

Đối với hệ thống đề xuất, phương pháp và cách thức đề xuất chính là yếu tố cốt lõi, thể hiện tính “thông minh” của hệ thống. Website tuyển dụng thông minh trong đồ án sử dụng phương pháp lọc dựa trên nội dung (Content-based Filtering – CBF) để đánh giá và so sánh mức độ phù hợp giữa các đối tượng, bao gồm ứng viên và tin tuyển dụng. Trong quá trình triển khai, một trong những khó khăn lớn nhất là việc chuẩn hóa và biểu diễn dữ liệu của người dùng và tin tuyển dụng nhằm đảm bảo quá trình so khớp và tính toán độ phù hợp được thực hiện chính xác và hiệu quả.

## **0.1 Chuẩn hóa dữ liệu**

### **0.1.1 Đặt vấn đề**

Đối với hệ thống tuyển dụng thông minh, dữ liệu đầu vào đóng vai trò đặc biệt quan trọng trong quá trình xây dựng và vận hành hệ thống đề xuất. Các dữ liệu này đến từ nhiều nguồn khác nhau như hồ sơ ứng viên, mong muốn nghề nghiệp, kỹ năng, hành vi tương tác của người dùng và thông tin từ các tin tuyển dụng. Tuy nhiên, các dữ liệu thu thập được thường không đồng nhất về định dạng, cấu trúc và mức độ đầy đủ, gây khó khăn cho quá trình so sánh và đánh giá mức độ phù hợp giữa ứng viên và tin tuyển dụng.

Trong hệ thống đề xuất sử dụng phương pháp lọc dựa trên nội dung Content-based Filtering), việc so khớp các đặc trưng giữa các đối tượng là yếu tố cốt lõi. Do đó, nếu dữ liệu không được chuẩn hóa một cách thống nhất, kết quả tính toán độ phù hợp có thể thiếu chính xác hoặc không phản ánh đúng năng lực của ứng viên cũng như yêu cầu của nhà tuyển dụng. Vì vậy, bài toán chuẩn hóa dữ liệu được đặt ra như một bước tiền xử lý bắt buộc nhằm đảm bảo chất lượng và hiệu quả của hệ thống đề xuất.

### **0.1.2 Giải pháp**

Dựa trên phương pháp lọc dựa trên nội dung (Content-based Filtering), hệ thống đề xuất trong đồ án biểu diễn cả ứng viên và tin tuyển dụng dưới dạng các vector đặc trưng có cùng cấu trúc. Việc sử dụng mô hình vector giúp hệ thống dễ dàng so sánh và tính toán mức độ tương đồng giữa các đối tượng, từ đó đưa ra kết quả đề xuất phù hợp.

Các vector đặc trưng được xây dựng từ những thuộc tính quan trọng nhất trong bài toán tuyển dụng, bao gồm: kỹ năng (skills), lĩnh vực hoặc ngành nghề quan tâm (tags), mức lương và địa điểm làm việc. Mỗi đặc trưng được gán một trọng số thể hiện mức độ quan trọng hoặc mức độ phù hợp của đặc trưng đó đối với từng đối tượng.

## a, **Ứng viên**

Trước khi thực hiện quá trình chuẩn hóa, dữ liệu của ứng viên được thu thập từ hai nguồn chính là hành vi cá nhân và thông tin mong muốn nghề nghiệp. Trong đó, hành vi cá nhân bao gồm các sự kiện được hệ thống ghi nhận trong quá trình ứng viên tương tác với nền tảng, chẳng hạn như ứng tuyển vào một công việc, thêm hoặc loại bỏ một công việc khỏi danh sách yêu thích, cũng như xem chi tiết thông tin của một tin tuyển dụng. Các sự kiện này lưu lại thông tin về những tin tuyển dụng mà ứng viên đã tương tác, từ đó được tổng hợp và chuyển đổi thành các đặc trưng tương ứng trong vector biểu diễn ứng viên.

Hành vi người dùng trong hệ thống được xây dựng dựa trên việc tổng hợp thông tin từ các tin tuyển dụng mà ứng viên đã tương tác, kết hợp với loại sự kiện phát sinh trong quá trình sử dụng hệ thống. Các dữ liệu này được hệ thống ghi nhận và lưu trữ trong bảng UserInterestHistory. Mỗi bản ghi hành vi phản ánh một lần tương tác cụ thể giữa ứng viên và một tin tuyển dụng, bao gồm các thông tin như mã ứng viên, mã tin tuyển dụng, tiêu đề công việc, địa điểm, mức lương trung bình, các lĩnh vực liên quan và loại sự kiện tương tác.

Các loại sự kiện hành vi phổ biến bao gồm: ứng viên xem chi tiết một tin tuyển dụng (viewed), ứng viên ứng tuyển vào công việc (applied), hoặc thêm và loại bỏ tin tuyển dụng khỏi danh sách yêu thích (favorite). Thông qua việc ghi nhận các sự kiện này, hệ thống có thể xác định mức độ quan tâm của ứng viên đối với từng tin tuyển dụng cụ thể.

Ví dụ, một bản ghi trong bảng UserInterestHistory của ứng viên được biểu diễn như sau:

```
{  
    "id": 349,  
    "user_id": 2,  
    "job_id": 37,  
    "job_title": "Chuyên Viên Kiểm Thủ Phần Mềm (QC/Tester -  
Tại Hà Nội)",  
    "location": "Hà Nội",  
    "avg_salary": 25000000,  
    "tags": [  
        { "name": "IT" },  
        { "name": "Tester" },  
        { "name": "QC" }  
    ]  
},
```

```

    "source": "viewed",
    "event_type": "open_detail"
}

```

Ngoài hành vi, ứng viên còn có một bảng dữ liệu ghi nhận mong muốn người dùng:

```

{
  "user_id": 1,
  "desired_title": "Front-End Developer",
  "desired_company": "Công ty TNHH Miyano Việt Nam",
  "desired_location": "Hồ Chí Minh",
  "desired_salary": 15000000,
  "tags": [
    { "id": 27, "name": "Javascript" },
    { "id": 28, "name": "FE" },
    { "id": 40, "name": "IT" },
    { "id": 41, "name": "Game Development" }
  ]
}

```

Từ hai nguồn dữ liệu trên kết hợp với dữ liệu về kỹ năng của người dùng, hệ thống tiến hành tổng hợp và sinh ra dữ liệu đặc trưng cho từng ứng viên dựa trên công thức tính toán cụ thể được trình bày tại Mục 2, Chương 5. Tập dữ liệu này nhằm phản ánh một cách toàn diện cả mong muốn nghề nghiệp khai báo và hành vi tương tác thực tế của ứng viên.

### **b, Nhà tuyển dụng**

Nhà tuyển dụng cũng là một người dùng trong hệ thống, tuy nhiên có vai trò khác với ứng viên, do đó được biểu diễn bằng một vector đặc trưng riêng. Vector này được xây dựng dựa trên dữ liệu về nhu cầu tuyển dụng do nhà tuyển dụng khai báo, nhằm phản ánh các yêu cầu và tiêu chí tuyển chọn ứng viên cho từng vị trí. Dữ liệu nhu cầu tuyển dụng sau khi được chuẩn hóa sẽ được sử dụng để xây dựng vector biểu diễn nhà tuyển dụng. Một ví dụ về dữ liệu nhu cầu tuyển dụng được trình bày như sau:

```

{
  "user_id": 1,
  "desired_location": "Hồ Chí Minh",

```

```

    "desired_salary_avg": 20000000,
    "desired_tags": [
        { "id": 2, "name": "ReactJS" },
        { "id": 14, "name": "React" },
        { "id": 28, "name": "FE" },
        { "id": 40, "name": "IT" }
    ],
    "required_skills": [
        { "id": 18, "name": "BA", "years_required": 1,
          "must_have": true },
        { "id": 21, "name": "FE", "years_required": 2,
          "must_have": false }
    ]
}

```

### c, Tin tuyển dụng

Tin tuyển dụng được biểu diễn dựa trên các thông tin tuyển dụng do nhà tuyển dụng khai báo cho từng vị trí công việc. Hệ thống sử dụng các trường dữ liệu này để xây dựng vector đặc trưng tương ứng cho mỗi tin tuyển dụng, bao gồm các yêu cầu về kỹ năng, lĩnh vực, địa điểm làm việc và mức lương.

Các vector của ứng viên và nhà tuyển dụng được chuẩn hóa theo cùng cấu trúc với vector tin tuyển dụng. Trên cơ sở đó, hệ thống áp dụng phương pháp lọc dựa trên nội dung (Content-based Filtering – CBF) để tính toán mức độ phù hợp và sinh ra lý do phù hợp giữa các đối tượng được trình bày tại Mục 3, Chương 5.

```

{
    "vector": {
        "job_id": 33,
        "skill_profile": [
            { "id": 1, "must": true, "weight": 0.44 },
            { "id": 21, "must": true, "weight": 0.48 }
        ],
        "tag_profile": [
            { "id": 28, "weight": 1 }
        ],
        "title_keywords": null,
        "location": "HN",
    }
}

```

```

    "salary_avg": 14000000,
    "updated_at": "2026-01-18T05:47:16.740Z"
  }
}

```

Trường *title\_keywords* trong vector đặc trưng được tính toán bằng phương pháp TF-IDF nhưng chưa được sử dụng trong đồ án do chưa hoàn thiện, và được định hướng phát triển trong các phiên bản tiếp theo của hệ thống.

### 0.1.3 Kết quả thực hiện

Các vector đặc trưng trong hệ thống được xây dựng theo cùng một cấu trúc nhằm đảm bảo tính nhất quán và thuận tiện cho việc so sánh giữa các đối tượng. Các trường dữ liệu chính trong mỗi vector bao gồm: mã định danh (*id*), kỹ năng (*skills*), lĩnh vực hoặc ngành nghề (*tags*), từ khóa chức danh (*title\_keywords*), địa điểm làm việc (*location*) và mức lương trung bình (*salary\_avg*).

```

{
  "user_id": 1,
  "skill_profile": [
    { "id": 1, "w": 0.64 },
    { "id": 18, "w": 0.36 },
    { "id": 21, "w": 0.52 }
  ],
  "tag_profile": [
    { "id": 95, "weight": 0.4476 },
    { "id": 28, "weight": 0.7344 },
    { "id": 40, "weight": 1 }
  ],
  "title_keywords": [
    { "keyword": "reactjs", "weight": 1 },
    { "keyword": "vuejs", "weight": 1 }
  ],
  "preferred_location": "HCM",
  "salary_expected": 17037818,
  "updated_at": "2026-01-18T06:18:00.907Z"
}

```

## 0.2 Tính hành vi người dùng

### 0.2.1 Đặt vấn đề

Như đã trình bày tại Mục 1, Chương 5, dữ liệu của ứng viên bao gồm nhiều thành phần khác nhau nhằm phản ánh đầy đủ mong muốn và đặc điểm của ứng viên, bao gồm dữ liệu hành vi và dữ liệu mong muốn nghề nghiệp. Do đó, cần có một phương pháp tính toán và tổng hợp các nguồn dữ liệu này thành một vector chuẩn cho mỗi ứng viên, làm cơ sở cho quá trình so khớp và đề xuất trong hệ thống.

### 0.2.2 Giải pháp

Dữ liệu hành vi của người dùng được thu thập từ nhiều nguồn khác nhau, trong đó mỗi loại hành vi phản ánh mức độ quan tâm của ứng viên ở các mức độ khác nhau. Do đó, hệ thống gán các trọng số khác nhau cho từng loại hành vi nhằm phản ánh đúng mức độ ảnh hưởng của chúng trong quá trình xây dựng vector đặc trưng của ứng viên.

Các trọng số này được xác định dựa trên kết quả khảo sát một số người dùng kết hợp với nhu cầu và phạm vi triển khai của đồ án. Cụ thể, hệ thống sử dụng các trọng số như sau: ứng viên ứng tuyển vào công việc có trọng số là 5, ứng viên đánh dấu yêu thích công việc có trọng số là 3, ứng viên xem tin tuyển dụng có trọng số là 1, và dữ liệu mong muốn nghề nghiệp do người dùng khai báo có trọng số là 4.

Từ các trọng số hành vi đã xác định, hệ thống tiến hành tính toán trọng số cho các trường dữ liệu tương ứng trong vector đặc trưng của ứng viên. Cách tính được thực hiện cho từng trường dữ liệu như sau:

- (i) **Từ khóa (title\_keywords)**: Các từ khóa được tổng hợp từ tiêu đề công việc và tên công ty trong bảng dữ liệu hành vi. Phương pháp TF-IDF được sử dụng để xác định trọng số của các từ khóa này. Tuy nhiên, trong phạm vi đồ án, trường dữ liệu này chưa được áp dụng để tính toán mức độ phù hợp và được định hướng phát triển trong các phiên bản tiếp theo của hệ thống.
- (ii) **Địa điểm (location)**: Địa điểm làm việc được xác định dựa trên tổng trọng số từ các hành vi liên quan. Ví dụ, ứng viên có các hành vi như: yêu thích công việc tại TP. Hồ Chí Minh, xem công việc tại Hà Nội và khai báo mong muốn làm việc tại Hà Nội. Khi đó, Hà Nội có tổng trọng số là 6, lớn hơn tổng trọng số 5 của TP. Hồ Chí Minh, do đó Hà Nội được chọn làm địa điểm đặc trưng của ứng viên.
- (iii) **Mức lương (salary)**: Mức lương được tính toán dựa trên trọng số hành vi, tuy nhiên có sự điều chỉnh so với cách cộng trọng số thông thường. Ví dụ (đơn vị: triệu đồng), với các hành vi gồm: yêu thích mức lương 20, mong muốn mức

lương 10 và ứng tuyển mức lương 12, giá trị mức lương đặc trưng được tính theo công thức:

$$0.7 \times \frac{20 \times 3 + 12 \times 5}{3 + 5} + 0.3 \times 10 = 13.5$$

Giá trị này phản ánh sự kết hợp giữa hành vi thực tế và mong muốn khai báo của ứng viên.

- (iv) **Ngành nghề (tags)**: Tương tự như cách xác định địa điểm làm việc, các ngành nghề được tính toán dựa trên tổng trọng số từ các hành vi tương tác của ứng viên. Hệ thống lựa chọn ba ngành nghề có tổng trọng số cao nhất làm các ngành nghề đặc trưng trong vector biểu diễn ứng viên.

### 0.2.3 Kết quả thực hiện

Dựa trên các phương pháp tính toán và trọng số đã trình bày, hệ thống tiến hành tổng hợp và xây dựng dữ liệu hành vi của người dùng dưới dạng cấu trúc chuẩn hóa như sau:

```
{  
    "user_id": 1,  
    "avg_salary": 17037818,  
    "main_location": "Hà Nội",  
    "tags": [  
        { "name": "IT", "weight": 1 },  
        { "name": "FE", "weight": 0.7343 },  
        { "name": "BE", "weight": 0.4478 }  
    ],  
    "keywords": [  
        { "name": "reactjs", "weight": 1 },  
        { "name": "vuejs", "weight": 1 }  
    ]  
}
```

## 0.3 Xây dựng công thức tính điểm phù hợp

### 0.3.1 Đặt vấn đề

Trong hệ thống tuyển dụng thông minh, việc xác định mức độ phù hợp giữa các đối tượng đóng vai trò then chốt, quyết định trực tiếp đến chất lượng của chức năng đề xuất. Điểm phù hợp (*Fit Score*) được sử dụng như một thước đo định lượng nhằm đánh giá mức độ đáp ứng giữa hồ sơ ứng viên và yêu cầu của tin tuyển dụng,

cũng như giữa ứng viên và nhu cầu tuyển dụng của nhà tuyển dụng. Bên cạnh việc tính toán điểm phù hợp, hệ thống còn sinh ra lý do đề xuất tương ứng với từng vai trò người dùng, qua đó giúp nâng cao trải nghiệm người dùng và tăng thuyết phục của các kết quả đề xuất.

Trong phần này, đồ án tập trung trình bày phương pháp và công thức tính toán điểm phù hợp dựa trên phương pháp lọc dựa trên nội dung (Content-based Filtering), làm cơ sở cho việc sắp xếp và sinh lý do đề xuất trong hệ thống.

### 0.3.2 Giải pháp

Để triển khai hai hướng đề xuất trên, hệ thống xét đến ba đối tượng liên quan gồm: ứng viên, nhà tuyển dụng và tin tuyển dụng. Mỗi đối tượng có các tiêu chí quan tâm khác nhau, do đó công thức tính điểm phù hợp được thiết kế linh hoạt theo từng ngữ cảnh so khớp.

Trên cơ sở đó, điểm phù hợp được tổng hợp từ các yếu tố quan trọng như mức độ đáp ứng về kỹ năng (bao gồm kỹ năng bắt buộc và kỹ năng bổ trợ), mức độ phù hợp về lĩnh vực và định hướng nghề nghiệp, cùng với các điều kiện ưu tiên như địa điểm làm việc và mức lương (trong trường hợp dữ liệu cho phép). Đồng thời, hệ thống sinh ra các lý do đề xuất tương ứng nhằm giúp người dùng hiểu rõ các điểm mạnh, các yếu tố còn thiếu và các gợi ý cải thiện, qua đó làm cho kết quả đề xuất trở nên minh bạch và thuyết phục hơn.

#### a, Dữ liệu đầu vào

Đối với ứng viên, ngoài dữ liệu hành vi đã trình bày tại Mục 2, Chương 5, vector ứng viên (*User Vector*) cần được bổ sung thêm hồ sơ kỹ năng (*skill profile*) của ứng viên, được lấy từ bảng lưu trữ kỹ năng trong hệ thống để hoàn thiện vector ứng viên.

Mỗi kỹ năng của ứng viên được gán một trọng số  $w \in [0, 1]$ , phản ánh mức độ phù hợp của kỹ năng đó. Trọng số này được xác định dựa trên mức trình độ (*level*) và số năm kinh nghiệm (*years*) mà ứng viên khai báo, qua đó giúp vector người dùng phản ánh chính xác hơn năng lực thực tế của ứng viên.

Trước hết, mỗi mức độ thành thạo được ánh xạ với một số năm kinh nghiệm tối đa tương ứng, dùng làm ngưỡng chuẩn. Trọng số kỹ năng được tính theo công thức:

$$w = 0.8 \times \frac{level}{5} + 0.2 \times \left( \min \left( \frac{years}{maxYears}, 1 \right) + bonus \right)$$

Trong đó:

- $w$ : trọng số của kỹ năng, có giá trị trong khoảng  $[0, 1]$ , tối đa là 1 nếu vượt quá

1;

- *level*: cấp độ kỹ năng do ứng viên khai báo, với giá trị từ 1 đến 5;
- *years*: số năm kinh nghiệm thực tế của ứng viên đối với kỹ năng tương ứng;
- *maxYears*: số năm kinh nghiệm tối đa tương ứng với từng mức độ thành thạo, được sử dụng làm ngưỡng chuẩn để đánh giá kinh nghiệm;
- *bonus*: hệ số thưởng khi số năm kinh nghiệm vượt quá ngưỡng *maxYears*, có giá trị tối đa là 0.2 nhằm tránh làm mất cân bằng trọng số tổng thể.

**Ví dụ:** giả sử một ứng viên có kỹ năng với mức độ thành thạo *level* = 3 và số năm kinh nghiệm *years* = 6. Với mức *level* = 3, số năm kinh nghiệm tối đa tương ứng là *maxYears* = 5. Khi đó:

$$levelScore = \frac{3}{5} = 0.6, \quad experienceRatio = 1, \quad bonus = 0.05$$

Suy ra trọng số kỹ năng:

$$w = 0.8 \times 0.6 + 0.2 \times (1 + 0.05) = 0.69$$

Giá trị này phản ánh việc ứng viên có mức độ thành thạo tốt và kinh nghiệm vượt ngưỡng yêu cầu đối với kỹ năng tương ứng.

Đối với nhà tuyển dụng và tin tuyển dụng, dữ liệu sử dụng để xây dựng vector đặc trưng chỉ xuất phát từ một nguồn dữ liệu khai báo. Do đó, quá trình chuẩn hóa và xây dựng vector cho hai đối tượng này tương đối đơn giản, và cấu trúc vector thu được tương tự như kết quả đã trình bày tại Mục 1, Chương 5.

### 0.3.3 Công thức tính điểm

Điểm phù hợp (*Fit Score*) được tính toán dựa trên bốn yếu tố chính, bao gồm:

- *S*: kỹ năng (*SkillScore*);
- *T*: ngành nghề (*TagScore*);
- *L*: địa điểm làm việc (*LocationScore*);
- *P*: mức lương (*SalaryScore*).

#### a, So khớp kỹ năng

Xét hai tập kỹ năng cần so khớp, trong đó  $K_J$  là tập kỹ năng yêu cầu của đối tượng tham chiếu (tin tuyển dụng hoặc nhà tuyển dụng) và  $K_U$  là tập kỹ năng của đối tượng được so khớp (ứng viên). Mỗi kỹ năng  $k \in K_J$  được gán một trọng số yêu cầu  $weight_k$  và được phân loại thành kỹ năng bắt buộc (*must*) hoặc kỹ năng bổ

trợ (*optional*).

Trọng số tối đa của mỗi kỹ năng  $k$  được xác định như sau:

$$W_k = \begin{cases} 5 \cdot weight_k, & \text{nếu } k \text{ là kỹ năng bắt buộc,} \\ 1 \cdot weight_k, & \text{nếu } k \text{ là kỹ năng bổ trợ.} \end{cases}$$

Nếu đối tượng  $U$  không có kỹ năng bắt buộc  $k$ , điểm kỹ năng được gán bằng 0:

$$P_k = 0$$

Nếu đối tượng  $U$  có kỹ năng bắt buộc  $k$  với trọng số  $w_k^U$ , điểm kỹ năng được tính theo:

$$base_k = \min\left(\frac{w_k^U}{weight_k}, 1\right)$$

$$bonus_k = \begin{cases} \min((w_k^U - weight_k) \cdot 0.2, 0.2), & w_k^U > weight_k, \\ 0, & \text{ngược lại} \end{cases}$$

$$P_k = (base_k + bonus_k) \cdot W_k$$

Trong đó, thành phần  $base_k$  phản ánh mức độ đáp ứng yêu cầu của kỹ năng, còn  $bonus_k$  là hệ số thuỷt nhầm phản ánh trường hợp đối tượng  $U$  có mức độ kỹ năng vượt quá yêu cầu ban đầu.

Nếu đối tượng  $U$  không có kỹ năng bổ trợ  $k$ :

$$P_k = 0$$

Nếu đối tượng  $U$  có kỹ năng bổ trợ  $k$  với trọng số  $w_k^U$ , điểm kỹ năng được xác định bởi:

$$base_k = \min\left(\frac{w_k^U}{weight_k}, 1\right)$$

$$bonus_k = \begin{cases} \min((w_k^U - weight_k) \cdot 0.1, 0.1), & w_k^U > weight_k, \\ 0, & \text{ngược lại} \end{cases}$$

$$P_k = (base_k + bonus_k) \cdot W_k$$

Điểm so khớp kỹ năng được chuẩn hóa theo tỷ lệ giữa tổng điểm đạt được và

tổng điểm tối đa:

$$SkillScore = \frac{\sum_{k \in K_J} P_k}{\sum_{k \in K_J} W_k}$$

Giá trị  $SkillScore$  (viết tắt là  $S$ ) thường nằm trong khoảng  $[0, 1]$ , và có thể tăng lên tối đa 1.2 trong trường hợp đối tượng được so khớp có mức độ kỹ năng vượt trội so với yêu cầu.

Ngoài điểm số tổng hợp, hệ thống ghi nhận các chỉ số thống kê sau để phục vụ việc sinh lý do đề xuất:

$$mustCount = |\{k \in K_J \mid k \text{ là kỹ năng bắt buộc}\}|$$

$$matchedMustCount = |\{k \in K_J \mid k \text{ là kỹ năng bắt buộc và } k \in K_U\}|$$

$$matchedOptionalCount = |\{k \in K_J \mid k \text{ là kỹ năng bổ trợ và } k \in K_U\}|$$

$$missingMust = \{k \in K_J \mid k \text{ là kỹ năng bắt buộc và } k \notin K_U\}$$

Các chỉ số này được sử dụng để sinh lý do đề xuất, giúp giải thích rõ những kỹ năng đã đáp ứng và những kỹ năng còn thiếu.

### b, So khớp lĩnh vực/ngành nghề

Việc so khớp lĩnh vực/ngành nghề được thực hiện giữa hai tập tag:  $T_J$  là tập tag của đối tượng tham chiếu (tin tuyển dụng hoặc nhà tuyển dụng) và  $T_U$  là tập tag của đối tượng được so khớp (ứng viên). Mỗi tag  $t \in T_J$  được gán trọng số  $w_t^J$  nhằm phản ánh mức độ quan trọng của lĩnh vực/ngành nghề đó đối với đối tượng tham chiếu; tương tự, mỗi tag  $t \in T_U$  có trọng số  $w_t^U$ .

Trong trường hợp một trong hai đối tượng không có dữ liệu tag (tức  $T_J = \emptyset$  hoặc  $T_U = \emptyset$ ), hệ thống gán điểm nền  $TagScore = 0.2$  để tránh triệt tiêu hoàn toàn ảnh hưởng của tiêu chí ngành nghề trong điểm phù hợp tổng thể.

Khi cả hai phía đều có dữ liệu tag, điểm so khớp được tính dựa trên mức độ giao nhau có xét trọng số. Với mỗi tag  $t \in T_J$ , điểm đạt được được xác định:

$$P_t = \begin{cases} \min(w_t^U, w_t^J), & \text{nếu } t \in T_U, \\ 0, & \text{ngược lại.} \end{cases}$$

Điểm tối đa tương ứng của tag đó là:

$$M_t = w_t^J.$$

Tổng điểm thô và tổng điểm tối đa lần lượt là:

$$P_{\text{raw}} = \sum_{t \in T_J} P_t, \quad M_{\text{raw}} = \sum_{t \in T_J} M_t.$$

Điểm thô được chuẩn hóa:

$$R = \frac{P_{\text{raw}}}{M_{\text{raw}}}.$$

Sau đó hệ thống áp dụng phép chuẩn hóa tuyến tính để đảm bảo tiêu chí ngành nghề luôn có ảnh hưởng tối thiểu:

$$\text{TagScore} = 0.2 + 0.8 \cdot R,$$

trong đó  $\text{TagScore} \in [0.2, 1]$ .

Ngoài điểm số, hệ thống ghi nhận danh sách các tag trùng khớp giữa hai đối tượng; để đảm bảo tính trực quan khi hiển thị, chỉ tối đa ba tag trùng khớp đầu tiên (theo thứ tự ưu tiên của  $T_J$ ) được sử dụng như các lý do đề xuất.

#### c, So khớp địa điểm

Nếu một trong hai phía thiếu dữ liệu vị trí (ứng viên hoặc đối tượng tham chiếu), hệ thống gán điểm nền và trạng thái trung tính:

$$\text{LocScore} = 0.3, \quad \text{matched} = \text{false}, \quad \text{level} = \text{neutral}.$$

Nếu có đủ dữ liệu, hệ thống kiểm tra mức độ trùng khớp theo tiêu chí so khớp chuỗi (ví dụ: một chuỗi chứa chuỗi còn lại). Khi đó:

$$\text{LocScore} = \begin{cases} 1, & \text{nếu vị trí trùng khớp (match),} \\ 0.3, & \text{nếu không trùng khớp.} \end{cases}$$

Giá trị  $\text{LocScore}$  (viết tắt là  $L$ ) nằm trong khoảng  $[0.3, 1]$ . Việc đặt ngưỡng nền 0.3 giúp điểm phù hợp tổng thể không bị giảm quá mạnh trong trường hợp địa điểm không trùng nhau hoặc thiếu dữ liệu vị trí.

#### d, So khớp mức lương

Tiêu chí so khớp mức lương được xây dựng dựa trên hai giá trị:

- $S_E$ : mức lương kỳ vọng của ứng viên (*expected*);
- $S_J$ : mức lương đại diện của đối tượng tham chiếu (tin tuyển dụng hoặc nhà tuyển dụng), ký hiệu *jobSalary*.

Hàm trả về bộ giá trị ( $SalaryScore$ ,  $level$ ,  $comparable$ ), trong đó:

- $SalaryScore$ : điểm so khớp lương;
- $level \in \{\text{near}, \text{lower}, \text{higher}, \text{null}\}$ : nhãn giải thích quan hệ giữa  $S_J$  và  $S_E$ ;
- $comparable$ : cờ logic cho biết hai giá trị có đủ dữ liệu để so sánh hay không.

Nếu một trong hai phía thiếu dữ liệu (tức  $S_E$  không tồn tại hoặc  $S_J$  không tồn tại), hệ thống không thể so sánh trực tiếp và gán trạng thái trung tính:

$$SalaryScore = 1, \quad level = \text{null}, \quad comparable = \text{false}.$$

Khi  $S_E$  và  $S_J$  đều tồn tại, đặt tỷ lệ:

$$r = \frac{S_J}{S_E}.$$

Khi đó  $comparable = \text{true}$  và  $SalaryScore$  được tính theo các quy tắc sau:

- Bằng nhau (near). Nếu  $S_J = S_E$ :

$$SalaryScore = 1, \quad level = \text{near}.$$

- Thấp hơn kỳ vọng (lower). Nếu  $S_J < S_E$ :

$$SalaryScore = \frac{S_J}{S_E} = r, \quad level = \text{lower}.$$

Giá trị này nằm trong  $(0, 1)$  và phản ánh mức độ “thiếu hụt” của lương so với kỳ vọng.

- Cao hơn kỳ vọng (higher). Nếu  $S_J > S_E$ , hệ thống áp dụng cơ chế thưởng theo ngưỡng tỷ lệ  $r$ :

$$SalaryScore = \begin{cases} 1.1, & 1 < r \leq 1.25, \\ 1.25, & 1.25 < r \leq 1.5, \\ 1.3, & 1.5 < r \leq 1.75, \\ 1.4, & 1.75 < r \leq 2, \\ 1.5, & r > 2, \end{cases} \quad level = \text{higher}.$$

Như vậy,  $SalaryScore$  có thể lớn hơn 1 trong trường hợp mức lương tham chiếu cao hơn kỳ vọng của ứng viên, nhằm thể hiện hiệu ứng “thưởng” khi cơ hội đáp

ứng tốt về mặt thu nhập.

#### e, Công thức tính FitScore theo từng ngữ cảnh đề xuất

Đánh giá mức phù hợp của ứng viên cho một tin tuyển dụng (ứng tuyển/đơn ứng tuyển). Trong ngữ cảnh này, tin tuyển dụng là đối tượng tham chiếu và mục tiêu là đánh giá ứng viên có đáp ứng yêu cầu của công việc hay không. Do đó tiêu chí kỹ năng được ưu tiên cao:

$$Fit_{U \rightarrow J} = 0.4S + 0.25T + 0.25P + 0.1L.$$

Gợi ý tin tuyển dụng cho ứng viên (đề xuất việc làm). Trong ngữ cảnh này, ứng viên là đối tượng tham chiếu và mục tiêu là đánh giá công việc có phù hợp mong muốn của ứng viên hay không. Vì vậy hệ thống tăng trọng số cho tag/ngành nghề và lương:

$$Fit_{J \rightarrow U} = 0.15S + 0.4T + 0.3P + 0.15L.$$

Gợi ý ứng viên cho nhà tuyển dụng (đề xuất ứng viên). Trong ngữ cảnh này, nhà tuyển dụng (hồ sơ nhu cầu tuyển dụng) là đối tượng tham chiếu và mục tiêu là đánh giá ứng viên có phù hợp với nhu cầu của nhà tuyển dụng hay không. Do đó kỹ năng được đặt trọng số cao nhất:

$$Fit_{U \rightarrow R} = 0.5S + 0.2T + 0.15P + 0.15L.$$

Trong cả ba công thức, các trọng số đều được chọn sao cho tổng bằng 1, nhằm đảm bảo FitScore là tổ hợp tuyển tính của các điểm thành phần và có thể so sánh trực tiếp giữa các đối tượng trong cùng một ngữ cảnh.

### 0.3.4 Kết quả thực hiện

The screenshot shows a search interface titled "Đề xuất dành cho bạn" (Job recommendations for you). It displays five job listings:

- Front End Developer (ReactJS, VueJS, HTML) - Tại Hà Nội**  
Công Ty TNHH LG CNS VIỆT NAM  
Ngày đăng 03-01-2026  
Cập nhật 03-01-2026  
Độ phù hợp: 72%  
Lương: 15.000.000 - 20.000.000 VND  
Hà Nội  
FE IT BE
- Quản Lý Dự Án (Project Manager) – Team Công Nghệ**  
CÔNG TY CỔ PHẦN GIẢI PHÁP SỐ MOBIFONE  
Ngày đăng 03-01-2026  
Cập nhật 03-01-2026  
Độ phù hợp: 66%  
Lương: 18.000.000 - 25.000.000 VND  
Hà Nội  
IT IT Project Manager
- Chuyên Viên Kiểm Thử Phần Mềm (QC/Tester) - Tại Hà Nội**  
CÔNG TY CỔ PHẦN GIẢI PHÁP SỐ MOBIFONE
- Korean Bridge Software Engineer - Kỹ Sư Cầu Nối Tiếng Hàn**  
Công Ty TNHH LG CNS VIỆT NAM

Hình 0.1: Điểm phù hợp và lý do đề xuất

Hình 0.1 là kết quả tính đề xuất một tin tuyển dụng cho ứng viên dựa theo công thức đã trình bày.

## 0.4 Gửi thông báo thông minh

### 0.4.1 Đặt vấn đề

Trong các hệ thống tuyển dụng, email là kênh tương tác phổ biến vì tận dụng được thông tin liên hệ mà người dùng đã đăng ký. Tuy nhiên, nếu hệ thống chỉ gửi email theo kiểu *thông báo thuần túy* mà không xét đến trạng thái và nhu cầu thực tế của người dùng, trải nghiệm sẽ nhanh chóng bị suy giảm.

Ví dụ, ứng viên đã tìm được việc làm nhưng vẫn tiếp tục nhận email giới thiệu công việc mới, gây cảm giác phiền hà. Tương tự, nhà tuyển dụng có thể nhận quá nhiều email không liên quan hoặc lặp lại, dẫn đến tình trạng *spam* và làm giảm mức độ tin cậy đối với hệ thống. Khi đó, hệ thống không còn được xem là một cơ chế đề xuất thông minh, mà chỉ là một kênh gửi thông tin đại trà.

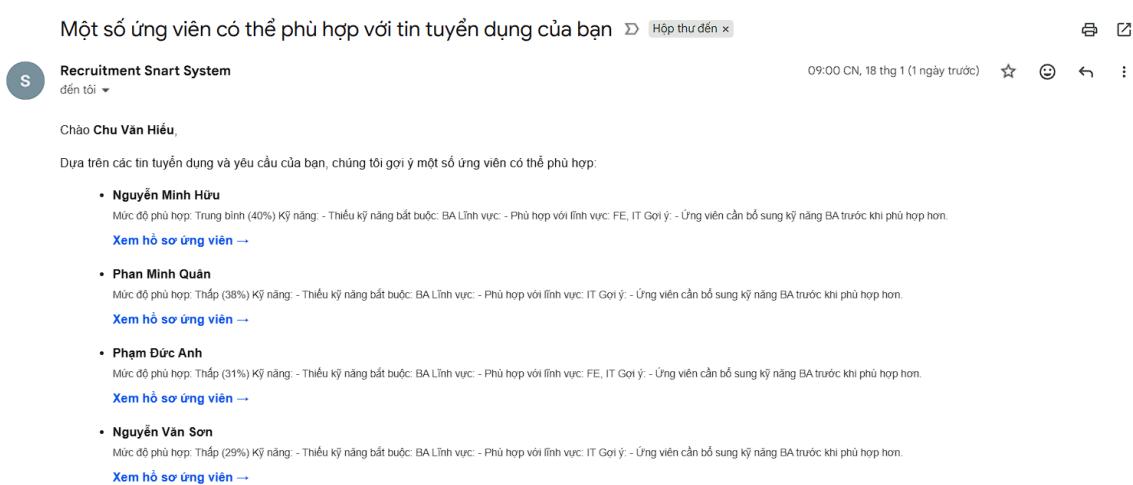
### 0.4.2 Giải pháp

Để đảm bảo email thực sự mang tính cá nhân hóa và hữu ích, hệ thống áp dụng các cơ chế sau:

- Tránh gửi trùng lặp: Hệ thống lưu vết các đề xuất đã gửi và gắn nhãn cho từng đối tượng đã được thông báo (tin tuyển dụng/ứng viên). Nhờ đó, cùng một nội dung sẽ không bị gửi lặp lại nhiều lần cho cùng một người dùng trong một khoảng thời gian nhất định.

- Cho phép người dùng chủ động quản lý thông báo: Hệ thống hỗ trợ bật/tắt nhận thông báo, đồng thời cho phép khóa tài khoản hoặc tạm dừng nhận email. Cơ chế này đảm bảo người dùng có quyền xác nhận rõ ràng về việc có muốn tiếp tục nhận thông báo từ hệ thống hay không.
- Tự động tắt thông báo khi người dùng không tương tác: Hệ thống theo dõi số lượng email đã gửi nhưng người dùng không có phản hồi gián tiếp (ví dụ: không đăng nhập/truy cập hệ thống trong một khoảng thời gian). Nếu vượt ngưỡng cho phép, hệ thống tự động tạm dừng gửi thông báo để hạn chế làm phiền.
- Xác thực hai chiều theo trạng thái nhận thông báo: Khi ứng viên hoặc nhà tuyển dụng tắt nhận thông báo, hệ thống không chỉ dừng gửi email cho người đó mà còn ngừng sử dụng hồ sơ của họ như một đối tượng để đề xuất cho phía còn lại. Cách tiếp cận này giúp tránh tình trạng đề xuất các đối tượng không có nhu cầu tương tác.
- Hai chế độ gửi: định kỳ và theo sự kiện: Hệ thống hỗ trợ gửi email định kỳ (tổng hợp theo thời gian) và gửi theo sự kiện (khi có tin tuyển dụng mới hoặc có đề xuất nổi bật). Việc kết hợp hai chế độ giúp hạn chế bỏ sót thông tin quan trọng nhưng vẫn kiểm soát tần suất gửi, đảm bảo thông báo đến đúng người quan tâm.

#### 0.4.3 Kết quả thực hiện



**Hình 0.2:** Email thông báo cho nhà tuyển dụng

Hình 0.2 minh họa nội dung email thông báo mà nhà tuyển dụng nhận được từ hệ thống.