ĐẠI HỌC QUỐC GIA TPHCM

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

Báo cáo về dữ liệu

Đề tài: Hệ thống gắn nhãn bài báo tự động

Môn học: Nhập môn học máy

Sinh viên thực hiện:

Tô Hữu Danh (23127336)

Nguyễn Đăng Hưng (23127050)

Lê Phú Cường (23127164)

Nguyễn Bá Đăng Khoa (23127392)

Giáo viên hướng dẫn:

TS. Bùi Tiến Lên

Ngày 22 tháng 11 năm 2025

Mục lục

Danh sách bảng

Danh sách hình vẽ

1 Giới thiệu

Trong kỷ nguyên số hoá hiện nay, khối lượng thông tin được tạo ra hàng ngày trên các phương tiện truyền thông trực tiếp đang tăng trưởng theo cấp số nhân. Đối với các toà soạn báo điện tử và các hệ thống tổng hợp tin tức tại Việt Nam, việc quản lý, tổ chức và phân phối nội dung đến đúng đối tượng độc giả là một thách thức lớn. Các phương pháp phân loại thủ công truyền thống không còn đáp ứng được yêu cầu về tốc độ và khả năng mở rộng, đồng thời dễ xảy ra sai sót do yếu tố chủ quan của con người. Hơn nữa, cách mà đọc giả tiếp cận thông tin hiện đại đã chuyển từ việc đọc thụ động theo chuyên mục sang tìm kiếm chủ động theo từ khoá và chủ đề.

Xuất phát từ nhu cầu đó, đồ án tập trung nghiên cứu và xây dựng một hệ thống phân loại văn bản tự động cho tin tức tiếng Việt. Mục tiêu cốt lõi là phát triển một giải pháp có khả năng gán nhãn chủ đề chính xác cho bài báo dựa trên nội dung văn bản, giúp tối ưu hoá trải nghiệm người dùng và hỗ trợ quá trình biên tập nội dung.

Để giải quyết bài toán phân loại văn bản tiếng Việt đa lớp (*Multi-class Text Classification*), đồ án thực hiện khảo sát và triển khai so sánh hiệu năng trên ba hướng tiếp cận đại diện cho các giai đoạn phát triển của xử lý ngôn ngữ tự nhiên (*Natural Language Processing*):

- **Mô hình Học máy truyền thống (Baseline):** sử dụng phương pháp biểu diễn văn bản **TF-IDF** (*Term Frequency-Inverse Document Frequency*) kết hợp với thuật toán **Support Vector Machine**. Đây là phương pháp nền tảng, giúp thiết lập mức chuẩn về độ chính xác thời gian huấn luyện.

- **Mô hình Học sâu (Deep Learning):** Sử dụng **FastText**, một thư viện được phát triển bởi Facebook AI Research, FastText có ưu điểm vượt trội về tốc độ huấn luyện và khả năng xử lý tốt các từ hiểm (*out-of-vocabulary*) nhờ vào việc sử dụng thông tin n-gram mức thứ tự, phù hợp với đặc điểm hình thái của tiếng Việt.

- **Mô hình Ngôn ngữ Tiền huấn luyện (State-of-the-art):** Ứng dụng **PhoBERT**, một mô hình dựa trên kiến trúc Transformer (tương tự RoBERTa) được huấn luyện chuyên biệt trên dữ liệu tiếng Việt quy mô lớn. PhoBERT tận dụng cơ chế Attention để nắm bắt ngữ cảnh sâu sắc của từ ngữ, hứa hẹn mang lại độ chính xác cao nhất.

2 Phương pháp và nguồn thu thập dữ liệu

2.1 Nguồn dữ liệu

Tập dữ liệu của nhóm được thu thập thông qua việc cào hơn 20,000 bài báo từ trang báo Thanh Niên (https://thanhnien.vn). Thông tin mỗi bài báo bao gồm đường link, tiêu đề, ngày đăng, nội dung, nhãn chính và các nhãn phụ nếu có.
Các bài báo được cào phân bổ đều theo chủ đề chính được gắn cho bài báo đó, gồm 7 tag chính: Thể thao, Thời sự, Chính trị, Kinh tế, Giáo dục, Sức khoẻ, Thế giới.

2.2 Phương pháp thu thập dữ liệu

Để thu thập được số lượng lớn bài báo mà vẫn đảm bảo phân loại chính xác chủ đề, nhóm cần giải quyết được 3 vấn đề chính:

- Khả năng mở rộng nguồn dữ liệu, tức là tự động hoá việc lấy thêm đường link các bài báo.
- Phân tích cấu trúc HTML của một bài báo và trích xuất thông tin như tag, tiêu đề, nội dung,...
- Tổ chức lưu trữ dữ liệu đã thu thập được sao cho dễ truy xuất và mở rộng.

2.2.1 Tìm kiếm khả năng mở rộng nguồn dữ liệu

Hầu hết các trang báo mạng sẽ có các chuyên mục để người dùng có thể đọc thể loại mình muốn. Bằng việc giới hạn phạm vi thu thập trong các chuyên mục định sẵn, nhóm sẽ giải quyết được bài toán gán nhãn dữ liêu và giảm bớt công việc trong quá trình làm sạch.
Một vấn đề với cách tiếp cận này các cơ chế phân trang truyền thống trên VnExpress hay Dân Trí thường áp đặt giới hạn cứng ở mức 20-30 trang lịch sử, gây khó khăn cho việc xây dựng tập dữ liệu lớn.
Tuy nhiên, thông qua kỹ thuật phân tích gói tin HTTP bằng công cụ `Burp Suite`, nhóm đã xác định được cơ chế truy xuất dữ liệu thông qua API ẩn của báo Thanh Niên. Khi nhấn vào nút "Xem Thêm", thay vì render lại toàn bộ trang web với nội dung mới, client sử dụng endpoint `GET timelinelist/{id}/{page_number}.htm` để trả về dữ liệu thô chứa đường link và hình ảnh của các bài báo trong một trang lịch sử và render nó ở cuối. Điểm đặc biệt nằm ở việc tham số

`page_number` có thể được tuỳ biến theo ý muốn, vượt qua con số 20 - 30, cho phép nhóm tự động hóa việc truy xuất các bài viết cũ nằm ngoài phạm vi hiển thị mặc định của giao diện người dùng.

Hình 1: Kết quả trả về từ API ẩn của báo Thanh Niên

2.2.2 Trích xuất dữ liệu

Nhóm sử dụng 3 thư viện Python cho việc trích xuất dữ liệu:

- `requests`: Thư viện dùng để gửi HTTP requests tới API cũng như đến các bài bào
- `Beautiful Soup`: Thư viện dùng để trích xuất thông tin từ dạng HTML, XML,.. theo điều kiện mong muốn.
- `sqlite3`: Connector tới sqlite, một hệ quản trị cơ sở dữ liệu SQL nhẹ, serverless, các database sẽ được xuất ra thành từng file, dùng cho việc lưu trữ các bài báo

Quy trình trích xuất dữ liệu từ một bài báo của nhóm gồm:

- Lặp qua các id của từng chuyên mục và số lượng page theo cài đặt (150 trong lần chạy của nhóm).
- Với mỗi `id` và `page_number`, gửi một HTTP request tới API ẩn của báo Thanh Niên, dùng `Beautiful Soup` lọc ra danh sách các đường link đến bài báo
- Gửi request tới đường link của từng bài báo đó nhận nội dung HTML trả về.
- Tiếp tục dùng `Beautiful Soup` để duyệt qua các HTML tag chứa nội dung bài báo gồm tiêu đề, ngày đăng, nội dung, các tag phụ và trích xuất chúng.
- Thực hiện lưu trữ nội dung đã trích xuất vào database.

2.2.3 Tổ chức lưu trữ dữ liệu

Nhóm đã sử dụng SQLite để tổ chức lưu trữ dữ liệu. Toàn bộ dữ liệu sau khi cào được nằm trong một file `articles.sqlite3` với 2 bảng `articles` và `sub_tags`

```
CREATE TABLE articles (
    link TEXT PRIMARY KEY,
    publication_date DATETIME,
    content TEXT,
    main_tag TEXT,
    source TEXT
);

CREATE TABLE sub_tags (
    link TEXT,
    tag TEXT,
    PRIMARY KEY (link, tag),
    FOREIGN KEY(link) REFERENCES articles(link)
);
```

Listing 1: Mã nguồn tạo bảng CSDL

Nhóm quyết định dùng mô hình cơ sở dữ liệu quan hệ (relational database) thay vì dùng file JSON hay CSV vì các lý do sau đây:

- CSV không thích hợp để chứa nội dung các bài báo, vốn gồm nhiều dấu phẩy (,) và dấu nháy kép (").

- Việc cập nhật dữ liệu vào file JSON truyền thống gặp khó khăn lớn khi kích thước file tăng lên, do quy trình này thường đòi hỏi phải xử lý lại toàn bộ nội dung file để đảm bảo đúng cấu trúc cú pháp.
- Với SQLite, nhóm có thể thiết lập quy trình ghi dữ liệu liên tục theo từng lô nhỏ (5 bài một lần insert) để đảm bảo hiệu suất mà có thể tránh các rủi ro kỹ thuật (mất mạng, sập nguồn) làm mất data trong quá trình cào bài báo.
- Việc dùng CSDL đã có ép kiểu (DATETIME) và đặt khoá chính từ trước sẽ giúp loại bỏ các dữ liệu trùng lặp hoặc gặp lỗi trong quá trình trích xuất, giúp làm giảm công việc cho khâu làm sạch.

2.3 Quy trình làm sạch và tiền xử lý dữ liệu

Trước khi đưa vào bất cứ mô hình nào, dữ liệu thô đều trải qua các bước làm sạch cơ bản để loại bỏ nhiễu:

- **Xử lý giá trị thiếu và trùng lặp:** Thực hiện loại bỏ các mẫu dữ liệu bị khuyết thiếu thông tin quan trọng (như tiêu đề hoặc nội dung) và các bài viết bị trùng lặp hoàn toàn để tránh hiện tượng rò rỉ dữ liệu (*data leakage*) giữa các tập huấn luyện và kiểm thử. Sau bước này, kích thước bộ dữ liệu còn lại là 20,777 bài viết.
- **Loại nhiễu định dạng:** Dữ liệu văn bản được quét để phát hiện và loại bỏ các đoạn mã HTML, CSS hoặc JavaScript không mang ngữ nghĩa, đảm bảo nội dung đầu vào hoàn toàn là văn bản tự nhiên.
- **Chuẩn hoá bảng mã Unicode:** Do đặc thù tiếng Việt có hai kiểu gõ phổ biến (dựng sẵn và tổ hợp), toàn bộ văn bản được chuẩn hoá về định dạng Unicode NFC để đảm bảo tính nhất quán trong biểu diễn ký tự.

Tuy nhiên, các mô hình có sự khác biệt đặc thù về cơ chế hoạt động, chủ yếu là sự khác biệt của Transformer so với phần còn lại. Vì thế giai đoạn tiền xử lý được chia thành hai luồng riêng biệt:

- Đối với các mô hình cơ bản (TF-IDF + SVM, FastText), việc giảm chiều dữ liệu và thống nhất từ vựng là ưu tiên hàng đầu:

  - **Chuẩn hoá ký tự thường (*Lowercasing*):** Chuyển toàn bộ văn bản về chữ thường để giảm kích thước bộ từ điển mà không làm mất đi quá nhiều ngữ nghĩa quan trọng.
  - **Tách từ (*Word Segmentation*):** Sử dụng thư viện `pyvi` để gộp các từ ghép tiếng Việt, giúp nhận diện đúng đơn vị từ vựng.
  - **Loại bỏ từ dừng (*Stopword Removal*):** Loại bỏ các hư từ (*stopwords*) dựa trên danh sách từ dừng tiếng Việt. Bước này giúp giảm nhiễu và tập trung vào các từ mang thông tin chính trị, kinh tế, xã hội đặc thù.

- Đối với mô hình ngôn ngữ tiền huấn luyện PhoBERT, ngữ cảnh và cấu trúc câu đóng vai trò then chốt. Do đó, chiến lược xử lý được điều chỉnh như sau:
  - **Giữ nguyên định dạng (*No Lowercasing/Stopword Removal*):** Chúng tôi không chuyển văn bản về chữ thường và không loại bỏ từ dừng. Việc này nhằm bảo toàn cấu trúc ngữ pháp và thông tin về tên riêng, giúp cơ chế *Attention* của PhoBERT hoạt động hiệu quả nhất.
  - **Tách từ (*Word Segmentation*):** Tương tự như luồng cơ bản, văn bản đầu vào cho PhoBERT cũng cần dược tách từ để đồng bộ với dữ liệu mà PhoBERT đã được huấn luyện trước đó.

Cuối cùng, các nhãn đầu ra phải được mã hoá bằng cách ánh xạ các nhãn (gồm 7 nhãn cho ngữ cảnh của đồ án này) thành các con só nguyên (từ 0 tới 6 cho ngữ cảnh của đồ án này) để phục vụ quá trình tính toán của máy.

2.4 Đảm bảo chất lượng dữ liệu

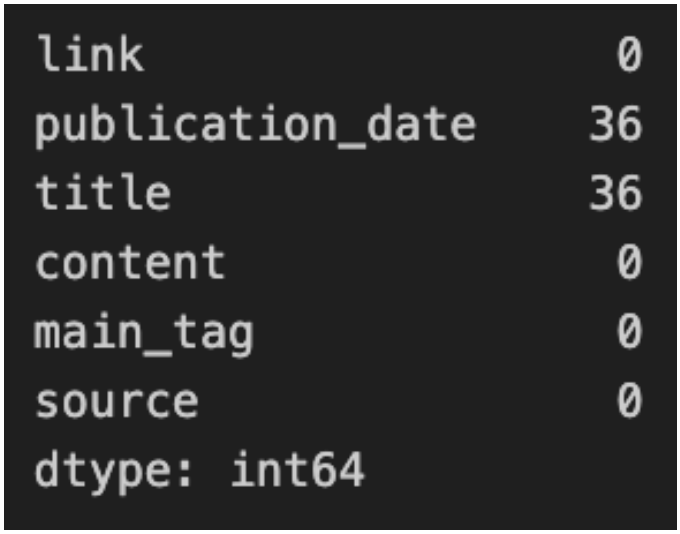
Để đảm bảo mô hình phân loại đạt hiệu suất cao và có khả năng tổng quát hoá tốt, bộ dữ liệu huấn luyện cần thoả mãn các tiêu chí:

- **Tính đầy đủ:** Các mẫu dữ liệu không được khuyết thiếu các trường thông tin quan trọng.
- **Tính sạch:** Dữ liệu văn bản không chứa các ký tự lạ, mã lỗi (HTML tags, JavaScript) hoặc các nội dung rác (quảng cáo hay link spam).
- **Tính nhất quán:** Dữ liệu phải đồng nhất về định dạng (Unicode NFC), ngôn ngữ (tiếng Việt) và cách gán nhãn chủ đề.

- **Tính cân bằng:** Số lượng mẫu dữ liệu giữa các nhãn không được chênh lệch quá lớn để tránh hiện tượng mô hình bị thiên vị (*bias*) về phía lớp đa số.

Quy trình kiểm tra gồm 2 bước chính:

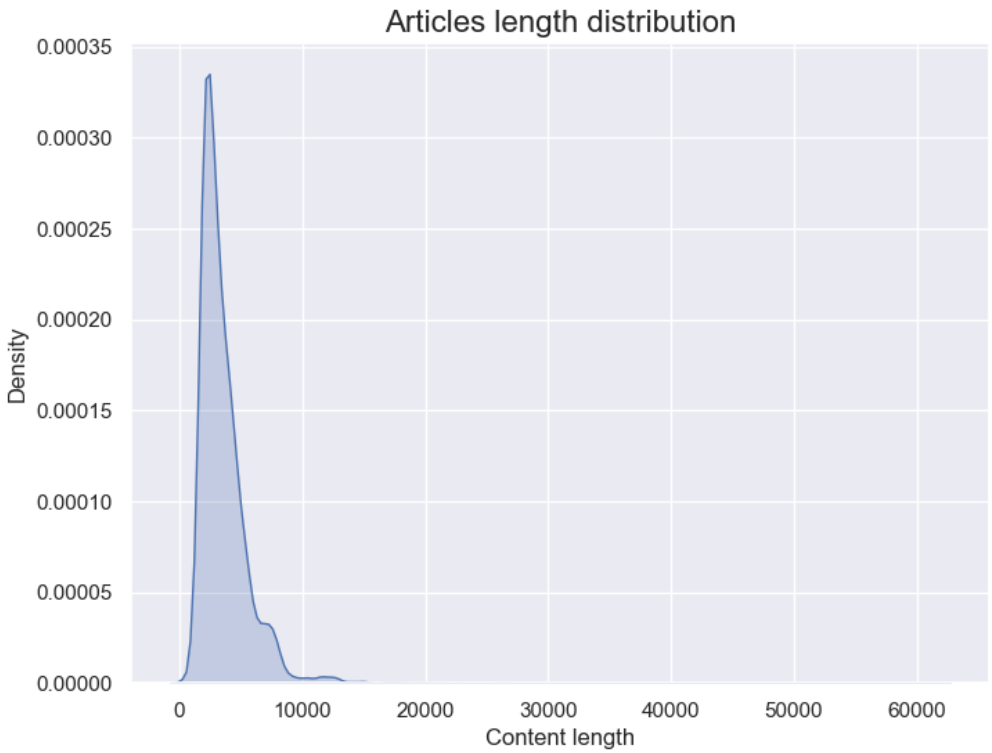
- **Kiểm tra tự động (*Automatic checks*):** Dựa trên các tiêu chí đã đề ra, xây dựng script để kiểm tra tự động quy trình EDA:
  - **Kiểm tra giá trị rỗng:** Sử dụng `pandas` để quét toàn bộ DataFrame. Kết quả phát hiện 36 mẫu bị thiếu thông tin nghi là các quảng cáo, các mẫu đã được loại bỏ tự động.
  - **Kiểm tra trùng lặp:** Rà soát dựa trên đường dẫn và nội dung bài báo để đảm bảo tính duy nhất.
  - **Kiểm tra độ dài văn bản:** Tính toán độ dài ký tự của từng bài báo để phát hiện các bài quá ngắn hoặc quá dài.
  - **Kiểm tra phân bố nhãn:** Thống kê số lượng bài viết theo từng chủ đề. Kết quả cho thấy dữ liệu đạt độ cân bằng rất tốt, không cần áp dụng các kỹ thuật tái lấy mẫu (*Resmapling*).
- **Kiểm tra thủ công (*Manual checks*):** Lấy ngẫu nhiên 5 bài viết từ các nhãn khác nhau. Đọc lướt nội dung để xác minh nhãn chủ đề ban đầu có khớp với nội dung thực tế không. Kết quả là dữ liệu đạt độ sạch hoàn hảo.



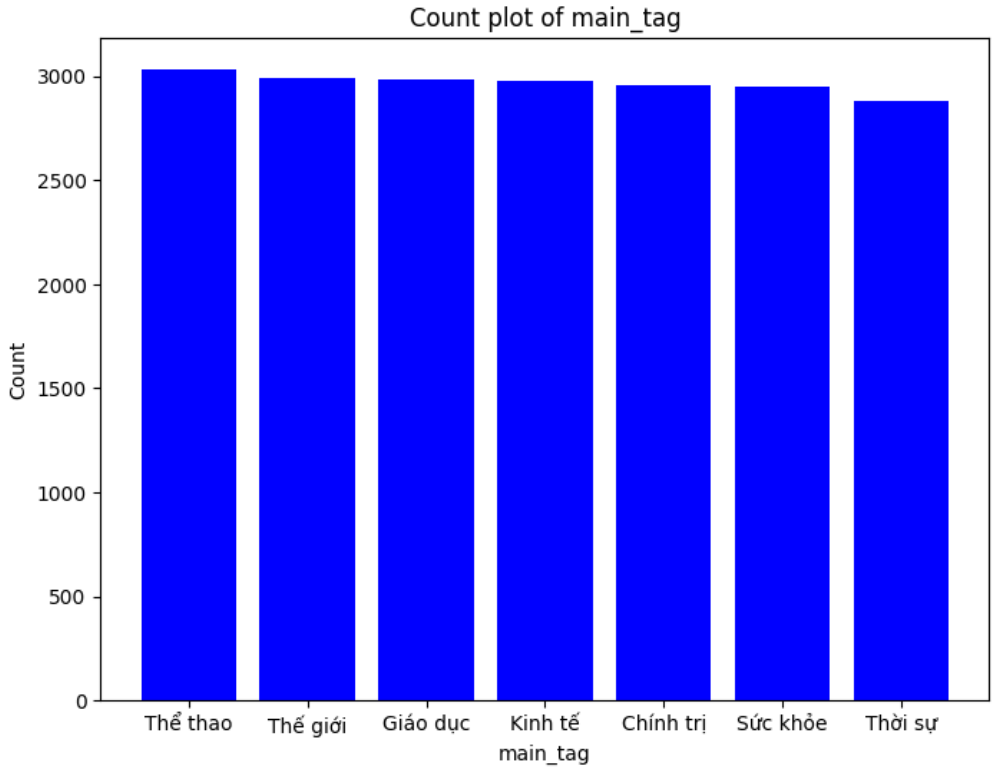
Hình 2: Đầu ra của bước kiểm tra giá trị rỗng. Cho thấy có 36 giá trị rỗng ở 2 cột `publication_date` và `title`.

```
link                20777
publication_date    19456
title               20755
content             20777
main_tag                7
source                  1
dtype: int64
```

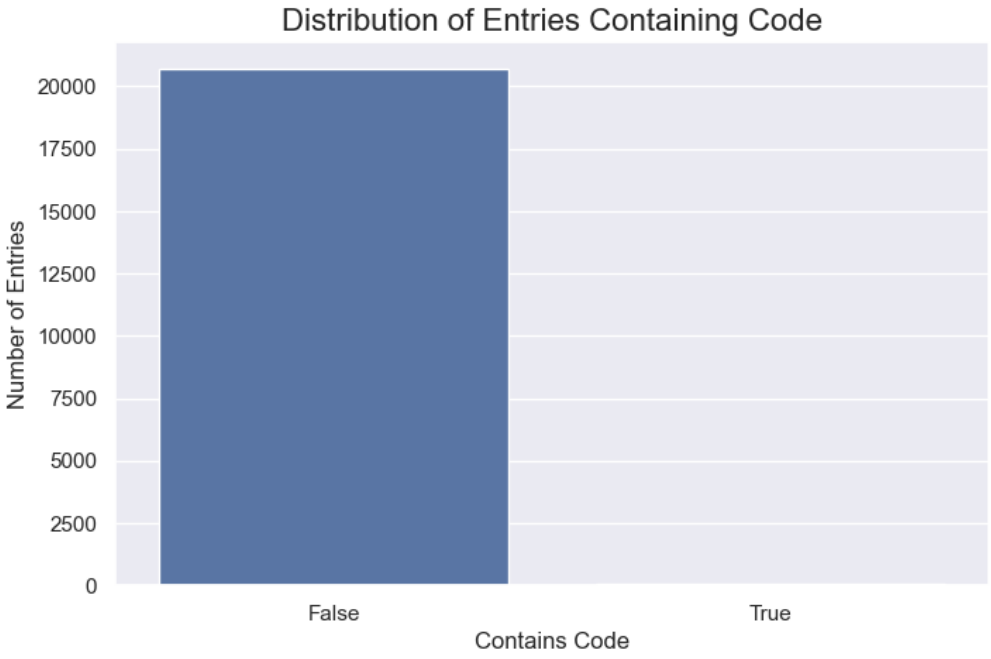
Hình 3: Đầu ra của bước kiểm tra trùng lặp. Cho thấy không có giá trị trùng lặp giữa các mẫu.



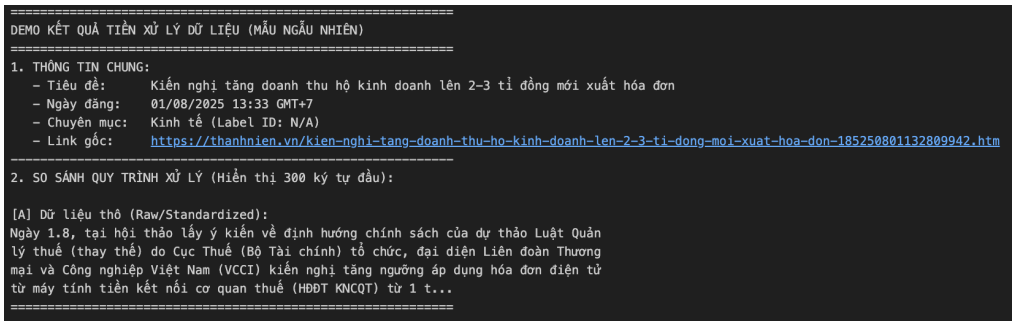
Hình 4: Biểu đồ thể hiện độ dài của các bài báo. Các bài báo trong tập dữ liệu có độ dài chủ yếu phân bố từ 3000-4000 từ.



Hình 5: Biểu đồ thể hiện phân bố nhãn của tập dữ liệu. Cho thấy sự phân bố đều một cách lý tưởng.



Hình 6: Biểu đồ thể hiện việc các bài báo có chứa mã thực thi hay không. Cho thấy không hề có mã thực thi trong dữ liệu.



Hình 7: Đầu ra của bước kiểm tra thủ công sau cùng. Chọn ngẫu nhiên một bài báo rồi hiện ra khoảng 300 từ đầu tiên trong bài báo. Kết quả cho thấy dữ liệu sau khi phân tích khám phá đã sạch sẽ.

2.5 Lưu trữ và quản lý dữ liệu

Để đâm bảo khả năng tái lập (*reproducibility*) và thuận tiện cho việc huấn luyện trên nhiều kiến trúc mô hình khác nhau, dữ liệu được quản lý theo quy trình 3 tầng:

- **Dữ liệu trung gian:** Là dữ liệu sau bước phân tích khám phá, được lưu dưới dạng Pickle (`.pkl`) để giữ nguyên các thuộc tính đối tượng Python (như DataFrame) sau khi làm sạch sơ bộ, giúp tiết kiệm thời gian tải dữ liệu của quá trình kế tiếp.
- **Dữ liệu huấn luyện:** Được chia tách và lưu trữ dưới dạng CSV trong các thư mục riêng biệt, sẵn sàng nạp vào mô hình.

Tập dữ liệu sau cùng sẽ được phân tách thành 3 tập riêng biệt là Train, Validation và Test với tỉ lệ 80%-10%-10%. Tuy nhiên cần áp dụng kỹ thuật lấy mẫu phân tầng (*Stratified Sampling*) để đảm bảo phân bố đều các nhãn ra các tập dữ liệu. Dưới đây là bảng phân bố các nhãn:

`label_encoded`	**Train**	**Validation**	**Test**
0	0.145960	0.145813	0.145813
1	0.143854	0.143888	0.143888
2	0.143794	0.143407	0.143888
3	0.143132	0.143407	0.142926
4	0.142410	0.142445	0.142445
5	0.142109	0.141963	0.142445
6	0.138740	0.139076	0.138595

Bảng 1: Bảng so sánh tỉ lệ phân bố của 3 tập dữ liệu

Dễ thấy từ bảng trên, tỉ lệ phân bố các nhãn của 3 tập dữ liệu khá đồng đều chứng tỏ quá trình phân chia dữ liệu đảm bảo được tính cân bằng.

3 Phân tích khám phá dữ liệu

4 Tài liệu tham khảo