảng `profile_info`, `experience`, `company_detail` và `job_posting`. Quá trình này giúp xác định cấu trúc dữ liệu, mối quan hệ giữa các thực thể như ứng viên, công việc, công ty và kỹ năng, từ đó làm nền tảng cho việc xây dựng mô hình.

(b) **Tiền xử lý dữ liệu**
Dữ liệu đã được làm sạch và chuẩn hóa kỹ lưỡng, bao gồm xử lý các giá trị thiếu, chuẩn hóa văn bản, và trích xuất các đặc trưng quan trọng như kỹ năng, kinh nghiệm làm việc, và yêu cầu công việc. Điều này đảm bảo dữ liệu đầu vào chất lượng cao cho các bước tiếp theo.

(c) **Xây dựng đồ thị tri thức**
Một đồ thị tri thức phong phú đã được tạo ra, kết nối các thực thể như ứng viên, công việc, kỹ năng và kinh nghiệm. Đồ thị này được trực quan hóa bằng công cụ Neo4j, giúp dễ dàng quan sát và khai thác các mối quan hệ phức tạp giữa các thực thể.

(d) **Phát triển mô hình KGAT**
Mô hình KGAT đã được triển khai để học biểu diễn vector cho các thực thể và dự đoán mức độ phù hợp giữa ứng viên và công việc. Nhờ cơ chế attention và khả năng khai thác kết nối bậc cao trong đồ thị tri thức, mô hình này vượt trội hơn các phương pháp truyền thống trong việc xếp hạng.

(e) **Thử nghiệm và đánh giá**
Kết quả thử nghiệm cho thấy mô hình KGAT đạt được các chỉ số hiệu quả cao như Recall@K và nDCG@K (với K = 5, 10, 20), chứng minh khả năng xếp hạng chính xác và phù hợp của hệ thống trong việc gợi ý ứng viên cho các vị trí công việc.

8.2 Hạn chế

Mặc dù dự án đã đạt được nhiều kết quả tích cực, vẫn tồn tại một số hạn chế cần được xem xét:

(a) **Chất lượng dữ liệu**
Dù đã qua tiền xử lý, dữ liệu vẫn có một số bất cập như thông tin không đầy đủ hoặc không chính xác, đặc biệt trong các trường văn bản tự do (ví dụ: mô tả công việc, kinh nghiệm làm việc). Điều này ảnh hưởng đến độ chính xác của mô hình trong một số trường hợp.

(b) **Tài nguyên tính toán**
Việc huấn luyện mô hình KGAT trên đồ thị tri thức lớn đòi hỏi tài nguyên tính toán đáng kể, bao gồm cả thời gian và phần cứng mạnh mẽ. Đây có thể là rào cản đối với các tổ chức có nguồn lực hạn chế.

(c) **Khả năng giải thích**
Mặc dù mô hình mang lại kết quả tốt, việc giải thích lý do tại sao một ứng viên được xếp hạng cao hơn ứng viên khác vẫn còn khó khăn. Điều này có thể làm giảm mức độ tin tưởng của người dùng đối với hệ thống.

8.3 Hướng phát triển tương lai

Để khắc phục các hạn chế và nâng cao hiệu quả của hệ thống, chúng tôi đề xuất một số hướng phát triển trong tương lai:

(a) **Cải thiện chất lượng dữ liệu**

Ứng dụng các kỹ thuật xử lý ngôn ngữ tự nhiên (NLP) tiên tiến hơn để trích xuất và chuẩn hóa thông tin từ văn bản tự do. Ngoài ra, có thể tích hợp dữ liệu từ các nguồn bên ngoài (ví dụ: API của Gemini) để làm giàu đồ thị tri thức.

(b) **Tối ưu hóa mô hình**
Nghiên cứu các phương pháp giảm chi phí tính toán của KGAT, như sử dụng kỹ thuật lấy mẫu đồ thị (graph sampling) hoặc tối ưu hóa thuật toán huấn luyện, nhằm giúp hệ thống hoạt động hiệu quả hơn trên các thiết bị có tài nguyên hạn chế.

(c) **Tăng cường khả năng giải thích**
Áp dụng các kỹ thuật AI giải thích được (XAI), chẳng hạn như SHAP hoặc LIME, để cung cấp lý do rõ ràng cho các dự đoán của mô hình. Điều này sẽ giúp người dùng hiểu rõ hơn về quá trình xếp hạng và tăng độ tin cậy của hệ thống.

(d) **Mở rộng ứng dụng**
Hệ thống có thể được áp dụng cho các nền tảng tuyển dụng khác ngoài LinkedIn hoặc mở rộng để hỗ trợ các tác vụ như gợi ý công việc phù hợp cho ứng viên dựa trên hồ sơ cá nhân của họ.

8.4 Tóm tắt

Tóm lại, dự án này đã đạt được mục tiêu ban đầu là xây dựng một hệ thống xếp hạng độ phù hợp giữa ứng viên và công việc, đồng thời đặt nền móng cho các cải tiến trong tương lai. Việc tiếp tục nghiên cứu và phát triển sẽ giúp hệ thống trở nên mạnh mẽ, hiệu quả và hữu ích hơn, mang lại giá trị thực tiễn cho cả nhà tuyển dụng và ứng viên trong thị trường lao động ngày càng cạnh tranh.

9

Phân công công việc

Bảng phân công công việc

Bảng 9.1: *Bảng phân công công việc*

MSV	Thành viên	Công việc
B21DCCN708	Phạm Văn Tiến	• Phân tích bộ dữ liệu • Định hướng giải quyết bài toán • Viết báo cáo.
B21DCCN668	Nguyễn Minh Thắng	• Tiền xử lý dữ liệu. • Trích xuất các thuộc tính bằng LLMs. • Viết báo cáo.
B21DCCN646	Nguyễn Đức Quỳnh	• Tìm hiểu tổng quan về các nghiên cứu. • Tạo Knowledge Graph • Viết báo cáo.
B21DCCN041	Nguyễn Thu Hà	• Tạo Knowledge Graph. • Viết báo cáo.

Bảng 9.1 – tiếp theo

B21DCCN046	Đồng Thị Hiền	• Xây dựng mô hình KGAT cho bài toán lọc hồ sơ ứng tuyển. • Huấn luyện mô hình trên tập dữ liệu đã chuẩn bị. • Thực hiện kiểm thử và đánh giá hiệu quả của mô hình. • Phát triển giao diện người dùng nhằm trực quan hóa kết quả và minh họa khả năng ứng dụng của mô hình.

Tài liệu tham khảo

Ali, Irfan et al. (2022). “Resume Classification System Using Natural Language Processing and Machine Learning Techniques”. In: *Mehran University Research Journal of Engineering and Technology* 41.1, pp. 65–79. url: https://publications.muet.edu.pk/index.php/muetrj/article/view/2353.

Daryani, Chirag et al. (2020). “An Automated Resume Screening System Using Natural Language Processing and Similarity”. In: *Topics in Intelligent Computing and Industry Design* 2.2, pp. 99–103. doi: 10.26480/etit.02.2020.99.103. url: https://doi.org/10.26480/etit.02.2020.99.103.

Gan, Chengguang, Qinghao Zhang, and Tatsunori Mori (2024). “Application of LLM Agents in Recruitment: A Novel Framework for Resume Screening”. In: *arXiv preprint arXiv:2401.08315*. url: https://arxiv.org/abs/2401.08315.

Priyanka, J. Himabindu and Nikhat Parveen (2024). “DeepSkillNER: An automatic screening and ranking of resumes using hybrid deep learning and enhanced spectral clustering approach”. In: *Multimedia Tools and Applications* 83.16, pp. 47503–47530. doi: 10.1007/s11042-023-17264-y. url: https://doi.org/10.1007/s11042-023-17264-y.

Tejaswini, K. et al. (2022). “Resume Screening with Natural Language Processing (NLP)”. In: *Alphanumeric Journal* 12.2. url: https://alphanumericjournal.com/media/Issue/volume-12-issue-2-2024/resume-screening-with-natural-language-processing-nlp.pdf.