**Abstract:**

Đối với các nhà tuyển dụng, việc đăng tuyển thông tin tìm kiếm ứng cử viên phù hợp một cách chính xác là một việc hết sức cần thiết. Nhất là thông tin về ngành nghề mình đang tuyển, để ứng viên có thể chọn lựa một cách chính xác. Tuy nhiên, trong một website tìm việc, luôn có một số lượng tương đối lớn nhóm các ngành nghề đánh dấu. Điều này làm cho nhà tuyển dụng dễ chọn thiếu, chọn sai khi tìm các ngành nghề phù hợp cho mô tả công việc đó. Do vậy, trong bài báo cáo lần này chúng tôi sẽ xây dựng mô hình phân lớp đa nhãn sử dụng các mô hình học máy như Linear Regression, SVM, SGD, các mô hình học sâu như TextCNN, Bi-LSTM, Bi-GRU cùng với các mô hình pretrained biến thể của BERT bao gồm phoBERT, XLMBERT và distilBERT. Kết quả tốt nhất chúng tôi đạt được đến từ mô hình … với EM score là … và F1-score là … trên tập test. Ngoài ra, với kết quả thực nghiệm này chúng tôi còn phân tích lỗi ở các mô hình nhằm cải thiện hiệu suất trong tương lai.

*Index Term: mô tả công việc, phân lớp đa nhãn, mô hình transformer, Neural Network, TextCNN, Bi-LSTM, Bi-GRU*

1. **Giới thiệu:**

Với sự phát triển vượt bậc của CNTT và các lĩnh vực liên quan, việc tìm kiếm việc làm thông qua internet đã là điều quá bình thường đối với người tìm việc nói chung và sinh viên nói riêng. Tuy nhiên, một điều không thể trách khỏi là lượng thông tin trong các mô tả công việc là vô cùng đa dạng và có thể diễn tả theo nhiều cách thức khác nhau. Đối với các nhà tuyển dụng, việc đăng tuyển thông tin tìm kiếm ứng cử viên phù hợp một cách chính xác là một việc hết sức cần thiết. Nhất là thông tin về ngành nghề mình đang tuyển, để ứng viên có thể chọn lựa một cách chính xác. Tuy nhiên, trong một website tìm việc, luôn có một số lượng tương đối lớn nhóm các ngành nghề đánh dấu. Điều này làm cho nhà tuyển dụng dễ chọn thiếu, chọn sai khi tìm các ngành nghề phù hợp cho mô tả công việc đó. Do đó, chúng ta cần một mô hình giúp phân loại các thông tin trên mỗi mô tả công việc và đưa ra một hoặc một vài ngành nghề phù hợp với mô tả công việc.

Bài toán phân loại mô tả công việc thành các lĩnh vực, ngành nghề là một bài toán phân loại văn bản (Text Classification). Đây là bài toán trong học máy và xử lý ngôn ngữ tự nhiên để dự đoán các ngành nghề, lĩnh vực dựa vào yêu cầu công việc, kiến thức, kỹ năng, sở thích… Tuy nhiên, đây là một bài toán đa nhãn vì các mô tả công việc thường sẽ có nhiều ngành nghề tương ứng. Bài toán có đầu vào là một văn bản chứa nội dung của mô tả công việc, mô hình sẽ cho ra đầu ra là dự đoán về các ngành nghề, lĩnh vực phù hợp nhất với mô tả công việc đó. Nhãn có thể là một ngành nghề, một lĩnh vực hoặc có thể có nhiều ngành nghề, nhiều lĩnh vực. Bảng 1 thể hiện các ví dụ về đầu vào và đầu ra của bài toán.

* **Đầu vào (Input)**: Một văn bản chứa mô tả công việc (Job Description) nhất định được thu thập từ các trang tìm việc làm.
* **Đầu ra (Output)**: Một danh sách các ngành nghề, lĩnh vực có liên quan nhất đến mô tả công việc đó.

Đầu tiên, để phát triển mô hình và giải quyết bài này, chúng tôi đã thu thập dữ liệu từ các trang tìm kiếm việc làm online ở Việt Nam. Sau khi thu thập, chúng tôi sẽ áp dụng các phương pháp tiền xử lý dữ liệu cho văn bản, các phương pháp học máy và các phương pháp học sâu như TextCNN, Bi-LSTM, Bi-GRU và các mô hình pretrained vào để giải quyết bài toán.

|  |  |  |
| --- | --- | --- |
| Input | Điều phối linh kiện từ kho đến các chi nhánh Liên hệ NCC để hỏi giá và lấy hàng Nhập/xuất hàng trên phần mềm Báo cáo công nợ cuối ngày Kiểm kê kho, liên hệ NCC để bảo hành linh kiện Các công việc khác theo yêu cầu của quản lý | - Thiết kế hiệu ứng, tạo chuyển động nhân vật 2D bằng Spine. - Phối hợp với họa sĩ thiết kế, tạo chuyển động theo ý tưởng. - Thực hiện các công việc khác theo yêu cầu của công ty |
| Output | Vận Tải - Lái xe - Giao nhận, Điện - Điện tử | IT Phần mềm, Thiết kế - Sáng tạo nghệ thuật, Thiết kế - Sáng tạo nghệ thuật |

Bảng 1. Ví dụ về đầu vào và đầu ra của bài toán phân lớp đa nhãn mô tả công việc

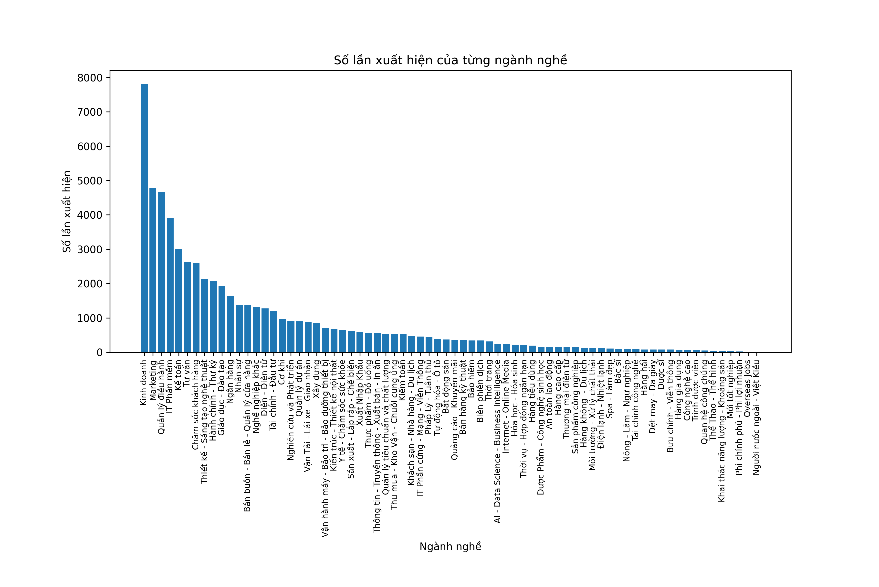
1. **Dữ liệu:**
2. ***Thu thập dữ liệu***:

Chúng tôi thu thập dữ liệu dạng văn bản từ 2 trang tìm kiếm việc làm online: **vietnameseworks** và **vieclam24h**. Dữ liệu chúng tôi đã thu thập bao gồm mô tả công việc, yêu cầu của công việc và các ngành công nghiệp trực tiếp liên quan đến công việc đó (có thể là một ngành nghề hoặc nhiều ngành nghề). Tập dữ liệu sau khi được thu thập có 40090 dòng, có 72 ngành nghề khác nhau (Xem chi tiết ở appendix A).

Bỏ vào appendix. Danh sách các ngành nghề (72 cái)

1. ***Phân tích thăm dò dữ liệu****:*

Sau khi thu thập dữ liệu chúng tôi tiến hành phân tích thăm dò dữ liệu nhằm hiểu thêm về tập dữ liệu này. Hình 1 thể hiện số lần xuất từng ngành nghề trên toàn bộ tập dữ liệu, chúng ta có thể thấy 1 vấn đề rằng dữ liệu đang bị mất cân bằng. Các ngành phổ biến trên thị thường công việc hiện nay như Công nghệ Thông tin, Marketing, … sẽ xuất hiện nhiều hơn các ngành ít phổ biến trên thị trường công việc như Thể thao – thể hình, Khai thác khoáng sản, .... Hình 2 là một phiên bản khác để thể hiện mức độ xuất hiện nhiều hay ít của các ngành nghề theo dạng wordmap.



Hình 1. Số lần xuất hiện theo từng nhãn trong tập dữ liệu.



Hình 2. Wordmap số lần xuất hiện theo từng nhãn trong tập dữ liệu.

1. ***Tiền xử lý dữ liệu***:

Sau khi thu thập dữ liệu hoàn chỉnh 40090 dòng dữ liệu, chúng tôi tiến hành tiền xử lý để xây dựng và chuẩn bị dữ liệu cho quá trình huấn luyện mô hình. Các bước tiền xử lý bao gồm:

+ Xóa tất cả các dòng tiếng Anh đi (1).

+ Lower case (2).

+ Tách từ bằng thư viện VNCoreNLP (3).

+ Xóa tất cả ký tự đặc biệt. (4)

+ Xóa tất cả các stopword của tiếng Việt (5).

Ở bước (1), chúng tôi xóa đi tất cả các dòng dữ liệu chỉ có mô tả bằng tiếng Anh. Sau bước này, dữ liệu chỉ còn lại khoảng 34000 mẫu dữ liệu. Ở bước (2), chúng tôi còn lowercase tất cả các dòng dữ liệu vì ngữ nghĩa của các từ viết hoa và viết thường thường không có thay đổi nhiều nhưng nếu mô hình phải xử lý cả hai thì làm tăng độ phức tạp. Tiếp theo ở bước (3), chúng tôi xóa tất cả các ký tự đặc biệt. Những thành phần này là các phần không quan trọng đối với mô hình và có khả năng gây nhiễu. Tuy nhiên, dấu chấm câu “.” với các mô tả công việc gồm nhiều câu thì đây là cách thức nhận biết đâu là kết câu hoặc các trường hợp đặc biệt khác. Ở bước (4), chúng tôi xóa tất cả các stopwords tiếng Việt. Việc xóa các stopwords giúp cho mô hình được đơn giản hóa, không cần xử lý đến và loại các yếu tố ít liên quan hoặc ít ảnh hưởng đến kết quả đầu ra. Chúng tôi sẽ áp dụng quy trình tiền xử lý này cho cả hai thông tin dạng văn bản là mô tả công việc (job description) và yêu cầu công việc (requirement) sau đó gộp cả hai thành một cột là thông tin công việc (information) tránh việc bỏ sót thông tin.

1. ***Chuẩn bị dữ liệu:***

Chúng tôi phân chia ngẫu nhiên các tập dữ liệu training, validation và test với tỉ lệ lần lượt là 70%, 10% và 20%. Bảng 2 thể hiện chi tiết về số lượng mẫu trong cả 3 tập.

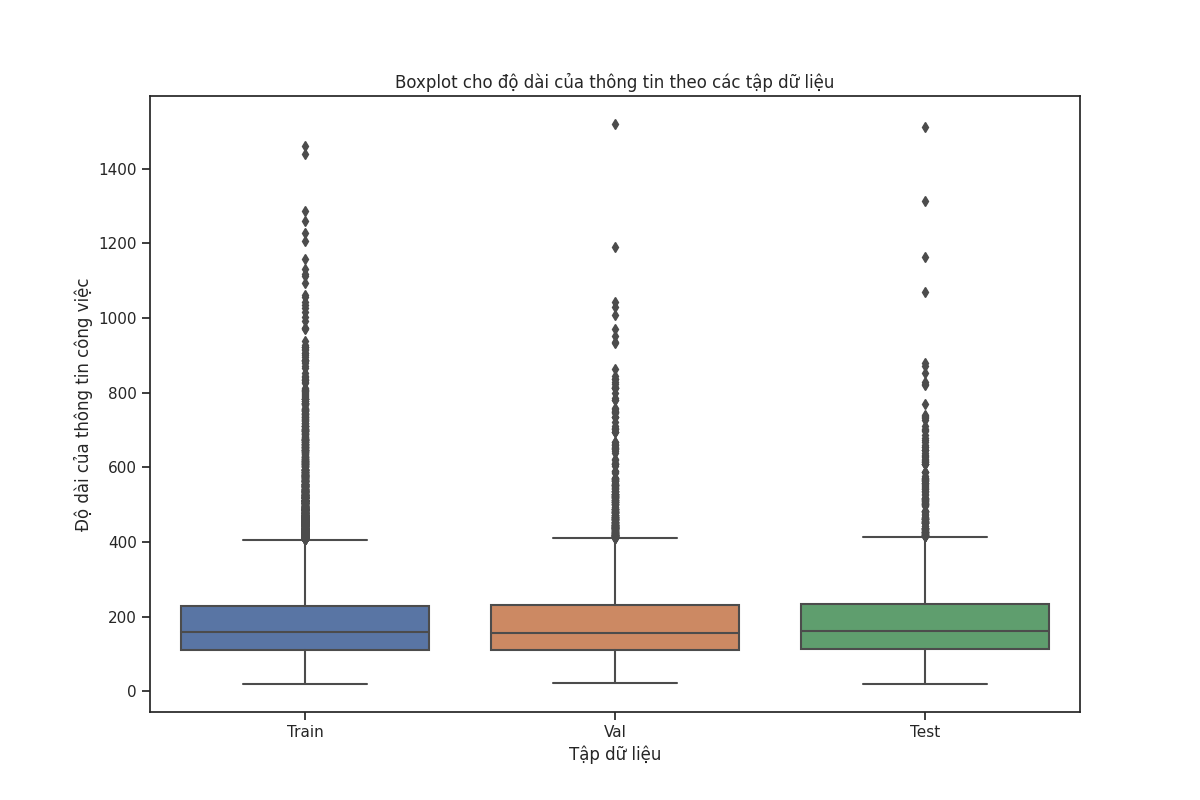
|  |  |
| --- | --- |
|  | Số lượng mẫu |
| Tập dữ liệu traning | 23978 |
| Tập dữ liệu validation | 6851 |
| Tập dữ liệu test | 3426 |

Bảng 2. Số lượng mẫu theo từng tập dữ liệu

Ngoài ra, để chọn ra các siêu tham số phù hợp cho mô hình, chúng tôi đã xem xét tới phân phối về độ dài mô tả công việc của từng tập dữ liệu. Độ dài của từng mô tả công việc được tính theo từng từ trong tiếng Việt. Bảng 3 thể hiện chi tiết thông tin này. Chúng ta có thể thấy rằng độ dài mô tả công việc trong cả 3 tập dữ liệu này khá tương đồng, mặc dù độ dài tối đa trên tập training có phần thấp hơn so với hai tập còn lại. Hình 3 thể hiện phân phân độ dài mô tả công việc trên cả ba tập.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Độ dài tối đa | Độ dài tối thiểu | Độ dài trung bình |
| Tập dữ liệu traning | 1460 | 20 | 187.7 |
| Tập dữ liệu validation | 1520 | 22 | 185.6 |
| Tập dữ liệu test | 1513 | 20 | 190 |

Bảng 3. Thống kê độ dài tối đa, độ dài tối thiểu, độ dài trung bình của các mô tả công việc từng tập dữ liệu

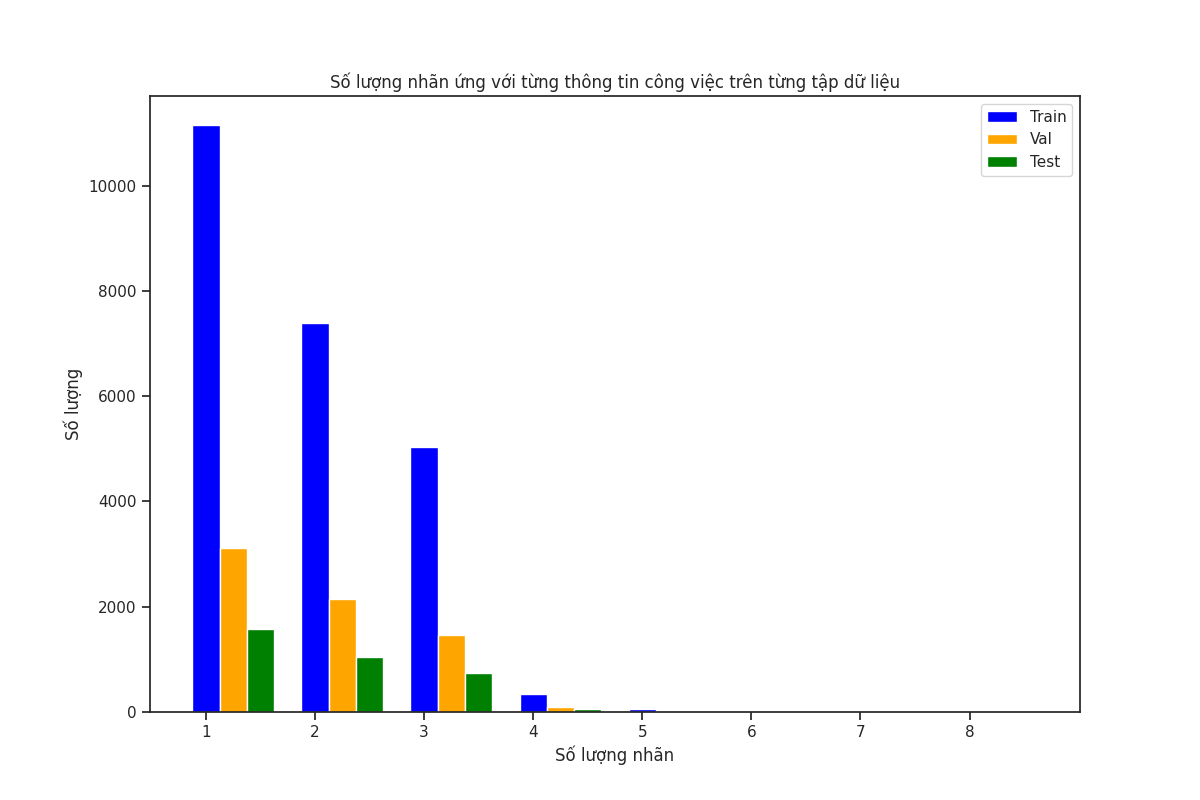


Hình 3. Phân phối độ dài mô tả công việc trên từng tập dữ liệu

Hơn nữa, bởi vì bài toán chúng tôi đang giải quyết là bài toán phân lớp đa nhãn, nên chúng ta cũng cần phải quan tâm đến số lượng nhãn cho từng mô tả công việc được phân bố ra sao. Bảng 4 tóm tắt phân phối số lượng nhãn đối với từng mô tả công việc trên cả ba tập. Chúng ta có thể dễ dàng thấy rằng, các mô tả công việc chỉ có 1 nhãn là nhiều nhất trên cả ba tập. Điều này chứng minh rằng các mô tả công việc có số lượng nhãn càng nhiều thì chiếm thiểu số trong tập dữ liệu.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Train | 11147 | 7391 | 5030 | 339 | 60 | 8 | 1 | 2 |
| Val | 3115 | 2154 | 1462 | 100 | 19 | 1 | 0 | 0 |
| test | 1583 | 1039 | 744 | 51 | 8 | 1 | 0 | 0 |

Bảng 4. Số lượng nhãn đối với từng mô tả công việc trên các tập Training, Validation và Test.



Hình 4. Barplot số lượng nhãn đối với từng mô tả công việc trên các tập training, validation và test

Bên cạnh đó, theo bảng 4 chúng ta có thể thấy rằng số nhãn tối đa của một mô tả công việc có thể có là 8 nhãn. Các mô tả công việc gồm có 8 nhãn chỉ xuất hiện trên tập dữ liệu training. Để thêm phần trực quan, hình 4 mô tả số lương nhãn đối với từng mô tả công việc trên các tập training, validation và test.

1. **Phương pháp tiếp cận:**
2. **Mô hình**

Trong đồ án này, chúng tôi thực nghiệm trên các mô hình học máy đơn giản như: *Linear Regression, Support Vector Machine, Stochastic Gradient Descent* và các mô hình học sâu được ứng dụng tốt cho dữ liệu dạng chuỗi như: *Neural Network, TextCNN, Bidirectional LSTM, Bidirectional GRU*. Các mô hình học máy trên cho kết quả về thời gian huấn luyện khá tốt, mặc khác các mô hình học sâu thì lại vượt trội về mặt xử lý ngữ nghĩa và các mối quan hệ trong câu.

**Linear Regression**

**Linear Regression (Hồi quy tuyến tính)** là một trong những thuật toán cơ bản và phổ biến nhất của**Supervised Learning (Học có giám sát),** trong đó **đầu ra dự đoán là liên tục**."Hồi quy tuyến tính" là một phương pháp thống kê để hồi quy dữ liệu với biến phụ thuộc có giá trị liên tục trong khi các biến độc lập có thể có một trong hai giá trị liên tục hoặc là giá trị phân loại. Nói cách khác "Hồi quy tuyến tính" là một phương pháp để dự đoán biến phụ thuộc (Y) dựa trên giá trị của biến độc lập (X). Nó có thể được sử dụng cho các trường hợp chúng ta muốn dự đoán một số lượng liên tục.

Thuật toán này thích hợp để dự đoán các giá trị đầu ra là các đại lượng liên tục như **doanh số hay giá cả** thay vì cố gắng phân loại chúng thành các đại lượng rời rạc như màu sắc và chất liệu của quần áo, hay xác định đối tượng trong một bức ảnh là mèo hay chó, …

**Stochastic Gradient Descent**

Stochastic Gradient Descent (SGD) là một thuật toán tối ưu hóa thường được sử dụng trong các ứng dụng học máy để tìm các tham số mô hình tương ứng với sự phù hợp nhất giữa đầu ra được dự đoán và thực tế. Từ ‘stochastic’ có nghĩa là một hệ thống hoặc một quá trình được liên kết với một xác suất ngẫu nhiên. **Gradient descent là một thuật toán lặp lại, bắt đầu từ một điểm ngẫu nhiên trên một hàm và đi xuống theo độ dốc của nó theo từng bước cho đến khi nó đạt đến điểm thấp nhất của hàm đó.** Do đó, trong Stochastic Gradient Descent, một vài mẫu được chọn ngẫu nhiên thay vì toàn bộ tập dữ liệu cho mỗi lần lặp.

**Support Vector Machine**

Support Vector Machine là một thuật toán phân loại đa lớp mạnh mẽ trong học máy. SVM có thể được sử dụng để phân loại dữ liệu vào nhiều lớp thông qua các biến thể của thuật toán, bao gồm "One-vs-One" và "One-vs-Rest". SVM tạo ra các siêu phẳng hoặc siêu mặt phân cách giữa các lớp dữ liệu bằng cách tối đa hoá khoảng cách giữa các điểm gần nhất từ các lớp khác nhau. Thuật toán này có khả năng xử lý cả bài toán phân loại tuyến tính và phi tuyến tính. Decision Tree là một thuật toán phân loại đa lớp phổ biến trong học máy. Nó dựa trên cấu trúc cây quyết định, trong đó mỗi nút đại diện cho một quyết định hoặc một thuộc tính để phân loại dữ liệu. Các quyết định được đưa ra tại mỗi nút dựa trên giá trị của thuộc tính và mục tiêu phân loại. Decision Tree có thể được sử dụng trực tiếp cho bài toán phân loại đa lớp và có khả năng xử lý cả dữ liệu rời rạc và liên tục.

**TextCNN**

TextCNN là một mô hình phân loại văn bản dựa trên mạng nơ-ron tích chập (CNN). Mô hình này sử dụng các lớp tích chập để trích xuất các đặc trưng ngữ nghĩa từ các cụm từ liên tiếp trong văn bản. Các đặc trưng này sau đó được kết hợp bằng cách sử dụng lớp gộp tối đa (max-pooling) để tạo ra một biểu diễn toàn cục của văn bản. Biểu diễn này sau đó được đưa vào một lớp kết nối đầy đủ (fully-connected) để thực hiện phân loại. TextCNN có thể được sử dụng để giải quyết nhiều nhiệm vụ phân loại văn bản khác nhau, như phân loại cảm xúc, phân loại chủ đề và phân loại tác giả.

**Bi-LSTM**

Bi-LSTM là viết tắt của Bidirectional Long Short-Term Memory, là một loại mạng nơ-ron hồi quy (RNN) hai chiều. Mô hình này bao gồm hai LSTM riêng biệt, một chạy từ trái sang phải và một chạy từ phải sang trái trên chuỗi đầu vào. Đầu ra của hai LSTM này sau đó được kết hợp lại để tạo ra biểu diễn cuối cùng cho mỗi phần tử trong chuỗi. Bi-LSTM có thể nắm bắt được thông tin ngữ cảnh từ cả hai phía của một phần tử trong chuỗi, do đó giúp cải thiện hiệu suất cho nhiều nhiệm vụ xử lý ngôn ngữ tự nhiên, như nhận dạng thực thể có tên, phân tích cảm xúc và dịch máy.

Bi-GRU

**Bi-GRU**

Bi-GRU là một kiến trúc mạng nơ-ron hồi tiếp (RNN) hai chiều (bidirectional) được sử dụng trong các bài toán xử lý ngôn ngữ tự nhiên (NLP). Nó kết hợp hai GRU (Gated Recurrent Unit) độc lập với nhau để xử lý dữ liệu đầu vào theo cả hai chiều. Mỗi GRU sẽ xử lý dữ liệu theo một chiều và đưa ra kết quả riêng. Sau đó, kết quả của hai GRU sẽ được kết hợp lại để đưa ra kết quả cuối cùng. Bi-GRU được sử dụng trong nhiều bài toán NLP như phân loại cảm xúc, trích xuất quan hệ và nhiều bài toán khác.

1. **Trích xuất đặc trưng**

Vì dữ liệu của chúng ta đang ở dạng văn bản cho nên chúng ta không thể trực tiếp đưa vào mô hình huấn luyện được mà cần phải qua một bước trích xuất đặc trưng hay còn gọi là quá trình ánh xạ chữ sang dạng số. Biểu diễn vector của các dữ liệu dạng văn bản đóng vai trò quan trọng trong các bài toán NLP có thể nâng cao hiệu suất của các mô hình phân loại bằng cách tăng khả năng nắm bắt và hiểu các từ trong ngữ cảnh.

Đối với các mô hình học máy, để đơn giản hóa cho bước này và quá trình huấn luyện, chúng tôi sử dụng phương pháp dựa trên thống kê: *TF-IDF*.

Đối với các mô hình học sâu, chúng tôi sử dụng các mô hình pretrained để có thể biểu diễn các câu ở dạng số có ngữ cảnh và mang các thông tin quan trọng hơn. Ở đây, chúng tôi sử dụng 3 biến thể của BERT:

* *phoBERT*
* *XLMBERT*
* *distilBERT*

### ***TF-IDF***

TF-IDF (Term Frequency-Inverse Document Frequency) là một phương pháp tính toán trọng số cho các từ trong một văn bản dựa trên tần suất xuất hiện của từ đó trong văn bản và trong toàn bộ tập dữ liệu. TF-IDF được sử dụng rộng rãi trong xử lý ngôn ngữ tự nhiên và trích xuất thông tin văn bản. Trọng số TF-IDF cho một từ trong một văn bản được tính bằng cách kết hợp hai thành phần:

* **Term Frequency (TF)**: Đây là đo lường tần suất xuất hiện của từ trong văn bản. Giá trị TF cao cho biết từ đó xuất hiện nhiều trong văn bản.
* **Inverse Document Frequency (IDF)**: Đây là đo lường độ quan trọng của từ đó trong toàn bộ tập dữ liệu. Giá trị IDF cao cho biết từ đó xuất hiện ít trong các văn bản khác.

A picture containing text, font, screenshot, line

Description automatically generated

Hình 20. Công thức tính TF-IDF.

**PhoBert**

**XLMBert**

XLMBert là một mô hình ngôn ngữ đa ngôn ngữ được đào tạo trước bằng cách sử dụng hai mục tiêu học có giám sát: Causal Language Modeling (CLM) và Masked Language Modeling (MLM). Mục tiêu CLM là dự đoán từ tiếp theo trong một câu dựa trên các từ trước đó. Mục tiêu MLM là dự đoán các từ bị che trong một câu dựa trên ngữ cảnh xung quanh. Bằng cách kết hợp hai mục tiêu này, XLMBert có thể học được các đặc trưng ngôn ngữ chung và cụ thể cho từng ngôn ngữ. XLMBert có thể được sử dụng để giải quyết nhiều nhiệm vụ xử lý ngôn ngữ tự nhiên khác nhau, như phân loại văn bản, dịch máy, trả lời câu hỏi và suy luận ngôn ngữ.

**distilBERT**

1. ***Cài đặt thực nghiệm***

Đối với việc tiền xử lý dữ liệu, chúng tôi sử dụng công cụ VNCoreNLP. Đây là công cụ tách từ cho tiếng Việt đem lại hiệu suất cao.

Đối với các mô hình học máy, chúng tôi sẽ áp dụng TF-IDF để trích xuất đặc trưng. Số lượng từ tối đa trong từ điển của TF-IDF là 5000. Từ điển này bao gồm 5000 từ phổ biến nhất trong toàn bộ tập dữ liệu. Chúng tôi sử dụng phương pháp phân tích câu thành các từ riêng lẻ để tính tần suất và trọng số của từng từ.

Đối với các mô hình học sâu, chúng tôi sẽ thực hiện 1 chuỗi các thực nghiệm bằng cách fine-tuning trên các biến thể của BERT bao gồm: phoBERT, XLMBERT và distilBER (đều được hỗ trợ bởi Huggingface) kết hợp với các mô hình như: MLP, TextCNN, Bi-LSTM và Bi-GRU để so sánh kết quả giữa chúng. Độ dài tối đa của câu đầu vào là 200 (200 tokens), 768 đơn vị cho LSTM và GRU. Tất cả các bộ tham số cần thiết của mô hình đều được áp dụng phương thức khởi tạo là Uniform Xavier. Hàm mất mát, thuật toán tối ưu, learning rate được chúng tôi sử dụng lần lượt là Binary Crossentropy, Adam và 2e-5. Ngoài ra, đối với learning rate chúng tôi sử dụng thêm phương pháp điều chỉnh tốc độ học *Linear Learning Rate Scheduler with Warmup* nhằm cân bằng tốc độ học và số bước huấn luyện, đồng thời giúp thuật toán nhanh hội tụ. Tất cả các mô hình trên đều được huấn luyện với batch\_size là 32 với 10 epochs.

1. **Phương pháp đánh giá:**

Trong tác vụ phân loại đa lớp hoặc đa nhãn, các khái niệm về độ đo precision, recall và f1-score có thể được áp dụng độc lập cho từng nhãn. Như vậy đối với mỗi lớp và mỗi độ đo sẽ ứng với một giá trị khác nhau. Có một số cách để kết hợp các kết quả đánh giá khác nhau trên các nhãn bằng cách lấy hàm trung bình trên tất cả các lớp khác nhau:

* + **Micro**: Tính các độ đo trên toàn cục bằng cách đếm tổng số các giá trị true positive, true negative, …
  + **Macro**: Tính các độ đo cho từng nhãn và tìm giá trị trung bình không trọng số của chúng. Phương pháp này bỏ qua việc xem xét đến sự mất cân bằng nhãn.
  + **Weighted**: Tính các độ đo cho từng nhãn và tìm trọng số trung bình của chúng theo trọng số support (số lượng mẫu đối với từng nhãn). Vì vậy, ngược lại với phương pháp ‘macro’ thì phương pháp có xem xét đến sự mất cân bằng nhãn.
  + **Samples**: Tính các độ đo cho từng mẫu dữ liệu và tìm giá trị trung bình (cách này chỉ có ý nghĩa trong việc phân lớp đa nhãn).

**Exact Match Ratio: …**

**Hamming score: …**

1. **Kết quả thực nghiệm và đánh giá:**

Bảng 5 thể hiện toàn bộ kết quả thực nghiệm. Cụ thể, chúng tôi đã tiến hành huấn luyện 15 mô hình bao gồm 3 mô hình học máy và 12 mô hình học sâu khác nhau. Mô hình kết hợp giữa việc fine-tuning XLMBERT và Bi-GRU cho kết quả cao nhất với Haming score, EM score, F1-score lần lượt là 0.57, 0.38 và 0.64. Hơn nữa đối với các mô hình học máy, SVM cho kết quả EM score là 0.38 khá cao so với các mô hình còn lại. Đối với các mô hình sử dụng BERT, XLMBERT cho kết quả cao nhất, tốt hơn cả phoBERT – là mô hình ngôn ngữ cho tiếng Việt và distilBERT.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Mô hình** | **Phương pháp trích xuất đặc trưng** | **Hamming score** | **EM score** | **F1-score** | **Precision** | **Recall** |
| Logistic Regression | TFIDF | 0.47 | 0.33 | 0.52 | 0.61 | 0.49 |
| SGD | TFIDF | 0.46 | 0.34 | 0.51 | 0.60 | 0.48 |
| SVM | TFIDF | 0.54 | **0.38** | 0.59 | 0.68 | 0.57 |
| MLP | phoBERT | 0.55 | 0.37 | 0.61 | **0.69** | 0.59 |
| TextCNN | phoBERT | 0.54 | 0.34 | 0.61 | 0.68 | 0.61 |
| Bi-LSTM | phoBERT | 0.47 | 0.32 | 0.52 | 0.62 | 0.49 |
| Bi-GRU | phoBERT | 0.55 | 0.36 | 0.62 | **0.69** | 0.61 |
| MLP | XLMBERT | 0.55 | 0.36 | 0.62 | 0.68 | 0.61 |
| TextCNN | XLMBERT | 0.56 | 0.35 | 0.62 | 0.67 | **0.64** |
| Bi-LSTM | XLMBERT | 0.52 | 0.36 | 0.59 | 0.67 | 0.56 |
| Bi-GRU | XLMBERT | **0.57** | **0.38** | **0.64** | **0.69** | **0.64** |
| MLP | distilBERT | 0.46 | 0.30 | 0.52 | 0.60 | 0.50 |
| TextCNN | distilBERT | 0.47 | 0.30 | 0.53 | 0.60 | 0.52 |
| Bi-LSTM | distilBERT | 0.42 | 0.29 | 0.46 | 0.55 | 0.43 |
| Bi-GRU | distilBERT | 0.48 | 0.32 | 0.54 | 0.62 | 0.52 |

Bảng 5. Kết quả thực nghiệm.

1. **Phân tích lỗi và hướng phát triển:**

Bảng 6 mô tả và thể hiện số lần dự đoán chính xác hoàn toàn của mô hình đối với các mô tả công việc từ 1 đến 6 nhãn bởi vì trên tập dữ liệu test chỉ có tối đa 6 nhãn. Chúng ta có thể thấy rằng với các mô tả công việc chỉ có đơn nhãn thì công việc dự đoán này sẽ có số lần chính xác cao hơn nhiều so với các mô tả công việc có đa nhãn. Điều này chứng tỏ rằng số lần chính xác sẽ tỉ lệ nghịch với số lượng nhãn của mô tả công việc nhất định. Chính thử thách này làm cản trở bài toán phân lớp đa nhãn (multi-label classification) bởi vì mô hình không chỉ cần một dự đoán chính xác mà phải dự đoán chính xác tất cả. Một điều mà chúng ta dễ nhận thấy rằng, đối với các mô hình học máy chúng tôi chỉ sử dụng TF-IDF (phương pháp trích xuất đặc trưng không có ngữ nghĩa) các mô hình này chỉ tập trung vào việc dự đoán đúng các mẫu chỉ có 1 nhãn. Vì vậy, với các mô tả công việc chỉ có 1 nhãn, số lượng dự đoán chính xác hoàn toàn của mô hình học máy khá cao. Ngược lại, đối với các mô hình học sâu, do sử dụng trích xuất đặc trưng là các biến thể của BERT, cho nên việc hiểu ngữ nghĩa khá tốt. Cho nên các dự đoán chính xác đối với các mô tả công việc có nhiều hơn 1 nhãn là vượt trội hơn hẳn.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Số lượng nhãn** | **1** | **2** | **3** | **4** | **5** | **6** |
| LR correction | 908 | 141 | 0 | 0 | 0 | 0 |
| SGD correction | 938 | 134 | 0 | 0 | 0 | 0 |
| SVM correction | **962** | 171 | 0 | 0 | 0 | 0 |
| MLP + phoBERT | 941 | 262 | 53 | 0 | 0 | 0 |
| TextCNN + phoBERT | 878 | 252 | 48 | 1 | 0 | 0 |
| Bi-LSTM + phoBERT | 946 | 156 | 4 | 0 | 0 | 0 |
| Bi-GRU + phoBERT | 933 | 253 | 50 | 2 | 0 | 0 |
| MLP + XLMBERT | 906 | 287 | 54 | 2 | 0 | 0 |
| TextCNN + XLMBERT | 840 | 288 | **78** | 2 | 0 | 0 |
| Bi-LSTM + XLMBERT | 935 | 242 | 42 | 1 | 0 | 0 |
| Bi-GRU + XLMBERT | 931 | **308** | 70 | **4** | 0 | 0 |
| MLP + distilBERT | 862 | 179 | 20 | 1 | 0 | 0 |
| TextCNN + distilBERT | 840 | 184 | 22 | 0 | 0 | 0 |
| Bi-LSTM + distilBERT | 859 | 137 | 10 | 0 | 0 | 0 |
| Bi-GRU + distilBERT | 879 | 179 | 26 | 0 | 0 | 0 |

Bảng 6. Số lần dự đoán chính xác hoàn toàn xếp theo số lượng nhãn của từng mô tả công việc.

Hơn nữa, chúng ta còn có thể dự đoán rằng các mô hình chưa hiểu hết mô tả công việc hoặc chưa nhận ra đủ thông tin để đưa ra dự đoán chính xác, một phần vì có quá nhiều ngành có mô tả công việc khá tương tự nhau.

1. **Kết luận:**

Trong bài báo cáo lần này, chúng tôi thực hiện 1 chuỗi các thực nghiệm dựa trên các mô hình học máy và học sâu.

1. **Tài liệu tham khảo:**

<https://github.com/vietnlp/etnlp>

https://github.com/vncorenlp/VnCoreNLP

https://huggingface.co/

Job Prediction: From Deep Neural Network Models to Applications - Tin Van Huynh1,2,\*, Kiet Van Nguyen1,2,† , Ngan Luu-Thuy Nguyen1,2,† , and Anh Gia-Tuan Nguyen1,2,†

Predicting Job Titles from Job Descriptions with Multi-label Text Classification – Hieu Trung Tran, Hanh Hong Phuc Vo, Son Thanh Luu

[Multi Label Classification | Solving Multi Label Classification problems (analyticsvidhya.com)](https://www.analyticsvidhya.com/blog/2017/08/introduction-to-multi-label-classification/)

[Multi Label Model Evaluation | Kaggle](https://www.kaggle.com/code/kmkarakaya/multi-label-model-evaluation)

[Metrics for Multilabel Classification | Mustafa Murat ARAT (mmuratarat.github.io)](https://mmuratarat.github.io/2020-01-25/multilabel_classification_metrics)

[Multi Label Classification | Solving Multi Label Classification problems (analyticsvidhya.com)](https://www.analyticsvidhya.com/blog/2017/08/introduction-to-multi-label-classification/)

**Bảng phân công**