HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN 1

ĐỀ CƯƠNG THỰC TẬP CƠ SỞ

Đề tài:
XÂY DỰNG HỆ THỐNG MLOPS END-TO-END CHO BÀI TOÁN PHÂN LOẠI TIN TỨC TIẾNG ANH

Giảng viên hướng dẫn: ThS. Kim Ngọc Bách
Sinh viên thực hiện: Nguyễn Hồng Quang
Mã sinh viên: B23DCVT361
Lớp: D23CQCE04 - B

Hà Nội, Tháng 2 năm 2026

Mục lục

1 GIỚI THIỆU DỰ ÁN

1.1 Lý do chọn đề tài

Trong bối cảnh bùng nổ thông tin toàn cầu, việc xử lý và phân loại tự động hàng triệu bài báo tiếng Anh mỗi ngày là nhu cầu cấp thiết của các doanh nghiệp và tổ chức truyền thông. Các phương pháp truyền thống dựa trên luật (Rule-based) hoặc thống kê cơ bản không còn đáp ứng được yêu cầu về độ chính xác và khả năng hiểu ngữ cảnh. Sự ra đời của các mô hình ngôn ngữ lớn (LLMs) dựa trên kiến trúc Transformer, điển hình là BERT, đã mở ra kỷ nguyên mới cho Xử lý ngôn ngữ tự nhiên (NLP).

Tuy nhiên, khoảng cách từ một mô hình thí nghiệm đến một sản phẩm thực tế là rất lớn. Các dự án sinh viên thường chỉ dừng lại ở bước huấn luyện mô hình (Model Training) mà bỏ qua khâu triển khai (Deployment) và vận hành (Operations). Xuất phát từ thực tế đó, em lựa chọn đề tài **"Xây dựng hệ thống MLOps End-to-End cho bài toán Phân loại tin tức Tiếng Anh"**, tập trung vào việc áp dụng quy trình chuẩn công nghiệp để đưa mô hình AI vào ứng dụng thực tiễn.

1.2 Ý nghĩa thực tiễn

- **Tự động hóa quy trình nghiệp vụ:** Hệ thống giúp tự động gán nhãn chủ đề (Thể thao, Kinh doanh, Công nghệ, Thế giới) cho các luồng tin tức đầu vào, giảm thiểu sức lao động thủ công.
- **Mô hình triển khai chuẩn:** Cung cấp một kiến trúc tham chiếu (Reference Architecture) cho việc xây dựng các ứng dụng AI theo mô hình Client-Server, tách biệt giữa việc huấn luyện và phục vụ người dùng.
- **Tối ưu hóa hiệu năng:** Sử dụng mô hình DistilBERT giúp cân bằng giữa độ chính xác cao và tốc độ phản hồi nhanh, phù hợp với tài nguyên phần cứng hạn chế.

1.3 Giá trị học thuật

Đề tài đóng góp các giá trị nghiên cứu và kỹ thuật cụ thể:

- **Ứng dụng Kỹ thuật Transfer Learning:** Chứng minh hiệu quả của việc tinh chỉnh (Fine-tuning) mô hình ngôn ngữ tiền huấn luyện (DistilBERT) trên miền dữ liệu cụ thể (AG News Dataset) thay vì huấn luyện từ đầu.
- **Kiến trúc MLOps hiện đại:** Xây dựng quy trình khép kín từ Dữ liệu (Data Engineering) $\rightarrow$ Huấn luyện (Model Engineering) $\rightarrow$ Triển khai (Deployment), áp dụng các nguyên lý CI/CD cho Machine Learning.
- **Phân tích thực nghiệm:** Đánh giá độ chính xác và hàm mất mát (Loss Function) qua các Epochs để hiểu rõ quá trình hội tụ của mô hình Deep Learning.

2 CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ

2.1 Cơ sở lý thuyết

2.1.1 Kiến trúc Transformer và Cơ chế Attention

Transformer [1] là nền tảng của các mô hình NLP hiện đại. Khác với RNN xử lý tuần tự, Transformer sử dụng cơ chế *Self-Attention* để xử lý toàn bộ câu cùng lúc, cho phép mô hình nắm bắt mối quan hệ ngữ nghĩa giữa các từ ở khoảng cách xa. Công thức tính Attention được định nghĩa là:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

Trong đó Q, K, V là các ma trận Query, Key, Value biểu diễn từ vựng.

2.1.2 DistilBERT và Knowledge Distillation

DistilBERT (Distilled BERT) [3] là phiên bản rút gọn của BERT. Nó sử dụng kỹ thuật *Knowledge Distillation* (Chưng cất tri thức), trong đó một mô hình nhỏ (Student) được huấn luyện để mô phỏng hành vi của mô hình lớn (Teacher).

- **Ưu điểm:** Giảm 40% kích thước tham số, tăng 60% tốc độ so với BERT gốc, nhưng vẫn giữ được 97% độ chính xác.
- **Lý do lựa chọn:** Phù hợp với bài toán yêu cầu phản hồi nhanh (Real-time inference) và môi trường thực tập có tài nguyên tính toán giới hạn.

2.1.3 Quy trình MLOps (Machine Learning Operations)

MLOps là sự kết hợp giữa Machine Learning, DevOps và Data Engineering nhằm mục đích triển khai và duy trì các hệ thống AI một cách tin cậy và hiệu quả [4]. Dự án tập trung vào mức độ *MLOps Level 1*: Pipeline Automation (Tự động hóa đường ống huấn luyện).

2.2 Công nghệ sử dụng

Thành phần	Công nghệ / Thư viện
Ngôn ngữ lập trình	Python 3.9+
Deep Learning Framework	**PyTorch**, Hugging Face Transformers
Mô hình lõi	**DistilBERT** (pre-trained: `distilbert-base-uncased`)
Bộ dữ liệu	**AG News** (News Topic Classification Dataset)
Backend API	**FastAPI** (High performance web framework)
Frontend Interface	**Streamlit** (Rapid prototyping UI)
Môi trường thực nghiệm	Anaconda / Virtualenv

Bảng 1: Danh sách công nghệ sử dụng trong dự án

3 PHÂN TÍCH YÊU CẦU DỰ ÁN

3.1 Yêu cầu chức năng

1. **Thu thập và Xử lý dữ liệu:** Hệ thống tự động tải bộ dữ liệu AG News, thực hiện tiền xử lý (làm sạch text, chuyển chữ thường, loại bỏ ký tự đặc biệt).
2. **Huấn luyện mô hình:** Thực hiện Fine-tuning mô hình DistilBERT để phân loại văn bản vào 4 nhóm: World, Sports, Business, Sci/Tech.
3. **Cung cấp API:** Xây dựng RESTful API cho phép các ứng dụng khác gửi văn bản và nhận về kết quả dự đoán kèm độ tin cậy.
4. **Giao diện người dùng:** Cung cấp giao diện Web trực quan để người dùng nhập liệu và xem kết quả phân tích.

3.2 Yêu cầu phi chức năng

- **Độ chính xác:** Mô hình đạt độ chính xác (Accuracy) trên tập kiểm thử $\geq 90\%$.
- **Hiệu năng:** Thời gian phản hồi API (Inference time) dưới 500ms cho một đoạn văn bản trung bình.
- **Tính tái lập (Reproducibility):** Mã nguồn được tổ chức module hóa, đảm bảo chạy ổn định trên các môi trường khác nhau.

4 KẾ HOẠCH THỰC HIỆN CHI TIẾT

Dự án được chia thành 4 giai đoạn chính, thực hiện theo mô hình phát triển phần mềm linh hoạt (Agile):

Giai đoạn 1: Data Engineering (Tuần 1-2)

- **Công việc:** Thiết lập môi trường, tải bộ dữ liệu AG News từ thư viện `datasets`.
- **Kỹ thuật:** Phân tích dữ liệu (EDA), xây dựng pipeline làm sạch dữ liệu (Text Cleaning, Lowercasing).
- **Kết quả:** Bộ dữ liệu sạch (`news_clean.csv`) sẵn sàng cho huấn luyện.

Giai đoạn 2: Model Engineering (Tuần 3-4)

- **Công việc:** Xây dựng Training Pipeline sử dụng `Trainer` API của Hugging Face.
- **Kỹ thuật:** Tokenization dữ liệu, Fine-tuning DistilBERT, Tối ưu hóa siêu tham số (Learning rate, Batch size, Epochs).
- **Kết quả:** File mô hình (`pytorch_model.bin`) đạt độ chính xác mục tiêu (Loss thấp).

Giai đoạn 3: Deployment & App Building (Tuần 5-6)

- **Công việc:** Xây dựng Backend API và Frontend Demo.
- **Kỹ thuật:**
  - Viết API endpoint `/predict` với FastAPI.
  - Xây dựng giao diện nhập liệu và hiển thị kết quả với Streamlit.
  - Tích hợp mô hình đã train vào hệ thống thực.
- **Kết quả:** Hệ thống Web App hoạt động hoàn chỉnh (End-to-End).

Giai đoạn 4: Documentation & Reporting (Tuần 7-8)

- **Công việc:** Tổng hợp kết quả, đánh giá hiệu năng, viết báo cáo.
- **Kết quả:** Báo cáo thực tập chi tiết, Slide thuyết trình, Video demo sản phẩm.

Tài liệu

[1] A. Vaswani *et al.*, ”Attention is all you need,”in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[2] J. Devlin, M. Chang, K. Lee, and K. Toutanova, ”BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,”*arXiv preprint arXiv:1810.04805*, 2018.

[3] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, ”DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter,”*arXiv preprint arXiv:1910.01108*, 2019.

[4] D. Kreuzberger, N. Kühl, and S. Hirschl, ”Machine Learning Operations (MLOps): Overview, Definition, and Architecture,”*IEEE Access*, vol. 11, pp. 31866-31879, 2023.

[5] S. Ramirez, ”FastAPI,”`https://fastapi.tiangolo.com`, 2018.