**Introduction:**

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.

+ Input: Một văn bản chứa mô tả công việc (Job Description) nhất định được thu thập từ các trang tìm việc làm.

+ 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, LSTM, GRU và cá pretrained mô hình vào để giải quyết bài toán.

Qua đồ án này, chúng tôi đã xây dựng được hoàn chỉnh:

+ Sử dụng dữ liệu về các mô tả công việc ứng với nhiều ngành nghề khác nha (đa nhãn) từ nhiều các website tìm việc làm khác nhau ở Việt Nam: vietnameseworks (9555 dòng) và vieclam24h (10445 dòng).

+ So sánh hiệu suất của nhiều các mô hình học sâu khác nhau cho bài toán phân lớp văn bản đa nhãn: TextCNN, LSTM và GRU cùng với nhiều mô hình pretrained word embedding như: XLMBERT và phoBERT.

**Related Work:**

**Data:**

***Data Collection***: Chúng tôi thu thập dữ liệu textual 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). Bảng A chứ tất cả 69 các ngành công nghiệp có trong tập dữ liệu. Tập dữ liệu sau khi được thu thập có 20000 dòng.

+ Bảng ví dụ về 2 sample dữ liệu

+ Bảng tất cả các industry labels (72 cái)

+ Barplot ra 69 cột onehot Số lần xuất hiện của các nhãn 🡺 Wordmap

***Tiền xử lý***: Sau khi thu thập dữ liệu hoàn chỉnh 20000 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: #, & \*, $, .. bằng RE (lưu ý: dấu “.” có thể có ảnh hưởng đến ý nghĩa trong nhiều trường hợp). (4)

+ Xóa tất cả các stopword của tiếng Việt (4). 🡺 Đơn giản mô hình, đơn giản xử lý và loại các yếu tố ít liên quan hoặc ít ảnh hưởng đến kết quả đầu ra.

+ Join

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 15.000 dòng. Ngoài ra chúng tôi còn lower case tất cả, xóa tất cả các ký tự đặc biệt, xóa tất cả các stopword của tiếng Việt với mong muốn giảm nhiễu trong dữ liệu và nâng cao hiệu suất của mô hình. Đây là những thành phần không quan trọng trong văn bản. Chúng tôi sẽ áp dụng quy trình tiền xử lý này cho cả 2 thông tin dạng text là mô tả công việc (jd) và yêu cầu công việc (requirement) sau đó sẽ gộp 2 chuỗi này lại thành 1 cột gọi tắt là thông tin công việc.

**Phân chia train, dev, test**

Sau đó, chúng tôi phân chia ngẫu nhiên các tập dữ liệu training, validation và testing với tỉ lệ 0.7:0.2:0.1 theo bảng sau.

- Train, dev, test 🡺 0.7,0.2,0.1 🡺

bảng số lượng sample theo từng tập

- maxlen, minlen, avglen từng tập

Bảng thống kê maxlen, minlen, avglen từng tập

Boxplot các len

- Số lượng sample có 1, 2, 3 nhãn theo từng tập

bảng thống kê số lượng label dựa trên các tập

Barplot ra

**Method:**

Machine Learning: tfidf 🡪 sgd, lr, svm

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

Để đơn giản hóa quá trình trích xuất đặc trưng cho mô hình xử lý ngôn ngữ tự nhiên này, chúng em sử dụng phương pháp trích xuất đặc trưng dựa vào thống kê – 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.

**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.

**XLMBert**

Mô hình 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.

**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 xúc1, trích xuất quan hệ2 và nhiều bài toán khác.

Deep Learning:

+ Embedding: pretrained: XLMBERT, DistilBERT, phoBERT, …

+ Loss: BinaryCrossEntropy

+ TextCNN + BERT (DistilBERT, XLMBERT, phoBERT)

+ Bi-GRU-CNN + BERT (DistilBERT, XLMBERT, phoBERT)

+ Bi-LSTM-CNN + BERT (DistilBERT, XLMBERT, phoBERT)

+ Bi-GRU-LSTM-CNN + BERT (DistilBERT, XLMBERT, phoBERT)

Experimental Settings:

Phương pháp khác:

+ Binary Relavance: Tách thành 69 cái mô hình khác nhau sau đó gộp kết quả lại

+ Classifier Chains:

+ Label Powerset

**Evaluation:**

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 TP, TF, …
  + **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 (hoàn toàn đúng)**:

Bảng 4 mô tả và thể hiện số dự đoán chính xác của mô hình. 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 multi-label classification bởi vì mô hình không chỉ cần một dự đoán chính xác mà phải dự đoán chính xác tất cả. …

**Error Analysis:**

Bảng 4 mô tả và thể hiện số dự đoán chính xác của mô hình. 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 multi-label classification bởi vì mô hình không chỉ cần một dự đoán chính xác mà phải dự đoán chính xác tất cả. …

Đưa ra một vài ví dụ về sai lầm của mô hình

- Ví dụ 1: có lẽ mô hình k hiểu hết về text

- Ví dụ 2: mô hình confused các ngành liên quan

- Ví dụ 3:

[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/)