BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA TOÁN – TIN HỌC

ĐỒ ÁN MÔN HỌC
XỬ LÝ NGÔN NGỮ TỰ NHIÊN

PHÂN BIỆT TIN GIẢ TRÊN VĂN BẢN TIẾNG ANH

Họ và tên: Kha Thái Hồ
MSSV: 22280025
Họ và tên: Lê Phan Ngọc Hiếu
MSSV: 22280023

Mục lục

PHẦN 1: GIỚI THIỆU ĐỀ TÀI

I. Đặt vấn đề

Sự lan truyền rộng rãi của tin giả và các hình thức tuyên truyền sai lệch đã gây ra những rủi ro nghiệm trọng đối với xã hội, bao gồm làm xói mòn niềm tin cộng đồng, gia tăng chia rẽ chính trị, thao túng bầu cử và đặc biệt nguy hiểm hơn trong tình huống đại dịch hay xung đột. Dưới góc nhìn của một nhà nghiên cứu Xử lý ngôn ngữ tự nhiên (NLP), xác định được tin giả vẫn luôn là một thách thức.

Về mặt ngôn ngữ, tin giả thường bắt chước văn phong và cấu trúc của báo chí chính thống, khiến cho các đặc trưng hình thức trở nên kém hiệu quả trong việc xác định thật giả. Sự thiếu vắng của các tập dữ liệu được gán nhãn đáng tin và theo kịp thời đại, đặc biệt đối với những ngôn ngữ và khu vực khác nhau, đã làm giảm độ hiệu quả của các mô hình học có giám sát. Sự linh hoạt và chống cự của tin giả khiến các tác nhân xấu liên tục thay đổi ngôn ngữ và phương pháp để tránh các hệ thống phát hiện. Ngữ cảnh, văn hóa hay thái độ, định kiến ngầm cũng góm phần gia tăng độ phức tạp trong phân tích.

Ngoài ra, các mô hình NLP còn có nguy cơ làm tăng độ lệch dữ liệu huấn luyện, dẫn đến phân loại thiếu công bằng và kiểm duyệt nội dung không chính thống. Những khó khăn này cho thấy sự cần thiết về các phương pháp tiếp cận thận trọng, có nhận thức ngữ cảnh; nếu không giải quyết chính xác, có thể vô tình góp phần lan truyền thông tin sai lệch.

II. Mục tiêu nghiên cứu:

Trước thực trạng phức tạp và tác động tiêu cực của tin giả đối với đời sống xã hội, nghiên cứu này hướng đến việc xây dựng một mô hình xử lý ngôn ngữ tự nhiên (NLP) hiệu quả nhằm phát hiện và phân loại tin giả một cách chính xác. Cụ thể, các mục tiêu chính của nghiên cứu bao gồm:

- **Khảo sát và phân tích đặc trưng ngôn ngữ của tin giả**, từ đó nhận diện các đặc điểm tiềm năng có thể hỗ trợ phân biệt với tin thật, bao gồm cấu trúc câu, từ vựng, cảm xúc và ngữ nghĩa ngầm ẩn.
- **Xây dựng và huấn luyện mô hình phân loại văn bản** dựa trên các kiến trúc Transformer tiên tiến, tiêu biểu như BERT, XLNet hay RoBERTa.
- **Đánh giá hiệu quả của mô hình trên tập dữ liệu thực tế,** tập trung vào độ chính xác, độ nhạy (recall), độ đặc hiệu (precision) và khả năng tổng quát hóa đối với các tin giả có ngôn ngữ tinh vi, khó phát hiện.

- **Phân tích các giới hạn và rủi ro của mô hình,** bao gồm việc phát hiện sai lệch do thiên kiến dữ liệu, hiểu sai ngữ cảnh văn hóa hoặc việc phân loại nhầm các thể loại nội dung đặc thù như châm biếm hay bình luận xã hội.
- **Đề xuất hướng cải tiến mô hình**, hướng đến việc xây dựng hệ thống phát hiện tin giả có khả năng thích nghi với ngôn ngữ thay đổi liên tục và đảm bảo tính công bằng trong xử lý thông tin.

PHẦN 2: BỘ DỮ LIỆU

I. Giới thiệu chung

Bộ dữ liệu gồm 2 datasets nhỏ, bao gồm:

- *MisinfoSuperset_TRUE.csv:* Bộ dữ liệu là tổng hợp của những bài báo chính thống đã qua kiểm duyệt và được đăng tải trên các tờ báo đáng tin cậy như *Reuters, The New York Times, The Washington Post,...*
- *MisinfoSuperset_FAKE.csv:* Bao gồm những đoạn tin giả tinh vi được đăng trên các nguồn như ***American right-wing extremist websites*** *(e.g., Redflag Newsdesk, Breitbart, Truth Broadcast Network), Ahmed, H., Traore, I., & Saad, S. (2017): "Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques" (Springer LNCS 10618).*

II. Thông tin dữ liệu:

- Dữ liệu TRUE bao gồm 34975 bản ghi, với mỗi bản ghi có 2 thuộc tính là '*Unnamed: 0'* dùng để đánh thứ tự và '*text'* là nội dung của bài báo.
- Dữ liệu FAKE bao gồm 43642 bản ghi, với mỗi bản ghi có 2 thuộc tính là '*Unnamed: 0'* dùng để đánh thứ tự và '*text'* là nội dung của bài báo.

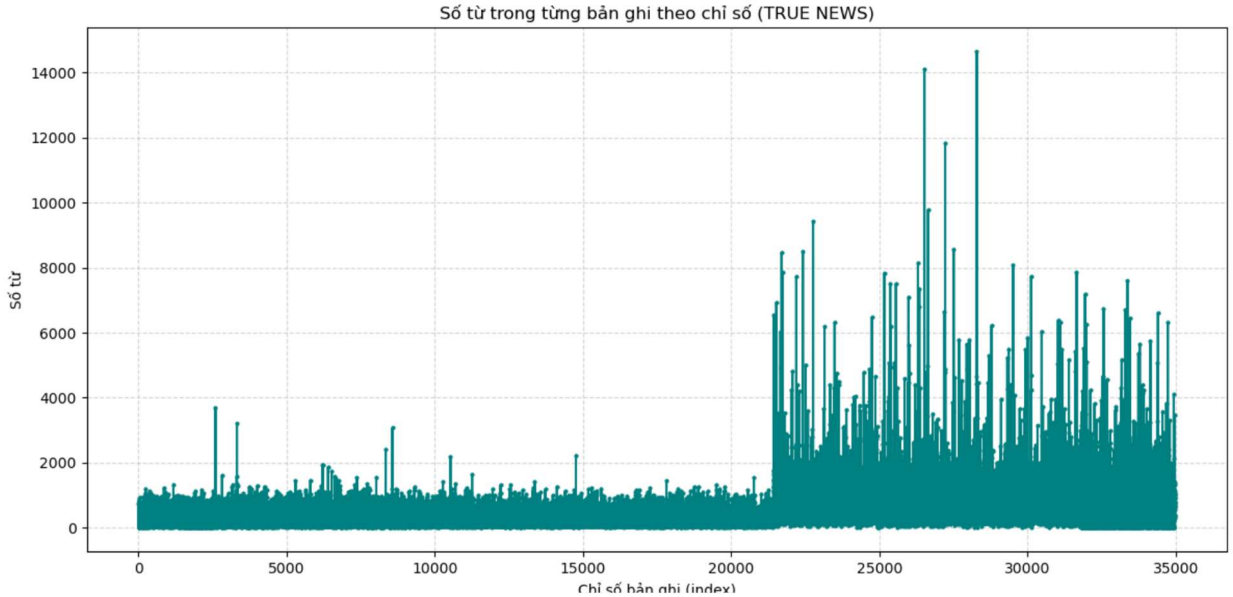
PHẦN 3: PHÂN TÍCH VÀ TIỀN XỬ LÝ DỮ LIỆU

I. Nhập dữ liệu:

- Dùng các hàm của thư viện Pandas để có thể truyền dữ liệu đã tải xuống ở máy: *read_csv(), read_table(),...*
- Gán nhãn "1" cho dữ liệu nhập từ File "*MisinfoSuperset_TRUE.csv"* và "0" cho dữ liệu từ File còn lại.
- Nối 2 DataFrame bằng lệnh *concat()*.

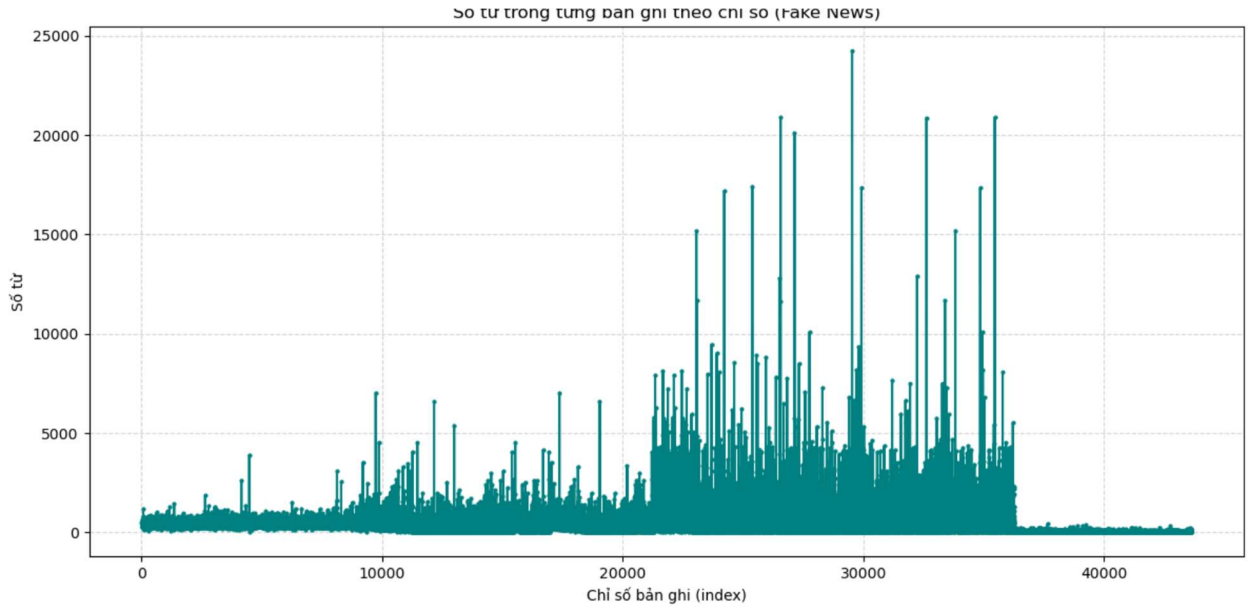
II. Khám phá và phân tích dữ liệu:

- Kiểm tra giá trị Null: có 29 giá trị Null
- Kiểm tra giá trị trùng lặp: 10 118 dữ liệu trùng lặp.
  ⇨ Cần phải xử lý các điểm dữ liệu này.
- Trực quan hóa biểu đồ số từ của mỗi điểm dữ liệu trên tập TRUE:



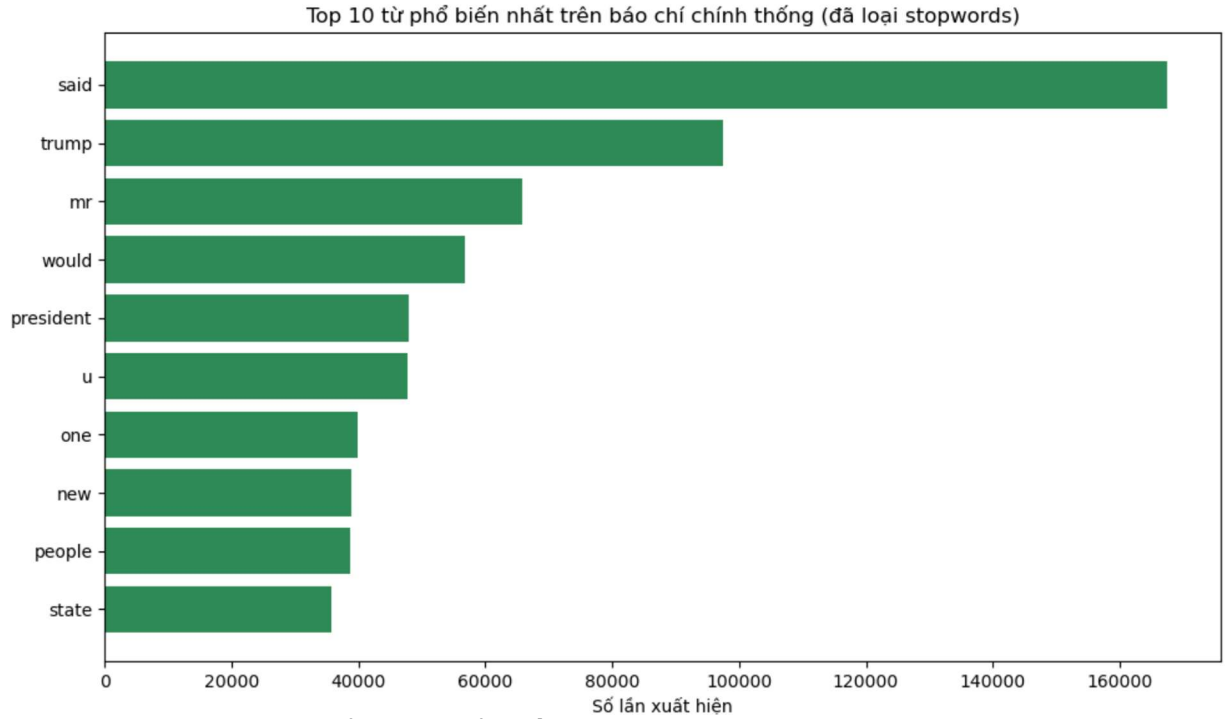
❖ **Nhận xét**: Mỗi bản ghi có ít nhất gần 1000 từ và nhiều nhất hơn 14 000 từ.

- Trực quan hóa biểu đồ số từ của mỗi điểm dữ liệu trên tập FAKE:



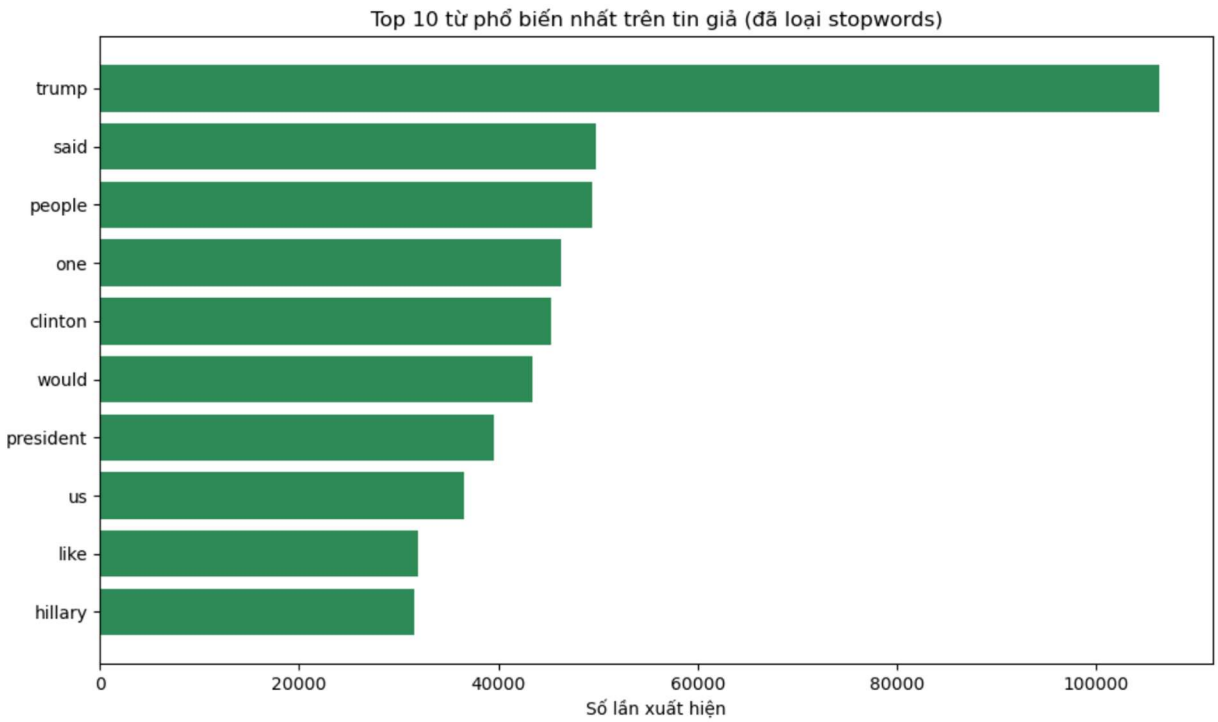
❖ **Nhận xét:** Số từ ít nhất của tập dữ liệu ít hơn và số từ tối đa lên đến ~ 25 000 từ.

- Trực quan hóa Top 10 từ phổ biến nhất trên tập TRUE:



❖ **Nhận xét:** Các từ nổi bật phổ biến là '*said*', '*trump*', '*mr*',...

- Trực quan hóa Top 10 từ phổ biến nhất trên tập FAKE:



❖ **Nhận xét:** Khác biệt khi 'trump' là từ xuất hiện nhiều nhất trên tập FAKE. '*like*' là từ xuất hiện trên top 10 của FAKE Data nhưng không xuất hiện trong TRUE Data.

III. Tiền xử lý dữ liệu:

- Tạo một hàm *clean_text()* để xóa các ký tự đặc biệt (\t, \n, \r,...).
- Loại bỏ các giá trị Null.
- Loại bỏ các giá trị trùng lặp.
  - ❖ Sau khi xử lý, dữ liệu còn lại 68 499 bản ghi và bao gồm 2 thuộc tính (Đoạn tin: 'Text' và Tính thật giả: 'target').

IV. Chia dữ liệu:

- Chia dữ liệu thành X (Các đoạn tin) và y (Mục tiêu).
- Chia dữ liệu thành Train và Test (Với Train Size 80%).
  - ❖ Tập Train có 54 799 dữ liệu và Test 13 700 dữ liệu.

PHẦN 4: HUẤN LUYỆN MÔ HÌNH

I. Các mô hình Học máy truyền thống:

1. TF-IDF Vectorizer:

- Trước khi sử dụng các mô hình học máy truyền thống, ta phải tìm được phương pháp có thể Vecto hóa các chuỗi từ trong dữ liệu bởi vì các mô hình học máy chỉ nhận vào là các ma trận số, ở đây là TF-IDF.
    - max_feature = 5000 (Ta chỉ giữ lại 5000 từ có độ phổ biến cao nhằm loại bỏ các từ không quan trọng và hiếm gặp).
    - n_gram = (1, 2) (Sử dụng cả Unigram Bigram để nắm bắt từ đơn lẻ hay theo cặp).
    - Với stopwords, ta hoàn toàn có thể loại bỏ các từ thông dụng vô giá trị như mạo từ, chỉ từ,... trong tiếng anh.

2. Logistic Regression:

- Ta sử dụng Logistic Regression của thư viện ScikitLearn kết hợp gridSearch để tìm ra bộ params có thể tối ưu hóa mô hình:
    - Các bộ parameters : C, Penalty, solver
    - Số lần lặp tối đa (max_iter) = 1000.
    - Điểm số: F1 Score.
- Kết quả:
    - Accuracy: 0.94.
    - F1 Score: 0.9457.
    - Chỉ có 364 + 391 = 755 nhãn cho là dự đoán chưa chính xác.

3. SVM:

- Tương tự, ta sử dụng mô hình linear SVM kết hợp gridSearch để tìm ra bộ Parameters đạt độ hiệu quả cao:
    - Parameters: C.
    - Số lần lặp tối đa: 1000.
    - Điểm số: F1 Score.
- Kết quả:
    - Accuracy: 0.94
    - F1 Score: 0.9452.
    - Có 364 + 398 = 762 nhãn dự đoán chưa chính xác.

II. Các mô hình Transformer

Với mỗi mô hình được sử dụng, ta đều sẽ thực hiện tuần tự các bước:

- Tokenizer.
- Data Loader.
- Model Building.
- Tuning Parameters.
- Model Training.
- Evaluate Performance.
- Inference.

Các lớp được định nghĩa:

- TextClassificationDataset và DataLoaderBuilder: Chuẩn hóa và thực hiện Tokenize cho các dữ liệu đầu vào.
- Trainer: bao gồm hàm train() để huấn luyện mô hình, evaluate() để đánh giá mô hình và predict() để dự đoán cho đầu vào mới.

1. BERT Model:

- Tokenizer: Dùng BertTokenizer với '*bert-base-uncased'*.
- Data Loader: Tạo Train và Test Loader với BertTokenizer.
- Model Bulding: Tạo mô hình BertForSequenceClassification với 'bert-base-uncased' và số lượng nhãn là 2.
- Tuning Hyperparameters: Điều chỉnh *learning_rate, max_length,...*
- Training: Train mô hình với 3 Epochs.
    - Thời gian trung bình cho 1 Epoch là 15p.
    - Train Lost giảm dần qua từng Epoch.
    - Test Lost giảm dần từ Epoch 1 đến 2, nhưng đến Epoch 3 thì tăng nhẹ.
    - Test Accuracy tăng dần qua từng Epochs.
- Performance Evaluation:
    - Mô hình dự dự đoán tốt trên tập Test.
    - Accuracy: ~ 0.99.
    - F1 Score: ~ 0.99.
    - Mô hình chỉ dự đoán sai 77 + 91 = 168 nhãn trên tập Test.
- Inference: Với dữ liệu được khởi tạo ngẫu nhiên bằng AI, mô hình chưa thể phán đoán. Nhưng nếu dữ liệu được tạo tương tự hoặc đã tồn tại trong Dataset, mô hình sẽ dự đoán đúng.

2. XLNet Model:

- Tokenizer: Dùng XLNetTokenizer với '*xlnet-base-cased'*.

- Data Loader: Tạo Train và Test Loader với **BertTokenizer**.
- Model Bulding: Tạo mô hình XLNetForSequenceClassification với 'xlnet-base-cased' và số lượng nhãn là 2.
- Tuning Hyperparameters: Điều chỉnh *learning_rate, max_length,...*
- Training: Train mô hình với 3 Epochs.
    - Thời gian trung bình cho 1 Epoch là 15p.
    - Train Lost giảm dần qua từng Epoch.
    - Test Lost dao động qua từng Epoch.
    - Test Accuracy dao động từ Epoch 1 đến 2 nhưng tăng ở Epoch 3.
- Performance Evaluation:
    - Mô hình dự dự đoán tốt trên tập Test.
    - Accuracy: ~ 0.99.
    - F1 Score: ~ 0.99.
    - Mô hình chỉ dự đoán sai 82 + 91 = 173 nhãn trên tập Test.
- Inference: Với dữ liệu được khởi tạo ngẫu nhiên bằng AI, mô hình chưa thể phán đoán chính xác. Nhưng nếu dữ liệu được tạo tương tự hoặc đã tồn tại trong Dataset, mô hình sẽ dự đoán đúng.

3. roBERTa Model:

- Tokenizer: Dùng **RobertaTokenizer** với '*roberta-base'*.
- Data Loader: Tạo Train và Test Loader với **RobertaTokenizer**.
- Model Bulding: Tạo mô hình RobertaForSequenceClassification với 'roberta-base' và số lượng nhãn là 2.
- Tuning Hyperparameters: Điều chỉnh *learning_rate, max_length,...*
- Training: Train mô hình với 3 Epochs.
    - Thời gian trung bình cho 1 Epoch là 15p.
    - Train Lost giảm dần qua từng Epoch.
    - Test Lost giữ ở mức 0.03 – 0,04.
    - Test Accuracy: Tăng dần và đạt 0.9899 ở Epoch 3.
- Performance Evaluation:
    - Mô hình dự dự đoán tốt trên tập Test.
    - Accuracy: ~ 0.99.
    - F1 Score: ~ 0.99.
    - Mô hình chỉ dự đoán sai 70 + 91 = 161 nhãn trên tập Test.

- Inference: Với dữ liệu được khởi tạo ngẫu nhiên bằng AI, mô hình chưa thể phán đoán chính xác. Nhưng nếu dữ liệu được tạo tương tự hoặc đã tồn tại trong Dataset, mô hình sẽ dự đoán đúng.

PHẦN 5: KẾT QUẢ ĐẠT ĐƯỢC

Xây dựng được các mô hình có thể phân biệt được tin giả với độ chính xác cao, cụ thể:

- Logistic Regression đặt Accuracy 0.945, F1 Score 0.94.
- SVM đạt Accuracy 0.94, F1 Score 0.94.
- BERT, XLNet và roBERTa đều cho ra kết quả 0.99 Accuracy và F1 Score.

PHẦN 6: KẾT LUẬN

Nhìn chung, các mô hình cho dù là truyền thống hay các mô hình có khả năng học sâu đều có thể huấn luyên và dự đoán tốt cho bộ dữ liệu, góp phần giúp con người có thể nhận biết và phòng chống nạn tin giả được tốt hơn. Các mô hình như BERT, XLNet hay roBERTa đều có thể hoạt động gần như tuyệt đối trên bộ dữ liệu đã cho, đặc biệt đối với ngôn ngữ tiếng Anh do đây là các mô hình đã được huấn luyện trước trên tập dữ liệu tiếng Anh.

Tuy nhiên vẫn còn một số khó khăn và thách thức:

- Tốn kém về thời gian, bộ nhớ cũng như tài nguyên sử dụng: Nhờ các nền tảng học thuật như Colab, Kaggle,... cung cấp môi trường sử dụng GPU để có thể huấn luyện mô hình, tuy nhiên vẫn có giới hạn thời gian. Nếu không, phải mất nhiều chi phí để phục vụ cho việc nghiên cứu.
- Bộ dữ liệu dùng để huấn luyện có thể đã cũ, nên khi thử dùng mô hình để dự đoán cho những thông tin thực tế bất kỳ, đôi khi mô hình lại hoạt động chưa chính xác do các thông tin cũ có thể đã sai hoặc không còn nữa.
- Tập dữ liệu chỉ gói gọn trong một vài lĩnh vực nhất định (Chính trị, Kinh tế, Tin tức,...) hoặc chỉ do một tờ báo biên soạn có thể dẫn đến việc văn phong giống nhau.
- Tập dữ liệu là ngôn ngữ tiếng Anh (Là thế mạnh của các mô hình ngôn ngữ đã dùng ở trên), nếu ta thay thế và sử dụng tập dữ liệu Tiếng Việt, hiệu quả có thể không bằng.

Đề xuất hướng giải quyết và phát triển trong tương lai:

- Mở rộng tập dữ liệu (Sử dụng nhiều nguồn tin từ các tờ báo khác nhau để nắm được văn phong khái quát của những tờ tin chính thống) và cập nhật dữ liệu thường xuyên.
- Xây dựng và phát triển dữ liệu đa ngôn ngữ.
- Có thể sử dụng thêm các mô hình ngôn ngữ khác như phoBERT,...
- Điều chỉnh các Hyperparameters cho phù hợp.