TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA CÔNG NGHỆ THÔNG TIN

TIỂU LUẬN CUỐI KỲ

HỌC PHẦN: KHOA HỌC DỮ LIỆU

NHẬN DẠNG TIÊU ĐỀ BÀI BÁO

HỌ VÀ TÊN SINH VIÊN	LỚP HỌC PHẦN	ĐIỂM BẢO VỆ
Lê Hữu Hưng	20TCLC-KHDL	
Hoàng Nguyên Bách	20TCLC-KHDL	
Đinh Huy Hoàng	20TCLC-KHDL	

GIẢNG VIÊN GIẢNG DẠY:

TS. Ninh Khánh Duy

Đà Nẵng, 05/2023

TÓM TẮT

Trong tiểu luận này, chúng tôi tập trung vào giải quyết vấn đề nhận dạng tiêu đề bài báo từ văn bản. Vấn đề này là một trong những thách thức quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên, do tiêu đề bài báo thường chứa thông tin quan trọng và đóng vai trò quan trọng trong việc thu hút sự chú ý của người đọc.
Phương pháp giải quyết của chúng tôi là sử dụng PhoBert, một mô hình nhúng từ vựng tiếng Việt dựa trên kiến trúc Transformer. PhoBert cho phép chúng tôi biểu diễn các từ trong tiêu đề bài báo dưới dạng các vectơ số thực, giúp mô hình có khả năng hiểu được ngữ nghĩa và tương quan giữa các từ trong tiêu đề.

Kết quả đạt được khi xây dựng mô hình nhận dạng tiêu đề bài báo là khả năng nhận dạng và phân loại chính xác các tiêu đề bài báo từ văn bản. Mô hình của chúng tôi đã được huấn luyện trên một tập dữ liệu đa dạng và đạt được độ chính xác cao trong việc nhận dạng và phân loại các tiêu đề bài báo. Điều này có thể hỗ trợ trong việc tổ chức và phân loại thông tin từ các bài báo, giúp người dùng tiết kiệm thời gian và nỗ lực trong việc tìm hiểu và tìm kiếm thông tin cần thiết từ các tiêu đề bài báo.

BẢNG PHÂN CÔNG NHIỆM VỤ

Sinh viên thực hiện	Các nhiệm vụ	Tự đánh giá theo 3 mức (Đã hoàn thành/Chưa hoàn thành/Không triển khai)
Lê Hữu Hưng	- Xử lý dữ liệu - Trích xuất đặc trưng	- Đã hoàn thành - Đã hoàn thành
Hoàng Nguyên Bách	- Crawl data	- Đã hoàn thành
Đinh Huy Hoàng	- Mô hình hóa dữ liệu	- Đã hoàn thành

MỤC LỤC

1. Giới thiệu

1.1 Lý do chọn đề tài

Trong thế giới thông tin kỹ thuật số ngày nay, hàng ngàn bài báo được xuất bản hàng ngày trên các nền tảng trực tuyến. Việc xác định và phân loại tiêu đề bài báo có thể giúp chúng ta nắm bắt thông tin nhanh chóng và tiết kiệm thời gian. Điều này có thể áp dụng trong nhiều lĩnh vực, bao gồm việc tìm kiếm thông tin, theo dõi xu hướng, và phân tích dữ liệu.

Việc sử dụng mô hình nhận dạng tiêu đề bài báo có thể giúp tự động hóa quy trình công việc liên quan đến xử lý và phân loại thông tin. Thay vì phải đọc từng tiêu đề bài báo một cách thủ công, ta có thể áp dụng mô hình để tự động phân loại và tổ chức thông tin dựa trên tiêu đề. Mô hình nhận dạng tiêu đề bài báo có thể được ứng dụng trong nhiều lĩnh vực thực tế, bao gồm phân loại tin tức, theo dõi sự kiện, quảng cáo và tiếp thị.

Chính vì vậy, nhóm chúng tôi thực hiện đề tài ‘Nhận dạng tiêu đề bài báo’ để có thể phân loại các bài báo theo các tiêu đề một cách tự động.

1.2 Giới thiệu bài toán

1.2.1 Mục tiêu

Xây dựng mô hình phân loại thể loại của các bài báo.

- Đầu vào là của một bài báo bất kì.
- Đầu ra trả về là chủ đề chính của bài báo(bài báo đó thuộc lĩnh vực nào).

1.2.2 Giải pháp

Phương pháp giải quyết của chúng tôi là sử dụng PhoBert, một mô hình nhúng từ vựng tiếng Việt dựa trên kiến trúc Transformer. PhoBert cho phép chúng tôi biểu diễn các từ trong tiêu đề bài báo dưới dạng các vectơ số thực, giúp mô hình có khả năng hiểu được ngữ nghĩa và tương quan giữa các từ trong tiêu đề.

Từ đó có thể xây dựng mô hình nhận dạng tiêu đề bài báo có độ chính xác từ các tiêu đề bài báo văn bản.

2. Thu thập và mô tả dữ liệu

2.1 Thu thập dữ liệu

Là quá trình thu thập các dữ liệu liệu cần thiết dùng để thực hiện dự án. Việc thu thập dữ liệu phục vụ cho xử lý ngôn ngữ tự nhiên có thể được thực hiện trên nhiều hình thức khác nhau. Phổ biến trong đó là thu thập dữ liệu từ các website.

Trang website nhóm chúng tôi thực hiện việc thu thập dữ liệu là: vtv.vn

Highlights | ĐT nữ Việt Nam 4-0 ĐT nữ Campuchia | Bán kết bóng đá nữ SEA Games 32

SEA Games 32 - 12/05/2023

VTV.vn - Tung ra sân đội hình có tới 8 sự thay đổi so với trận đấu trước, tuy nhiên ĐT nữ Việt Nam vẫn tỏ ra quá mạnh so với đối thủ và giành chiến thắng đậm.

Mỗi bài báo trên website sẽ gồm có những nội dung như sau:

- Tiêu đề.
- Ngày viết bài.
- Chủ đề.
- Mô tả ngắn gọn.

Từ những nội dung trên website nhóm chúng tôi quyết định sẽ thu thập nội dung để thực hiện việc huấn luyện mô hình là:

- Tiêu đề.
- Chủ đề.

2.2 Mô tả dữ liệu

2.2.1. Mô tả

Từ những dữ liệu trên nhóm đã thu thập thành công được hai tập dữ liệu.

- Small data: gồm 1374 mẫu.
- Big data: gồm 10981 mẫu.

Số lượng nhãn(tiêu đề) của từng tập dữ liệu bao gồm:

- Big data

Tiêu đề	Số lượng
kinh-te	2326
truyen-hinh	2164
chinh-tri	1856
van-hoa-giai-tri	1564
doi-song	1456
xa-hoi	1005

cong-nghe	548

- Small data

Tiêu đề	**Số lượng**
kinh-te	566
truyen-hinh	213
chinh-tri	189
van-hoa-giai-tri	156
doi-song	113
xa-hoi	102
cong-nghe	36

2.2.2. Nhận xét

Tỷ lệ tiêu đề bài không có sự chênh lệch lớn. Số lượng số lớp còn ít.

3. Trích xuất đặc trưng

3.1 Chọn đặc trưng

Trong bài tiểu luận cuối kỳ, việc xử lý ngôn ngữ tự nhiên là một phần quan trọng để đạt được kết quả chính xác và hiệu quả. Một trong những công đoạn quan trọng trong xử lý ngôn ngữ là loại bỏ các từ dừng (stop words).

Các từ dừng là những từ phổ biến và không mang ý nghĩa quan trọng trong một ngôn ngữ. Chúng thường xuất hiện rất nhiều trong văn bản như "và", "là", "của", "được", và nhiều từ khác. Trong quá trình xử lý ngôn ngữ, ta có thể loại bỏ các từ này để giảm kích thước của văn bản và tập trung vào những từ quan trọng hơn.

Bộ stop words được chúng tôi sử dụng trong đề tài này là thư viện nltk để import stopwords và tạo danh sách stop words cho tiếng Việt. Quy trình loại bỏ stop words trong xử lý ngôn ngữ tự nhiên có thể được áp dụng như sau:

- Phân tách câu thành các từ riêng lẻ.
- So sánh từng từ với danh sách stop words và loại bỏ những từ tương ứng.
- Kết hợp lại các từ còn lại thành câu mới sau khi đã loại bỏ stop words.

3.2 Làm sạch dữ liệu

Tập dữ liệu sẽ được tiền xử lý qua các công đoạn sau:

- Đưa toàn bộ dữ liệu về chữ thường
- Xóa các các kí hiệu đặc biệt
- Xóa các ký tự số
- Xóa các ký tự không cần thiết

3.3 Trích lọc đặc trưng

3.3.1. Các khái niệm

Trong đề tài này chúng nhóm không đề cập sâu vào từng các mô hình mà chỉ đi tìm hiểu sơ lược so sánh ưu nhược điểm của từng loại mô hình.

Thuật toán	Đặc điểm chính	Ưu điểm	Nhược điểm
Word2Vec	Mô hình dựa trên mạng nơ-ron sử dụng cơ chế skip-gram hoặc CBOW để học các biểu diễn từ	● Tính nhất quán: Các từ có nghĩa tương đồng sẽ có biểu diễn gần nhau trong không gian ● Tính tương quan văn bản: Các từ cùng xuất hiện trong ngữ liệu sẽ có biểu diễn gần nhau ● Tính tương tự từ: Có thể tính toán được độ tương tự giữa các từ dựa trên khoảng cách cosine giữa các biểu diễn từ	● Không xử lý được từ không có trong từ điển ● Không học được đại diện từ có ý nghĩa khác nhau dựa trên ngữ cảnh
FastText	Mô hình dựa trên Word2Vec với mục tiêu học biểu diễn từ và cụm từ (subword level)	● Học được biểu diễn cho từ không có trong từ điển ● Tính tương quan văn bản: Các từ cùng xuất hiện trong ngữ liệu sẽ có biểu diễn gần nhau ● Tính tương tự từ: Có thể tính toán được độ tương tự giữa các từ dựa trên khoảng cách cosine giữa các biểu diễn từ	● Mất đi một số tính nhất quán so với Word2Vec ● Cần thời gian và tài nguyên tính toán cao hơn để huấn luyện mô hình

			vì cần xử lý các cụm từ
BERT	Mô hình ngôn ngữ tự nhiên dựa trên Transformer sử dụng pretraining và fine-tuning để học biểu diễn từ và câu	● Khả năng hiểu được ngữ cảnh và ý nghĩa của từ và câu ● Học biểu diễn từ và câu đồng thời, đảm bảo tính toàn vẹn ngữ nghĩa ● Đạt được kết quả tốt trên nhiều tác vụ ngôn ngữ tự nhiên	● Đòi hỏi tài nguyên tính toán và bộ nhớ lớn ● Yêu cầu lượng dữ liệu huấn luyện lớn ● Yêu cầu quá trình pretrain dài và phức tạp
RoBERTa	Mô hình ngôn ngữ tự nhiên dựa trên Transformer, là phiên bản cải tiến của BERT với việc sử dụng phương pháp huấn luyện mở rộng hơn	● Hiểu được ngữ cảnh và ý nghĩa của từ và câu ● Độ chính xác cao và kết quả tốt trên nhiều tác vụ ngôn ngữ tự nhiên ● Xử lý được nhiều ngôn ngữ ● Đạt được kết quả tốt mà không cần tác vụ tiền huấn luyện đặc thù	● Đòi hỏi tài nguyên tính toán và bộ nhớ lớn ● Yêu cầu lượng dữ liệu huấn luyện lớn ● Yêu cầu quá trình huấn luyện lâu hơn so với BERT

3.3.2. Giới thiệu PhoBERT

PhoBERT là một mô hình ngôn ngữ dựa trên kiến trúc Transformer, được phát triển để xử lý ngôn ngữ tiếng Việt. Tên gọi "PhoBERT" kết hợp từ "Phở" - một món ăn đặc trưng của Việt Nam và "BERT" - viết tắt của "Bidirectional Encoder Representations from Transformers". Đây là một mô hình tiên tiến trong lĩnh vực xử lý ngôn ngữ tự nhiên.

3.3.3. Đặc điểm và ưu điểm của PhoBERT

PhoBERT được xây dựng dựa trên kiến trúc Transformer, đây là một mạng nơ-ron sử dụng cơ chế tự chú ý (self-attention) để hiểu và biểu diễn ngôn ngữ. Kiến trúc này cho phép PhoBERT hiểu các mối quan hệ ngữ nghĩa phức tạp và tạo ra các biểu diễn ngôn ngữ chất lượng cao.

Huấn luyện trên dữ liệu tiếng Việt: PhoBERT được huấn luyện trên một lượng lớn dữ liệu tiếng Việt từ nhiều nguồn khác nhau, bao gồm các bài viết trên mạng, tin tức, sách và các tài liệu văn bản khác. Điều này giúp PhoBERT nắm bắt được đặc thù của ngôn ngữ tiếng Việt và có khả năng xử lý hiệu quả các tác vụ liên quan đến ngôn ngữ.

Tính linh hoạt và đa nhiệm: PhoBERT có thể được sử dụng cho nhiều tác vụ xử lý ngôn ngữ tự nhiên như phân loại văn bản, dịch máy, rút trích thông tin và nhiều tác vụ khác. Điều này giúp giảm thiểu công sức huấn luyện và tăng tính linh hoạt trong việc áp dụng PhoBERT cho các bài toán khác nhau.

3.3.4. Số lớp Neural và số tham số của mô hình.

PhoBERT được xây dựng với một số layer được xếp chồng lên nhau, trong phiên bản PhoBERT-base, tổng số layer là 12. Mô hình PhoBERT-base với 12 layer và kích thước embedding 768, có khoảng 110 triệu tham số.

```
RobertaModel(
  (embeddings): RobertaEmbeddings(
    (word_embeddings): Embedding(64001, 768, padding_idx=1)
    (position_embeddings): Embedding(258, 768, padding_idx=1)
    (token_type_embeddings): Embedding(1, 768)
    (LayerNorm): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
    (dropout): Dropout(p=0.1, inplace=False)
  )
  (encoder): RobertaEncoder(
    (layer): ModuleList(
      (0-11): 12 x RobertaLayer(
        (attention): RobertaAttention(
          (self): RobertaSelfAttention(
            (query): Linear(in_features=768, out_features=768, bias=True)
            (key): Linear(in_features=768, out_features=768, bias=True)
            (value): Linear(in_features=768, out_features=768, bias=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
          (output): RobertaSelfOutput(
            (dense): Linear(in_features=768, out_features=768, bias=True)
            (LayerNorm): LayerNorm((768,), eps=1e-05, elementwise_affine=Tru
e)
            (dropout): Dropout(p=0.1, inplace=False)
          )
        )
        (intermediate): RobertaIntermediate(
          (dense): Linear(in_features=768, out_features=3072, bias=True)
          (intermediate_act_fn): GELUActivation()
        )
        (output): RobertaOutput(
          (dense): Linear(in_features=3072, out_features=768, bias=True)
          (LayerNorm): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
          (dropout): Dropout(p=0.1, inplace=False)
        )
      )
    )
  )
  (pooler): RobertaPooler(
    (dense): Linear(in_features=768, out_features=768, bias=True)
    (activation): Tanh()
  )
)
```

3.4 Giảm chiều dữ liệu

PCA là một thuật toán phân tích thành phần chính được sử dụng để giảm chiều dữ liệu từ không gian có số chiều cao xuống không gian có số chiều thấp hơn. Đây là một phương pháp quan trọng trong xử lý và khai thác dữ liệu, đặc biệt là trong việc trực quan hóa và nén dữ liệu.

Thuật toán PCA hoạt động bằng cách tìm các thành phần chính của dữ liệu, đó là các vectơ riêng ứng với các giá trị riêng của ma trận hiệp phương sai của dữ liệu. Các thành phần chính này biểu diễn các hướng chính của phương sai trong dữ liệu và giúp chúng ta hiểu được mối quan hệ giữa các điểm dữ liệu.

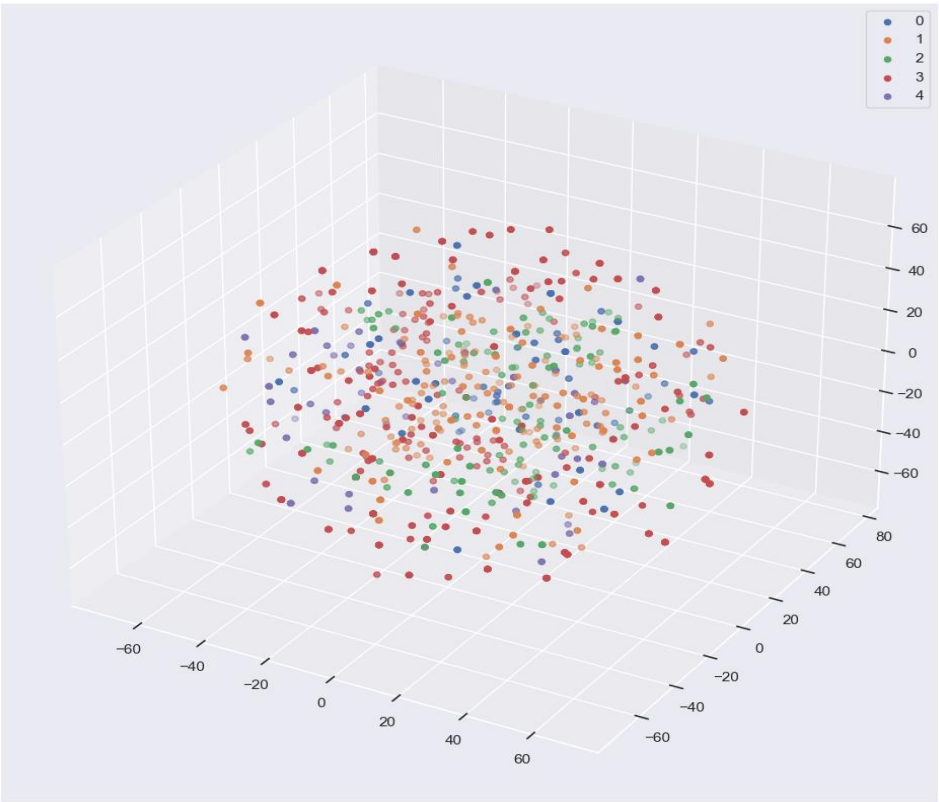
Trong đề tài này nhóm sử dụng PCA để giảm chiều dữ liệu vector embedding 768 chiều về 128 và 512 chiều để so sánh độ chính của mô hình đối với các chiều dài vector khác nhau.

3.5 Trực quan hóa dữ liệu

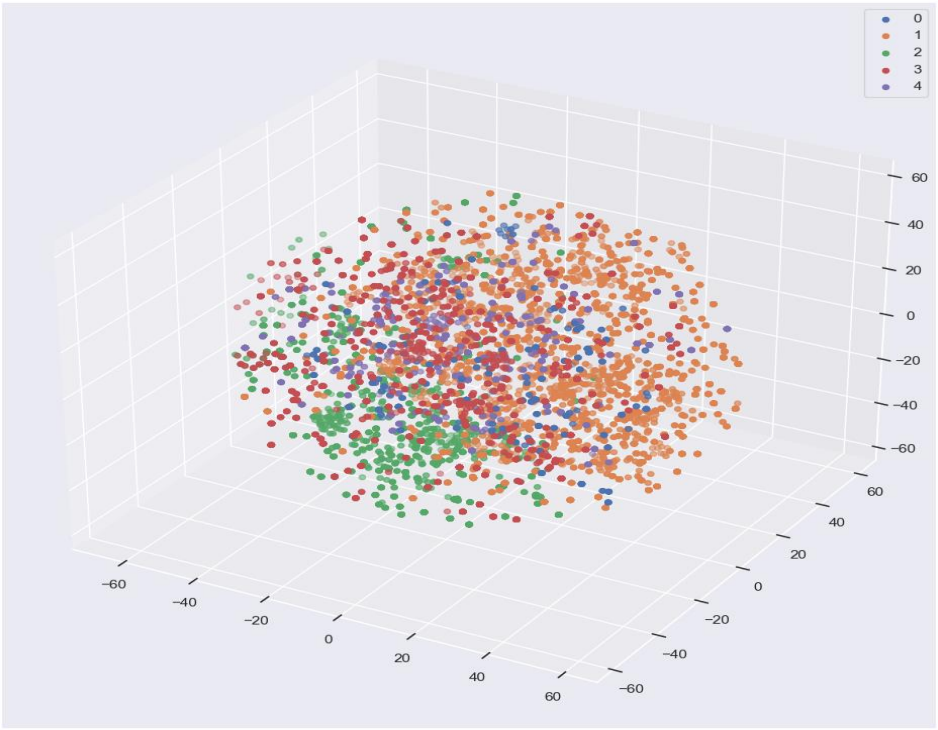
Trực quan hóa dữ liệu từ tập dữ liệu huấn luyện và được vector hóa bằng PhoBert.

Sử dụng thuật toán giảm chiều t-SNE để trực quan vector 768 chiều khi mã hóa bằng PhoBert ở không gian 3 chiều.

- Small data

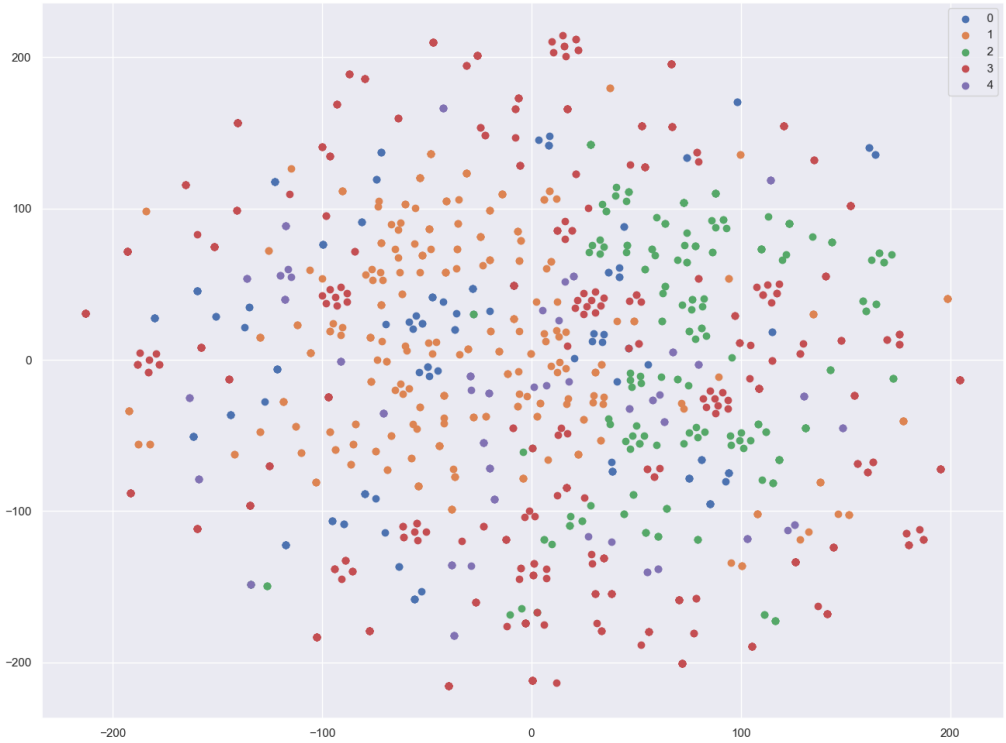


- Big data

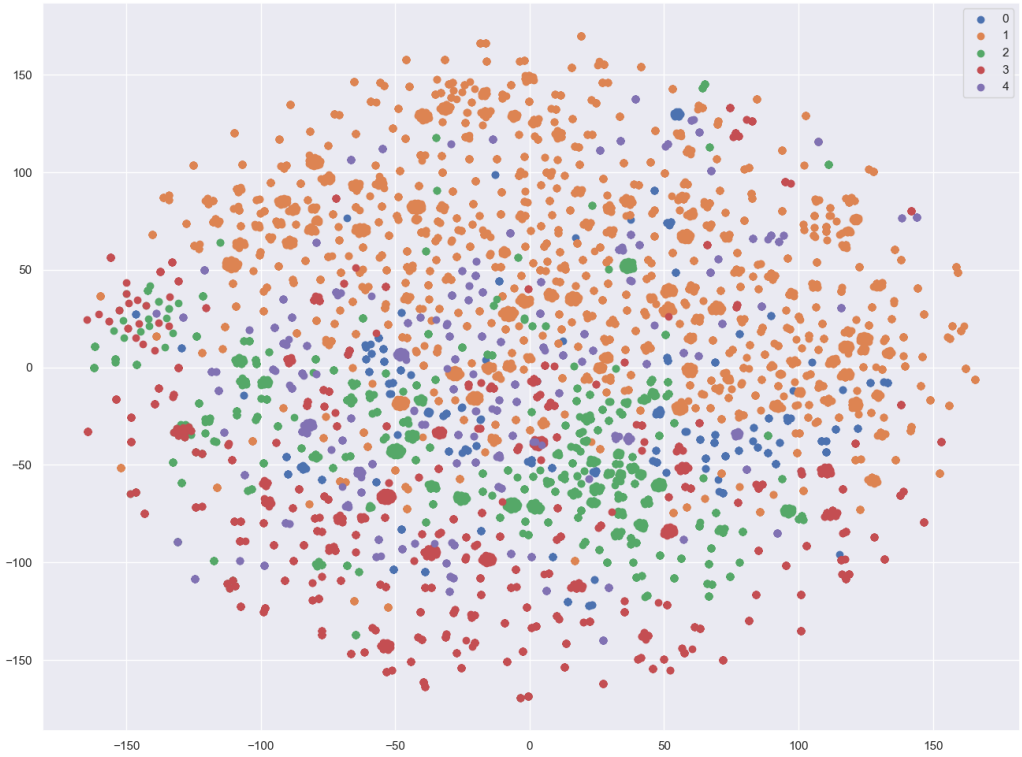


Sử dụng thuật toán giảm chiều t-SNE để trực quan vector 768 chiều khi mã hóa bằng PhoBert ở không gian 2 chiều.

- Small data



- Big data

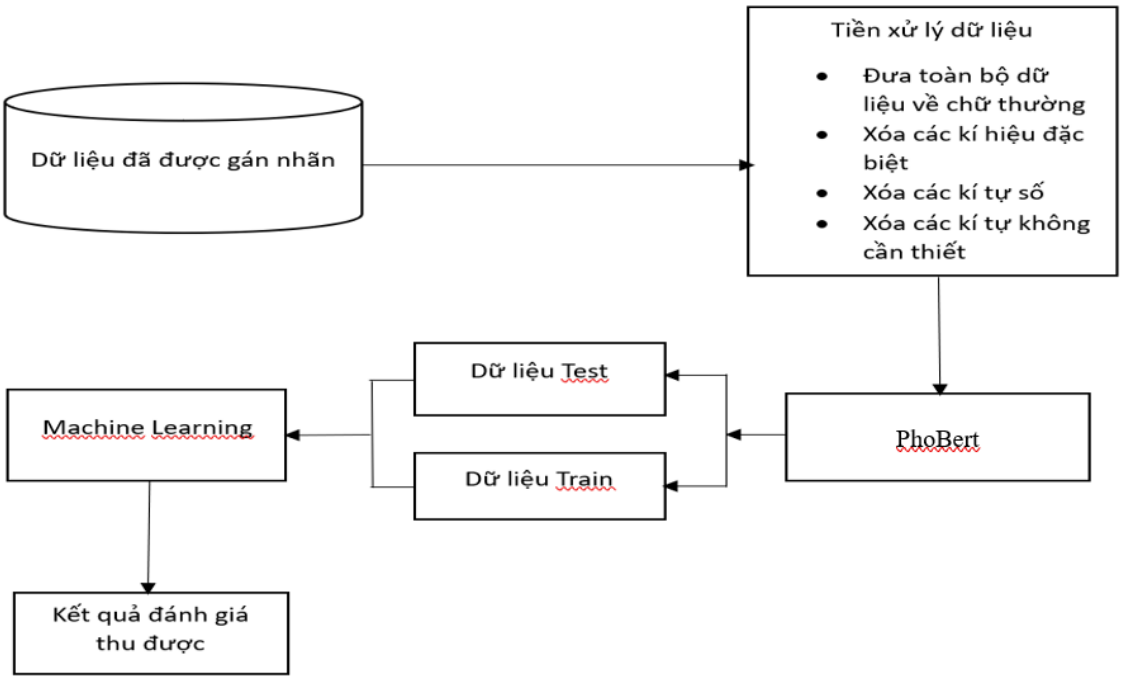


4. Mô hình hóa dữ liệu

4.1. Phát biểu bài toán

Đầu vào: Văn bản (tiêu đề bài báo).

Đầu ra: Dự đoán của mô hình (lĩnh vực của bài báo)



4.2. Phân Chia dữ liệu

4.2.1. Repeated hold-out

Repeated hold-out: Đây là một phương pháp đơn giản trong đó dữ liệu được chia thành hai phần, một phần được sử dụng để huấn luyện mô hình và phần còn lại được sử dụng để đánh giá hiệu suất của mô hình. Quá trình này được lặp lại nhiều lần để đảm bảo độ tin cậy của kết quả. Repeated hold-out thường được sử dụng khi dữ liệu huấn luyện là hạn chế và không đủ để thực hiện phương pháp cross-validation.

4.2.2. Cross-validation

Repeated hold-out: Đây là một phương pháp đơn giản trong đó dữ liệu được chia thành hai phần, một phần được sử dụng để huấn luyện mô hình và phần còn lại được sử dụng để đánh giá hiệu suất của mô hình. Quá trình này được lặp lại nhiều lần để đảm bảo độ tin cậy của kết quả. Repeated hold-out thường được sử dụng khi dữ liệu huấn luyện là hạn chế và không đủ để thực hiện phương pháp cross-validation.

4.3. Thuận toán

4.3.1. Support Vector Machine (SVM)

SVM là một thuật toán giám sát, nó có thể sử dụng cả việc phân loại hoặc đệ quy. Tuy nhiên nó thường được sử dụng cho việc phân loại. Thuật toán SVM chia các lớp dữ liệu bằng một siêu mặt phẳng d-1 chiều khi số chiều của dữ liệu huấn luyện là d.

Cho trước 1 tập huấn luyện được biểu diễn trong không gian vector, trong đó mỗi tiêu đề bài báo là một điểm, phương pháp này tìm ra một siêu mặt phẳng. Hiệu quả xác định siêu mặt này được quyết định bởi khoảng cách của các điểm gần mặt phẳng nhất của mỗi lớp . Khoảng cách càng lớn thì mặt phẳng quyết định càng tốt đồng nghĩa với việc phân loại càng chính xác và ngược lại. Mục đích cuối cùng của phương pháp là tìm khoảng cách biên lớn nhất.

- Mục đích: Dùng để phân lớp dữ liệu mới thuộc lớp nào.
- Đầu vào: Dữ liệu huấn luyện là các bài báo đã gán nhãn với các lớp là các chủ đề của nó.
- Đầu ra: Dùng để tìm ra nhãn cho các dữ liệu mới.

Khoảng cách từ một điểm (vecto) có tọa độ x0 tới siêu mặt phẳng có phương trình $\mathbf{w}^T\mathbf{x} + b = 0$ được xác định bởi:

$$\frac{|\mathbf{w}^T\mathbf{x}_0 + b|}{||\mathbf{w}||_2}$$

Với $||\mathbf{w}||_2 = \sqrt{\sum_{i=1}^{d} w_i^2}$ với d là số chiều của không gian.

Margin được tính là khoảng cách gần nhất từ 1 điểm đến mặt phẳng đó (bất kể điểm nào trong các lớp với cặp dữ liệu bất kỳ (xn, yn)):

$$\text{margin} = \min_n \frac{y_n(\mathbf{w}^T\mathbf{x}_n + b)}{||\mathbf{w}||_2}$$

Bài toán tối ưu trong SVM chính là bài toán tìm w và b sao cho margin này đạt giá trị lớn nhất:

$$(\mathbf{w}, b) = \arg\max_{\mathbf{w},b} \left\{ \min_n \frac{y_n(\mathbf{w}^T\mathbf{x}_n + b)}{||\mathbf{w}||_2} \right\} = \arg\max_{\mathbf{w},b} \left\{ \frac{1}{||\mathbf{w}||_2} \min_n y_n(\mathbf{w}^T\mathbf{x}_n + b) \right\} \quad (1)$$

Bài toán (1) có thể đưa về bài toán tối ưu có ràng buộc sau đây:

$$(\mathbf{w}, b) = \arg\max_{\mathbf{w},b} \frac{1}{||\mathbf{w}||_2}$$
$$\text{subject to:} \quad y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1, \forall n = 1, 2, \ldots, N \quad (2)$$

Bằng một biến đổi đơn giản, ta có thể đưa bài toán này về bài toán dưới đây:

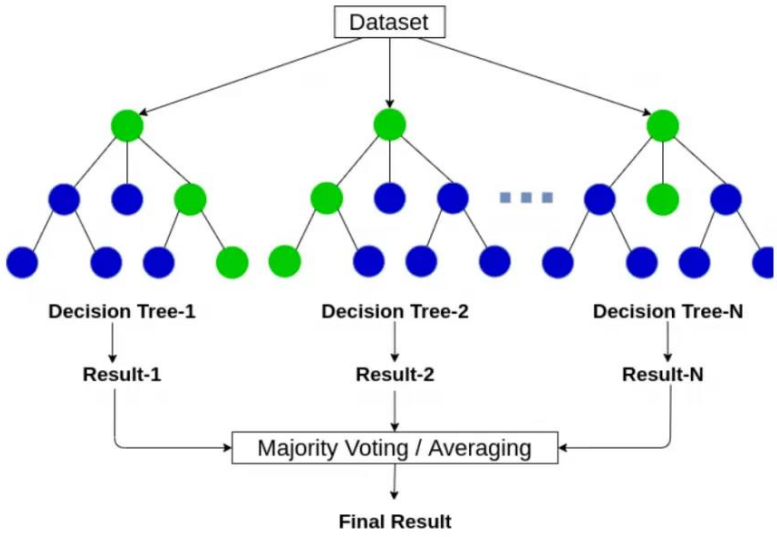
$$(\mathbf{w}, b) = \arg\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||_2^2$$
$$\text{subject to:} \quad 1 - y_n(\mathbf{w}^T\mathbf{x}_n + b) \leq 0, \forall n = 1, 2, \ldots, N \quad (3)$$

Xác định lớp cho một điểm dữ liệu mới: Sau khi tìm được mặt phân cách $\mathbf{w}^T\mathbf{x} + b = 0$, lớp của bất kì một điểm nào sẽ được xác định đơn giản bằng cách:

$$\text{class}(\mathbf{x}) = \text{sgn}(\mathbf{w}^T\mathbf{x} + b)$$

4.3.2. Random Forest Classifier

Random forest là một phương pháp thống kê mô hình hóa bằng máy dùng để phục vụ các mục đích phân loại, tính hồi quy và các nhiệm vụ khác bằng cách xây dựng nhiều cây quyết định. Một cây quyết định là một cách đơn giản để biểu diễn một giao thức. Nói cách khác, cây quyết định biểu diễn một kế hoạch, trả lời câu hỏi phải làm gì trong một hoàn cảnh nhất định. Mỗi Node của cây là một thuộc tính và các nhánh là giá trị lựa chọn của thuộc tính đó. Bằng cách đi theo các giá trị thuộc tính trên cây, cây quyết định sẽ cho ta biết giá trị dự đoán. Phương pháp này cho thấy hiệu quả hơn so với thuật toán phân loại thường được sử dụng vì nó có khả năng tìm ra các thuộc tính nào quan trọng hơn so với các thuộc tính khác. Trên thực tế, nó còn có thể chỉ ra rằng một số thuộc tính là không có tác dụng trong cây quyết định.



Thuật toán lấy mẫu cho phương pháp random forest ứng dụng cho các phương pháp sử dụng thuật toán mô tả thống kê để ước lượng số lượng từ một mẫu dữ liệu (bagging). Ví dụ như một tập mẫu X = x1, ..., xn với các câu trả lời Y= y1, ..., yn, lấy giá trị trung bình (B lần), chọn một mẫu ngẫu nhiên từ bộ mẫu phù hợp với cây quyết định:

Lặp $b = 1, \ldots, B$:

n mẫu từ giá trị tọa độ (X, Y); gọi là (Xb, Yb);

4.4. Đánh giá thuật toán

4.4.1. Accuracy

Độ bao phủ đo lường tỷ lệ các mẫu dương tính được phân loại chính xác so với tổng số mẫu dương tính. Nó đánh giá khả năng của mô hình trong việc bao phủ tất cả các mẫu dương tính.

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$

4.4.2. Recall

Độ bao phủ đo lường tỷ lệ các mẫu dương tính được phân loại chính xác so với tổng số mẫu dương tính. Nó đánh giá khả năng của mô hình trong việc bao phủ tất cả các mẫu dương tính

$$recall = \frac{TP}{TP + FN}$$

4.4.3. F1 Score

Độ bao phủ đo lường tỷ lệ các mẫu dương tính được phân loại chính xác so với tổng số mẫu dương tính. Nó đánh giá khả năng của mô hình trong việc bao phủ tất cả các mẫu dương tính

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

4.4.4. Ma trận nhầm lẫn

Ma trận nhầm lẫn (confusion matrix) là một bảng đa chiều sử dụng để đánh giá hiệu suất của một mô hình phân loại trong bài toán học máy. Ma trận này cho thấy sự so sánh giữa nhãn thực tế và nhãn dự đoán của mô hình trên tập dữ liệu kiểm tra.

- TP (True Positive): Đại diện cho số lượng các mẫu dương tính (positive) được dự đoán chính xác là dương tính. Đây là những trường hợp mà mô hình dự đoán chính xác các mẫu thuộc lớp dương tính và nhãn thực tế của chúng cũng là dương tính.
- FP (False Positive): Đại diện cho số lượng các mẫu âm tính (negative) được dự đoán là dương tính sai. Đây là những trường hợp mà mô hình dự đoán mẫu thuộc lớp dương tính, nhưng nhãn thực tế của chúng lại là âm tính.
- FN (False Negative): Đại diện cho số lượng các mẫu dương tính bị dự đoán là âm tính sai. Đây là những trường hợp mà mô hình dự đoán mẫu thuộc lớp âm tính, nhưng nhãn thực tế của chúng lại là dương tính.
- TN (True Negative): Đại diện cho số lượng các mẫu âm tính được dự đoán chính xác là âm tính. Đây là những trường hợp mà mô hình dự đoán chính xác các mẫu thuộc lớp âm tính và nhãn thực tế của chúng cũng là âm tính.

5. Kết luận

5.1 Kết quả đạt được

- Small data

	Dataset	Normalize	Dimension	Model	evaluation	accuracy	recall	f1
0	DataFrame	None	128	SVC	training_model_repeat_holdout	0.932446	0.913474	0.918216
1	DataFrame	None	128	SVC	training_model_cross_val	0.941818	0.921828	0.924988
2	DataFrame	None	128	RandomForestClassifier	training_model_repeat_holdout	0.929540	0.898803	0.915835
3	DataFrame	None	128	RandomForestClassifier	training_model_cross_val	0.947636	0.923873	0.938936
4	DataFrame	None	256	SVC	training_model_repeat_holdout	0.936804	0.916962	0.923382
5	DataFrame	None	256	SVC	training_model_cross_val	0.944000	0.923152	0.926640
6	DataFrame	None	256	RandomForestClassifier	training_model_repeat_holdout	0.922518	0.894282	0.909681
7	DataFrame	None	256	RandomForestClassifier	training_model_cross_val	0.945455	0.922584	0.936908
8	DataFrame	None	512	SVC	training_model_repeat_holdout	0.938257	0.918937	0.925011
9	DataFrame	None	512	SVC	training_model_cross_val	0.944727	0.924442	0.928273
10	DataFrame	None	512	RandomForestClassifier	training_model_repeat_holdout	0.921308	0.893304	0.908483
11	DataFrame	None	512	RandomForestClassifier	training_model_cross_val	0.952000	0.927861	0.941752
12	DataFrame	None	768	SVC	training_model_repeat_holdout	0.938257	0.918937	0.925011
13	DataFrame	None	768	SVC	training_model_cross_val	0.944727	0.924442	0.928273
14	DataFrame	None	768	RandomForestClassifier	training_model_repeat_holdout	0.935109	0.904388	0.922613
15	DataFrame	None	768	RandomForestClassifier	training_model_cross_val	0.952000	0.926520	0.943280
16	DataFrame	Norm	128	SVC	training_model_repeat_holdout	0.811138	0.719598	0.750605
17	DataFrame	Norm	128	SVC	training_model_cross_val	0.830545	0.753908	0.785808
18	DataFrame	Norm	128	RandomForestClassifier	training_model_repeat_holdout	0.924213	0.895064	0.912620
19	DataFrame	Norm	128	RandomForestClassifier	training_model_cross_val	0.941818	0.919397	0.935462
20	DataFrame	Norm	256	SVC	training_model_repeat_holdout	0.842857	0.763052	0.794309
21	DataFrame	Norm	256	SVC	training_model_cross_val	0.864000	0.799051	0.827717
22	DataFrame	Norm	256	RandomForestClassifier	training_model_repeat_holdout	0.921308	0.893477	0.909015
23	DataFrame	Norm	256	RandomForestClassifier	training_model_cross_val	0.944000	0.921092	0.935504
24	DataFrame	Norm	512	SVC	training_model_repeat_holdout	0.844552	0.767694	0.798750
25	DataFrame	Norm	512	SVC	training_model_cross_val	0.864727	0.799657	0.829067
26	DataFrame	Norm	512	RandomForestClassifier	training_model_repeat_holdout	0.921792	0.894188	0.909672
27	DataFrame	Norm	512	RandomForestClassifier	training_model_cross_val	0.952727	0.929075	0.943232
28	DataFrame	Norm	768	SVC	training_model_repeat_holdout	0.844552	0.767694	0.798750
29	DataFrame	Norm	768	SVC	training_model_cross_val	0.864727	0.799657	0.829067
30	DataFrame	Norm	768	RandomForestClassifier	training_model_repeat_holdout	0.929782	0.899610	0.917705
31	DataFrame	Norm	768	RandomForestClassifier	training_model_cross_val	0.947636	0.923687	0.939737

- Big data

	Dataset	Normalize	Dimension	Model	evaluation	accuracy	recall	f1
0	DataFrame	None	128	SVC	training_model_repeat_holdout	0.932688	0.913221	0.918423
1	DataFrame	None	128	SVC	training_model_cross_val	0.941818	0.921567	0.925894
2	DataFrame	None	128	RandomForestClassifier	training_model_repeat_holdout	0.931477	0.902062	0.918583
3	DataFrame	None	128	RandomForestClassifier	training_model_cross_val	0.944000	0.920720	0.936579
4	DataFrame	None	256	SVC	training_model_repeat_holdout	0.937046	0.917252	0.923379
5	DataFrame	None	256	SVC	training_model_cross_val	0.944727	0.924442	0.927438
6	DataFrame	None	256	RandomForestClassifier	training_model_repeat_holdout	0.922276	0.894554	0.910036
7	DataFrame	None	256	RandomForestClassifier	training_model_cross_val	0.950545	0.927304	0.940755
8	DataFrame	None	512	SVC	training_model_repeat_holdout	0.938257	0.918937	0.925011
9	DataFrame	None	512	SVC	training_model_cross_val	0.944727	0.924442	0.928273
10	DataFrame	None	512	RandomForestClassifier	training_model_repeat_holdout	0.921308	0.893304	0.908483
11	DataFrame	None	512	RandomForestClassifier	training_model_cross_val	0.952000	0.927861	0.941752
12	DataFrame	None	768	SVC	training_model_repeat_holdout	0.938257	0.918937	0.925011
13	DataFrame	None	768	SVC	training_model_cross_val	0.944727	0.924442	0.928273
14	DataFrame	None	768	RandomForestClassifier	training_model_repeat_holdout	0.935109	0.904388	0.922613
15	DataFrame	None	768	RandomForestClassifier	training_model_cross_val	0.952000	0.926520	0.943280
16	DataFrame	Norm	128	SVC	training_model_repeat_holdout	0.815738	0.723587	0.756102
17	DataFrame	Norm	128	SVC	training_model_cross_val	0.834182	0.755735	0.787360
18	DataFrame	Norm	128	RandomForestClassifier	training_model_repeat_holdout	0.925182	0.894831	0.912228
19	DataFrame	Norm	128	RandomForestClassifier	training_model_cross_val	0.944727	0.921336	0.937553
20	DataFrame	Norm	256	SVC	training_model_repeat_holdout	0.842131	0.762001	0.793301
21	DataFrame	Norm	256	SVC	training_model_cross_val	0.865455	0.800907	0.830001
22	DataFrame	Norm	256	RandomForestClassifier	training_model_repeat_holdout	0.922518	0.894714	0.910048
23	DataFrame	Norm	256	RandomForestClassifier	training_model_cross_val	0.946182	0.923537	0.937430
24	DataFrame	Norm	512	SVC	training_model_repeat_holdout	0.844552	0.767694	0.798750
25	DataFrame	Norm	512	SVC	training_model_cross_val	0.864727	0.799657	0.829067
26	DataFrame	Norm	512	RandomForestClassifier	training_model_repeat_holdout	0.921792	0.894188	0.909672
27	DataFrame	Norm	512	RandomForestClassifier	training_model_cross_val	0.952727	0.929075	0.943232
28	DataFrame	Norm	768	SVC	training_model_repeat_holdout	0.844552	0.767694	0.798750
29	DataFrame	Norm	768	SVC	training_model_cross_val	0.864727	0.799657	0.829067
30	DataFrame	Norm	768	RandomForestClassifier	training_model_repeat_holdout	0.929782	0.899610	0.917705
31	DataFrame	Norm	768	RandomForestClassifier	training_model_cross_val	0.947636	0.923687	0.939737

5.2 Nhận xét

5.2.1. Trên cùng một tập dữ liệu

Chia tập dữ liệu để huấn luyện mô hình bằng 2 phương pháp: repeat hold-out và cross-validate thì các metric đánh giá độ chính xác cho ra kết quả chênh lệch không lớn (1-2%). Vậy nên các metric đánh giá là đáng tin cậy, và việc chia dữ liệu bằng phương pháp nào cũng hiệu quả. Nhưng trong các trường hợp cross-validate luôn nhỉnh hơn về độ chính xác so với repeat hold-out.

Sau khi trích xuất đặc trưng ta sẽ thực hiện chuẩn hóa các vector đặc trưng: thì các metric đánh giá độ chính xác giảm đi khá nhiều(5- 7%). Nguyên nhân là vì đề tài sử dụng

mô hình trích xuất đặc trưng pre-train đã quá tốt, nên khi chuẩn hóa sẽ làm giảm độ chính xác của các metric đánh giá đo được.

Đối với giảm chiều dữ liệu:

- 512(dimension): **95.20%** (RandomForestClassifier - cross-validate)
- 256(dimension): **94.54%** (RandomForestClassifier - cross-validate)
- 128(dimension): **94.76%** (RandomForestClassifier - cross-validate)
- So với 768(dimension) không giảm chiều: 95.20% (RandomForestClassifier - cross-validate).

Vậy giảm chiều dữ liệu không làm giảm đi độ chính xác của mô hình nên nhóm khuyến khích sử dụng giảm chiều dữ liệu để cho mô hình xử lý nhanh và hiệu quả hơn.

Tổng kết: Trong đề tài này chúng tôi đã thực hiện mô hình hóa dữ liệu với nhiều phương pháp khác nhau để có thể so sánh được mức độ hiệu quả hay sự chênh lệch về độ chính xác của từng phương pháp với nhau dựa trên bảng tổng hợp phía trên.

5.2.2. Trên hai tập dữ liệu khác nhau

Tổng kết:

- Trong đề tài này chúng tôi đã thực hiện mô hình hóa dữ liệu với hai tập dữ liệu khác nhau là small data (~1000 mẫu) và big data (~10000 mẫu) thì chúng tôi thấy độ chính khác khi áp dụng các phương pháp giống nhau cho cả hai tập thì độ chính xác chênh lệch không lớn. Chênh lệch từ (2-3%) đối với mỗi phương pháp.
- Tập dữ liệu lớn (big data) luôn luôn có độ chính xác cao hơn, bởi lẽ mô hình được huấn luyện nhiều hơn nên có tính khái quát hóa cao hơn.
- Thời gian huấn luyện của tập dữ liệu lớn sẽ nhiều hơn so với tập dữ liệu nhỏ.
- Độ chính xác của mô hình sẽ đáng tin cậy hơn trên tập dữ liệu lớn so với tập dữ liệu nhỏ.

5.3 Giới hạn của đề tài

Về cơ bản, đề tài đã thực hiện hoàn thành yêu cầu đặt ra là giúp người dùng có thể phân loại thể loại các bài báo. Tuy nhiên tập dữ liệu ban đầu chỉ ở mức thử nghiệm.

Điểm hạn chế là đề tài đang sử dụng mô hình embedding chưa tinh chỉnh để phù hợp hơn với yêu cầu bài toán.

6. Tài liệu tham khảo

[1] BERT, RoBERTa, PhoBERT, BERTweet: Ứng dụng state-of-the-art pre-trained model cho bài toán phân loai văn bản (viblo.asia)

[2] Nhận diện cảm xúc văn bản với PhoBERT, Hugging Face - Mì AI (miai.vn)