Báo cáo project

Sinh viên thực hiện: Đỗ Trần Sáng

Lớp: 22KDL

Ngày: 12 tháng 6 năm 2025

1 Giới thiệu

1.1 Bối cảnh và vấn đề

Trong thời đại số, tin giả lan truyền nhanh chóng trên các nền tảng trực tuyến, gây ra nhiều hệ lụy nghiêm trọng. Đồ án này nhằm xây dựng một hệ thống tự động phân biệt tin thật và tin giả dựa trên nội dung văn bản, sử dụng các kỹ thuật Xử lý Ngôn ngữ Tự nhiên (NLP) và Học máy. Dự án tuân theo quy trình khoa học dữ liệu, từ chuẩn bị dữ liệu đến đánh giá mô hình.

1.2 Mục tiêu

Mục tiêu chính là phát triển hệ thống phân loại bài báo thành "Tin thật" hoặc "Tin giả" với độ chính xác cao, thông qua hai phương pháp:

- **Học máy cổ điển:** Sử dụng Support Vector Machine (SVM) kết hợp đặc trưng TF-IDF.
- **Học sâu:** Áp dụng mô hình BERT (bert-base-uncased) để khai thác ngữ nghĩa sâu sắc.

1.3 Dữ liệu

Bộ dữ liệu gồm các bài báo từ nguồn uy tín (Reuters, The New York Times,...) và các trang tin giả, cực đoan. Sau khi làm sạch, bộ dữ liệu cuối cùng có 68,380 bài báo, với nhãn cân bằng (50.5% tin thật, 49.5% tin giả).

2 Tiền xử lý và phân tích dữ liệu

2.1 Làm sạch dữ liệu

Quy trình làm sạch bao gồm:

- Gán nhãn: Tin thật (1), tin giả (0).
- Xử lý dữ liệu thiếu và trùng lặp: Loại bỏ 29 giá trị thiếu và 10,012 bản ghi trùng (448 tin thật, 9,564 tin giả).
- Kết hợp dữ liệu thành một tập duy nhất.
- Tạo đặc trưng: Thêm cột `word_count` (số từ) và `text_length` (độ dài văn bản).
- Lọc nhiễu: Loại bỏ bài báo dưới 6 từ.

2.2 Phân tích dữ liệu khám phá (EDA)

- **Phân phối nhãn:** Dữ liệu cân bằng, không cần xử lý mất cân bằng.
- **Độ dài văn bản:** Tin thật dài hơn tin giả (537 từ so với 429 từ, p-value ≈ 0 từ T-test), cho thấy `word_count` là đặc trưng hữu ích nhưng chưa đủ để phân loại chính xác.

Kết luận: Dữ liệu sạch, cân bằng, nhưng cần phân tích nội dung sâu hơn bằng TF-IDF và BERT.

3 Xây dựng và huấn luyện mô hình

3.1 Phương pháp 1: Support Vector Machine (SVM)

- **Đặc trưng:** Kết hợp TF-IDF (`max_features=10000`, `ngram_range=(1,2)`) và đặc trưng số (`word_count`, `char_count`) được chuẩn hóa.
- **Huấn luyện:** Sử dụng GridSearchCV để tìm siêu tham số tối ưu (`kernel='rbf'`, `C=10.0`, `gamma='scale'`).
- **Kết quả trên tập kiểm thử (20%):**

	Precision	Recall	F1-score	Support
Tin giả	0.95	0.95	0.95	6773
Tin thật	0.95	0.95	0.95	6903
Accuracy			0.95	13676

Bảng 1: Kết quả mô hình SVM

Mô hình SVM đạt hiệu suất cao và cân bằng, với F1-score 0.95 cho cả hai lớp.

3.2 Phương pháp 2: BERT (bert-base-uncased)

- **Kiến trúc:** Sử dụng `transformers` và `torch` để token hóa, tạo DataLoader và huấn luyện.
- **Huấn luyện:** 3 epochs, `batch_size=16`, `learning_rate=3e-5`. Loss giảm từ 0.0992 xuống 0.0044.
- **Kết quả trên tập kiểm thử (20%):**

BERT vượt trội với F1-score 0.98, nhờ khả năng nắm bắt ngữ nghĩa phức tạp.

	Precision	Recall	F1-score	Support
Tin giả	0.98	0.98	0.98	6746
Tin thật	0.98	0.98	0.98	6975
Accuracy			0.98	13721

Bảng 2: Kết quả mô hình BERT

4 Kiểm thử trên dữ liệu mới

Kiểm thử trên 10 bài báo mới (5 thật, 5 giả):

- **SVM:** Độ chính xác 60%, phân loại sai 4/5 tin thật thành tin giả, cho thấy khả năng tổng quát hóa hạn chế.
- **BERT:** Độ chính xác 90%, chỉ sai 1/10 bài (tin thật về NASA), thể hiện khả năng hiểu ngữ cảnh vượt trội.

5 Kết luận và hướng phát triển

5.1 Tổng kết

Cả SVM (F1-score 0.95) và BERT (F1-score 0.98) đều đạt hiệu quả cao trên tập kiểm thử. Tuy nhiên, BERT vượt trội trong kiểm thử định tính nhờ khả năng tổng quát hóa tốt hơn, phù hợp với các bài toán NLP phức tạp.

5.2 Hạn chế

- SVM phụ thuộc nhiều vào dữ liệu huấn luyện, dễ bị overfitting.
- BERT đòi hỏi tài nguyên tính toán lớn (GPU, thời gian).

5.3 Hướng phát triển

- Cải thiện SVM bằng embedding (Word2Vec, GloVe).
- Thử nghiệm các mô hình lớn hơn như RoBERTa, DeBERTa.
- Áp dụng LIME/SHAP để diễn giải quyết định của BERT.
- Mở rộng dữ liệu với nhiều nguồn và chủ đề.
- Triển khai BERT thành API hoặc ứng dụng web để phân loại tin tức theo thời gian thực.