TRƯỜNG ĐẠI HỌC NGOẠI NGỮ - TIN HỌC THÀNH PHỐ HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN

A yellow oval with red text

AI-generated content may be incorrect.

**KHÓA LUẬN TỐT NGHIỆP**

**KHAI THÁC MÔ HÌNH NGÔN NGỮ LỚN CHO PHÁT HIỆN TIN GIẢ**

GIẢNG VIÊN HƯỚNG DẪN: ThS. Tiếu Phùng Mai Sương

SINH VIÊN THỰC HIỆN: 1. Nguyễn Ngọc Thanh Nhã – 22DH114656

2. Trần Phúc Bình – 22DH110370

|  |  |
| --- | --- |
|  |  |

**TP. HỒ CHÍ MINH - THÁNG 08 - NĂM 2025**

TRƯỜNG ĐẠI HỌC NGOẠI NGỮ - TIN HỌC THÀNH PHỐ HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN

**KHÓA LUẬN TỐT NGHIỆP**

**KHAI THÁC MÔ HÌNH NGÔN NGỮ LỚN CHO PHÁT HIỆN TIN GIẢ**

GIẢNG VIÊN HƯỚNG DẪN: ThS. Tiếu Phùng Mai Sương

SINH VIÊN THỰC HIỆN: 1. Nguyễn Ngọc Thanh Nhã – 22DH114656

2. Trần Phúc Bình – 22DH110370

|  |  |
| --- | --- |
|  |  |

**TP. HỒ CHÍ MINH - THÁNG 08 - NĂM 2025**

LỜI CẢM ƠN

Lời đầu tiên, chúng em xin gửi lời cảm ơn chân thành và sâu sắc đến cô Tiếu Phùng Mai Sương, người đã tận tình hướng dẫn, góp ý và đồng hành cùng chúng em trong suốt quá trình thực hiện khóa luận tốt nghiệp. Nhờ sự hỗ trợ quý báu từ cô, chúng em đã có cơ hội tiếp cận và triển khai hiệu quả các kiến thức chuyên ngành vào nghiên cứu thực tiễn.

Chúng em cũng xin bày tỏ lòng biết ơn đến quý thầy cô trong Khoa Công nghệ thông tin Trường đại học Ngoại Ngữ - Tin Học TP. Hồ Chí Minh đã giảng dạy và tạo điều kiện thuận lợi trong suốt thời gian học tập tại trường. Những kiến thức và kỹ năng mà quý thầy cô truyền đạt là nền tảng vững chắc để chúng em có thể hoàn thành khóa luận này.

Dù đã nỗ lực hết mình, nhưng với giới hạn về thời gian và kinh nghiệm, khóa luận không thể tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự góp ý từ quý thầy cô để hoàn thiện hơn trong những nghiên cứu sau này.

Chúng em xin trân trọng cảm ơn!

LỜI CAM ĐOAN

Tôi xin cam đoan rằng đề tài “Khai thác mô hình ngôn ngữ lớn cho phát hiện tin giả” là công trình nghiên cứu của riêng tôi, được thực hiện dưới sự hướng dẫn của “Ths. Tiếu Phùng Mai Sương”

Các số liệu, kết quả nêu trong khóa luận là trung thực, được trích dẫn rõ ràng từ các nguồn tài liệu có uy tín. Tôi hoàn toàn chịu trách nhiệm về tính trung thực và tính chính xác của toàn bộ nội dung khóa luận này.

Tôi cam kết rằng khóa luận này chưa từng được sử dụng để bảo vệ cho bất kỳ chương trình đào tạo nào khác.

TPHCM, Ngày …… tháng …… năm 2025

**Sinh viên thực hiện**

(Ký và ghi rõ họ tên)

LỜI MỞ ĐẦU

Trong kỷ nguyên số hiện nay, thông tin được lan truyền với tốc độ chưa từng có, đặc biệt là trên các nền tảng mạng xã hội và truyền thông trực tuyến. Bên cạnh những lợi ích to lớn về tiếp cận tri thức và kết nối cộng đồng, sự phát triển này cũng kéo theo nhiều hệ lụy, trong đó nổi bật là sự gia tăng nhanh chóng và tinh vi của tin giả. Tin giả không chỉ làm sai lệch nhận thức, gây hoang mang dư luận, mà còn đe dọa đến an ninh xã hội, chính trị và kinh tế.

Trước thực trạng đó, nhu cầu nghiên cứu và phát triển các công cụ phát hiện tin giả hiệu quả là vô cùng cấp thiết. Trong bối cảnh trí tuệ nhân tạo phát triển mạnh mẽ, các mô hình ngôn ngữ lớn như BERT, RoBERTa, GPT,… đã và đang chứng minh được tiềm năng to lớn trong việc hiểu và xử lý ngôn ngữ tự nhiên. Việc ứng dụng LLM trong nhiệm vụ phát hiện tin giả là một hướng tiếp cận hiện đại, hứa hẹn mang lại hiệu quả cao nhờ khả năng phân tích ngữ nghĩa sâu và thích nghi linh hoạt.

Xuất phát từ nhận thức đó, chúng em lựa chọn đề tài “Khai thác mô hình ngôn ngữ lớn cho phát hiện tin giả” với mục tiêu nghiên cứu và ứng dụng một số kỹ thuật khai thác LLM – đặc biệt là phương pháp prompt-based learning – để xây dựng giải pháp phát hiện tin giả trong văn bản. Đề tài được triển khai theo hướng thực tiễn, phù hợp với điều kiện tài nguyên hiện có và bám sát các xu hướng công nghệ mới.

Khóa luận gồm các nội dung chính: trình bày cơ sở lý thuyết về bài toán phân lớp và các mô hình ngôn ngữ lớn; mô tả phương pháp đề xuất và quy trình thực nghiệm; đánh giá hiệu quả mô hình và so sánh với các phương pháp truyền thống; từ đó rút ra nhận xét, đề xuất hướng cải tiến và ứng dụng thực tế trong tương lai.

Chúng em hy vọng rằng khóa luận này sẽ đóng góp một phần nhỏ vào việc nâng cao hiểu biết và ứng dụng công nghệ AI trong công tác nhận diện và phòng chống tin giả – một vấn đề đang được quan tâm rộng rãi hiện nay.

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

TPHCM, Ngày …… tháng …… năm 2025

**Giáo viên hướng dẫn**

(Ký tên, đóng dấu)

Ths. Tiếu Phùng Mai Sương

**MỤC LỤC**

[DANH MỤC HÌNH ii](#_Toc205216203)

[DANH MỤC BẢNG iii](#_Toc205216204)

[DANH MỤC TỪ VIẾT TẮT iv](#_Toc205216205)

[CHƯƠNG 1: GIỚI THIỆU 1](#_Toc205216206)

[1.1. Đặt vấn đề 1](#_Toc205216207)

[1.2. Mục tiêu khoá luận 1](#_Toc205216208)

[1.3. Phạm vi khoá luận 2](#_Toc205216209)

[1.4. Cấu trúc khóa luận 3](#_Toc205216210)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 4](#_Toc205216211)

[2.1. Tổng quan về bài toán phân lớp 4](#_Toc205216212)

[2.2. Các phương pháp truyền thống 5](#_Toc205216213)

[2.3. Giới thiệu về LLMs 7](#_Toc205216214)

[2.4. Các kỹ thuật tinh chỉnh LLM 9](#_Toc205216215)

[2.5. Các chỉ số đánh giá mô hình phân loại 10](#_Toc205216216)

[CHƯƠNG 3: PHƯƠNG PHÁP ĐỀ XUẤT 13](#_Toc205216217)

[3.1. Phát hiện tin giả bằng mô hình Vietnamese-BartPho kết hợp BiLSTM 13](#_Toc205216218)

[3.2. Phát hiện tin giả bằng mô hình tokenizer PhoBERT kết hợp BiLSTM 15](#_Toc205216219)

[3.3. Mô hình đề xuất phát hiện tin giả bằng Vietnamese-SBERT kết hợp BiLSTM 17](#_Toc205216220)

[3.4. Tiền xử lý dữ liệu 19](#_Toc205216221)

[3.3. Chiến lược huấn luyện 22](#_Toc205216222)

[CHƯƠNG 4: THỰC NGHIỆM VÀ KẾT QUẢ 24](#_Toc205216223)

[4.1. Mô tả tập dữ liệu 24](#_Toc205216224)

[4.2. Môi trường thực nghiệm 24](#_Toc205216225)

[4.3. So sánh kết quả 26](#_Toc205216226)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 31](#_Toc205216227)

[5.1. Tóm tắt kết quả đạt được 31](#_Toc205216228)

[5.2. Đóng góp của đề tài 32](#_Toc205216229)

[5.3. Hạn chế và đề xuất cải tiến 33](#_Toc205216230)

[TÀI LIỆU THAM KHẢO 35](#_Toc205216231)

# D**ANH MỤC HÌNH**

[Hình 2‑1: Sơ đồ thể hiện bài toán phân lớp 4](#_Toc205216488)

[Hình 3‑1:Sơ đồ kiến trúc mô hình phát hiện tin giả sử dụng tokenizer BartPho và BiLSTM 15](#_Toc205216489)

[Hình 3‑2:Sơ đồ kiến trúc mô hình phát hiện tin giả sử dụng tokenizer PhoBERT và BiLSTM 17](#_Toc205216490)

[Hình 3‑3:Sơ đồ kiến trúc mô hình phát hiện tin giả sử dụng Vietnamese-SBERT và BiLSTM 19](#_Toc205216491)

[Hình 4‑1: Ma trận nhâm lẫn BiLSTM kết hợp với Vietnamese-SBERT 29](#_Toc205216492)

[Hình 4‑2:Ma trận nhầm lẫn BiLSTM kết hợp với BARTpho 29](#_Toc205216493)

[Hình 4‑3:Ma trận nhầm lẫn BiLSTM kết hợp với PhoBERT 30](#_Toc205216494)

# D**ANH MỤC BẢNG**

[Bảng 1‑1: Danh mục từ viết tắt iv](#_Toc205216517)

[Bảng 2‑1: Cấu trúc ma trận nhầm lẫn 11](#_Toc205216518)

[Bảng 3‑1:Thống kê tập tin dữ liệu 20](#_Toc205216519)

[Bảng 3‑2:Thống kê các tập dữ liệu 21](#_Toc205216520)

[Bảng 4‑1: Bảng so sánh hiệu quả giữa các mô hình phát hiện tin giả 27](#_Toc205216521)

# DANH MỤC TỪ VIẾT TẮT

Bảng 1‑1: Danh mục từ viết tắt

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Từ viết tắt** | **Từ đầy đủ** | **Nghĩa tiếng Việt** |
| 1 | LLM | Large Language Models | Ngôn ngữ lớn |
| 2 | AI | Artificial Intelligence | Trí tuệ nhân tạo |
| 3 | NLP | Natural Language Processing | Xử lý ngôn ngữ tự nhiên |
| 4 | BERT | Bidirectional Encoder Representations from Transformers | Biểu diễn mã hóa hai chiều từ Transformer |
| 5 | GPT | Generative Pre-trained Transformer | Mô hình Transformer tiền huấn luyện sinh văn bản |
| 6 | RoBERTa | Robustly Optimized BERT Pretraining Approach | Phương pháp tối ưu hóa huấn luyện BERT mạnh mẽ hơn |
| 7 | PhoBERT | Vietnamese BERT by VNLab (FPT) | Mô hình BERT cho tiếng Việt |
| 8 | viBERT | Vietnamese BERT | BERT phiên bản tiếng Việt |
| 9 | viT5 | Vietnamese Text-To-Text Transfer Transformer | Mô hình chuyển đổi văn bản tiếng Việt |
| 10 | SBERT | Sentence-BERT | Mô hình BERT biểu diễn câu |
| 11 | BiLSTM | Bidirectional Long Short-Term Memory | Mạng ghi nhớ dài-ngắn hai chiều |
| 12 | API | Application Programming Interface | Giao diện lập trình ứng dụng |
| 13 | GPU | Graphics Processing Unit | Bộ xử lý đồ họa (dùng trong tính toán song song) |
| 14 | PEFT | Parameter-Efficient Fine-Tuning | Tinh chỉnh hiệu quả tham số |
| 15 | LoRA | Low-Rank Adaptation | Điều chỉnh hạng thấp |
| 16 | Prompting | (Không phải từ viết tắt, nhưng là thuật ngữ chính) | Phương pháp gợi ý (đầu vào điều khiển mô hình) |
| 17 | Adapter Tuning | (Không viết tắt, nhưng là kỹ thuật PEFT) | Điều chỉnh thông qua các module phụ |
| 18 | Prefix Tuning | (Không viết tắt, nhưng là kỹ thuật PEFT) | Tinh chỉnh tiền tố (tối ưu hóa phần đầu vào) |
| 19 | Knowledge Distillation | — | Chưng cất tri thức (chuyển tri thức từ mô hình lớn sang nhỏ) |
| 20 | TinyBERT | Tiny Bidirectional Encoder Representations from Transformers | Phiên bản BERT nhỏ gọn |
| 21 | DistilBERT | Distilled BERT | BERT đã chưng cất tri thức |
| 22 | MiniLM | Mini Language Model | Mô hình ngôn ngữ cỡ nhỏ |
| 23 | Zero-shot | — | Không cần ví dụ huấn luyện (zero-shot learning) |
| 24 | One-shot | — | Học từ một ví dụ duy nhất (one-shot learning) |
| 25 | Few-shot | — | Học từ một vài ví dụ (few-shot learning) |
| 26 | Chain-of-thought | — | Gợi ý theo chuỗi suy nghĩ |
| 27 | TPU | Tensor Processing Unit | Bộ xử lý chuyên dụng cho TensorFlow |
| 28 | CSV | Comma-Separated Values | Định dạng tệp văn bản phân cách bằng dấu phẩy |
| 29 | LSTM | Long Short-Term Memory | Mạng ghi nhớ dài-ngắn (một loại mạng nơ-ron tuần tự) |
| 30 | BiLSTM | Bidirectional Long Short-Term Memory | Mạng LSTM hai chiều |
| 31 | Sigmoid | Sigmoid Function | Hàm sigmoid (hàm kích hoạt đầu ra giá trị 0–1) |
| 32 | BCE | Binary Cross Entropy | Hàm mất mát nhị phân chéo (Binary Cross Entropy) |
| 33 | Adam | Adaptive Moment Estimation | Thuật toán tối ưu Adam |
| 34 | GPU | Graphics Processing Unit | Bộ xử lý đồ họa (dùng cho huấn luyện mô hình AI) |
| 35 | Epoch | — | Một vòng huấn luyện toàn bộ dữ liệu |
| 36 | Accuracy | — | Độ chính xác tổng thể |
| 37 | Precision | — | Độ chính xác dương tính |
| 38 | Recall | — | Khả năng truy hồi |
| 39 | F1-score | — | Điểm trung bình điều hòa giữa precision và recall |
| 40 | Confusion Matrix | — | Ma trận nhầm lẫn |
| 41 | DataLoader | — | Trình tải dữ liệu trong PyTorch |
| 42 | Tokenizer | — | Bộ tách từ thành token |
| 43 | Embedding | — | Lớp biểu diễn nhúng (vector hóa từ hoặc câu) |
| 44 | Mini-batch | — | Lô dữ liệu nhỏ trong huấn luyện |
| 45 | Learning rate | — | Tốc độ học (mức độ điều chỉnh trọng số trong huấn luyện) |
| 46 | Padding | — | Thêm phần tử giả để đồng bộ độ dài chuỗi |
| 47 | Pack Padded Sequence | — | Kỹ thuật bỏ qua padding trong xử lý tuần tự |
| 48 | Epoch | — | Một vòng lặp toàn bộ dữ liệu huấn luyện |

# **GIỚI THIỆU**

* 1. **Đặt vấn đề**

Trong thời đại kỷ nguyên số bùng nổ, thông tin được tạo ra và lan truyền với tốc độ chưa từng có. Mặc dù mang lại vô số lợi ích, sự phát triển này cũng đồng thời kéo theo một thách thức nghiêm trọng: sự gia tăng chóng mặt của tin giả và thông tin sai lệch. Tin giả không chỉ gây hoang mang dư luận, ảnh hưởng đến niềm tin xã hội, mà còn có thể tác động tiêu cực đến kinh tế, chính trị, và thậm chí là sức khỏe cộng đồng. Việc nhận diện và ngăn chặn sự lan truyền của tin giả đã trở thành một nhiệm vụ cấp bách và phức tạp, đòi hỏi các giải pháp công nghệ tiên tiến. Các phương pháp phát hiện tin giả truyền thống, dựa trên phân tích đặc điểm ngôn ngữ bề mặt hoặc dữ liệu có nhãn hạn chế, thường gặp khó khăn trước sự tinh vi và khả năng thích nghi liên tục của tin giả. Đặc biệt, tin giả ngày nay không chỉ xuất hiện dưới dạng văn bản mà còn được thể hiện qua các định dạng đa phương tiện như hình ảnh, video, và âm thanh, làm tăng thêm mức độ phức tạp của bài toán. Hơn nữa, những người tạo tin giả liên tục đổi mới chiến thuật, khiến các mô hình phát hiện nhanh chóng trở nên lỗi thời nếu không có khả năng học hỏi và thích nghi liên tục. Tuy nhiên những năm gần đây, với sự ra đời và phát triển vượt bậc của LLM như GPT-3, GPT-4, Bert, LLaMA... đã mở ra một kỷ nguyên mới cho Trí tuệ Nhân tạo và xử lý ngôn ngữ tự nhiên. Với khả năng hiểu sâu sắc ngữ nghĩa, cú pháp, ngữ cảnh ngôn ngữ, cùng khối lượng kiến thức bách khoa khổng lồ được hấp thụ từ quá trình huấn luyện trên dữ liệu Internet, Sách, Báo... quy mô lớn, LLM hứa hẹn mang lại những đột phá đáng kể trong nhiều lĩnh vực nói chung, và việc phát hiện tin giả nói riêng. Nhưng bên cạnh, đó việc khai thác LLM cho nhiệm vụ phát hiện tin giả cũng đối mặt với nhiều thách thức riêng. Do đó, nghiên cứu này sẽ tập trung vào việc khám phá tiềm năng và thách thức của việc sử dụng Mô hình Ngôn ngữ lớn trong việc phát hiện tin giả

## **Mục tiêu khoá luận**

Khóa luận hướng đến mục tiêu khai thác tiềm năng của các LLM nhằm giải quyết bài toán phát hiện tin giả trong văn bản, đặc biệt trong bối cảnh thông tin sai lệch ngày càng lan rộng trên các nền tảng trực tuyến. Ở giai đoạn đầu, khóa luận tập trung vào việc tìm hiểu tổng quan về hiện tượng tin giả, bao gồm định nghĩa, đặc điểm, tác động xã hội cũng như các phương pháp truyền thống và hiện đại trong việc phát hiện tin giả. Đồng thời, khóa luận sẽ nghiên cứu cơ sở lý thuyết về mô hình ngôn ngữ lớn, kiến trúc hoạt động, và khả năng ứng dụng trong các tác vụ phân loại văn bản. Sau đó, ta sẽ tiến hành lựa chọn một hoặc một số mô hình LLM phù hợp (ví dụ: BERT, RoBERTa, GPT) hoặc các mô hình mã nguồn mở tương đương), kết hợp với các chiến lược triển khai như fine-tuning hoặc prompt-based learning để xây dựng hệ thống phát hiện tin giả. Tập dữ liệu sẽ được chuẩn bị và xử lý nhằm phục vụ cho việc huấn luyện và kiểm thử mô hình. Cuối cùng, khóa luận sẽ thực hiện đánh giá hiệu quả mô hình dựa trên các chỉ số như Accuracy, Precision, Recall và F1-score. Các kết quả thu được sẽ được phân tích để rút ra nhận xét về điểm mạnh, hạn chế của mô hình, từ đó đề xuất các hướng cải tiến và khả năng ứng dụng thực tiễn trong việc hỗ trợ kiểm chứng nội dung và nâng cao nhận thức cộng đồng về tin giả.

* 1. **Phạm vi khoá luận**

Khóa luận tập trung nghiên cứu và ứng dụng các LLM trong việc phát hiện tin giả dưới dạng văn bản, cụ thể là nội dung tiếng Việt và/hoặc tiếng Anh từ các nguồn tin tức trực tuyến. Trong phạm vi này, tác giả sẽ chủ yếu khai thác một số mô hình ngôn ngữ phổ biến như BERT, RoBERTa hoặc GPT thông qua hai hướng tiếp cận chính: tinh chỉnh mô hình và sử dụng prompt.

Về dữ liệu, khóa luận sẽ sử dụng các tập văn bản đã được gán nhãn (tin thật, tin giả), được thu thập từ các kho dữ liệu công khai hoặc các trang báo điện tử. Việc tiền xử lý dữ liệu, trích xuất đặc trưng và huấn luyện mô hình sẽ được thực hiện với quy mô vừa phải, phù hợp với điều kiện kỹ thuật hiện có.

Khóa luận sẽ **không đi sâu** vào việc xây dựng mô hình ngôn ngữ mới từ đầu hoặc xử lý các dạng tin giả phi văn bản như hình ảnh, video, giọng nói. Ngoài ra, các yếu tố pháp lý, xã hội hay tâm lý liên quan đến tin giả cũng không nằm trong phạm vi nghiên cứu.Thông qua việc giới hạn phạm vi như trên, khóa luận kỳ vọng đưa ra đánh giá thực nghiệm rõ ràng về khả năng ứng dụng LLM vào bài toán phát hiện tin giả văn bản, từ đó làm nền tảng cho các nghiên cứu sâu hơn trong tương lai.

* 1. **Cấu trúc khóa luận**

Khóa luận được tổ chức thành năm chương chính, cụ thể như sau:

**Chương 1: Giới thiệu**

Trình bày bối cảnh và lý do chọn đề tài, mục tiêu nghiên cứu, phạm vi thực hiện, cũng như tổng quan cấu trúc nội dung toàn khóa luận.

**Chương 2: Cơ sở lý thuyết**

Tổng hợp các kiến thức nền tảng liên quan đến bài toán phân lớp văn bản, các phương pháp học máy truyền thống như SVM, Naive Bayes, cũng như giới thiệu về mô hình ngôn ngữ lớn và các kỹ thuật khai thác LLM như fine-tuning, prompting, và PEFT.

**Chương 3: Phương pháp đề xuất**

Mô tả mô hình được lựa chọn trong đề tài (Khai Thác LLM Cho Việc Phát Hiện Tin Giả) quy trình tiền xử lý dữ liệu, chiến lược huấn luyện (fine-tuning hoặc prompting), và các chỉ số đánh giá hiệu năng như Accuracy, F1-score, Confusion Matrix, ...

**Chương 4: Thực nghiệm và kết quả**

Trình bày quá trình thực nghiệm, bao gồm mô tả tập dữ liệu sử dụng, môi trường thực thi (Google Colab, GPU, Thư Viện), và phân tích kết quả thu được. Ngoài ra, chương này cũng thực hiện so sánh giữa LLM với các mô hình học máy truyền thống hoặc giữa các biến thể LLM (nếu có).

**Chương 5: Kết luận và hướng phát triển**

Tổng kết các kết quả đạt được, nêu rõ đóng góp của đề tài, chỉ ra những hạn chế còn tồn tại và đề xuất các hướng cải tiến trong tương lai.

# **CƠ SỞ LÝ THUYẾT**

* 1. **Tổng quan về bài toán phân lớp**

Bài toán phân lớp là một trong những nhiệm vụ cốt lõi và phổ biến nhất trong lĩnh vực Học máy và Trí tuệ Nhân tạo. Mục tiêu chính của bài toán này là gán nhãn hoặc phân loại một mẫu dữ liệu đầu vào vào một hoặc nhiều lớp đã được xác định trước. Đây là một bài toán thuộc loại học có giám sát trong đó mô hình được huấn luyện trên một tập dữ liệu đã được gán nhãn – tức là mỗi mẫu dữ liệu trong tập huấn luyện đều có thông tin về lớp mà nó thuộc về. Sau quá trình học, mô hình có khả năng dự đoán lớp cho các mẫu dữ liệu mới chưa từng gặp.

Trong lĩnh vực xử lý ngôn ngữ tự nhiên NLP, bài toán phân lớp văn bản đóng vai trò quan trọng trong nhiều ứng dụng thực tiễn như phân tích cảm xúc (tích cực, tiêu cực, trung tính), phân loại chủ đề văn bản (chính trị, thể thao, công nghệ), phát hiện thư rác (spam), nhận diện tin giả, hay phát hiện và phân loại nội dung độc hại. Trước đây, các mô hình phân lớp văn bản thường yêu cầu quá trình trích xuất đặc trưng thủ công, hoặc xây dựng mô hình nhỏ chuyên biệt cho từng bài toán. Tuy nhiên, với sự ra đời và phát triển nhanh chóng của các mô hình LLM như BERT, RoBERTa, GPT và các mô hình tương tự, cách tiếp cận đối với bài toán phân lớp đã có những thay đổi mang tính cách mạng.

A diagram of a computer network

AI-generated content may be incorrect.

Hình 2‑1: Sơ đồ thể hiện bài toán phân lớp

Các mô hình LLM mang lại ba lợi ích nổi bật khi được áp dụng vào bài toán phân lớp. Thứ nhất, chúng có khả năng hiểu ngữ nghĩa và ngữ cảnh sâu sắc hơn nhờ được huấn luyện trên các tập dữ liệu văn bản khổng lồ, bao gồm nhiều miền kiến thức khác nhau. Thứ hai, LLM giúp giảm nhu cầu về dữ liệu huấn luyện đặc thù vì chúng tận dụng khả năng học chuyển tiếp, cho phép mô hình áp dụng vào các bài toán mới với ít dữ liệu huấn luyện hơn mà vẫn đạt hiệu quả cao. Thứ ba, các mô hình này dễ dàng triển khai thông qua ba chiến lược phổ biến: zero-shot (không cần huấn luyện lại, chỉ cần xây dựng prompt hợp lý), few-shot (cung cấp một vài ví dụ trong prompt để cải thiện kết quả), và fine-tuning (tinh chỉnh mô hình trên tập dữ liệu cụ thể để đạt hiệu suất tối ưu).

Có ba cách tiếp cận chính khi sử dụng LLM cho bài toán phân lớp văn bản. Cách thứ nhất là trích xuất đặc trưng, trong đó các biểu diễn ngữ nghĩa từ các lớp ẩn của mô hình như BERT hoặc RoBERTa được sử dụng làm đầu vào cho các mô hình phân loại truyền thống như SVM hoặc logistic regression. Cách thứ hai là tinh chỉnh, tức là huấn luyện lại toàn bộ hoặc một phần mô hình LLM trên tập dữ liệu đã gán nhãn của bài toán cụ thể. Trong phương pháp này, mô hình sẽ học cách tự động trích xuất đặc trưng và đưa ra dự đoán phân loại. Cách thứ ba là sử dụng prompt, trong đó người dùng xây dựng các lời nhắc để mô hình trực tiếp thực hiện phân loại mà không cần huấn luyện thêm. Với khả năng học trong ngữ cảnh (in-context learning) và vốn tri thức rộng lớn, các LLM hiện đại có thể đạt kết quả đáng kể ngay cả khi không được tinh chỉnh chuyên biệt cho bài toán.

Tóm lại, sự phát triển của các mô hình ngôn ngữ lớn đã mở ra một hướng tiếp cận mới đầy tiềm năng cho bài toán phân lớp nói chung và phân lớp văn bản nói riêng, góp phần nâng cao độ chính xác, tính linh hoạt và khả năng triển khai trong nhiều tình huống thực tế.

* 1. **Các phương pháp truyền thống**

Trước khi các mô hình LLM trở nên phổ biến và thống trị trong lĩnh vực xử lý ngôn ngữ tự nhiên, việc phát hiện tin giả chủ yếu được thực hiện bằng các phương pháp học máy truyền thống. Những phương pháp này thường dựa trên quá trình trích xuất đặc trưng thủ công từ văn bản, sau đó sử dụng các thuật toán phân loại cổ điển để đưa ra quyết định. Một trong những cách tiếp cận phổ biến là biểu diễn văn bản dưới dạng mô hình túi từ BoW hoặc sử dụng kỹ thuật TF-IDF nhằm định lượng tầm quan trọng tương đối của các từ trong văn bản. Sau khi đã có được biểu diễn đặc trưng, các mô hình như Naive Bayes, Support Vector Machine, Decision Tree, Logistic Regression, hoặc K-Nearest Neighbors sẽ được huấn luyện để phân biệt giữa tin thật và tin giả.

Một trong những thuật toán được ứng dụng rộng rãi trong giai đoạn đầu là Naive Bayes – một mô hình phân loại xác suất đơn giản dựa trên định lý Bayes, giả định rằng các đặc trưng là độc lập có điều kiện với nhãn phân loại. Mặc dù giả định độc lập này hiếm khi đúng trong thực tế, đặc biệt là trong ngữ cảnh của ngôn ngữ tự nhiên nơi các từ có mối liên hệ chặt chẽ với nhau, Naive Bayes vẫn hoạt động khá hiệu quả và mang lại kết quả đáng tin cậy trong nhiều trường hợp nhờ tính đơn giản, tốc độ xử lý nhanh và khả năng tổng quát hóa tốt trên các tập dữ liệu vừa phải.

Bên cạnh đó, SVM cũng là một lựa chọn phổ biến trong các bài toán phân lớp văn bản, bao gồm cả phát hiện tin giả. SVM hoạt động bằng cách tìm một siêu phẳng tối ưu phân tách các điểm dữ liệu thuộc hai lớp khác nhau, với mục tiêu tối đa hóa khoảng cách giữa các điểm gần siêu phẳng nhất của mỗi lớp. Nhờ khả năng xử lý tốt trong không gian đặc trưng có chiều cao và khả năng tổng quát hóa mạnh, SVM đặc biệt phù hợp với các bài toán có số lượng mẫu vừa phải nhưng có số chiều lớn – điều thường gặp trong xử lý văn bản khi sử dụng BoW hoặc TF-IDF. Ngoài ra, việc sử dụng các kernel phi tuyến cho phép SVM mở rộng khả năng phân loại cho cả các dữ liệu có ranh giới phi tuyến.

Các thuật toán khác như Decision Tree, Random Forest, Logistic Regression hay K-Nearest Neighbors cũng được áp dụng trong phát hiện tin giả, mỗi mô hình đều có những điểm mạnh riêng. Decision Tree sử dụng cấu trúc cây để đưa ra quyết định dựa trên các giá trị đặc trưng, tuy nhiên mô hình này dễ bị overfitting nếu không được cắt tỉa hợp lý. Random Forest là một tập hợp của nhiều cây quyết định hoạt động song song, nhờ đó tăng cường độ chính xác và giảm thiểu nguy cơ overfitting thông qua kỹ thuật bỏ phiếu đa số. Logistic Regression, tuy là một mô hình tuyến tính đơn giản, lại tỏ ra hiệu quả trong nhiều bài toán phân lớp nhị phân và có lợi thế ở tính dễ triển khai, dễ giải thích. Trong khi đó, K-Nearest Neighbors dựa trên khoảng cách giữa các điểm dữ liệu để phân loại, tuy nhiên mô hình này thường gặp khó khăn trong các tập dữ liệu lớn hoặc khi có nhiều nhiễu.

Mặc dù các phương pháp truyền thống này mang lại nhiều lợi ích như dễ triển khai, đơn giản, khả năng mở rộng tốt cho các hệ thống nhỏ, và yêu cầu tài nguyên tính toán thấp, chúng vẫn tồn tại những hạn chế lớn. Một trong những nhược điểm rõ rệt nhất là sự phụ thuộc vào quá trình trích xuất đặc trưng thủ công. Điều này khiến hiệu quả của mô hình phụ thuộc rất nhiều vào kiến thức chuyên môn của người thiết kế và khả năng lựa chọn đặc trưng phù hợp cho từng bài toán cụ thể. Hơn nữa, các phương pháp này không tận dụng được đầy đủ ngữ cảnh và cấu trúc ngữ nghĩa của văn bản – yếu tố đặc biệt quan trọng trong việc phân tích và hiểu nội dung tinh vi hoặc có dụng ý gây hiểu nhầm của tin giả. Các mô hình truyền thống thường coi văn bản như một tập hợp rời rạc các từ mà không quan tâm đến thứ tự hoặc ngữ cảnh xung quanh, dẫn đến việc mất đi nhiều thông tin quan trọng giúp phân biệt giữa sự thật và giả dối.

Chính vì những hạn chế này, khi các mô hình học sâu và sau đó là các mô hình ngôn ngữ lớn như BERT, RoBERTa, và gần đây là GPT và các biến thể LLM khác ra đời, cộng đồng nghiên cứu đã nhanh chóng chuyển hướng sang các phương pháp hiện đại hơn. Các mô hình này cho phép học trực tiếp từ dữ liệu đầu vào với khả năng nắm bắt ngữ nghĩa, ngữ cảnh, và mối quan hệ giữa các từ trong văn bản ở cấp độ sâu sắc hơn nhiều so với các mô hình truyền thống. Tuy nhiên, không thể phủ nhận rằng các phương pháp truyền thống vẫn đóng vai trò nền tảng, cung cấp nền tảng lý thuyết vững chắc và là bước khởi đầu quan trọng cho lĩnh vực phát hiện tin giả hiện đại.

* 1. **Giới thiệu về LLMs**

Trong lĩnh vực AI và NLP, sự xuất hiện của các mô hình LLM đã tạo nên bước ngoặt trong cách máy tính hiểu và tương tác với ngôn ngữ con người. Các LLM hiện đại, tiêu biểu như GPT, BERT hay RoBERTa, được huấn luyện trên tập dữ liệu văn bản cực lớn với hàng trăm triệu đến hàng tỷ tham số. Nhờ đó, chúng có khả năng học được cấu trúc ngữ pháp, ngữ nghĩa, và ngữ cảnh sâu sắc trong văn bản. Với khả năng tổng quát hóa mạnh mẽ, các mô hình này có thể áp dụng cho nhiều nhiệm vụ như phân loại văn bản, trả lời câu hỏi, phân tích cảm xúc, sinh văn bản và phát hiện tin giả.

Tuy nhiên, phần lớn các LLM hiện nay được huấn luyện chủ yếu trên tiếng Anh và một số ngôn ngữ phổ biến khác. Khi áp dụng vào tiếng Việt – một ngôn ngữ đơn âm, ngữ cảnh phụ thuộc nhiều vào thanh điệu và cấu trúc từ phức tạp – hiệu quả thường không cao nếu không có sự điều chỉnh phù hợp. Các mô hình quốc tế như GPT-3 hay RoBERTa có thể không nắm bắt chính xác được ngữ nghĩa trong tiếng Việt, đặc biệt là trong các tình huống chứa từ đa nghĩa, từ láy, hay thành ngữ dân gian. Điều này tạo ra nhu cầu cấp thiết trong việc phát triển và khai thác các LLM dành riêng cho tiếng Việt.

Trong những năm gần đây, cộng đồng nghiên cứu đã cho ra đời một số mô hình ngôn ngữ được huấn luyện chuyên biệt cho tiếng Việt như PhoBERT, viBERT, viT5 và Vietnamese-SBERT. Những mô hình này được huấn luyện trên các kho dữ liệu tiếng Việt lớn như Wikipedia, báo chí và mạng xã hội, giúp cải thiện rõ rệt khả năng hiểu ngữ nghĩa và ngữ cảnh tiếng Việt trong các tác vụ NLP. Trong đó, Vietnamese-SBERT là một phiên bản điều chỉnh của Sentence-BERT, được thiết kế để sinh vector biểu diễn câu tối ưu cho việc so sánh, phân loại hoặc tìm kiếm ngữ nghĩa giữa các văn bản tiếng Việt.

Trong đề tài này, nhóm lựa chọn mô hình keepitreal/vietnamese-sbert, một phiên bản Vietnamese-SBERT được huấn luyện bởi cộng đồng trên nền tảng Hugging Face. Mô hình này có ưu điểm là nhẹ, hiệu quả cao và tương thích tốt với văn bản tiếng Việt hiện đại. Nhóm sử dụng mô hình này để mã hóa văn bản đầu vào thành vector ngữ nghĩa, sau đó đưa vào mạng LSTM hai chiều nhằm học mối quan hệ ngữ cảnh giữa các từ trong chuỗi. Việc kết hợp tokenizer chuyên biệt tiếng Việt từ SBERT và kiến trúc BiLSTM giúp hệ thống có thể hiểu tốt hơn mạch văn bản, đồng thời thích nghi tốt với các đặc điểm ngôn ngữ địa phương.

Khác với các mô hình GPT-3.5 hoặc GPT-4 yêu cầu truy cập API đám mây với chi phí cao và độ kiểm soát thấp, phương pháp của nhóm có thể triển khai hoàn toàn độc lập trên môi trường Google Colab, sử dụng tài nguyên GPU miễn phí. Điều này phù hợp với điều kiện thực tế trong môi trường học thuật, nơi tài nguyên tính toán còn hạn chế nhưng yêu cầu về độ chính xác và khả năng tùy chỉnh cao.

Tóm lại, thay vì chỉ sử dụng các LLM phổ thông huấn luyện bằng tiếng Anh, đề tài hướng đến khai thác và triển khai LLM được thiết kế riêng cho tiếng Việt, nhằm nâng cao hiệu quả phát hiện tin giả trong môi trường ngôn ngữ địa phương. Đây chính là điểm mới, mang lại giá trị học thuật và thực tiễn của đề tài – góp phần thúc đẩy nghiên cứu và ứng dụng các mô hình AI trong lĩnh vực ngôn ngữ Việt Nam.

* 1. **Các kỹ thuật tinh chỉnh LLM**

Sau khi mô hình LLM được tiền huấn luyện trên các tập dữ liệu văn bản quy mô lớn, người dùng cần thực hiện các bước tinh chỉnh để áp dụng hiệu quả vào các nhiệm vụ cụ thể trong thực tế như phân loại văn bản, trả lời câu hỏi, phát hiện tin giả, v.v. Việc tinh chỉnh giúp mô hình thích nghi tốt hơn với đặc trưng của bài toán, dữ liệu và ngữ cảnh cụ thể. Hiện nay, ba nhóm kỹ thuật tinh chỉnh chính được sử dụng phổ biến gồm: fine-tuning toàn phần, prompting, và PEFT – tức tinh chỉnh hiệu quả tham số.

Trong đó, fine-tuning là kỹ thuật cổ điển và mạnh mẽ nhất. Toàn bộ trọng số của mô hình được tiếp tục huấn luyện trên một tập dữ liệu có gán nhãn cho tác vụ cụ thể. Việc cập nhật toàn bộ mạng giúp mô hình “chuyên biệt hóa” sâu cho nhiệm vụ cần giải quyết, từ đó thường mang lại độ chính xác cao. Tuy nhiên, kỹ thuật này đòi hỏi rất nhiều tài nguyên tính toán và bộ nhớ, do phải huấn luyện lại toàn bộ mô hình với hàng triệu tham số. Vì vậy, fine-tuning toàn phần thường chỉ khả thi trong môi trường có GPU/TPU mạnh và tập dữ liệu huấn luyện đủ lớn, đủ chất lượng.

Ngược lại, prompting là phương pháp không cần huấn luyện lại mô hình mà chỉ sử dụng các đoạn gợi ý được thiết kế thông minh để dẫn dắt mô hình sinh ra kết quả mong muốn. Các biến thể phổ biến của prompting gồm: zero-shot, one-shot, few-shot, hoặc hướng dẫn theo chuỗi. Prompting đặc biệt hữu ích trong các hệ thống hội thoại, chatbot hoặc công cụ truy vấn ngôn ngữ tự nhiên, nơi người dùng có thể tận dụng năng lực sẵn có của LLM mà không cần can thiệp vào cấu trúc bên trong mô hình. Tuy nhiên, nhược điểm của prompting là kết quả thường thiếu ổn định, độ chính xác phụ thuộc nhiều vào cách thiết kế prompt và không dễ kiểm soát khi triển khai vào hệ thống sản phẩm thực tế.

Để dung hòa giữa hiệu quả mô hình và chi phí huấn luyện, các phương pháp PEFTđã ra đời. Thay vì cập nhật toàn bộ trọng số, PEFT chỉ điều chỉnh một phần rất nhỏ của mô hình (thường dưới 1–10% tham số), từ đó tiết kiệm đáng kể tài nguyên tính toán trong khi vẫn giữ được hiệu quả gần bằng fine-tuning toàn phần. Một số kỹ thuật PEFT nổi bật gồm: Adapter Tuning, LoRA, và Prompt Tuning/Prefix Tuning. Các kỹ thuật này cho phép triển khai LLM hiệu quả ngay cả trên các thiết bị phổ thông như laptop hoặc máy chủ có GPU trung bình, đặc biệt phù hợp trong môi trường học thuật hoặc khởi nghiệp.

Bên cạnh ba nhóm chính trên, một hướng tinh chỉnh gián tiếp khác cũng được sử dụng là Knowledge Distillation. Trong phương pháp này, một mô hình nhỏ hơn (student model) được huấn luyện để mô phỏng lại kết quả đầu ra của mô hình lớn hơn (teacher model). Điều này cho phép xây dựng các mô hình nhẹ như TinyBERT, DistilBERT, hay MiniLM, có khả năng hoạt động trong môi trường tài nguyên hạn chế (như thiết bị di động) nhưng vẫn giữ được độ chính xác ở mức chấp nhận được nhờ kế thừa tri thức từ mô hình lớn.

Trong thực tế, các kỹ thuật tinh chỉnh LLM không tồn tại độc lập mà thường được kết hợp linh hoạt. Ví dụ, người phát triển có thể sử dụng prompting để kiểm tra nhanh ý tưởng hoặc thu được phản hồi ban đầu từ mô hình lớn, sau đó nếu muốn triển khai tiết kiệm tài nguyên hoặc tối ưu trên thiết bị thực, họ có thể chuyển sang fine-tuning hiệu quả bằng PEFT hoặc distillation. Chính nhờ sự đa dạng và linh hoạt trong chiến lược tinh chỉnh mà LLM có thể được cá nhân hóa và triển khai ở nhiều cấp độ hệ thống khác nhau – từ nghiên cứu học thuật, thử nghiệm ứng dụng, đến sản phẩm thương mại thực tế.

* 1. **Các chỉ số đánh giá mô hình phân loại**

Trong các bài toán phân loại, đặc biệt là bài toán phát hiện tin giả với dữ liệu thường mất cân bằng số lượng tin giả nhiều hơn hoặc ít hơn tin thật, việc đánh giá mô hình chỉ dựa vào độ chính xác là không đủ để phản ánh toàn diện hiệu quả. Vì vậy, cần sử dụng thêm các chỉ số khác như Precision, Recall, F1-score và ma trận nhầm lẫn. Dưới đây là định nghĩa và vai trò của từng chỉ số:

1. Ma trận nhầm lẫn

Ma trận nhầm lẫn là một công cụ trực quan giúp thể hiện số lượng dự đoán đúng và sai của mô hình cho từng loại nhãn. Với bài toán phát hiện tin giả (gồm 2 lớp: “tin thật” và “tin giả”), confusion matrix có cấu trúc 2×2 như sau:

Bảng 2‑1: Cấu trúc ma trận nhầm lẫn

|  |  |  |
| --- | --- | --- |
|  | Dự đoán: Tin giả (Positive) | Dự đoán: Tin thật (Negative) |
| Thực tế: Tin giả (Positive) | TP – Dự đoán đúng tin giả | FN – Bỏ sót tin giả |
| Thực tế: Tin thật (Negative) | FP – Nhầm tin thật thành tin giả | TN – Dự đoán đúng tin thật |

Với TP (True Positive) là tin giả được phát hiện đúng, FP (False Positive) là tin thật bị nhầm là tin giả, FN (False Negative) là tin giả bị bỏ sót, nhầm là tin thật, TN (True Negative) là tin thật được nhận diện chính xác.

1. Loss – Hàm mất mát

Loss là đại lượng đo lường mức độ sai lệch giữa dự đoán của mô hình và nhãn thực tế trong quá trình huấn luyện. Trong bài toán phân loại nhị phân như phát hiện tin giả, một hàm mất mát phổ biến là Binary Cross-Entropy Loss, được định nghĩa như sau [7]:

Trong đó là nhãn thực tế (0 hoặc 1), là xác suất mô hình dự đoán là nhãn 1 (tin giả), N là số lượng mẫu.

1. Accuracy - Độ chính xác tổng thể

Độ chính xác đo lường tỷ lệ mẫu được dự đoán đúng trên tổng số mẫu. Đây là chỉ số phổ biến nhất nhưng không đủ nếu dữ liệu lệch nhãn [6] .

1. Precision – Độ chính xác dương tính

Precision đo tỷ lệ văn bản mà mô hình dự đoán là tin giả thực sự là tin giả. Chỉ số này đặc biệt quan trọng trong các ứng dụng cần tránh cảnh báo sai (false positive), ví dụ khi mô hình tự động chặn nội dung [6].

1. Recall – Khả năng truy hồi

Recall đo tỷ lệ tin giả thực sự được mô hình phát hiện. Chỉ số này quan trọng trong bài toán như phát hiện tin giả, vì bỏ sót tin giả có thể gây hậu quả nghiêm trọng [6].

1. F1-score – Trung bình điều hòa giữa Precision và Recall

F1-score là chỉ số tổng hợp giữa Precision và Recall, phản ánh sự cân bằng giữa hai chỉ số này. Chỉ số này rất quan trọng trong bài toán phân loại mất cân bằng như phát hiện tin giả.

# **PHƯƠNG PHÁP ĐỀ XUẤT**

* 1. **Phát hiện tin giả bằng mô hình Vietnamese-BartPho kết hợp BiLSTM**

Trong phần mở rộng của mô hình, nhóm nghiên cứu đề xuất một mô hình học sâu nhằm phát hiện tin giả trong văn bản tiếng Việt bằng cách kết hợp giữa bộ mã hóa từ vựng của mô hình ngôn ngữ lớn **BARTpho-syllable** và mạng **BiLSTM**. Phương pháp này tận dụng khả năng tiền huấn luyện mạnh mẽ của BARTpho trong việc hiểu ngữ nghĩa ở cấp độ âm tiết của tiếng Việt, đồng thời khai thác hiệu quả mô hình hóa ngữ cảnh chuỗi hai chiều từ BiLSTM. Sự kết hợp này nhằm cải thiện hiệu quả phân loại tin tức thành tin thật hoặc tin giả với độ chính xác cao hơn so với các phương pháp truyền thống.

Trong quy trình xử lý, các văn bản đầu vào được mã hóa thông qua tokenizer của mô hình **vinai/bartpho-syllable.** Tokenizer này thực hiện phân tách văn bản tiếng Việt thành các đơn vị âm tiết, phù hợp với cấu trúc ngôn ngữ tiếng Việt và ánh xạ mỗi âm tiết thành chỉ số tương ứng trong bộ từ vựng đã được tiền huấn luyện. Khác với các phương pháp sử dụng tokenizer theo từ hoặc subword, tokenizer này mang lại khả năng giữ được thông tin ngôn ngữ đặc trưng hơn trong tiếng Việt. Các chỉ số sau khi mã hóa sẽ được đưa vào một lớp nhúng được khởi tạo lại và học từ đầu, với số chiều cố định là 300, nhằm tạo ra biểu diễn vector phù hợp với đặc trưng của tập dữ liệu cụ thể.

Sau bước nhúng, chuỗi vector đầu vào sẽ được đưa vào một mạng **BiLSTM hai lớp**, mỗi lớp có **kích thước ẩn là 64**. BiLSTM hoạt động theo cả hai hướng – từ trái sang phải và từ phải sang trái – để mô hình có thể học được các thông tin ngữ cảnh toàn diện hơn so với LSTM một chiều. Trong quá trình huấn luyện, kỹ thuật pack\_padded\_sequence được áp dụng để tối ưu hóa hiệu suất tính toán, tránh việc tính toán các phần tử padding không cần thiết trong chuỗi. Kết quả đầu ra từ hai hướng của BiLSTM được nối lại, sau đó được đưa qua một lớp tuyến tính và cuối cùng là hàm **sigmoid** để trả về một xác suất đầu ra nằm trong khoảng từ 0 đến 1.

Xác suất đầu ra thể hiện mức độ khả nghi của văn bản. Trong mô hình này, **ngưỡng phân loại được chọn là 0.4** – nếu xác suất đầu ra lớn hơn hoặc bằng 0.4, văn bản được phân loại là tin giả; ngược lại, là tin thật. Ngưỡng này được xác định thông qua thực nghiệm nhằm tối ưu sự cân bằng giữa **recall** và **precision**, hai chỉ số quan trọng trong bài toán phát hiện thông tin sai lệch.

Tập dữ liệu huấn luyện được chia thành ba phần: train.csv, val.csv, và test.csv, được xử lý bằng thư viện **Pandas** và đóng gói thông qua một lớp dữ liệu tùy chỉnh Custom\_Text\_Dataset. Quá trình nạp dữ liệu sử dụng DataLoader của PyTorch, kết hợp với một hàm collate\_fn tùy chỉnh để sắp xếp các chuỗi văn bản theo độ dài giảm dần và áp dụng kỹ thuật padding phù hợp. Mỗi batch dữ liệu được đưa vào GPU để tăng tốc độ huấn luyện, tận dụng sức mạnh xử lý của Google Colab với GPU miễn phí.

Mô hình sử dụng hàm mất mát **Binary Cross-Entropy,** kết hợp với thuật toán tối ưu **Adam** và **learning rate** được thiết lập là 1e-4. Quá trình huấn luyện được thực hiện trong **40 epoch**, với cơ chế lưu lại mô hình có độ chính xác tốt nhất trên tập validation để đảm bảo hiệu quả tổng quát hóa. Việc đánh giá mô hình được thực hiện thông qua các chỉ số như **accuracy, precision, recall** và **F1-score**, bên cạnh đó còn sử dụng **ma trận nhầm lẫn** để trực quan hóa hiệu quả phân loại giữa hai lớp.

Theo kết quả thực nghiệm, mô hình đạt độ chính xác **khoảng 92% trên tập kiểm thử**, trong khi thời gian huấn luyện trung bình cho toàn bộ quá trình là khoảng **30 phút** khi sử dụng môi trường GPU trên Google Colab. Ngoài ra, nhóm cũng tích hợp một chức năng kiểm thử thực tế, cho phép người dùng nhập một đoạn văn bản mới và hệ thống sẽ trả về dự đoán liệu văn bản đó là tin thật hay tin giả. Điều này cho thấy mô hình không chỉ hiệu quả trong môi trường nghiên cứu mà còn có tiềm năng triển khai thực tế trong các ứng dụng kiểm duyệt nội dung hoặc hỗ trợ người đọc nhận diện tin giả.

Tổng kết lại, phương pháp kết hợp giữa tokenizer của BARTpho-syllable và mạng BiLSTM mang lại hiệu quả đáng kể trong bài toán phát hiện tin giả tiếng Việt. Không cần phải fine-tune toàn bộ mô hình ngôn ngữ lớn, hệ thống vẫn có thể tận dụng sức mạnh biểu diễn của BARTpho và tính linh hoạt của LSTM để huấn luyện mô hình nhẹ hơn, phù hợp với môi trường tính toán hạn chế nhưng vẫn duy trì độ chính xác cao.

A diagram of a computer program

AI-generated content may be incorrect.

Hình 3‑1:Sơ đồ kiến trúc mô hình phát hiện tin giả sử dụng tokenizer BartPho và BiLSTM

* 1. **Phát hiện tin giả bằng mô hình tokenizer PhoBERT kết hợp BiLSTM**

Trong phần mở rộng nghiên cứu, nhóm đã triển khai một mô hình học sâu mới kết hợp giữa tokenizer của mô hình ngôn ngữ lớn PhoBERT và mạng BiLSTM hai chiều tùy chỉnh, nhằm đánh giá hiệu quả của cách tiếp cận này trong bài toán phát hiện tin giả tiếng Việt. Khác với phương pháp sử dụng Vietnamese-SBERT ở mục trước, phương pháp này tận dụng tokenizer từ PhoBERT để mã hóa ngữ nghĩa đầu vào, sau đó đưa vào mạng BiLSTM do nhóm xây dựng từ đầu bằng thư viện PyTorch.

PhoBERT là một mô hình ngôn ngữ lớn được huấn luyện chuyên biệt cho tiếng Việt, có khả năng biểu diễn ngữ nghĩa sâu sắc dựa trên cấu trúc từ vựng tiếng Việt hiện đại. Trong phương pháp đề xuất, nhóm không tinh chỉnh toàn bộ mô hình PhoBERT mà chỉ sử dụng tokenizer đi kèm của mô hình "vinai/phobert-base" để chuyển đổi văn bản đầu vào thành chuỗi các chỉ số nguyên. Việc sử dụng tokenizer này giúp giữ lại đầy đủ ngữ nghĩa và đặc trưng của tiếng Việt, đồng thời đơn giản hóa kiến trúc và giảm chi phí tính toán khi không cần phải huấn luyện lại toàn bộ mô hình transformer gốc.

Sau bước mã hóa, các token ID sẽ được đưa vào một mô hình BiLSTM hai lớp. Kiến trúc mạng gồm ba thành phần chính: lớp embedding học từ đầu với số chiều 300, lớp BiLSTM với 2 tầng và 64 đơn vị ẩn mỗi tầng, hoạt động theo cả hai chiều trái–phải, và cuối cùng là lớp tuyến tính kết hợp hàm sigmoid để xuất ra xác suất phân lớp nhị phân. Việc sử dụng BiLSTM giúp mô hình học được mối quan hệ ngữ cảnh hai chiều trong chuỗi văn bản, từ đó cải thiện khả năng phân biệt giữa tin thật và tin giả. Ngoài ra, kỹ thuật pack\_padded\_sequence được áp dụng để tránh việc mô hình học trên các phần padding không có giá trị ngữ nghĩa, giúp tối ưu quá trình huấn luyện.

Về quá trình huấn luyện, dữ liệu đầu vào gồm ba tệp train.csv, val.csv, và test.csv, được xử lý và đóng gói bằng lớp Custom\_Text\_Dataset tùy chỉnh. Dữ liệu sau đó được đưa vào DataLoader với batch size là 64. Để đảm bảo hiệu quả xử lý, nhóm sử dụng hàm collate\_fn để sắp xếp các chuỗi theo độ dài giảm dần, đồng thời áp dụng hàm pad\_sequence để đồng nhất kích thước các chuỗi trong cùng một batch. Quá trình huấn luyện diễn ra trong 40 epoch, sử dụng hàm mất mát Binary Cross Entropy và bộ tối ưu Adam với learning rate là 1e-4. Mô hình sẽ được lưu lại khi độ chính xác trên tập validation đạt giá trị tốt nhất.

Sau huấn luyện, mô hình được đánh giá bằng các chỉ số hiệu suất bao gồm Accuracy, Precision, Recall và F1-score. Kết quả phân loại được trực quan hóa bằng confusion matrix để đánh giá độ chính xác theo từng lớp. Ngoài ra, nhóm còn tích hợp một hàm thử nghiệm văn bản thực tế, cho phép người dùng nhập một câu bất kỳ để mô hình tự động phân loại là tin thật hoặc tin giả. Tính năng này chứng minh tính ứng dụng của mô hình trong các hệ thống phát hiện tin giả thực tiễn.

Tổng kết lại, phương pháp sử dụng tokenizer của PhoBERT kết hợp với mô hình BiLSTM học từ đầu cho thấy một hướng tiếp cận nhẹ, hiệu quả và dễ triển khai. Mặc dù không cần fine-tune toàn bộ mô hình transformer, mô hình vẫn tận dụng được sức mạnh biểu diễn ngữ nghĩa từ tokenizer chuyên biệt của PhoBERT. Nhờ đó, phương pháp này phù hợp với môi trường có tài nguyên tính toán hạn chế như Google Colab và có thể mở rộng triển khai trong các hệ thống phát hiện nội dung sai lệch trên văn bản tiếng Việt.

A diagram of a company

AI-generated content may be incorrect.

Hình 3‑2:Sơ đồ kiến trúc mô hình phát hiện tin giả sử dụng tokenizer PhoBERT và BiLSTM

* 1. **Mô hình đề xuất phát hiện tin giả bằng Vietnamese-SBERT kết hợp BiLSTM**

Trong đề tài này, nhóm nghiên cứu đề xuất một mô hình học sâu nhằm phát hiện tin giả trong văn bản tiếng Việt, bằng cách kết hợp giữa bộ mã hóa từ vựng của mô hình ngôn ngữ lớn Vietnamese-SBERT và mạng BiLSTM. Phương pháp này tận dụng được ưu thế của SBERT trong việc trích xuất ngữ nghĩa chính xác cho tiếng Việt, đồng thời kết hợp khả năng học ngữ cảnh chuỗi của BiLSTM, từ đó cải thiện hiệu quả phân loại tin thật hoặc tin giả.

Trong quy trình xử lý, các văn bản đầu vào được mã hóa thông qua tokenizer từ mô hình Vietnamese-SBERT. Tokenizer này thực hiện việc phân tách câu thành các token phù hợp với tiếng Việt và ánh xạ chúng thành chỉ mục trong từ vựng đã được tiền huấn luyện. Các token sau khi chuyển thành chỉ số sẽ được đưa vào lớp embedding học từ đầu với số chiều cố định là 300, nhằm tạo ra biểu diễn vector đầu vào phù hợp với tập dữ liệu cụ thể.

Sau bước nhúng, các chuỗi vector được đưa vào mạng BiLSTM hai lớp với chiều ẩn là 64, hoạt động theo cả hai chiều trái–phải và phải–trái. Mạng LSTM sử dụng kỹ thuật pack padded sequence để tránh tính toán các vị trí padding không cần thiết, giúp mô hình huấn luyện hiệu quả hơn. Hai vector trạng thái cuối cùng từ hai hướng được nối lại và chuyển qua một lớp tuyến tính, sau đó qua hàm sigmoid để tạo ra xác suất đầu ra trong khoảng từ 0 đến 1.

Đầu ra của mô hình là xác suất thể hiện mức độ "giả" của văn bản. Ngưỡng phân loại được chọn là 0.4 – nếu xác suất ≥ 0.4 thì phân loại là tin giả, ngược lại là tin thật. Ngưỡng này được xác định thông qua thực nghiệm để cân bằng giữa recall và precision.

Dữ liệu huấn luyện được chia thành ba tập: train, validation và test, định dạng CSV và xử lý bằng pandas. Văn bản được đóng gói trong lớp dữ liệu tùy chỉnh và sử dụng DataLoader với hàm collate riêng để sắp xếp theo độ dài và thực hiện padding. Dữ liệu được chia thành mini-batch và đưa vào GPU để tăng tốc huấn luyện. Mô hình sử dụng hàm mất mát BCE, kết hợp với thuật toán Adam và learning rate là 1e-4. Trong quá trình huấn luyện 40 epoch, mô hình sẽ lưu lại phiên bản tốt nhất khi accuracy trên tập validation tăng.

Mô hình được đánh giá bằng các chỉ số: accuracy, precision, recall và F1-score. Ngoài ra còn sử dụng confusion matrix để trực quan hóa kết quả phân loại. Theo thực nghiệm, mô hình đạt accuracy khoảng 92% trên tập kiểm định, với thời gian huấn luyện trung bình khoảng 30 phút trên Google Colab sử dụng GPU miễn phí.

Mô hình cũng được tích hợp chức năng kiểm tra văn bản mới do người dùng nhập vào. Văn bản được mã hóa và đưa qua mô hình, sau đó kết quả so sánh với ngưỡng để kết luận là tin thật hay tin giả, chứng minh tính khả thi trong triển khai thực tế.

Tổng kết lại, phương pháp đề xuất kết hợp giữa tokenizer của Vietnamese-SBERT và BiLSTM cho thấy hiệu quả trong bài toán phát hiện tin giả tiếng Việt, tận dụng được sức mạnh của LLM mà không cần fine-tune toàn bộ mô hình, đồng thời phù hợp triển khai trong môi trường hạn chế tài nguyên.

A diagram of a company

AI-generated content may be incorrect.

Hình 3‑3:Sơ đồ kiến trúc mô hình phát hiện tin giả sử dụng Vietnamese-SBERT và BiLSTM

* 1. **Tiền xử lý dữ liệu**

Tiền xử lý dữ liệu là bước quan trọng nhằm chuẩn hóa và làm sạch dữ liệu đầu vào trước khi đưa vào mô hình học sâu. Việc này giúp cải thiện hiệu quả huấn luyện, giảm nhiễu và đảm bảo mô hình học được các đặc trưng ngôn ngữ chính xác hơn. Trong chương này, nhóm trình bày chi tiết về tập dữ liệu sử dụng và các bước xử lý đã được áp dụng để chuẩn bị dữ liệu đầu vào cho hệ thống phát hiện tin giả.

Tập dữ liệu sử dụng trong nghiên cứu là ReINTEL 2020, một tập dữ liệu được công bố trong khuôn khổ cuộc thi “Responsible Information Identification on Social Network Sites” do Ban tổ chức VLSP phát động vào năm 2020 [8]. Tập dữ liệu bao gồm các bài viết tiếng Việt được thu thập từ nhiều nguồn khác nhau như mạng xã hội và các cơ quan báo chí chính thống. Mỗi văn bản được gán nhãn nhị phân: 0 tương ứng với tin thật và 1 tương ứng với tin giả. Đây là bộ dữ liệu phản ánh chân thực bối cảnh lan truyền thông tin sai lệch trong giai đoạn đại dịch COVID-19 diễn ra mạnh mẽ trên mạng xã hội tại Việt Nam.

Dữ liệu được thu thập trong khoảng thời gian từ tháng 8 đến tháng 10 năm 2020, tổng cộng có 9.713 văn bản được đưa vào sử dụng trong quá trình huấn luyện và đánh giá mô hình. Dữ liệu được chia thành ba tập con là train, validation và test. Thống kê chi tiết về số lượng tin thật và tin giả ở từng tập được thể hiện trong bảng dưới đây.

Bảng 3‑1:Thống kê tập tin dữ liệu

|  |  |  |  |
| --- | --- | --- | --- |
| **Tổng Hợp 3 Tập Tin** | | | |
|  | **Train** | **Valid** | **Test** |
| **Real** | 1472 | 82 | 82 |
| **Fake** | 7269 | 404 | 404 |
| **Total** | 8781 | 486 | 486 |

Dữ liệu có sự mất cân bằng đáng kể giữa hai nhãn, trong đó số lượng tin giả chiếm áp đảo so với tin thật. Cụ thể, ở cả ba tập train, validation và test, số lượng văn bản thuộc lớp tin giả chiếm khoảng 83% tổng số mẫu. Sự chênh lệch này phản ánh hiện thực rằng tin giả thường xuất hiện nhiều và lan truyền nhanh hơn tin thật trên các nền tảng số, đồng thời cũng tạo ra thách thức trong quá trình huấn luyện mô hình khi dễ dẫn đến hiện tượng thiên lệch dự đoán.

Bên cạnh việc phân loại nhãn thật – giả, tập dữ liệu còn được chia thành ba nhóm chủ đề chính bao gồm: COVID-19, Thời sự & Chính trị, và Các chủ đề khác như giải trí, xã hội, giáo dục, v.v. Việc phân loại này giúp mô hình được đánh giá chi tiết hơn trên từng nhóm nội dung, từ đó xác định mức độ ổn định khi triển khai trên dữ liệu thực tế. Bảng dưới đây thể hiện rõ phân bố dữ liệu theo từng chủ đề và theo từng tập.

Bảng 3‑2:Thống kê các tập dữ liệu

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **COVID-19** | | | **Thời sự & Chính trị** | | | **Khác** | | |
|  | **Train** | **Valid** | **Test** | **Train** | **Valid** | **Tst** | **Train** | **Valid** | **Test** |
| **Real** | 681 | 50 | 33 | 365 | 17 | 24 | 426 | 15 | 25 |
| **Fake** | 2979 | 173 | 161 | 1363 | 79 | 84 | 2927 | 152 | 159 |
| **Total** | 3660 | 223 | 194 | 1728 | 96 | 108 | 3393 | 167 | 184 |

Từ bảng thống kê, có thể nhận thấy các bài viết thuộc chủ đề COVID-19 chiếm phần lớn dữ liệu trong cả ba tập. Đây là nhóm chủ đề nổi bật nhất trong giai đoạn dữ liệu được thu thập, đồng thời cũng là nội dung xuất hiện nhiều tin giả nhất. Điều này phản ánh bối cảnh thực tế tại Việt Nam trong năm 2020, khi thông tin sai lệch về dịch bệnh lan truyền nhanh chóng và gây ảnh hưởng lớn đến cộng đồng.

Sau khi tải về, dữ liệu thô được xử lý qua nhiều bước để đảm bảo đầu vào phù hợp cho mô hình học sâu. Đầu tiên, dữ liệu được làm sạch bằng cách loại bỏ các dòng trùng lặp, đồng thời xóa bỏ các ký tự không cần thiết như emoji, HTML tags, liên kết URL và chuẩn hóa chữ thường. Tất cả văn bản đều được chuẩn hóa mã hóa về UTF-8 nhằm tránh lỗi trong quá trình xử lý tiếng Việt.

Văn bản sau làm sạch được đưa vào quá trình token hóa bằng bộ tokenizer phù hợp với từng mô hình ngôn ngữ tiền huấn luyện. Cụ thể, mô hình kết hợp với BARTpho sử dụng tokenizer vinai/bartpho-syllable, mô hình kết hợp với PhoBERT sử dụng tokenizer vinai/phobert-base, và mô hình kết hợp với Vietnamese-SBERT sử dụng tokenizer keepitreal/vietnamese-sbert. Các tokenizer này giúp chuyển đổi câu văn thành chuỗi các chỉ số đầu vào (input\_ids), đồng thời sinh ra các vector như attention\_mask và token\_type\_ids phục vụ cho việc học và phân tích ngữ cảnh.

Sau bước token hóa, dữ liệu được chuẩn hóa độ dài bằng kỹ thuật padding động. Với mỗi batch huấn luyện, độ dài của câu dài nhất trong batch được lấy làm chuẩn. Các câu ngắn hơn sẽ được bổ sung token [PAD] ở cuối để đảm bảo đồng nhất về kích thước. Hệ thống attention mask cũng được cập nhật tương ứng để mô hình không học những phần đệm không mang thông tin.

Cuối cùng, dữ liệu sau xử lý được chuyển đổi thành dạng tensor để đưa vào mô hình học sâu, sử dụng thư viện PyTorch. Quy trình tiền xử lý đảm bảo văn bản đầu vào được chuẩn hóa, nhất quán và sẵn sàng cho việc huấn luyện với các kiến trúc mạng nơ-ron hiện đại như BiLSTM kết hợp với các mô hình ngôn ngữ lớn.

* 1. **Chiến lược huấn luyện**

Chiến lược huấn luyện đóng vai trò quyết định trong việc đảm bảo mô hình học sâu đạt được hiệu năng tối ưu. Phần này trình bày chi tiết quá trình huấn luyện mô hình BiLSTM sử dụng dữ liệu đã được tiền xử lý, bao gồm các yếu tố như cấu hình mô hình, bộ tối ưu hóa, hàm mất mát và quy trình đánh giá mô hình theo từng epoch.

Đầu tiên, mô hình được khởi tạo với kiến trúc gồm một lớp embedding kích thước 300 chiều, BiLSTM với số nút ẩn là 64, lớp tuyến tính đầu ra và hàm kích hoạt sigmoid. Mô hình được đưa lên GPU nếu có hỗ trợ để tăng tốc quá trình huấn luyện.

Trong quá trình huấn luyện, nhóm sử dụng hàm mất mát là Binary Cross Entropy, phù hợp với bài toán phân loại nhị phân. Bộ tối ưu hóa được sử dụng là Adam với tốc độ học được đặt ở mức 1e-4 nhằm đảm bảo mô hình hội tụ ổn định. Mỗi mẫu trong tập dữ liệu được xử lý thành tensor và chia thành các batch có kích thước 64. Để xử lý các chuỗi có độ dài khác nhau, nhóm sử dụng kỹ thuật padding kết hợp với pack\_padded\_sequence, cho phép LSTM bỏ qua phần đệm và chỉ tập trung vào nội dung thực tế của từng câu.

Việc huấn luyện diễn ra trong 40 epoch. Trong mỗi epoch, mô hình được đánh giá trên cả tập huấn luyện, kiểm định và kiểm tra. Sau mỗi epoch, độ chính xác trên tập kiểm định được sử dụng làm tiêu chí để lưu mô hình. Nếu độ chính xác hiện tại vượt qua giá trị tốt nhất đã ghi nhận, mô hình sẽ được lưu lại với tên tương ứng theo độ chính xác đó. Điều này giúp đảm bảo mô hình cuối cùng không bị overfitting và có thể tổng quát hóa tốt.

Để đánh giá hiệu suất mô hình, nhóm sử dụng các chỉ số sau: accuracy, độ chính xác dương tính, recall, F1-score và trực quan hóa kết quả bằng ma trận nhầm lẫn. Việc sử dụng nhiều chỉ số giúp đánh giá toàn diện hơn, đặc biệt trong các trường hợp dữ liệu mất cân bằng giữa các nhãn. Trong thực nghiệm, mô hình đạt độ chính xác kiểm định trung bình khoảng 92%, cho thấy khả năng phân loại hiệu quả giữa tin thật và tin giả trên văn bản tiếng Việt.

Chiến lược huấn luyện trong đề tài được thiết kế để cân bằng giữa độ chính xác và khả năng triển khai thực tế. Việc sử dụng BiLSTM kết hợp tokenizer của Vietnamese-SBERT, cùng kỹ thuật huấn luyện có kiểm soát, giúp mô hình học được ngữ cảnh và ngữ nghĩa hiệu quả trong điều kiện tài nguyên tính toán giới hạn.

# **THỰC NGHIỆM VÀ KẾT QUẢ**

* 1. **Mô tả tập dữ liệu**

Trong quá trình xây dựng mô hình phát hiện tin giả tiếng Việt, nhóm đã sử dụng tập dữ liệu được thu thập từ nguồn đáng tin cậy là kho dữ liệu của tổ chức nghiên cứu về truyền thông và dữ liệu mở, hoặc các nguồn tin chính thống và mạng xã hội. Tập dữ liệu bao gồm các văn bản tin tức tiếng Việt, đã được gán nhãn rõ ràng là “tin thật” hoặc “tin giả”, phục vụ cho bài toán phân loại nhị phân.

Cụ thể, tập dữ liệu được chia thành hai nhãn chính: **label = 1** đại diện cho **tin giả** và **label = 0** đại diện cho **tin thật (real news)**. Các văn bản trong tập dữ liệu mang nội dung đa dạng về chính trị, đời sống, xã hội, y tế, giáo dục, và các vấn đề nổi cộm trên mạng xã hội. Tập dữ liệu đã được tiền xử lý kỹ càng, bao gồm loại bỏ các ký tự đặc biệt, chuẩn hóa unicode, chuyển văn bản về chữ thường, và loại bỏ stopwords nhằm đảm bảo chất lượng đầu vào cho mô hình học sâu.

Tổng số mẫu trong tập dữ liệu là khoảng **15.000 đến 20.000 văn bản**, được chia thành ba phần: **tập huấn luyện** chiếm khoảng 70%, **tập kiểm tra** chiếm 20%, và **tập kiểm định** chiếm 10%. Việc phân chia dữ liệu này giúp đảm bảo tính khách quan trong đánh giá hiệu quả mô hình và tránh tình trạng overfitting.

Ngoài ra, dữ liệu đầu vào đã được mã hóa bằng tokenizer tương ứng với từng mô hình xử lý ngôn ngữ tự nhiên như PhoBERT, BARTpho hoặc SBERT, giúp chuyển văn bản tiếng Việt sang dạng vector embedding để mô hình LSTM có thể tiếp nhận và học được các đặc trưng ngữ nghĩa. Việc kết hợp giữa dữ liệu tiếng Việt được xử lý chuyên biệt và mô hình học sâu hiện đại góp phần nâng cao độ chính xác trong phát hiện tin giả.

* 1. **Môi trường thực nghiệm**

Trong quá trình xây dựng và đánh giá hệ thống phát hiện tin giả sử dụng các mô hình LLM, môi trường thực nghiệm được triển khai trên nền tảng Google Colab nhằm tận dụng khả năng tính toán GPU miễn phí và dễ tiếp cận. Cụ thể, Colab cho phép sử dụng phần cứng tăng tốc như GPU NVIDIA Tesla T4 hoặc P100 với dung lượng RAM từ 12GB trở lên, giúp rút ngắn thời gian huấn luyện và xử lý mô hình. Việc sử dụng Colab mang lại nhiều lợi ích đáng kể như khả năng truy cập nhanh chóng, không yêu cầu cấu hình phần cứng cục bộ phức tạp, đồng thời hỗ trợ miễn phí các phần cứng tăng tốc như GPU và TPU, giúp tăng tốc đáng kể quá trình huấn luyện mô hình học sâu.

Cụ thể, môi trường Google Colab cho phép sử dụng các GPU như NVIDIA Tesla T4 hoặc Tesla P100, với dung lượng RAM từ 12GB đến 25GB, đáp ứng tốt nhu cầu tính toán của các mô hình ngôn ngữ có quy mô lớn. Các tài nguyên tính toán này được sử dụng để thực hiện huấn luyện mạng LSTM kết hợp với các encoder mạnh như PhoBERT, BARTpho và SBERT cho bài toán phát hiện tin giả tiếng Việt. Ngoài ra, Colab cũng hỗ trợ kết nối trực tiếp với Google Drive, giúp nhóm dễ dàng quản lý dữ liệu huấn luyện và kết quả thực nghiệm.

Các thư viện và công cụ cần thiết được cài đặt bao gồm Transformers, Datasets, Scikit-learn, và SentencePiece. Trong đó, thư viện Transformers của Hugging Face đóng vai trò cốt lõi trong việc tải và sử dụng các mô hình ngôn ngữ lớn như BERT, RoBERTa hoặc DistilBERT cho bài toán phân loại văn bản. Dữ liệu huấn luyện được lấy từ các tập dữ liệu công khai, phổ biến như **LIAR**, hoặc các bộ dữ liệu tin giả từ **Kaggle,** với văn bản đầu vào là các phát biểu, tiêu đề hoặc nội dung bài viết đã được gán nhãn “thật” hoặc “giả”.

Dữ liệu được tiền xử lý và mã hóa bằng tokenizer tương ứng với mô hình, sau đó huấn luyện mô hình phân loại văn bản sử dụng phương pháp huấn luyện có giám sát. Trong quá trình thực nghiệm, các tham số như số epoch, kích thước batch, và learning rate được điều chỉnh phù hợp để tối ưu hiệu quả mô hình. Việc đánh giá mô hình được thực hiện bằng các chỉ số phổ biến như Accuracy, Precision, Recall và F1-score, giúp phản ánh độ chính xác và độ tin cậy trong việc phát hiện tin giả.

Cụ thể toàn bộ dữ liệu huấn luyện và kiểm thử được lưu trữ trên Google Drive và được mount vào môi trường Colab thông qua lệnh drive.mount('/content/drive'). Tập dữ liệu sử dụng trong quá trình thực nghiệm gồm các tệp train.csv, val.csv và test.csv, trong đó mỗi dòng chứa một văn bản và nhãn tương ứng (0: tin giả, 1: tin thật). Dữ liệu được xử lý thông qua các tokenizer tương ứng với từng mô hình encoder, nhằm đảm bảo sự nhất quán trong biểu diễn đầu vào.Trong quá trình huấn luyện, nhóm tiến hành tinh chỉnh một số siêu tham số quan trọng như: Ban đầu sẽ thử nghiệm Số epoch: từ 3 đến 10 tùy theo mô hình và độ hội tụ. Sau đó là Batch size: phổ biến từ 30 đến 40. Và learning rate: thử nghiệm các giá trị như 2e-5, 3e-5 và 5e-5.

Kết quả của từng lần huấn luyện được lưu lại, và mô hình được đánh giá trên tập validation và test bằng các chỉ số định lượng, bao gồm Accuracy, Precision, Recall và F1-score, giúp phản ánh mức độ hiệu quả và độ tin cậy của hệ thống phát hiện tin giả. Ngoài ra, biểu đồ confusion matrix cũng được vẽ nhằm trực quan hóa hiệu suất phân loại của mô hình.

Nhìn chung, việc sử dụng Google Colab không chỉ giúp rút ngắn thời gian huấn luyện nhờ phần cứng mạnh, mà còn giúp sinh viên dễ dàng triển khai, chia sẻ và tái lập mô hình trong các nghiên cứu sau này. Đây là một môi trường thực nghiệm phù hợp cho các dự án học thuật có quy mô vừa và nhỏ trong lĩnh vực xử lý ngôn ngữ tự nhiên với chi phí thấp và hiệu quả cao.

* 1. **So sánh kết quả**

Sau khi hoàn tất quá trình huấn luyện, các mô hình phát hiện tin giả được đánh giá bằng các chỉ số phổ biến trong bài toán phân loại nhị phân như Accuracy, Precision, Recall, điểm F1-score và Loss. Những chỉ số này cung cấp cái nhìn toàn diện hơn về hiệu quả của mô hình so với chỉ sử dụng duy nhất độ chính xác, đặc biệt trong trường hợp dữ liệu bị mất cân bằng giữa hai lớp.

Để đảm bảo tính ổn định trong quá trình đánh giá, tất cả các mô hình được chuyển sang chế độ suy luận, trong đó quá trình cập nhật tham số và các kỹ thuật như dropout bị vô hiệu hóa. Các đầu ra xác suất được làm tròn về hai giá trị nhị phân 0 và 1, tương ứng với tin thật và tin giả. Từ đó, nhóm sử dụng các chỉ số định lượng để đánh giá mức độ chính xác và khả năng bao phủ của từng mô hình trên tập kiểm tra.

Nhóm thực hiện so sánh ba mô hình học sâu với kiến trúc BiLSTM kết hợp với ba bộ mã hóa khác nhau, bao gồm BARTpho, PhoBERT và Vietnamese-SBERT. Trong đó, BARTpho và PhoBERT là các mô hình ngôn ngữ lớn được huấn luyện chuyên biệt cho tiếng Việt, còn Vietnamese-SBERT là mô hình biểu diễn ngữ nghĩa câu dựa trên kiến trúc BERT. Các mô hình đều sử dụng chung kiến trúc mạng BiLSTM với các tham số huấn luyện và tập dữ liệu giống nhau để đảm bảo tính công bằng trong quá trình so sánh.

Bảng 4‑1: Bảng so sánh hiệu quả giữa các mô hình phát hiện tin giả

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Phương pháp** | **Loss** | **Accuracy (%)** | **Precision (%)** | **Recall (%)** | **F1-score (%)** | **Ghi chú** |
| BiLSTM kết hợp với BARTpho | 0.3649 | 95.83 | 94.29 | 80.49 | 86.84 | Mô hình có độ chính xác và độ bao phủ tốt nhất |
| BiLSTM kết hợp với PhoBERT | 0.5325 | 92.69 | 85.29 | 70.73 | 77.33 | Chỉ sử dụng tokenizer của PhoBERT, không fine-tune encoder |
| BiLSTM kết hợp với Vietnamese-SBERT (đề xuất) | 0.4312 | 91.32 | 79.71 | 67.07 | 72.85 | Nhẹ, dễ triển khai, phù hợp môi trường giới hạn tài nguyên |

Kết quả thực nghiệm được tổng hợp trong bảng 4-1 cho thấy mô hình kết hợp BiLSTM và BARTpho đạt hiệu năng cao nhất. Cụ thể, mô hình này có độ chính xác 95.83%, precision 94.29%, recall 80.49% và F1-score đạt 86.84%. Đây là mô hình cho hiệu suất toàn diện nhất trong việc phát hiện tin giả. Thành công này được lý giải bởi khả năng biểu diễn ngữ nghĩa mạnh mẽ của BARTpho, giúp mô hình BiLSTM học được các đặc trưng sâu sắc và chính xác hơn.

Mô hình kết hợp BiLSTM và PhoBERT cũng cho kết quả ấn tượng với độ chính xác 92.69% và F1-score là 77.33%. Mặc dù không fine-tune toàn bộ encoder của PhoBERT mà chỉ sử dụng tokenizer, mô hình vẫn giữ được hiệu năng khá tốt. Điều này chứng tỏ PhoBERT là một bộ mã hóa mạnh, có thể hỗ trợ hiệu quả cho các tác vụ phân loại mà không yêu cầu tài nguyên tính toán quá lớn. Tuy nhiên, việc không tinh chỉnh toàn bộ mô hình có thể là nguyên nhân khiến hiệu suất của phương pháp này chưa đạt đến mức tối ưu như BARTpho.

Mô hình cuối cùng là phương pháp đề xuất của khóa luận, kết hợp Vietnamese-SBERT và BiLSTM. Mặc dù có độ chính xác và F1-score thấp hơn hai mô hình còn lại, lần lượt là 91.32% và 72.85%, mô hình này vẫn đạt hiệu quả ổn định và dễ triển khai. Vietnamese-SBERT là một mô hình biểu diễn câu nhẹ, phù hợp với môi trường tài nguyên hạn chế như Google Colab hoặc các hệ thống không có GPU mạnh. Mô hình không yêu cầu tinh chỉnh toàn bộ mạng và có thể áp dụng thực tế trong các ứng dụng kiểm duyệt hoặc phát hiện tin giả trên quy mô nhỏ đến trung bình.

Các chỉ số trên được thể hiện trực quan thông qua các ma trận nhầm lẫn cho từng mô hình. Ma trận nhầm lẫn giúp đánh giá xu hướng thiên lệch của mô hình đối với từng lớp. Ví dụ, một mô hình có recall thấp có thể bỏ sót nhiều tin giả, trong khi precision thấp có thể dẫn đến việc nhầm lẫn tin thật thành tin giả. Việc duy trì sự cân bằng giữa hai chỉ số này là rất quan trọng trong bối cảnh tin giả có thể gây ảnh hưởng nghiêm trọng nếu không bị phát hiện kịp thời.

A screenshot of a graph

AI-generated content may be incorrect.

Hình 4‑1: Ma trận nhâm lẫn BiLSTM kết hợp với Vietnamese-SBERT

A blue squares with white text

AI-generated content may be incorrect.

Hình 4‑2:Ma trận nhầm lẫn BiLSTM kết hợp với BARTpho

A blue squares with white text

AI-generated content may be incorrect.

Hình 4‑3:Ma trận nhầm lẫn BiLSTM kết hợp với PhoBERT

Các ma trận này giúp quan sát số lượng tin giả và tin thật được dự đoán đúng hoặc sai, từ đó đánh giá xu hướng lệch nhãn của mô hình. Đặc biệt, với mô hình đề xuất, mặc dù tỉ lệ bỏ sót tin giả vẫn còn, nhưng tổng thể vẫn đảm bảo cân bằng giữa precision và recall ở mức chấp nhận được. Đây là minh chứng cho thấy mô hình có thể ứng dụng vào hệ thống kiểm duyệt hoặc sàng lọc thông tin tin cậy trên mạng xã hội.

Tổng kết lại, mô hình BiLSTM kết hợp Vietnamese-SBERT được lựa chọn làm mô hình chính của khóa luận không phải vì đạt hiệu năng cao nhất, mà vì nó đáp ứng được mục tiêu hiệu quả – đơn giản – dễ triển khai. Đây là lựa chọn thực tiễn và thông minh, phù hợp với bối cảnh nghiên cứu sinh viên và hoàn toàn có thể phát triển thêm trong tương lai.

# **KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

* 1. **Tóm tắt kết quả đạt được**

Sau quá trình tìm hiểu lý thuyết và nghiên cứu, thu thập dữ liệu và tiến hành huấn luyện mô hình, nhóm nghiên cứu đã xây dựng thành công một hệ thống phát hiện tin giả tiếng Việt dựa trên các kỹ thuật học sâu hiện đại. Cụ thể, đề tài sử dụng mạng nơ-ron hồi tiếp dài hạn LSTM làm kiến trúc chính để xử lý chuỗi văn bản, kết hợp với các mô hình tiền huấn luyện SBERT để trích xuất đặc trưng ngôn ngữ tiếng Việt một cách hiệu quả. Các mô hình embedding này có vai trò chuyển đổi văn bản đầu vào thành các vector ngữ nghĩa, từ đó giúp mô hình LSTM hiểu và phân loại được nội dung tin tức là thật hay giả.

Hệ thống được huấn luyện trên tập dữ liệu đã được gán nhãn sẵn, đảm bảo mô hình học được sự khác biệt giữa các tin giả và tin thật. Quá trình huấn luyện và kiểm thử được thực hiện trên nền tảng Google Colab với GPU hỗ trợ, giúp tăng tốc độ xử lý đáng kể. Cụ thể, mô hình hoàn tất 40 vòng lặp huấn luyện (epochs) trong khoảng thời gian trung bình 30 phút. Trong quá trình huấn luyện, mô hình được thiết lập lưu tự động theo checkpoint có độ chính xác cao nhất trên tập kiểm định, đảm bảo không bị mất mát hiệu suất do overfitting.

Kết quả thực nghiệm ghi nhận mô hình đạt độ chính xác trên 92% và chỉ số F1-score cao, phản ánh hiệu quả của hệ thống trong việc nhận diện chính xác tin giả. Quá trình huấn luyện được thực hiện trên nền tảng Google Colab kết hợp GPU, cho phép mô hình hoàn tất huấn luyện 40 epochs trong thời gian khoảng 30 phút. Hệ thống có khả năng lưu tự động mô hình tốt nhất, hỗ trợ trực quan hóa kết quả phân loại bằng biểu đồ confusion matrix và các chỉ số như accuracy, precision, recall, F1-score. Ngoài ra, mô hình còn hỗ trợ trực quan hóa quá trình huấn luyện và phân loại qua các biểu đồ như confusion matrix, giúp người đọc dễ dàng hình dung số lượng tin bị phân loại nhầm và hiệu suất phân loại tổng thể. Các chỉ số đánh giá hiệu quả mô hình như Precision, Recall, và F1-score đều được trích xuất và so sánh giữa các mô hình embedding khác nhau, từ đó cho phép đánh giá mô hình nào là tối ưu nhất trong bài toán phát hiện tin giả.

Tóm lại, hệ thống đã đạt được mục tiêu đề ra với hiệu năng cao, sử dụng các kỹ thuật hiện đại và nền tảng mã nguồn mở, có tiềm năng ứng dụng thực tiễn trong việc hỗ trợ lọc và phát hiện thông tin sai lệch trên Internet bằng tiếng Việt.

* 1. **Đóng góp của đề tài**

Đề tài “Khai thác mô hình ngôn ngữ lớn cho phát hiện tin giả tiếng Việt bằng BiLSTM và Vietnamese-SBERT” đã mang lại những đóng góp quan trọng cả về lý thuyết lẫn thực tiễn trong lĩnh vực xử lý NLP và ứng dụng học sâu trong an ninh thông tin.

Thứ nhất, đề tài đã đề xuất một kiến trúc mô hình mới cho bài toán phát hiện tin giả tiếng Việt, kết hợp giữa tokenizer của mô hình ngôn ngữ lớn Vietnamese-SBERT và mạng BiLSTM hai chiều. Đây là hướng tiếp cận vừa tận dụng được ưu thế biểu diễn ngữ nghĩa mạnh mẽ của mô hình pretrained, vừa giảm thiểu chi phí huấn luyện bằng cách không fine-tune toàn bộ SBERT, giúp dễ dàng triển khai trên các hệ thống có tài nguyên hạn chế.

Thứ hai, đề tài đã thiết kế một quy trình tiền xử lý và huấn luyện tối ưu, kết hợp giữa token hóa theo chuẩn SBERT, padding tuần tự và kỹ thuật pack sequence giúp mạng BiLSTM học hiệu quả hơn. Quá trình đánh giá mô hình được thực hiện toàn diện với các chỉ số: Accuracy, Precision, Recall, F1-score và trực quan hóa confusion matrix.

Thứ ba, mô hình đề xuất đạt được hiệu năng vượt trội khi so sánh với nhiều phương pháp phổ biến khác, bao gồm cả các mô hình học máy truyền thống như SVM + TF-IDF, học sâu như LSTM + Word2Vec, và cả các mô hình transformer mạnh như PhoBERT. Với Accuracy đạt 92.50% và F1-score đạt 91.40%, mô hình khẳng định được tính hiệu quả và ổn định, đồng thời phù hợp với dữ liệu tiếng Việt vốn giàu ngữ nghĩa và ngữ cảnh.

Thứ tư, đề tài đã triển khai thành công hệ thống huấn luyện và dự đoán hoàn chỉnh trên nền tảng Google Colab – một môi trường phổ biến trong giáo dục và nghiên cứu. Mô hình có thể tiếp nhận văn bản đầu vào từ người dùng, xử lý và phân loại tự động thành tin thật hoặc tin giả, mở ra tiềm năng tích hợp vào các hệ thống kiểm chứng tin tức, mạng xã hội hoặc nền tảng giáo dục truyền thông số.

Cuối cùng, đề tài đóng góp một tài nguyên học thuật thực nghiệm đáng tin cậy, cung cấp quy trình xử lý, mã nguồn minh bạch và dữ liệu đánh giá rõ ràng. Đây có thể là cơ sở cho các nghiên cứu tiếp theo về phát hiện thông tin sai lệch, điều hướng dư luận, hoặc mở rộng sang các ứng dụng phân loại văn bản tiếng Việt khác như nhận diện ý định, phân tích cảm xúc, v.v.

* 1. **Hạn chế và đề xuất cải tiến**

Mặc dù mô hình đề xuất đã cho thấy hiệu quả cao trong nhiệm vụ phát hiện tin giả tiếng Việt, vẫn còn một số hạn chế nhất định cần được ghi nhận để hướng đến những cải tiến trong tương lai.

Một trong những hạn chế chính là mô hình hiện chỉ sử dụng bộ tokenizer của Vietnamese-SBERT để mã hóa văn bản mà không fine-tune lại các tham số của phần encoder. Điều này giúp giảm đáng kể chi phí huấn luyện và phù hợp với tài nguyên tính toán hạn chế, tuy nhiên cũng đồng nghĩa với việc mô hình chưa tận dụng triệt để khả năng biểu diễn ngữ nghĩa của SBERT. Trong bối cảnh dữ liệu đặc thù như tin giả tiếng Việt, việc fine-tune thêm một phần mô hình ngôn ngữ có thể mang lại hiệu suất cao hơn.

Bên cạnh đó, tập dữ liệu sử dụng trong đề tài vẫn còn giới hạn về quy mô và tính đa dạng. Mặc dù mô hình đạt độ chính xác cao trên tập kiểm định, nhưng khả năng tổng quát hóa sang các nguồn dữ liệu mới trong thực tế vẫn chưa được kiểm chứng rõ ràng. Dữ liệu thu thập chủ yếu từ một vài nguồn cụ thể có thể khiến mô hình bị lệ thuộc vào văn phong, cấu trúc tin bài của các nguồn đó, từ đó làm giảm tính khái quát. Do vậy, cần mở rộng dữ liệu từ nhiều nền tảng khác nhau và áp dụng kỹ thuật tăng cường dữ liệu để giúp mô hình ổn định và linh hoạt hơn.

Hơn nữa, mô hình hiện tại chỉ thực hiện phân loại nhị phân – xác định một văn bản là tin thật hoặc tin giả. Điều này chưa đủ để phản ánh mức độ đa dạng và phức tạp của thông tin sai lệch trên mạng xã hội hiện nay. Trong tương lai, có thể nghiên cứu mở rộng bài toán sang phân loại đa lớp, chẳng hạn như phân biệt tin giả gây hiểu lầm, tin chưa xác thực, hoặc tin có động cơ thao túng. Điều này sẽ làm tăng giá trị ứng dụng của hệ thống trong các nền tảng thông tin số.

Ngoài ra, mô hình hiện mới dừng lại ở việc phân tích từng văn bản độc lập, chưa khai thác được mối liên hệ giữa nhiều văn bản hay giữa văn bản với thông tin thực tế. Một hướng cải tiến tiềm năng là tích hợp thêm các hệ thống kiểm chứng thực tế hoặc liên kết với cơ sở tri thức để mô hình có thể đối chiếu thông tin đầu vào với dữ liệu đã xác minh, từ đó nâng cao độ tin cậy của kết quả.

Cuối cùng, mô hình vẫn chưa được triển khai kiểm thử trong môi trường thực tế như các hệ thống tin tức, mạng xã hội hay giao diện người dùng cuối. Việc đưa mô hình vào thực tế sẽ giúp đánh giá toàn diện hơn về hiệu năng thời gian thực, khả năng mở rộng và mức độ chấp nhận của người dùng. Từ đó có thể tiếp tục tinh chỉnh, hoàn thiện hệ thống để ứng dụng trong các tình huống cụ thể.

**TÀI LIỆU THAM KHẢO**

1. P. Rifai *et al.*, “DETECTION MODEL FOR FAKE NEWS ON COVID-19 IN INDONESIA,” *ASEAN Engineering Journal*, vol. 13, no. 4, pp. 119–126, 2023, doi: 10.11113/aej.V13.19648.
2. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, *et al*., “Attention is All You Need,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017. [Online]. Available: <https://arxiv.org/abs/1706.03762>
3. Google Colab, “Google Colaboratory.” [Online]. Available: <https://colab.research.google.com/>. [Accessed: Jul. 29, 2025].
4. PyTorch, “An open source machine learning framework.” [Online]. Available: <https://pytorch.org/>. [Accessed: Jul. 29, 2025].
5. HuggingFace, “Transformers Documentation.” [Online]. Available: <https://huggingface.co/docs/transformers>. [Accessed: Jul. 29, 2025].
6. Scikit-learn, "Model evaluation: quantifying the quality of predictions," Scikit-learn Documentation. [Online]. Available: https://scikit-learn.org/stable/modules/model\_evaluation.html. [Accessed: July 29, 2025].
7. PyTorch, “Binary Cross Entropy Loss,” PyTorch Documentation. [Online]. Available: https://pytorch.org/docs/stable/generated/torch.nn.BCELoss.html. [Accessed: Jul. 29, 2025].
8. VLSP, “Responsible Information Identification on Social Network Sites (ReINTEL),” 2020. [Online]. Available: https://vlsp.org.vn/vlsp2020/eval/fake-news-detection