TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN/ĐỒ ÁN CUỐI KÌ MÔN**

**XỬ LÝ NGÔN NGỮ TỰ NHIÊN**

**FINE-TUNE VÀ ĐÁNH GIÁ MÔ HÌNH TRANSFORMER TRÊN DỮ LIỆU TIẾNG VIỆT**

*Người hướng dẫn*: **TS LÊ ANH CƯỜNG**

*Người thực hiện*: **Lê Như Đạt – 52200160**

**Cao Minh Quân – 52200136**

Lớp **: 22050301**

Khoá  **: 26**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2025**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN/ĐỒ ÁN CUỐI KÌ MÔN**

**XỬ LÝ NGÔN NGỮ TỰ NHIÊN**

**FINE-TUNE VÀ ĐÁNH GIÁ MÔ HÌNH TRANSFORMER TRÊN DỮ LIỆU TIẾNG VIỆT**

Người hướng dẫn: **TS LÊ ANH CƯỜNG**

Người thực hiện: **Lê Như Đạt - 52200160**

**Cao Minh Quân - 52200136**

Lớp **: 22050301**

Khoá  **: 26**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2025**

**LỜI CẢM ƠN**

Trong suốt quá trình học tập và thực hiện đồ án môn *Xử lý Ngôn ngữ Tự nhiên*, chúng em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến **TS Lê Anh Cường**. Thầy không chỉ truyền đạt kiến thức một cách rõ ràng, hệ thống và đầy cảm hứng, mà còn giúp chúng em hiểu được giá trị thực tiễn cũng như tầm quan trọng của NLP trong các ứng dụng hiện đại.

Nhờ sự tận tâm hướng dẫn của thầy, chúng em có cơ hội tiếp cận nhiều mô hình và kỹ thuật tiên tiến như Transformer, PhoBERT, T5, cũng như cách áp dụng chúng vào bài toán thực tế. Thầy luôn khuyến khích chúng em suy nghĩ độc lập, tìm tòi, thử nghiệm và không ngừng hoàn thiện sản phẩm nghiên cứu của mình. Đây là nền tảng quan trọng giúp chúng em trưởng thành hơn trong tư duy, kỹ năng và định hướng chuyên môn.

Chúng em xin trân trọng cảm ơn thầy vì những góp ý, định hướng và sự hỗ trợ quý báu trong suốt học kỳ. Những kiến thức và kinh nghiệm mà thầy mang đến chắc chắn sẽ là hành trang quan trọng với chúng em trong con đường học tập và nghiên cứu sau này.

Xin kính chúc thầy thật nhiều sức khỏe, niềm vui và luôn tiếp tục truyền cảm hứng cho các thế hệ sinh viên tiếp theo.

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi / chúng tôi và được sự hướng dẫn của TS Lê Anh Cường;. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng năm*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Đạt*

*Lê Như Đạt*

*Quân*

*Cao Minh Quân*

**PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN**

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**TÓM TẮT**

Báo cáo trình bày về các bài toán cần tìm hiểu và các mô hình fine tune trong transformer, cũng như đánh giá.

**MỤC LỤC**

[LỜI CẢM ƠN i](#_qa7u2ufzxr3l)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN i](#_f8qlgp6bzxa8)

[TÓM TẮT i](#_5kuorq7o7j12)

[MỤC LỤC 1](#_lu7pdjfdv2kw)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 1](#_lyx7gjcs8fq6)

[CHƯƠNG 1 – MỞ ĐẦU 1](#_lrccpqef7n8w)

[1.1 Tiểu mục cấp 1 1](#_uif8rtizmkom)

[1.1.1 Tiểu mục cấp 2 1](#_c983ldwe903r)

[1.1.1.1 Tiểu mục cấp 3 1](#_v6u1ypch6oe3)

[1.1.1.2 Tiểu mục cấp 3 tiếp theo. 1](#_lg36r56ffzb0)

[1.1.2 Tiểu mục cấp 2 tiếp theo 1](#_rvmnzhgltd1b)

[1.2 Nội dung của chương này 1](#_i49rmpsjsyzp)

[CHƯƠNG 2 – TỔNG QUAN 1](#_ud14k7koijuk)

[1.1 Trình bày công thức toán học 1](#_sv0ktt96qerv)

[1.2 Trình bày một hình vẽ, sơ đồ 1](#_mkx6dwljsxg0)

[CHƯƠNG 3 – CƠ SỞ LÝ THUYẾT / NGHIÊN CỨU THỰC NGHIỆM 1](#_kejs8uhihfyx)

[3.1 Chèn bảng: 1](#_kdqsewba5kb3)

[3.2 Viết tắt 1](#_kgp3wcp76zkz)

[3.3 Trích dẫn 1](#_fwm9sqfcarxm)

[3.3.1 Tài liệu tham khảo và cách trích dẫn 1](#_uier8ubogib9)

[3.3.2 Qui định của Khoa Công nghệ thông tin 1](#_jcclnnjpu27x)

**CHƯƠNG 1 – GIỚI THIỆU VÀ TỔNG QUAN MÔ HÌNH**

**1.1 Giới thiệu và mục tiêu dự án**

***1.1.1 Encoder only***

Trong bối cảnh lượng tin tức tiếng Việt trên Internet ngày càng tăng mạnh, nhu cầu tự động phân loại chủ đề bài báo trở nên rất quan trọng đối với các hệ thống báo điện tử, cổng thông tin hay nền tảng mạng xã hội. Việc gắn nhãn chủ đề giúp tổ chức, tìm kiếm và gợi ý nội dung chính xác hơn cho người dùng, đồng thời hỗ trợ các bài toán phía sau như lọc nội dung, phân tích xu hướng, hay xây dựng hệ thống hỏi–đáp theo lĩnh vực. Tuy nhiên, tiếng Việt có nhiều đặc thù về dấu, từ ghép và cách dùng từ, khiến các phương pháp truyền thống dựa trên bag-of-words hoặc mô hình máy học đơn giản khó đạt chất lượng cao.

Để giải quyết hạn chế đó, các mô hình ngôn ngữ tiền huấn luyện (pretrained language models) như PhoBERT đã chứng minh hiệu quả vượt trội trong nhiều bài toán xử lý ngôn ngữ tự nhiên tiếng Việt. Dựa trên tiền đề này, bài toán mà nhóm thực hiện là phân loại chủ đề tin tức tiếng Việt với 21 nhãn (ví dụ: Chính trị, Kinh doanh, Đời sống, Bất động sản, Góc nhìn, Podcasts, Thư giãn,…), sử dụng mô hình encoder-only PhoBERT làm backbone. Dữ liệu được thu thập từ bộ dữ liệu tin tức gốc (khoảng gần 600.000 mẫu), sau đó làm sạch, loại trùng, cân bằng lại và cắt xuống còn 50.000 mẫu để thuận tiện cho huấn luyện và đánh giá. Trên cơ sở đó, nhóm tiếp tục bổ sung dữ liệu synthetic cho các nhãn ít mẫu bằng cách sinh văn bản mới, nhằm kiểm tra xem dữ liệu tăng cường có thực sự giúp mô hình phân loại tốt hơn hay không.

Mục tiêu của dự án là xây dựng một pipeline hoàn chỉnh cho bài toán phân loại chủ đề tin tức tiếng Việt với 21 nhãn dựa trên mô hình PhoBERT, bao gồm các bước tiền xử lý và chuẩn hóa dữ liệu (làm sạch văn bản, chuẩn hóa Unicode, tách từ bằng ViTokenizer, mã hoá nhãn, chia train/val/test có chiến lược), sau đó huấn luyện và fine-tune mô hình trên hai kịch bản dữ liệu gốc và dữ liệu gốc + synthetic cho các nhãn ít mẫu, rồi dùng cả các thước đo truyền thống (Accuracy, Macro-F1) lẫn đánh giá dựa trên LLM (Gemini chấm coherence, relevance, factuality, accept) để so sánh khách quan hiệu quả hai mô hình, qua đó rút ra kết luận về mức độ cải thiện và ảnh hưởng thực tế của việc tăng cường dữ liệu synthetic trong bài toán phân loại văn bản tiếng Việt.

***1.1.3 Encoder - Decoder***

***1.1.4 Decoder only***

**1.2 Mô hình Transformer**

Transformer là kiến trúc mạng nơ-ron sâu được đề xuất để xử lý chuỗi mà không cần RNN hay CNN, thay vào đó dùng hoàn toàn cơ chế self-attention.

Mỗi từ trong câu được ánh xạ thành vector embedding, cộng thêm positional encoding để mô hình biết vị trí tương đối.

Trong một lớp Multi-Head Self-Attention, mỗi từ sinh ra 3 vector: *Query, Key, Value*. Điểm tương đồng giữa Query của một từ với Key của tất cả các từ khác sẽ quyết định mức “tập trung” (attention weight) vào các vị trí đó.

Nhờ cơ chế self-attention, Transformer có thể nắm bắt quan hệ dài hạn trong câu tốt hơn RNN truyền thống, đồng thời tính toán song song trên toàn bộ chuỗi nên huấn luyện nhanh hơn.

So sánh Encoder, Encoder – Decoder, Decoder:

* Encoder: gồm stack nhiều lớp Encoder (self-attention hai chiều + FFN). Mỗi token được nhìn cả trái lẫn phải (bidirectional), nên biểu diễn ngữ cảnh rất giàu thông tin. Phù hợp cho các bài toán hiểu ngôn ngữ: phân loại văn bản, gán nhãn chuỗi, trích xuất thông tin.
* Encoder – Decoder: Encoder mã hoá câu nguồn → chuỗi vector ngữ cảnh. Decoder nhận thông tin từ encoder và sinh ra chuỗi đích từng bước, có *cross-attention* từ decoder sang encoder. Dùng cho các bài toán biến đổi chuỗi sang chuỗi: dịch máy, tóm tắt, sinh mô tả ảnh.
* Decoder: Chỉ gồm các lớp Decoder với causal self-attention (mỗi token chỉ nhìn về bên trái). Thích hợp cho sinh ngôn ngữ tự do (LLM kiểu GPT, ChatGPT, v.v.).

**CHƯƠNG 2 – MÔ TẢ DỮ LIỆU & SYNTHETIC DATA**

**2.1 Giới thiệu bộ dữ liệu gốc (encoder only)**

***2.1.1 Nguồn dữ liệu và cách thu thập***

Bộ dữ liệu sử dụng được lấy từ hugging face là bộ tập tin tức Tiếng Việt đa chủ đề có tổng dung lượng 1,04GB. Có tổng cộng 5 cột nhưng chỉ sử dụng 2 cột chính là content và label. Dữ liệu sau khi tải về sẽ được xử lý chỉ lấy 2 cột và chuyển về định dạng json có cấu trúc content – label.

***2.1.2 Hệ thống nhãn chủ đề***

Dữ liệu được gán nhãn theo 21 chủ về khác nhau, phản ánh nhiều mảng nội dung báo chí phổ biến như: ngôi sao, thế giới, giải trí giới trẻ, thể thao, kinh doanh, sức khỏe, bất động sản, thư giãn, padcasts, ý kiến, góc nhìn, đời sống, thời sự, giải trí, tâm sự, pháp luật, khoa học, số hóa, giáo dục, du lịch, xe.

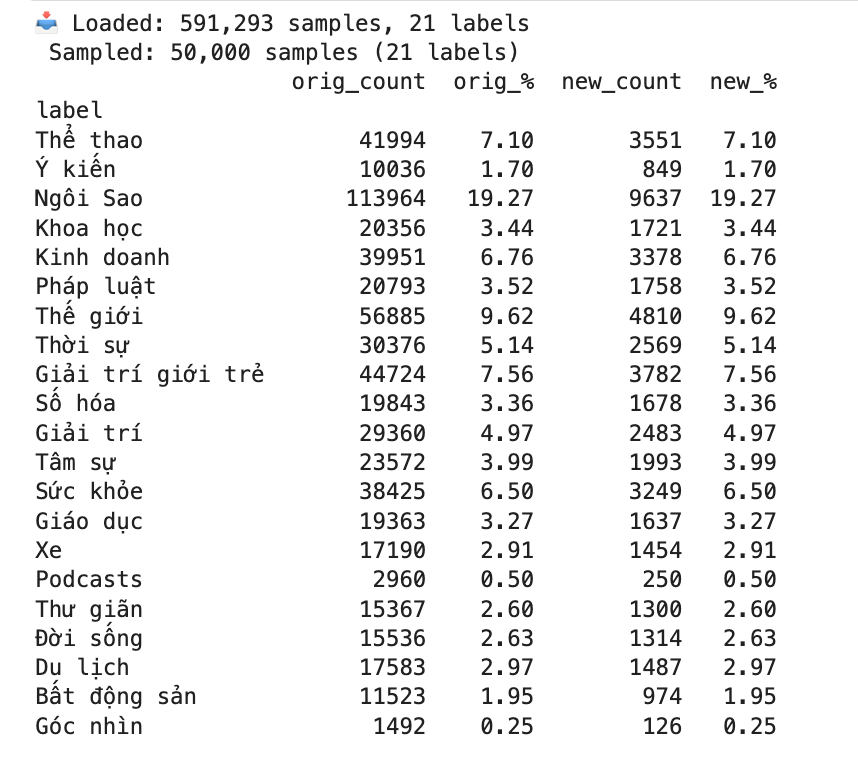
Mỗi bản ghi content đều có đúng một nhãn chủ đề

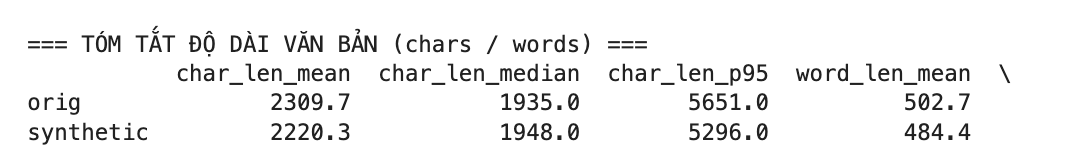
***2.1.3 Kích thước và định dạng dữ liệu***

Kích thước bộ dữ liệu gốc sử dụng khoảng gần 600k mẫu, nhưng chỉ cắt về khoảng 50k mẫu nhưng vẫn dữ độ tỉ lệ mẫu của nhãn tương ứng bộ gốc đẻ không làm thay đổi đi tính đặc trưng của bộ dữ liệu.

**2.2 Phân tích thống kê và chất lượng dữ liệu gốc (encoder only)**

***2.2.1 Phân bố số lượng mẫu theo nhãn***





***2.2.2 Làm sạch nhẹ và xóa trùng***

Dữ liệu sẽ được làm sạch nhẹ và lọc trùng trước khi được đưa vào tiền xử lý.

**2.3 Thiết kế dữ liệu synthetic (encoder only)**

***2.3.1 Mục tiêu sinh dữ liệu synthetic***

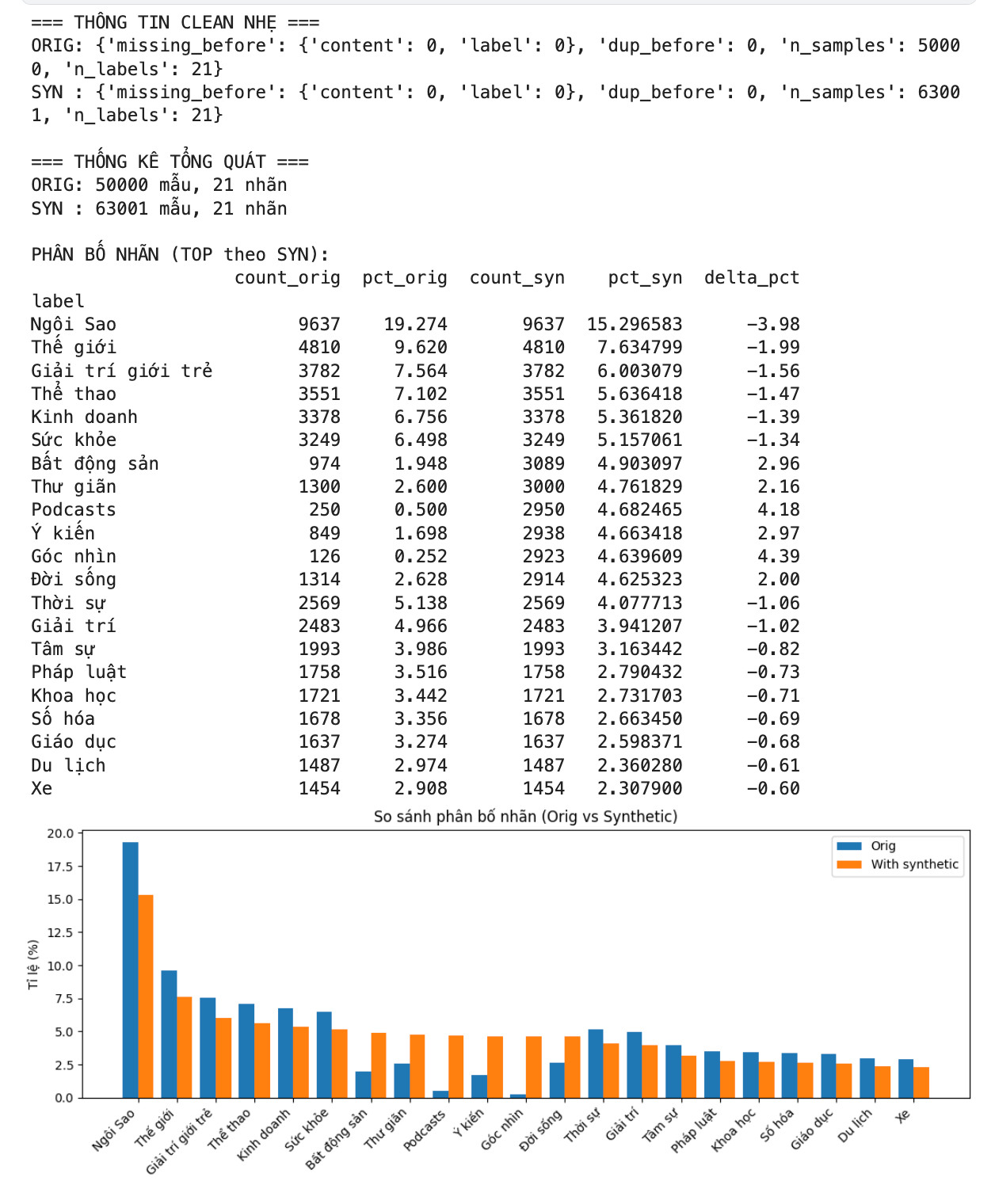
Mục tiêu chính của phần bổ sung dữ liệu synthetic bằng các phương pháp LLM là bổ sung dữ liệu cho các nhãn có ít ví dụ trong bộ gốc, giúp giảm mức độ mất cân bằng nhãn, giúp mô hình học tốt hơn các chủ đề hiếm, giữ chất lượng văn bản, với đề bài phần này dựa vào bộ data thì sẽ bổ sung synthetic cho 6 label yếu nhất là đời sống, thư giãn, bất động sản, ý kiến, podcasts, góc nhìn.

***2.3.2 Mô hình và cách thức sinh dữ liệu***

Để thực hiện phần này, nhóm sử dụng mô hình ngôn ngữ lớn (LLM) chạy local trên máy với 2 mô hình là DeepSeek-chat-7b và GPT-OSS-20B ở định dạng GGUF.

***2.3.3 Thiết kế prompt và chiến lược sinh***

Với mỗi label tương ứng dựa vào phần trực quan hóa dữ liệu đã có thông tin về số lượng mẫu của label yếu, trung bình kí tự, trung bình của từ xuất hiện trong label đó từ đó với mỗi label đưa ra các thông số phù hợp khác nhau.



Ví dụ label góc nhìn đang có khoảng 126 mẫu cần sổ sung khoảng 2800 mẫu để trung bình các mẫu được khoảng 3000. Và số từ sinh ra không được quá ít so sánh số từ trung bình của label này, tránh làm lạc dữ liệu ban đầu gây mất cân bằng thêm.

Quy trình sinh cho mỗi nhãn:

* Chuẩn bị prompt tương ứng với label đó, mô tả chi tiết, đa dạng hóa các nội dung sinh ra và đáp ứng các tiêu chí về số từ và kí tự, tổng sô lượng, trành lặp lại,..
* Sử dụng mô hình gpt-oss-20b tương ứng thực hiện chạy và sinh theo cấu trúc như đã thực hiện với dữ liệu trước đó là content – label và có cấu trúc dạng json. Với mỗi label khi chạy xong có 1 file kết quả tương ứng sau khi đã đủ số lượng cho các label cần thiết thì tiến hành lọc trùng và tiến hành trộn vào với file dữ liệu gốc trước đó để chuẩn bị cho quá trình tiền xử lý.

**2.1 Giới thiệu bộ dữ liệu gốc (encoder -decoder)**

Bộ dữ liệu “*nam194/vietnews*” trên Hugging Face là một tập dữ liệu tin tức tiếng Việt quy mô trung bình, gồm khoảng **144.000 bài báo** được lưu ở định dạng **Parquet** và tổ chức thành ba tập **train/validation/test** phục vụ trực tiếp cho các bài toán xử lý ngôn ngữ tự nhiên như phân loại, tóm tắt hoặc mô hình ngôn ngữ. Mỗi bản ghi thường bao gồm các trường như **title**, **abstract** (tóm tắt ngắn) và **article** (toàn văn bài báo).

Trong dự án này nhóm em chỉ sử dụng 10k sample cho bộ train, 2k sample cho bộ validation, 2k sample cho bộ test. Bởi vì chi phí để fine-tuning toàn bộ bộ dữ liệu kia tốn gần 40 tiếng.



**2.2 Giới thiệu bộ dữ liệu synthetic data**

Bộ dữ liệu này được em sinh bằng phương pháp sử dụng 1 model decoder là “Qwen/Qwen2.5-1.5B-Instruct”. Em sinh với 5 trial và thu được 2k8 sample.

**2.3 Mô tả bài toán và mục tiêu đầu ra**

Trong đề tài này, ta coi đây là **bài toán tóm tắt văn bản tiếng Việt**: đầu vào là nội dung bài báo (article, có thể kèm title), đầu ra là một đoạn tóm tắt ngắn gọn, súc tích, tương đương với trường abstract trong bộ dữ liệu. Cụ thể, mô hình cần học được cách rút trích và diễn đạt lại thông tin quan trọng của bài báo bằng ngôn ngữ tự nhiên, đảm bảo giữ đúng ý chính, ngữ pháp và phong cách tin tức.

**Mục tiêu đầu ra**

* Xây dựng một mô hình học sâu có khả năng **tự động sinh tóm tắt** cho các bài báo tiếng Việt mới, không có trong tập huấn luyện.
* Đầu ra mong muốn là một đoạn văn ngắn (vài câu) **mô tả đầy đủ nội dung chính** của bài báo, độ dài tương đương hoặc ngắn hơn trường abstract trong dữ liệu.
* Đánh giá chất lượng mô hình bằng các thước đo chuẩn cho bài toán tóm tắt như **ROUGE-1, ROUGE-2, ROUGE-L**, so sánh tóm tắt sinh ra với abstract gốc để kiểm tra mức độ trùng khớp nội dung và độ bao phủ thông tin.

**2.4 Giới thiệu bộ data gốc decoder only**

Bộ dữ liệu bkai-foundation-models/vi-alpaca trên Hugging Face là một tập 50.000 cặp instruction–input–output bằng tiếng Việt, được xây dựng theo phong cách Stanford Alpaca / Self-Instruct, dùng để huấn luyện và tinh chỉnh các mô hình ngôn ngữ hướng chỉ dẫn (instruction-tuned LLM).

Dữ liệu được BKAI-HUST Foundation Models Lab tạo ra bằng cách sinh các nhiệm vụ hạt giống (seed tasks) tiếng Việt và dùng GPT-4, GPT-3.5… để sinh ra đa dạng lệnh (instruction) và đáp án, từ đó rất phù hợp cho các bài toán chatbot tiếng Việt, trợ lý ảo, hỏi–đáp, và sinh văn bản theo yêu cầu.



**2.4 Giới thiệu bộ synthetic data cho decoder only**

Bộ dữ liệu này nhóm em dùng 2 model “gpt-oss-20b” và “deepseek-llm-chat7b-Q6\_K”. Model gpt nhóm em sinh 1k sample. Model deepseek nhóm em sinh 4k sample. Tổng là 5k sample được generate thêm.

**2.5 So sánh giữa các LLM dùng để generate synthetic data**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mô hình | Kích thước & kiến trúc | Ngôn ngữ & context | Chất lượng sinh dữ liệu  (ước lượng tương đối) | Tốc độ & tài nguyên | Gợi ý cách dùng để sinh data |
| Qwen/Qwen2.5-1.5B-Instruct | ~1.5B tham số, Transformer, RoPE, SwiGLU, RMSNorm; bản instruct, tối ưu cho thiết bị yếu. | Multilingual >29 ngôn ngữ (có Tiếng Việt), context tối thiểu 32K, một số triển khai cho phép tới 128K token. | Thấp–Trung bình: dễ bị lỗi format, lẫn ngôn ngữ như bạn quan sát; cần prompt chặt chẽ, data đơn giản (FAQ, câu ngắn, template cố định). | Rất nhanh, chạy được trên GPU yếu hoặc thậm chí CPU; phù hợp generate số lượng lớn với chi phí thấp. | Hợp để sinh dữ liệu “thô”: câu hỏi ngắn, câu ví dụ, paraphrase; nên kèm thêm bước lọc / hậu kiểm (regex, rule) để sửa lỗi format và loại mẫu kém. |
| gpt-oss-20b | ~21B tham số (3.6B active MoE), open-weight, tối ưu reasoning và chain-of-thought. | Model general-purpose, context dài tới ~131K token, thiết kế cho tác vụ reasoning, tool-use, hỗ trợ prompt dài và phức tạp. | Cao: văn bản mạch lạc hơn, tuân thủ format tốt hơn, phù hợp sinh dataset phức tạp (multi-step reasoning, hội thoại dài, bài viết dài). | Chậm hơn 1.5B/7B rõ rệt, yêu cầu tối thiểu khoảng 16GB VRAM để chạy local; độ trễ cao hơn khi sinh nhiều token. | Hợp để sinh dữ liệu chất lượng cao: bài giải có giải thích bước, bài đọc dài, hội thoại multi-turn, dữ liệu cần lập luận; nên dùng cho batch nhỏ, phần “core” quan trọng của dataset. |
| deepseek-llm-7b-chat7b-Q6\_K | 7B tham số gốc, bản chat; bản Q6\_K GGUF (6-bit quantization) giúp giảm RAM nhưng hơi giảm chất lượng. | Chủ yếu hỗ trợ tiếng Anh và tiếng Trung, context khoảng 4K token. | Trung bình–Khá: tốt hơn 1.5B về mạch lạc và tuân thủ hướng dẫn, nhưng không bằng 20B; bản quant Q6\_K đôi khi sinh văn bản hơi lặp hoặc lệch hơn bản full-precision. | Tốc độ trung bình: nhanh hơn 20B, chậm hơn 1.5B; bản Q6\_K tối ưu cho máy RAM/GPU vừa phải (Ollama, llama.cpp, v.v.). | Hợp để sinh dữ liệu tầm trung: hội thoại, Q&A, mô tả ngắn–vừa; nếu sinh tiếng Việt thì nên kiểm tra kỹ và có bước post-edit vì model gốc tập trung en/zh. |

**CHƯƠNG 3 – THỰC NGHIỆM**

**3.1 Phương pháp và chi tiết huấn luyện (encoder only)**

***3.1.1 Mô hình và quy trình tổng quát***

Huấn luyện PhoBERT trên cả 2 bộ data gốc và bộ data đã thêm synthetic. 2 bộ dữ liệu đều sử dụng chung 1 ‘pipline như sau:

- tiền xử lý dữ liệu (làm sạch văn bản và chuẩn hóa)

- chia dữ liệu train/val/test

- word segmentation bằng Vitokenizer để chuẩn bị cho input PhoBERT.

- mã hóa nhãn sang id

- xây dựng data, tokenizer PhoBERT và fine tune mô hình phân loại.

- đánh giá và so sánh kết quả.

***3.1.2 Tiền xử lý văn bản***

- Làm sạch giá trị thiếu và trùng lặp.

- Làm sạch nội dung bằng regex

- Giới hạn độ dài để tránh tốn tài nguyên khi tokenize

- chuẩn hóa nhãn

***3.1.3 Word segmentation và chia train/validation/test***

Dữ liệu sau khi làm sạch sẽ được tiếp tục xử lý word segmentation với ViTokenizer để ghép các cụm từ Tiếng Việt bằng dấu gạch dưới.

- Mã hóa nhãn sử dụng LabelEncoder.

- Chia train/validdation/test với tỉ lệ 80/10/10.

***3.1.4 Mô hình PhoBERT***

PhoBERT là mô hình Transformer encoder-only được VinAI huấn luyện trước trên một lượng lớn corpora Tiếng Việt có kiến trúc:

- gồm nhiều lớp self-attention và feed-forward xếp chồng.

- hoạt động theo cơ chế masked language modeling trong giai đoạn pre-train.

- input được token hóa theo subword BPE trên dữ liệu đã word segmentation.

Trong bài nhóm sử dụng checkpoint vinai/phobert-base từ thư viện transformer.

***3.1.5 Xây dựng dataset và pipline fine-tune***

Mỗi dataset gồm: input\_ids, attention\_mask, label là id nhãn.

***3.1.6 Cấu hình huấn luyện (fine tune)***

Dùng chung hyper-parameters cho cả 2 bộ dữ liệu. Ý nghĩa các tham số

* batch\_size=24, num\_train\_epochs=4: kích thước batch và số epoch trung bình, tránh overfitting nhưng vẫn đủ hội tụ.
* learning\_rate=2e-5: LR chuẩn cho fine-tune BERT-like, tránh phá hỏng pretrained weights.
* warmup\_ratio=0.06: vài step đầu LR được tăng dần, giúp huấn luyện ổn định.
* weight\_decay=0.01: regularization nhẹ để giảm overfitting.
* lr\_scheduler\_type="linear": LR giảm tuyến tính về 0 trong suốt quá trình.
* fp16=True: nếu có GPU hỗ trợ, dùng half-precision để tiết kiệm VRAM.
* save\_strategy="epoch": mỗi epoch lưu một checkpoint đầy đủ; cuối cùng lưu OUT\_DIR làm final model.

Huấn luyện bằng Trainer.

**3.2 Đánh giá kết quả (metric + LLM evaluation) (encoder only)**

Xây dựng quy trình đánh giá và dự đoán sau fine-tune.

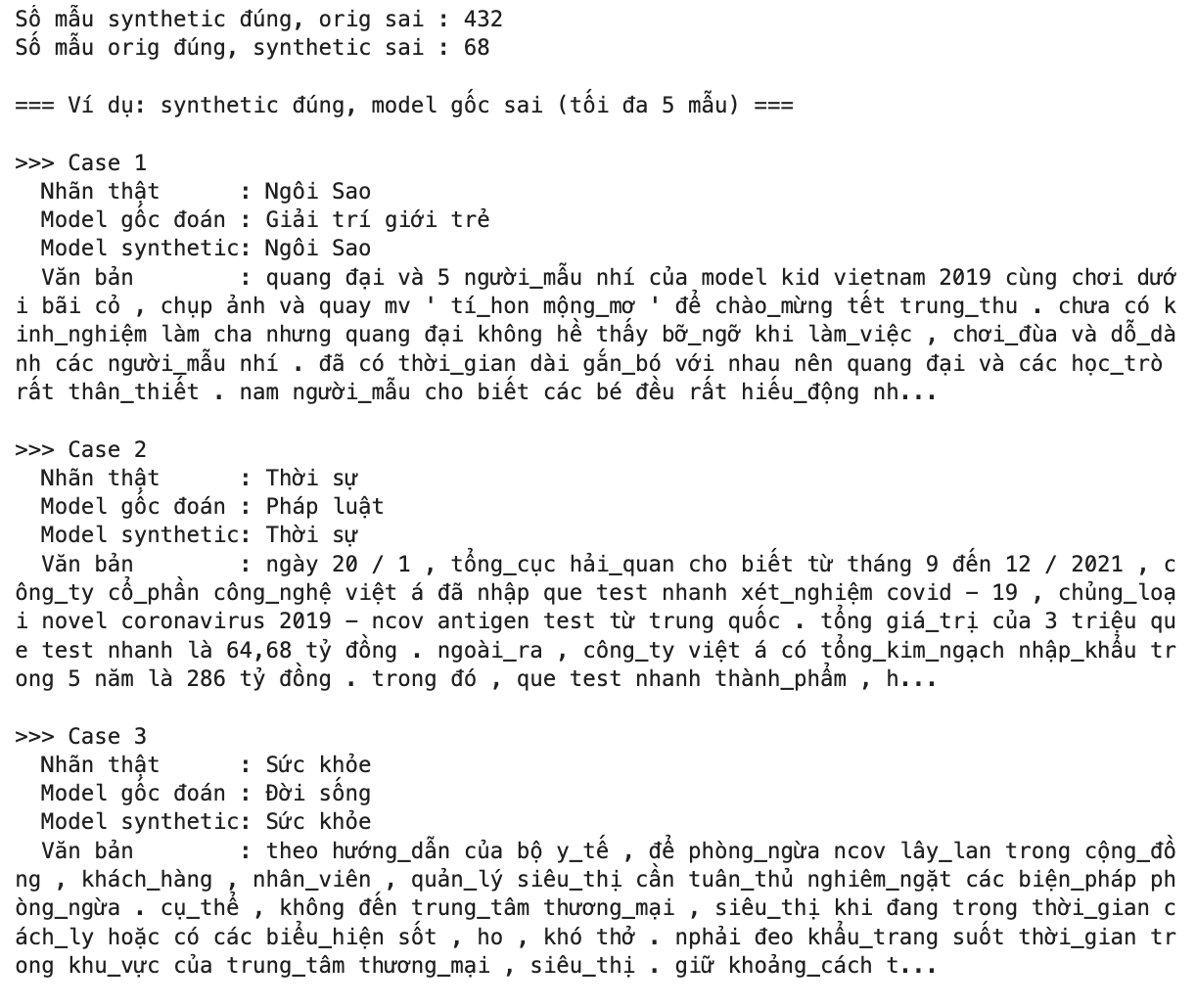
- Chuẩn bị hàm FrameDataset và evaluate\_one\_experiment: load dữ liệu và mapping nhãn, load model đã fine-tune, chuẩn bị dataloader cho test, chạy dự đoán, tính các metric tuyền thống, lưu kết quả để so sánh.

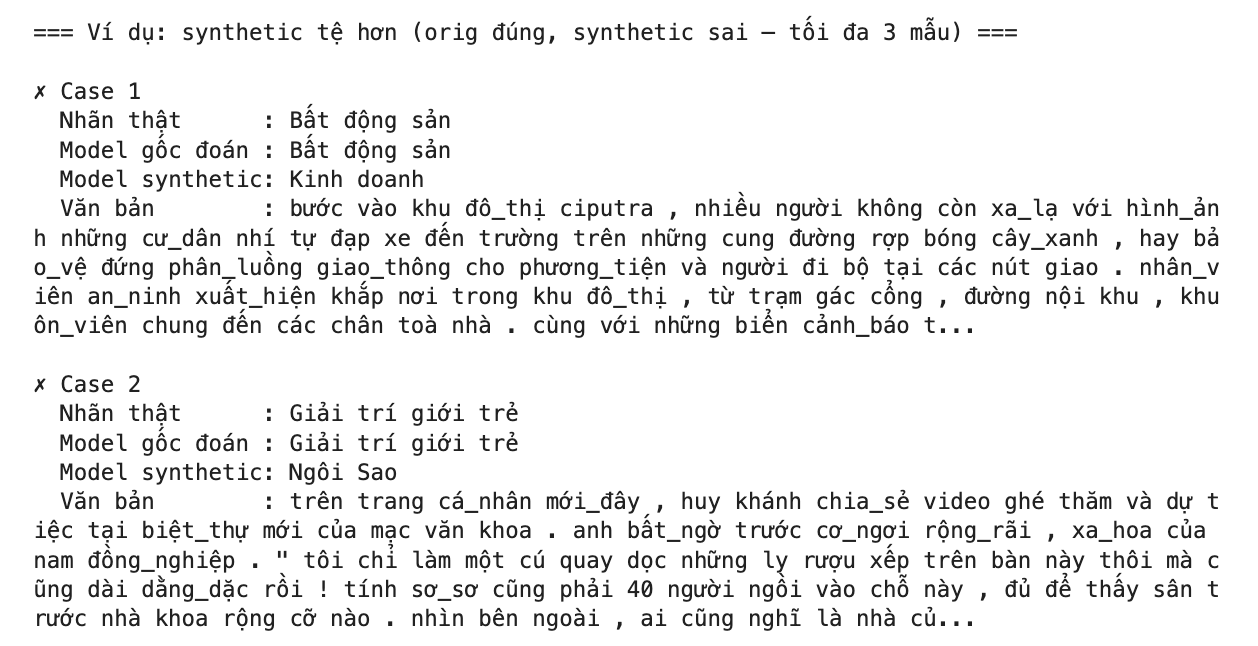
- Quy trình xây dựng dự đoán và đánh giá đều được sử dụng cho cùng cả 2 bộ dữ liệu.



Kết quả cho thấy mô hình với dữ liệu gốc cho kết quả khá tốt nhưng với bộ data đã thêm synthetic thậm chí còn tốt hơn chứng tỏ rằng mô hình đã học được các đặc trưng của các label hiếm khi dữ liệu được cân bằng.

1 số mẫu so sánh giữa 2 kết quả dự đoán.





Xây dựng quy trình đánh giá bằng LLM (gemini): Chọn 1 model bằng cách gọi API của model đó (gemini-2.0-flash) để mô hình này làm giám khảo để chấm cho chất lượng phân loại chủ đề ở các khía cạnh góc nhìn ngữ nghĩa có hợp lý không, bài viết có mạch lạc không, có thực sự hợp lý hay không.

- Chuẩn bị dữ liệu cho đầu vào của gemini: chạy model để lấy dự đoán + ghép với nhãn thật.

- Chọn mẫu cân bằng theo nhãn để gửi LLM.

- Thiết kế tiêu chí chấm và prompt cho Gemini:

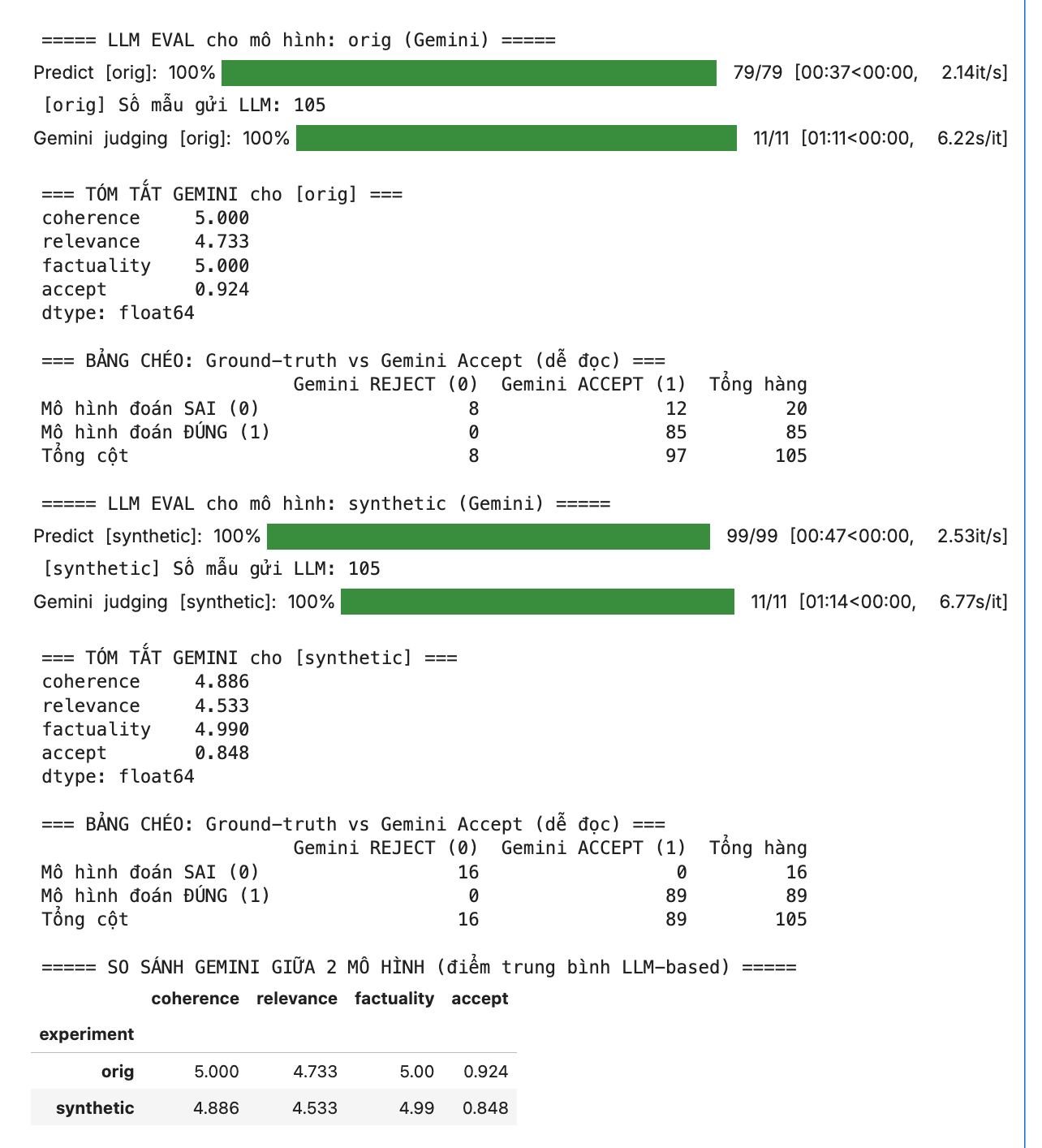
+ coherence: độ mạch lạc, logic của bài viết.

+ relevance: bài có phù hợp với chủ đề được gán hay không.

+ factuality: ý chính có đúng nội dung ở mức khái quát hay không.

+ verdict: ACCEPT – Gemini cho rằng việc gán chủ đề này chấp nhận được về mặt ngữ nghĩa, REJECT – sai lệch rõ hoặc không hợp lý.

Kết quả chấm như sau:



Với từng bộ dữ liệu cho ra số điểm trên thang 5đ.

Bảng chéo của origin mô hình dự đoán sai 20 mẫu nhưng gemini chấp nhận 12 mẫu nghĩa là 12 mẫu này nhãn sai nhưng về ngữ nghĩa thì lại chấp nhận. Còn với bộ synthetic thì mô hình dự đoán sai 16 mẫu và gemini thấy những lần sai này là rõ ràng, không chấp nhận được về mặt chủ đề.

Synthetic data giúp mô hình giảm số lỗi thuần túy theo ground-truth, nhưng khi mô hình dự đoán sai thì lỗi thường nặng hơn, lệch hẳn chủ đề.

Còn với dữ liệu gốc thì có ít điểm accuracy hơn, nhưng nhiều lỗi của nó vẫn được LLM gemini đánh giá là chấp nhận được về mặt ngữ nghĩa.

**3.3 Kết luận (encoder only)**

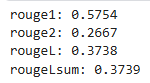
Sau quá trình fine-tune, dự đoán và đánh giá thì về số liệu truyền thống khi bổ sung synthetic data sẽ cải thiện accuracy và đặc biệt là macro-F1, nhất là với các nhãn thiểu số. Synthetic đã giúp mô hình học thêm các đặc trưng của các label yếu.

Cả 2 kết quả trên 2 bộ dữ liệu đối với mô hình đều cho ra kết quả đánh giá dựa trên Gemini rất cao. Tỷ lệ accept của mô hình synthetic hơi thấp hơn origin phản ánh mô hình khi chọn nhãn rất đúng rubric hơn nhưng gemini lại đánh giá theo thiên hướng hợp lý về mặt chủ quan như ngữ nghĩa.

Synthetic không chỉ cải thiện mectric mà còn mở ra cơ hội phân tích sâu hơn bằng LLM. Kết quả cho thấy việc kết hợp đánh giá truyền thống và mô hình ngôn ngữ lớn là cần thiết cho các bài toán NLP hiện đại.

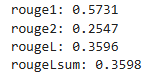
**3.4 Phương pháp và chi tiết huấn luyện (encoder - decoder)**

* **Tiền xử lý:** Bộ dữ liệu “nam194/vietnews” cơ bản đã hoàn toàn sạch sẽ nên em chỉ thêm bước xoá xuống dòng thừa, thay \_ bằng khoảng trắng.
* **Mô hình:** Mô hình và bộ tokenizer em sử dụng là “VietAI/vit5-base”
* **Metrics:** rouge1, rouge2, rougeL
* Kết quả đánh giá theo Rouge:



**3.5 Phương pháp và chi tiết huấn luyện synthetic\_data (encoder - decoder)**

* **Tiền xử lý:** Bộ dữ liệu cơ bản đã hoàn toàn sạch sẽ nên em chỉ thêm bước xoá xuống dòng thừa, thay \_ bằng khoảng trắng.
* **Mô hình:** Mô hình và bộ tokenizer em sử dụng là “VietAI/vit5-base”
* **Metrics:** rouge1, rouge2, rougeL
* Kết quả đánh giá theo Rouge:



**3.6 Phương pháp và chi tiết huấn luyện (decoder only)**

* **Tiền xử lý:** Do đây là mô hình decoder only nên em phải gộp 3 cột ‘instruction’, ‘input’, ‘output’ thành 1 message để đưa vào model:
* **Mô hình:** Mô hình và bộ tokenizer em sử dụng là “VietAI/vit5-base”
* **Metric**: Perplexity



**3.7** **Phương pháp và chi tiết huấn luyện synthetic\_data (decoder only)**

* **Tiền xử lý:** Do đây là mô hình decoder only nên em phải gộp 3 cột ‘instruction’, ‘input’, ‘output’ thành 1 message để đưa vào model:
* **Mô hình:** Mô hình và bộ tokenizer em sử dụng là “VietAI/vit5-base”
* **Metric**: Perplexity



3.8 RAG

Trong phần này nhóm em demo nhiều phiên bản RAG khác nhau như:

* Sử dụng File search + gemini
* RAG from scratch + advance technique (**Hybrid retrieval (dense + sparse) + Reciprocal Rank Fusion (RRF), Multi-query / Query rewriting với LLM + RRF lần 2, cross-encoder reranker)**
* RAG (langchain, basic RAG)
* RAG (langchain, basic RAG) + decoder tự fine tune
* RAG (langchain, basic RAG) + decoder tự fine tune trên bộ synthetic data

**TÀI LIỆU THAM KHẢO**

**Tiếng Việt**

1. Trần Q. Nhật, Trần N. Minh, và Q. M. Đỗ, “PhoBERT: Mô hình ngôn ngữ tiếng Việt dựa trên RoBERTa,” VinAI Research, 2020.Bộ nông nghiệp & PTNT (1996), *Báo cáo tổng kết 5 năm (1992-1996) phát triển lúa lai,* Hà Nội.
2. Phan M. Cường, “Bài giảng Xử lý Ngôn ngữ Tự nhiên,” Đại học Bách Khoa TP.HCM, 2021.
3. Trần V. Quang, “Tổng quan về mô hình Transformer và ứng dụng trong NLP tiếng Việt,” Báo cáo khoa học, 2021.

**PHỤ LỤC**

Link source code encoder-only <https://www.kaggle.com/code/caoquan415414/nlp-ver3>

Phần nâng cao RAG:

<https://drive.google.com/drive/folders/1-KwzeSv5TIuWsXVa4sUERtgTxulIfruN>

Decoder\_non\_synthetic

<https://www.kaggle.com/code/lndat1811/fine-tune-qwen1-5b>

Decoder\_synthetic

<https://www.kaggle.com/code/lndat1811/fine-tune-qwen1-5b-augmented-data>

Encoder-Decoder\_non\_synthetic

<https://www.kaggle.com/code/lndat181104/fine-tune-vit5>

Encoder\_Decoder\_synthetic

<https://www.kaggle.com/code/lndat181104/fine-tune-vit5-augmented-data>

Link github của dự án

<https://github.com/EvoMinhQuan04/NLP_FINAL>